Projektgruppe

Show_Me_MoP

Endbericht

Jun.-Prof. Susanne Boll
Universität Oldenburg
Department für Informatik
Juniorprofessur Multimedia
und Internet-Technologien

Sabine Thieme
Ansgar Scherp
Oldenburger Forschungs- und
Entwicklungsinstitut für
Informatik-Werkzeuge und
Systeme (OFFIS)
Abt. Multimedia und Internet-
Informationsdienste

In Kooperation mit cewe color

Inhaltsverzeichnis

1 Einleitung **5**

2 Anforderungsdefinition **8**
2.1 Einleitung 9
2.1.1 Was ist eine Projektgruppe? 9
2.1.2 Was ist DVB? 10
2.1.3 Was ist MHP? 11
2.1.4 Projektgruppe Show_Me_MHP 12
2.1.5 Aufbau der Anforderungsdefinition 16
2.2 Systemeinsatz und -umgebung 17
2.2.1 Zielsystem 17
2.2.2 Entwicklungssystem 18
2.3 Nichtfunktionale Anforderungen 20
2.3.1 Personalisierung und Datenschutz 20
2.3.2 Antwortzeiten und Datenverkehr 20
2.3.3 Benutzungsoberfläche und Navigation 21
2.3.4 Dokumentation 21
2.4 Mediengalerie 23
2.4.1 Benutzungsschnittstelle 24
2.4.2 Anwendungsfälle 28
2.5 Tipp2Win 34
2.5.1 Benutzungsschnittstellen 35
2.5.2 Anwendungsfälle 38
2.6 Electronic Program Guide 45
2.6.1 Benutzungsschnittstelle 46
2.6.2 Anwendungsfälle 51
2.7 Abnahmekriterien 57
2.8 Vorgehensmodell und Zeitplan 58
Glossar 61
Literatur 65
Unterschriften 67

3 Zwischenbericht **68**
3.1 Kurzer überblick über die folgenden Kapitel 69
3.2 Projektplanung 69
3.2.1 Zeitplan 69
3.2.2 Vorgehensmodell 72
3.2.3 Werkzeugeinsatz 75
3.3 änderung an der Anforderungsdefinition 79
3.3.1 Allgemeines 79
3.3.2 LoginScreen (Screenshot) 80
3.3.3 PartnerLogin (Screenshot) 81
3.3.4 Thumbnails (Screenshot) 82
3.3.5 Vollbild (Screenshot) 83
3.3.6 RegisterScreen (Screenshot) 84
3.3.7 Ausblick (noch zu implementierende Oberflächen) . . . 84
3.4 Entwurf der Mediengalerie MeGa 85
3.4.1 Einleitung 85
3.4.2 Programmierrichtlinien 86
3.4.3 Architektur 87
3.4.4 Detailliertes statisches Modell 91
3.4.5 Detailliertes dynamisches Modell 113
3.4.6 Nutzungsvoraussetzungen 121
Literatur 122

4 Abschlussbericht **124**
4.1 Kurzer Überblick über die folgenden Kapitel 125
4.2 Überarbeiteter Entwurf der Mediengalerie MeGa 126
4.2.1 Einleitung 126
4.2.2 Programmierrichtlinien 127
4.2.3 Architektur 127
4.2.4 Detailliertes statisches Modell 132
4.2.5 Detailliertes dynamisches Modell 164
4.2.6 Nutzungsvoraussetzungen 170
4.3 Entwurf des Tippspiels Tipp2Win 172
4.3.1 Einleitung 172
4.3.2 Architektur 172
4.3.3 Detailliertes statisches Modell 175
4.3.4 Detailliertes dynamisches Modell 202
4.4 Gemeinsam genutzte Klassen (common-Paket) 207
4.4.1 Unterteilung in Komponenten/Schichten 207
4.4.2 IHelpScreen 208
4.4.3 IInputAssistance 208

4.4.4 IOSDInfo 209
4.5 Technisches Handbuch 210
4.5.1 Kompilieren des Quellcodes 210
4.5.2 Nutzung der MeGa auf der IRT Referenzimplementierung 214
4.5.3 Inbetriebnahme der Fernbedienung für die Referenzimplementierung 216
4.5.4 Nutzung der MeGa auf der Galaxis Set-Top-Box 218
4.5.5 Unterschiede zwischen der Referenzimplementierung und der Set-Top-Box von Galaxis 222
4.5.6 Vorhandene Fakes in der Mediengalerie MeGa 223
4.5.7 ResourceBundles 224
4.6 Handbuch Mediengalerie MeGa 225
4.6.1 Allgemeines 225
4.6.2 Beschreibung der einzelnen Bildschirme 226
4.7 Handbuch Tippspiel Tipp2Win 242
4.7.1 Allgemeines 242
4.7.2 Beschreibung der einzelnen Bildschirme 244
4.8 Mögliche Weiterentwicklungen 257
4.8.1 Mediengalerie 257
4.8.2 Tipp2Win 259
4.9 Erfahrungsbericht der Projektgruppe 260
4.9.1 Allgemeines 260
4.9.2 Freizeitaktivitäten und PG-Fahrt 261
4.9.3 Danksagung 261

Anhang **263**
A Gesamtglossar 263
B Seminararbeiten 271

Kapitel 1

Einleitung

Der Endbericht der Projektgruppe Show_Me_MHP enthält eine vollständige Zusammenstellung aller Dokumente und Ausarbeitungen, die im Laufe des einen Jahres der Projektgruppenarbeit erstellt wurden. Darunter fallen folgende Dokumente:

1. Anforderungsdefinition: Vorstellung von möglichen auf der MHP basierenden Applikationen und Erarbeitung eines Kataloges von Funktionalitäten und Anforderungen an diese potenziellen Anwendungen.

2. Zwischenbericht: Enthält eine Beschreibung des Werkzeugeinsatzes, Aktualisierung der Anforderungsdefinition und des Zeitplans sowie die Erläuterung des Vorgehensmodells.

3. Entwurf: Der Fokus der Projektgruppenarbeit lag zunächst auf der Entwicklung der Mediengalerie, da diese Anwendung auf Grund des Kooperationspartners CeWe Color höchste Priorität hatte. Daher wurde zu Beginn nur der Entwurf für die Mediengalerie fertig gestellt. Aus den durch die Entwicklungsarbeiten an der Mediengalerie gewonnenen Erfahrungen wurde klar, dass nur die Implementierung einer weiteren Applikation in Angriff genommen werden konnte. Augrund der Aktualität der EM2004 entschied sich die PG für die Entwicklung der Applikation Tipp2Win und gegen den EPG.

   - Mediengalerie: Vorstellung des Mediatiorprinzips, der Paketstruktur, der Programmierrichtlinien und des genaunen statischen und dynamischen Modells. Zusätzlich wurden die Eigenschaften und die Funktion des Common Paketes herausgearbeitet.
   - Tippspiel: Vorstellung der Client-Server Architektur, des Datenbankmodells sowie des groben statischen und dynamischen Modells.

4. Handbücher: In den Handbüchern zu den beiden Applikationen *MeGa und Tipp2Win* befinden sich Screenshots mit den entsprechenden Beschreibungen.

5. Abschlussbericht: In diesem sind die zuvor beschriebenen Dokumente *Überarbeiteter Entwurf der Mediengalerie MeGa*, *Entwurf des Tippspiels Tipp2Win*, die Handbücher für das Tippspiel und die Mediengalerie aufgeführt. Des Weiteren befindet sich im Abschlussbericht ein technisches Handbuch, eine Beschreibung der gemeinsam genutzten Klassen (common-Paket), mögliche Weiterentwicklungen für beide Applikationen sowie ein Bericht der PG Fahrt nach Berlin und der Projektgruppenarbeit.

6. Anhang: Im Anhang befinden sich die Seminararbeiten der Projektgruppenmitglieder zu den Einarbeitungsthemen in die MHP. Die Ausarbeitung von Christian Wegener *Personalisierung und Multimedia auf dem Fernseher* wird krankheitsbedingt nachgereicht. Im Folgenden werden die Themengebiete aufgelistet und kurz erklärt.

    - DVB und die MHP: Zusammenhang zwischen den DVB Empfangsstandards und der MHP
    - Die Multimedia Home Platform: Genaue Definition und Erläuterung der Eigenschaften bzw. Funktionalitäten der MHP
    - Multimedia on TV: Vergleich von Standards wie MHEG und OpenTV mit der MHP
    - Prozessmodelle zur Multimedia/Software-Entwicklung: Vorstellung verschiedener Modelle zur Softwareentwicklung
    - Marktforschung Hardware: Analyse der sich auf dem Markt befindlichen MHP fähigen Hardware
    - Marktforschung Software: Analyse der sich auf dem Markt befindlichen Software zur Entwicklung von auf der MHP basierenden Applikationen
    - Personalisierung und Multimedia auf dem Fernseher: Vorstellung von vorhandenen personalisierten Anwendungen
    - Geschäftsmodelle für Multimedia on TV / eCommerce: Vorstellung verschiedener Geschäftsmodelle
    - Sicherheit mit der MHP: Erläuterung von sicherheitsrelevanten Aspekten die MHP betreffend

Kapitel 2

Anforderungsdefinition

2.1 Einleitung

Thema dieses Dokumentes ist die Anforderungsdefinition der Projektgruppe Show_Me_MHP. In diesem Kapitel soll zu Beginn erläutert werden, was unter einer Projektgruppe zu verstehen ist. Danach soll auf die beiden Standards DVB und MHP eingegangen werden, die für diese Arbeit von besonderem Interesse sind. Im darauf folgenden Abschnitt 2.1.4 geht es um die Aufgabenstellung der Projektgruppe Show_Me_MHP. Es werden die Rahmenbedingungen aufgezeigt und die einzelnen Implementationsvorhaben beschrieben. Abschließend soll dann im Abschnitt 2.1.5 der weitere Aufbau der Anforderungsdefinition vorgestellt werden.

2.1.1 Was ist eine Projektgruppe?

Die Projektgruppe (PG) ist im Standard-Studienplan für den Diplomstudiengang Informatik dem 7. und 8. Semester zugeordnet. Sie besteht in der Regel aus sechs bis zwölf Teilnehmerinnen und Teilnehmern, die gemeinsam ein komplettes Projekt im Umfang von 4 Modulen (jeweils zwei pro Semester) durchführen. Eine Projektgruppe integriert mindestens drei Lehrformen: Ein Seminar, ein Fortgeschrittenenpraktikum und eine Studienarbeit. Ziel ist es, anhand eines gegebenen Problems die vollständige Entwicklung von der Problemanalyse bis hin zur Realisierung des Systems durchzuführen. Neben Methoden und Inhalten des Studienfaches erlernen die Studierenden berufstypische Arbeitsweisen wie folgende:

- Arbeiten im Team,
- Arbeitsteilung,
- Übernahme von Verantwortung,
- Aufbereitung von Inhalten,
- Zielorientiertes Argumentieren und
- Förderung von Präsentations- und Urteilsfähigkeit.

In zunehmendem Maße sollen die Fähigkeiten des selbstständigen Arbeitens, Arbeiten in einem Team und die Bewertung von Ideen und Produkten gefördert werden, welches die Projektgruppe ermöglicht [Bol04].

2.1.2 Was ist DVB?

DVB steht für Digital Video Broadcasting. Das sogenannte DVB-Projekt [DVB03], ein Konsortium von über 300 Programmanbietern, Geräteherstellern und anderen Organisationen, hat verschiedene Standards im Zusammenhang mit digitalem Fernsehen entwickelt.
DVB ist ein Standard zur Verbreitung von digitalem Fernsehen in einem MPEG-Datenstrom. Dabei sind verschiedene Übertragungswege möglich: Dies sind zum Beispiel die Verbreitung über Satellit (DVB-S), Kabel (DVB-C) oder auch über terrestrische Antennen (DVB-T). In diesem MPEG-2-Datenstrom finden neben den eigentlichen Fernsehinhalten auch Anwendungen und weitere Daten ihren Platz und können auf diesem Wege zum Zuschauer übertragen werden. Broadcasting bedeutet also, dass jeder Teilnehmer alle gesendeten Daten empfängt, die Daten werden also nicht an einen bestimmten Zuschauer gesendet. Durch eine automatische Fehlerkorrektur des Signals, die mit Hilfe von sogenannten Paritätsbits[1] durchgeführt wird, kann die Bild- und Tonqualität verbessert werden. Die erwähnte MPEG-2-Datenkomprimierung ermöglicht es, in einem ehemaligen analogen Programmplatz in den meisten Fällen vier digitale Programme übertragen zu können. Digitales Fernsehen bedeutet also sowohl eine bessere Bild- und Tonqualität als auch ein umfangreicheres Programmangebot und zusätzliche Datendienste.
Für den Empfang des digitalen Fernsehens ist neben einem Fernseher eine so genannte Set-Top-Box erforderlich, die auf den jeweiligen Übertragungsweg zugeschnitten ist. Für die in der Projektgruppe zu entwickelnden Anwendungen ist zudem eine Set-Top-Box erforderlich, die über eine spezielle Schnittstelle, die Multimedia Home Platform, sowie über einen Rückkanal verfügt. Über den Rückkanal, einer Verbindung zu beispielsweise einem Server, können zusätzliche Daten empfangen, aber vor allen Dingen auch gesendet werden (dies ist über Broadcasting nicht möglich). Der Begriff *Rückkanal* ist irreführend, da er eine Verbindung in nur eine Richtung, nämlich zurück, suggeriert. Tatsache ist aber, dass der Rückkanal ein bidirektionaler Kanal ist, über den Daten in beide Richtungen gesendet werden können. Der Rückkanal kann über Modem, ISDN oder auch DSL realisiert werden.
Zukünftig können also Anwendungen, wie auf einem PC, auch auf einer Set-Top-Box ausgeführt werden. Der Bildschirm wird dann als Monitor und die Fernbedienung als Eingabegerät verwendet. Die Interaktivität bzw. der Internet-Zugang wird durch den Rückkanal ermöglicht.

Die Abbildung 2.1 zeigt auf der linken Seite ein sogenanntes *Playout-Center*:

[1] Bits, die an den Datenstrom angehängt werden und mit denen die Daten auf Fehler überprüft werden können.

hier werden die Daten (Videodaten, Audiodaten, Anwendungen, etc.) durch einen Multiplexer zu einem MPEG-2-Datenstrom zusammengeführt und über die verschiedenen Übertragungswege per Broadcast gesendet.
Auf der rechten Seite sind die wichtigsten Empfangsmöglichkeiten dargestellt: hier gelangt der Datenstrom via Satelliten-Schüssel, Kabelnetz oder Antenne in die Set-Top-Box und kann für die Darstellung auf dem Fernseher verarbeitet werden.

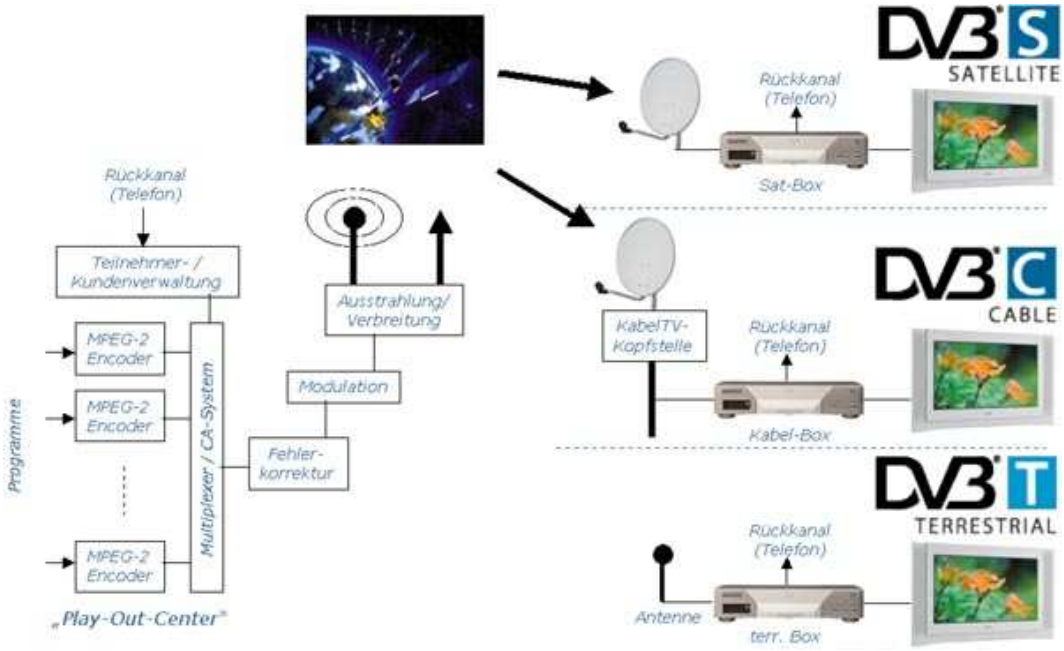

Abbildung 2.1: DVB-Schema

2.1.3 Was ist MHP?

Der Begriff MHP steht für *Multimedia Home Platform* [Mul03] und beschreibt eine offene und einheitliche Schnittstelle zur Entwicklung von multimedialen und interaktiven Diensten auf der Basis von DVB [DVB03][SC01]. Motivation bei der Festsetzung des Standards MHP war es, den in Zukunft immer umfangreicher werdenden Anforderungen des Digitalen Fernsehens gerecht zu werden. Besondere Aufmerksamkeit wurde dabei auf die Gestaltung interaktiver Inhalte gelegt, deswegen wurden unter anderem ein Rückkanal und die Nutzung des Internets durch die MHP spezifiziert.
Ein wichtiges Element innerhalb der MHP-Spezifikation [Eur01][ETS01] ist die hardwareunabhängige *Virtual Machine*, welche die in Java geschriebenen

und über den DVB-Stream gesendeten oder auch aus dem Internet geladenen Applikationen auf den unterschiedlichen Geräten ausführt. Die MHP vereint damit die in Abschnitt 2.1.2 erwähnten Vorteile des digitalen Fernsehens, die in einer verbesserten Bild- und Tonqualität bestehen, mit Interaktivität, die durch eine Anbindung an das Internet ermöglicht wird. Der Konsument profitiert von diesem Standard, da er mit nur einer MHP-tauglichen digitalen Set-Top-Box oder einem digitalen TV-Gerät mit integrierter MHP-Lösung alle multimedialen und interaktiven Anwendungen nutzen kann. Genauso ermöglicht er es unserer Projektgruppe und auch Unternehmen, die multimediale Inhalte oder Anwendungen anbieten wollen, dieses auf Basis eines Standards zu unternehmen und nicht für unterschiedliche Plattformen mehrere proprietäre Lösungen implementieren zu müssen.
Zuerst veröffentlicht wurde der MHP Standard in der Version 1.0 als Ergänzung zum bereits bestehenden sehr erfolgreichen DVB-Standard im Februar 2000. Im Juli 2001 wurde die MHP Version 1.1 eingeführt, durch sie wurden Zugriffsmöglichkeiten auf das Internet ergänzt. Unterstützt wird die Multimedia Home Platform mittlerweile von 300 DVB-Mitgliedern aus 37 Ländern, mit Unternehmen wie z. B. Philips, Nokia, Panasonic und der Deutschen Telekom. Mit der internationalen Markteinführung der Multimedia Home Platform befassen sich die MHP MarCom Group und die MHP Implementers Group in Zusammenarbeit mit dem DVB-Konsortium. Nähere Informationen lassen sich z. B. in [Eva98], [Sed01] und [FKT00] finden.

Die Abbildung 2.2 zeigt den Zusammenhang von DVB und MHP: das Playout-Center und die Verteilnetze sorgen dabei für den Datentransport zum Zuschauer. Im unteren Teil ist die Set-Top-Box mit MHP und einem Rückkanal zu sehen. Ein Server ist im Zusammenspiel mit einem sogenannten Objekt-Karussell in der Lage, dem Datenstrom Anwendungen und Daten hinzuzufügen. Die Set-Top-Box kann diese durch Broadcast gesendeten Daten empfangen und über den Rückkanal mit einem Server bidirektional kommunizieren. Damit können beispielsweise Wünsche oder Befehle an den Server übermittelt werden, aber auch ein Download von Daten oder Anwendungen vom Server ist möglich. In das Objekt-Karussell können die zu verbreitenden Daten eingestellt werden, diese werden dann sofort oder z. B. in regelmäßigen Abständen dem Datenstrom hinzugefügt.

2.1.4 Projektgruppe Show_Me_MHP

Die Projektgruppe Show_Me_MHP wird vom Department für Informatik der Fakultät 2 der Universität Oldenburg veranstaltet. Sie setzt sich aus neun Studierenden des Studiengangs Informatik zusammen und entspricht

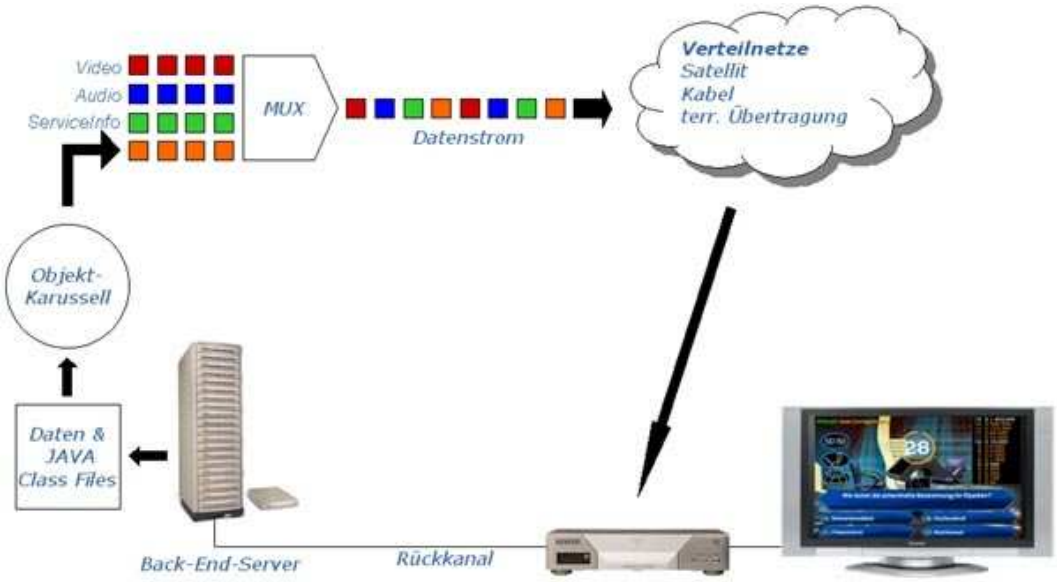

Abbildung 2.2: MHP-Schema

damit den Vorgaben des Fachbereiches. Die Aufgabe der Studierenden ist es, im Rahmen der Projektgruppe, ein auf dem MHP Standard basierendes Softwareprojekt zu realisieren. Als konkrete Aufgabe steht zunächst die Einarbeitung in den MHP Standard und die Analyse und Evaluierung der Möglichkeiten dieses Standards an. Dies geschieht durch eine Reihe von Vorträgen innerhalb der Projektgruppe, die jedem Mitglied ein fundiertes Wissen über die gesamte Thematik vermitteln sollen:

- DVB-T/S/C und die MHP,
- Die Multimedia Home Platform (MHP),
- Multimedia on TV,
- Prozessmodelle zur Multimedia/Software-Entwicklung,
- Marktforschungsanalyse Hardware,
- Marktforschung Software,
- Personalisierung und Multimedia auf dem Fernseher,
- Geschäftsmodelle für Multimedia on TV/eCommerce und
- Sicherheit mit der MHP.

Jedes Mitglied der Projektgruppe hält einen Vortrag über ein Themengebiet und erstellt eine 10-25 Seiten umfassende schriftliche Ausarbeitung, die alle themenspezifischen Details beinhaltet. Anhand der Vorträge lassen sich dann die hard -und softwaretechnischen *Rahmenbedingungen* des Projektes abstecken:

2.1.4.1 Rahmenbedingungen

Es sind verschiedene Rahmenbedingungen einzuhalten, zu denen zum einen die hard- und softwaretechnischen, zum anderen aber auch die aus der Aufgabenstellung resultierenden gehören. Die *softwaretechnischen* Rahmenbedingungen sind durch den MHP Standard fest vorgegeben. Bei allen MHP Standards handelt es sich um ein Stück Software, das auf einer Java Klassenbibliothek basiert. Damit Applikationen kompiliert und getestet werden können, müssen auf der *Entwicklungsplattform* (z.B. LINUX Workstation) und auf der *Einsatzplattform* (z. B. MHP-TV) das SDK (Software Development Kit) der Firma SUN und die entsprechenden MHP Klassenbibliotheken installiert sein. Das SDK enthält sowohl die nötigen Klassen, um Java Dateien zu kompilieren, als auch die JRE (Java Runtime Environment), die die Java Virtual Maschine bildet und die Ausführung von Java Classfiles ermöglicht. Beispiele für Entwicklungsumgebungen, die speziell auf den MHP Standard zugeschnitten sind, sind die Referenzimplentierung des IRT (Institut für Rundfunktechnik) oder eines der Developer Packages von Panasonic.
Bei den *hardwaretechnischen* Rahmenbedingungen muss zwischen der Entwicklungsplattform und der Einsatzplattform unterschieden werden. Für die Entwicklung kommen Workstations (PCs mit üblicher Peripherie) in Betracht, die Windows oder Linux basiert sein können. Als Endgeräte, auf denen die Applikationen letztendlich laufen sollen, dienen entweder Set-Top-Boxen (für analoge Fernseher), MHP-Fernseher oder mobile Lösungen wie PCMCIA-Karten für Notebooks.
Auch die *Aufgabenstellung* der Projektgruppe, die Realisierung von zwei bis drei auf dem MHP Standard beruhenden Anwendungen, muss als Rahmenbedingung betrachtet werden, da sie maßgebend bei der Zeitplanung und Aufgabeneinteilung ist. Eine weitere Rahmenbedingung ist die Kooperation mit CeWe Color. Durch diese Zusammenarbeit ist die Entwicklung der Mediengalerie vorgegeben.

2.1.4.2 Anwendungsfelder und konkrete Implementationsvorhaben

Beispiele für Anwendungsfelder sind EPGs (elektronische Programmführer) oder interaktive Quizshows [Fp03]. Die von uns geplanten Anwendungen resultieren zum einen aus den Rahmenbedingungen, die in Abschnitt 2.1.4.1 beschrieben wurden, und zum anderen aus eigenen Überlegungen. Konkret denken wir an die Implementierung einer *personalisierten Mediengalerie*, eines *EPGs* und eines *Fußball Tippspieles*. Grundfunktionalitäten des Tippspieles sollen dabei das Tippen der einzelnen Gruppenspiele bzw. Finals, ein individuelles Benutzerprofil und eine Ranglistenfunktion sein. Die Entwicklung eines weiteren EPGs könnte auf den ersten Blick überflüssig erscheinen, jedoch sind bisherige EPGs auf ein Programm beschränkt. Funktionen, die eine Anpassung des geplanten Programms an individuelle Bedürfnisse bieten und die alle Programme berücksichtigen, sind Neuheiten. Die Implementierung der personalisierten Mediengalerie geschieht in Zusammenarbeit mit CeWe Color. Die Mediengalerie soll Funktionen bieten, die das Anschauen, Bearbeiten und Bestellen von Fotos bequem vom Wohnzimmer aus ermöglichen. Wichtig ist hier vor allem die Einhaltung der Schnittstellen, die vom Kooperationspartner vorgegeben werden, da z. B. die Archivierung von Fotos auf hauseigenen Servern von CeWe realisiert werden soll.
Für detailliertere Informationen zu den drei Anwendungen soll an dieser Stelle auf die Kapitel 2.4, 2.5 und 2.6 verwiesen werden.

2.1.4.3 Gemeinsamkeiten der Anwendungsfelder

Bei einer oberflächlichen Betrachtung der geplanten Implementationsvorhaben erscheinen sie vom Aufbau und Konzeption grundverschieden. Es existieren aber, besonders im Hinblick auf die Personalisierung, einige Gemeinsamkeiten. Eine Arbeitsersparnis und Kapselung von Funktionen planen wir durch eine gemeinsame Datenhaltung der personalisierten Benutzerdaten zu erreichen. Die Daten sollen auf einem gemeinsamen Datenbankserver abgespeichert und Anfragen sowie Einträge von einer gemeinsamen Datenbankklasse vorgenommen werden. Weitere Möglichkeiten, das Projekt generisch und anpassbar zu halten, sind die Verwendung von gemeinsamen Klassen für Menüelemente (z. B. Buttons) oder Soundeffekte. Da alle Oberflächen für die MHP auf die HAVI[2] zurückgreifen, ist es sicherlich auch in dem Bereich möglich, gemeinsame Klassen zu verwenden. Sollte entschieden werden, GUI-Man[3] bei der Erstellung der Oberflächen zu verwenden, ist es wahrscheinlich

[2]Home Audio Video Interoperability
[3]Entwicklung der Technischen Universität Braunschweig

auch hier möglich, Code wiederzuverwenden. Genaueres hierzu wird sich aber bei der Erstellung des Entwurfes herausstellen.

2.1.5 Aufbau der Anforderungsdefinition

Die hier vorgelegte Anforderungsdefinition stellt die Arbeitsgrundlage für das weitere Vorgehen der Projektgruppe Show_Me_MHP dar. Dieses erste Kapitel bietet bereits einige grundlegende Informationen über das Vorhaben der Gruppe. Im Kapitel 2.2 wird zunächst das Zielsystem beschrieben, auf dem die Anwendungen später zum Einsatz kommen sollen, und anschließend das Entwicklungssystem, mit dem die vorgestellten Anwendungen entwickelt werden.
Da sich die nichtfunktionalen Anforderungen der Anwendungen kaum unterscheiden, sind diese in einem eigenen Kapitel 2.3 in dieser Anforderungsdefinition untergebracht.
In den anschließenden drei Kapiteln erfolgen Spezialisierungen der einzelnen Applikationen. Das 2.4., das 2.5. und das 2.6. Kapitel beschreiben im wesentlichen die funktionalen Anforderungen der Mediengalerie, dem Tippspiel und dem EPG, welche jeweils anhand von möglichen Benutzungsschnittstellen und Anwendungsfällen erläutert werden. Jede Anwendung ist dabei jeweils einem Kapitel zugeordnet.
Im 2.7. Kapitel sind die Abnahmekriterien zu finden. Diese beschreiben, welche Anforderungen für ein erfolgreiches Abschneiden der Projektgruppe erfüllt werden müssen.
Das abschließende 2.8. Kapitel befasst sich mit dem weiteren Vorgehen, wobei ein geeignetes Vorgehensmodell mit einem dazu passenden Zeitplan vorgestellt wird.

2.2 Systemeinsatz und -umgebung

In diesem Abschnitt soll auf die Hard- und Software eingegangen werden, in deren Umgebung unsere Anwendungen zum Einsatz kommen sollen. Dabei muss unterschieden werden zwischen *dem Entwicklungssystem*, also der Umgebung, in der die Anwendungen entwickelt werden und dem *Zielsystem* der Umgebung, in der die Anwendungen später als fertiges Produkt Einsatz finden sollen.

2.2.1 Zielsystem

Das Zielsystem basiert auf einer Client-Server Architektur. Allen drei Anwendungen ist gemeinsam, dass sie einen Datenbankserver benötigen, um die anfallenden Datenmengen zentral zu speichern. Bei allen Anwendungen sind das Benutzerdaten, die zur Wiedererkennung und zum Einloggen in das jeweilige Programm verwendet werden. Die Daten des Tippspieles bestehen außerdem aus den abgegebenen Tipps der einzelnen Benutzer, die auf einem Server abgespeichert und ausgewertet werden müssen. Bei dem EPG ist der Server dafür verantwortlich, Informationen von Dienstleistern wie z. B. TV-Movie zu sammeln, um diese dann in geeigneter Form den Clients wieder zur Verfügung zu stellen. Auf dem Server der Mediengalerie müssen die Bilddaten gespeichert werden. Nur mit einem zentralen Server ist es in diesem Fall möglich, Bilder auch anderen Nutzern zur Verfügung zu stellen. In dem Spezialfall *Mediengalerie* muss zusätzlich darauf geachtet werden, dass die Schnittstelle zu diesem Server mit der XTCI [4] Schnittstelle zum Server des Kooperationspartners CeWe Color, die auf XML basiert, konform ist.

Für die Verbindung von Server und Client gibt es den Broadcast und den sogenannten Rückkanal, der im Grunde aber ein bidirektionaler Kanal ist, d h. er kann in beide Richtungen genutzt werden. Für das Tippspiel ist die Nutzung des Rückkanals unverzichtbar, um die von den Benutzern abgegebenen Tipps an den Server übermitteln zu können. Für den Weg vom Server zum Client kann wahlweise entweder der Rückkanal oder der Broadcast verwendet werden. Im Fall des EPG ist dagegen auch eine Variante vorstellbar, in der kein Rückkanal verwendet wird. Hier gibt es dann nur eine Verbindung in eine Richtung, nämlich vom Server zum Client. Über den Broadcast sollen dann, in gewissen Zeitabständen, die vom Server gesammelten Informationen an die Clients übertragen werden. Allerdings muss bei einer solchen Variante auf eine zentrale Benutzerverwaltung verzichtet werden. Für eine

[4] external Client Interface

Benutzerverwaltung, die zentral auf einem eigens dafür vorgesehenen Server abgewickelt werden soll, ist der Einsatz eines Rückkanals unumgänglich. Für die Mediengalerie gilt das gleiche wie für das Tippspiel. Auch hier muss der Rückkanal eingesetzt werden, über den der Benutzer sich am Server anmeldet, um Zugriff auf seine Bilder zu erhalten. Der Broadcast kann hier dagegen nur wenig sinnvoll eingesetzt werden. Die Daten, die sich auf dem Server befinden, sind im Unterschied zu den beiden anderen Anwendungen zu kundenspezifisch, als dass sie an eine breite Masse gesendet werden sollten. Lediglich zur Verbreitung der Anwendung kann der Broadcast verwendet werden, wobei immer bedacht werden muss, dass eine Benutzung des Broadcast für den Anbieter, der sein Produkt über diesen verteilen möchte, sehr kostspielig ist.

Der Client besteht also in allen drei Fällen aus einem MHP- fähigen Gerät in jeder Form, sei es eine Set-Top-Box, ein Fernseher oder eine PCMCIA-Karte z. B. im Notebook. Damit die drei Anwendungen bei ihrem Start nicht jedes Mal erst heruntergeladen werden müssen, ist eine Festplatte im Gerät sinnvoll, nicht aber zwingend notwendig. Damit es überhaupt möglich ist, heruntergeladene Anwendungen permanent auf dem Client zu speichern, muss außerdem eine MHP in der Version 1.1 vorliegen.

Nähere Informationen zum Zusammenhang von MHP und DVB und der Architektur, die diese beiden Standards verbindet, können im Kapitel 2.1.2 und 2.1.3 nachgelesen werden.

2.2.2 Entwicklungssystem

Zum Entwicklungssystem gehören die Hard- und Software, die zum einen zum Erstellen und Implementieren der Programme benutzt wird, zum anderen aber auch als Testsystem fungieren soll.

Da unsere Anwendungen plattformunabhängige Lösungen im Bereich des digitalen Fernsehens sein sollen, ist eine Rahmenbedingung, die uns bei der Implementierung vorgegeben ist, die Verwendung der auf Java basierenden MHP-API. MHP ist vom European Telecommunications Standards Institute (ETSI) in Europa als Standard erklärt worden, siehe dazu Abschnitt 2.1.3, dies gewährleistet uns das gewünschte hohe Maß an Plattform- und Hardwareunabhängigkeit. Unsere Anwendungen sollten somit auf allen MHP fähigen Geräten ausgeführt werden können. Zum Implementieren der Programme steht uns eine Referenzimplementierung [Bre01] vom IRT, siehe dazu Abschnitt 2.1.4.1, zur Verfügung, welche die benötigten Klassen und Biblio-

theken enthält, mit denen sich die Funktionen der MHP ansprechen lassen. Bei der Entwicklung der graphischen Benutzungsoberflächen gibt es die Möglichkeit auf den Oberflächeneditor GUIMan, eine Entwicklung der Technischen Universität Braunschweig, zurückzugreifen. GUIMan ermöglicht es, Benutzungsoberflächen unter Verwendung von XML zu generieren. Inwiefern diese damit erstellbaren Oberflächen aber unseren Anforderungen genügen, muss sich bei der späteren Entwicklung eines Prototyps noch herausstellen und kann an dieser Stelle deshalb noch nicht beantwortet werden. Zusätzlich soll eine geeignete Entwicklungsumgebung eingesetzt werden, die uns Hilfestellung bei der Programmierung leisten kann. Wir haben uns in diesem Punkt für TogetherJ von Borland entschieden, da dies ein komfortables und für Studenten frei verfügbares Entwicklungstool ist. Um unsere Anwendungen zu testen genügt zunächst ein Rechner mit installierter Java Virtual Machine (JVM). Für weitere Tests sollte dann aber eine MHP fähige Set-Top-Box verwendet werden, um die korrekte Funktion der Anwendungen auch in einer realitätsnahen Umgebung zu untersuchen. Weiterhin muss beachtet werden, dass in der Entwicklungsphase noch kein Broadcast vorhanden ist, diesen gilt es also auf geeignete Weise zu simulieren. Wie diese Simulation im Detail aussehen soll, muss erst untersucht werden.

2.3 Nichtfunktionale Anforderungen

Nachfolgend werden nun die nichtfunktionalen Anforderungen beschrieben. Dies sind Anforderungen, die sich nicht wie die funktionalen Anforderungen direkt in Funktionen ausdrücken lassen, sondern etwas abstrakter und allgemeiner beschrieben werden. Sie gelten daher auch für alle drei Anwendungen gleichermaßen. Vorweg soll noch einmal bemerkt werden, dass das Projekt von der Juniorprofessur Multimedia und Internet-Technologien veranstaltet wird. Daher sollen in den Anwendungen auf jeden Fall die Aspekte Personalisierung, Multimedia und Interaktivität berücksichtigt werden.

2.3.1 Personalisierung und Datenschutz

Für die Personalisierung müssen Daten abgespeichert werden. Dies kann entweder lokal auf der Set-Top-Box oder auf einem zentralen Server geschehen. Eine lokale Speicherung würde durch sogenannte Cookies erfolgen. Dabei speichern die Applikationen Daten auf der Set-Top-Box ab, welche beim nächsten Start der Applikation wieder abgerufen werden können. Die Personalisierung erfolgt dabei für ein Gerät, denn jedes Gerät hat seinen eigenen Speicher für Cookies. Einem Benutzer, der sein Gerät wechselt, würden die Daten des anderen Gerätes negativerweise nicht zur Verfügung stehen. Nachteilig könnte außerdem die Beschränkung der Datenmenge sein. Der Vorteil einer lokalen Speicherung liegt darin, dass keine Verbindung zu einem Server aufgebaut werden muss. Weil jedoch für die Anwendungen der Projektgruppe eine Verbindung mit einem Server ohnehin unvermeidbar ist, sollen die Personalisiereungsdaten zentral auf einer Datenbank gespeichert werden. Dabei ist zu beachten, dass Personalisierungsdaten häufig persönliche und sensible Daten wie Zugangsdaten oder Bankverbindungen enthalten können und die Möglichkeit einer verschlüsselten Übertragung daher in Betracht gezogen werden sollte.

2.3.2 Antwortzeiten und Datenverkehr

Für den Datenverkehr der Set-Top-Box kann sowohl der Broadcast als auch der (bidirektionale) Rückkanal genutzt werden. Welche Daten auf welchem Übertragungsweg gesendet werden sollen, muss jeweils geeignet entschieden werden. Sollen die Applikationen über den Broadcast ausgestrahlt werden, hat das den Vorteil, dass sie von allen Anwendern gleichzeitig mit einer hohen Bandbreite empfangen werden können. Damit auch möglichst viele Personen dieses Programm herunterladen und der Broadcast somit optimal genutzt wird, ist es in diesem Fall sinnvoll, die Fernsehzuschauer z. B. in einer Wer-

bung über die Ausstrahlung der Applikation zu informieren. Zu beachten ist, dass die Daten im Broadcast, siehe dazu Abschnitt 2.1.3, nur etwa alle fünf bis zehn Minuten ausgestrahlt werden. Aufgrund dieser langen Reaktionszeit eignet sich der Broadcast weniger für Daten, die schnell zur Verfügung stehen müssen. Über den Rückkanal sollten die Daten gesendet werden, die nur für wenige oder gar einen Anwender bestimmt sind und schnell zur Verfügung stehen sollen.

2.3.3 Benutzungsoberfläche und Navigation

Gerade bei Anwendungen für die MHP, die in der Regel auf dem heimischen Fernseher im Wohnzimmer ablaufen, ist auf eine einfache Bedienung zu achten, wie man sie vom Fernseher z. B. beim Videotext kennt. Für diesen Zweck soll eine Graphische Benutzungsoberfläche (GUI) realisiert werden, die intuitiv und logisch gestaltet ist. Dabei muss berücksichtigt werden, dass in den meisten Fällen keine Tastatur oder Maus vorhanden ist. Die Bedienung muss also möglichst über standardisierte Tasten erfolgen können, die auf jeder Fernbedienung einer MHP-fähigen Set-Top-Box vorhanden sind.
Ebenso wichtig in diesem Bereich ist das Fehlerverhalten der Anwendungen. Die Fehlerakzeptanz der Fernsehzuschauer, die eine Anwendung im laufenden Fernsehbetrieb benutzen, ist überaus gering einzuschätzen. Die Anwendungen müssen daher genügend getestet werden, um eine nahezu 100-prozentige Stabilität zu gewährleisten. Im Fall, dass dennoch ein Fehler auftreten sollte, darf dieser auf keinen Fall die restliche Funktion der MHP beeinträchtigen oder das laufende Fernsehprogramm unterbrechen.

2.3.4 Dokumentation

Die Dokumentation für die Anwendungen wird für die Endnutzer in Form eines Handbuches erhältlich sein. Das Handbuch wird dafür in einer druckfähigen Variante, als PDF- und PS-Dokument, auf der Homepage unserer Projektgruppe veröffentlicht, so dass jeder Nutzer freien Zugriff auf das Handbuch hat und die Möglichkeit besitzt, dieses herunterzuladen oder auszudrucken. In dem Handbuch werden zur Illustration einige Abbildungen zu den implementierten Anwendungen zu finden sein. Das Handbuch soll die folgende Struktur besitzen, um den Benutzern einen einfacheren Einstieg in die Bedienung des Systems zu bieten:

Systemvoraussetzungen: Hier werden die Systemvoraussetzungen sowohl für die Client- als auch für die Serverseite in Bezug auf Soft- und Hard-

ware aufgelistet. Hierbei ist festzuhalten, dass sich die Erklärung auf der Serverseite an den Administrator richtet und die des Client an den Endnutzer.

Programminstallation und Programmstart: Für den Administrator finden sich in diesem Teil des Dokumentes Informationen zum Einrichten des Serversystems inklusive der Datenbank.

Bedienungsanleitung: Dieses Kapitel soll dem Nutzer einen leichteren Einstieg in die Bedienung der Programme ermöglichen und enthält sämtliche Funktionsbeschreibungen der Anwendungen. Dazu werden unter anderem Screenshots zu sehen sein, um die einzelnen Funktionen dem Benutzer näher zu bringen.

Fehlermeldungen: Da Fehlermeldungen nicht ausgeschlossen werden können, soll dem Anwender die Möglichkeit gegeben werden, auf diese angemessen zu reagieren und sie eigenständig zu beheben. Dazu wird an dieser Stelle eine Auflistung der möglichen Fehlermeldungen und eine passende Beschreibung zur Lösung dieses Fehlers in tabellarischer Form zu finden sein.

2.4 Mediengalerie

In diesem und den beiden folgenden Kapiteln 2.5 und 2.6 sollen nun die drei Anwendungen, von denen im Rahmen dieser Projektgruppe auf jeden Fall die Mediengalerie und mindestens eine der anderen beiden implementiert werden soll, genauer vorgestellt werden. Da sich alle drei Anwendungen in ihren funktionalen Anforderungen und den Benutzungsschnittstellen unterscheiden, sollen diese Punkte für jedes Modul getrennt behandelt werden. Im Folgenden soll in jedem dieser drei Kapitel einleitend eine genauere Beschreibung des Moduls erfolgen. Danach wird jeweils auf die Benutzungsschnittstellen und abschließend auf die funktionalen Anforderungen, die an die jeweilige Anwendung gestellt werden, eingegangen. Als erstes wird nun die Mediengalerie beschrieben, wobei einige gute Ideen aus [Ste03] und [Sch03] mit eingegangen sind.

Wie in vielen anderen Bereichen hat auch in der Fotografie die Digitalisierung Einzug gehalten. Bilder und Fotos liegen nicht mehr nur in Papierform sondern zunehmend auch digital vor. Bisher ist es möglich, diese digitalen Bilder auf dem Computer anzuschauen. Durch die Einführung des digitalen Fernsehens in Verbindung mit der MHP ergeben sich nun wiederum ganz neue Möglichkeiten.
So soll es in Zukunft auch möglich sein, diese Bilder ganz bequem im Wohnzimmer vor dem heimischen Fernseher anzuschauen. Durch einen Rückkanal, der in die MHP-fähige Set-Top-Box integriert ist, und den dadurch möglichen Zugriff auf das Internet wird es ermöglicht, die eigenen Fotos, Freunden und Verwandten in aller Welt zur Verfügung zu stellen.
Dies zu verwirklichen ist die Intention der Mediengalerie. Dazu stelle man sich folgendes beispielhafte Szenario vor:

Manfred Mustermann gerade aus einem Südsee Urlaub heimgekommen, ist zu Gast bei seinen Eltern. Die gesamte Familie sitzt versammelt im Wohnzimmer vor dem Fernseher und unterhält sich. Alle Anwesenden wollen natürlich interessiert wissen, wie der Urlaub war. Da fällt Manfred ein, dass er die Fotos ja schon längst über die MHP-Box auf einen Server im Internet hochgeladen hat. Das einzige, was er nun noch tun muss, ist den Fernseher anzuschalten, sich mit Hilfe der MHP zu verbinden und schon kann er allen Anwesenden stolz seine Fotos zeigen.
Plötzlich klingelt das Telefon, Manfreds Schwester ist am Apparat, auch sie erkundigt sich nach dem Urlaub und kann es kaum noch erwarten, endlich die Fotos zu sehen. Manfred schlägt ihr einfach vor doch mal ihren Fernseher anzumachen. Er übermittelt ihr den Zugang zum Server und schon kann auch

sie sich die Fotos anschauen, da Manfred in der Zwischenzeit ihr Zugriffsrecht auf die Fotos eingestellt hat.
Durch einen Firmenpartner wie CeWe Color ist es der Schwester zudem auch möglich gleich Abzüge von den gewünschten Bildern zu bestellen, die sie sich bequem per Post nach Hause schicken lassen kann.

Die Mediengalerie ist also eine Weiterentwicklung der digitalen Fotoarchive und wird für die MHP entwickelt. Sie unterstützt die herkömmlichen Funktionen eines Fotoarchives, zu denen die Möglichkeit gehört, Fotos zu verwalten, indem sie z. B. zu Alben zusammengefasst werden können und Fotos zu betrachten, indem diese wahlweise in einer übersichtlichen Thumbnailansicht oder in der Vollbildansicht angezeigt werden können. Als Weiterentwicklung ist sie aber um die oben genannten Funktionen erweitert, also um ein Foto-Sharing, sowie die Möglichkeit, die digitalen Fotos direkt bei einem Online-PhotoService, wie dem von CeWe Color, entwickeln und dann zuschicken zu lassen.

Im nächsten Abschnitt 2.4.1 soll durch die Benutzungsschnittstellen ein erster Eindruck von der Benutzerführung und der Gestaltung der Oberflächen gegeben werden. Im darauf folgenden Abschnitt 2.4.2 wird dann detaillierter auf die Funktionalität der Anwendung eingegangen.

2.4.1 Benutzungsschnittstelle

Die Benutzungsschnittstelle wird aus vier Fenstern bestehen. Diese sind das Login-, das Partnerwahl-, das Thumbnail- und das Vollbildfenster. Die Fenster sind hier in der Reihenfolge angegeben, in der sie dem Benutzer später im laufenden Programm auch in Erscheinung treten werden.

2.4.1.1 Login

Sobald die Anwendung Mediengalerie im Elektronischen Programmführer ausgewählt und gestartet wurde, ist das erste, was der Benutzer zu Gesicht bekommen soll, der Anmeldebildschirm, (Abbildung 2.3) auch Loginfenster genannt. Dieser Anmeldebildschirm soll kundenneutral gestaltet sein. In diesem Fenster werden sich im wesentlichen zwei Textfelder und eine Checkbox befinden. Die beiden Textfelder dienen zum Einloggen und Identifizieren des Benutzers. In das Feld mit der Beschriftung *Email* muss vom Benutzer die entsprechende E-Mail eingetragen werden, mit der er bei einem der CeWe-Kunden (z. B. Schlecker oder Fotopoint) registriert ist. In dem Feld *Password* muss das entsprechende Passwort eingegeben werden. Durch das Markieren

der Checkbox stimmt der Benutzer zu, dass seine Logindaten der beiden Textfelder gespeichert werden. Bei einer erneuten Sitzung werden diese Kenndaten dann automatisch eingetragen. Es ist auch möglich, diesen Bildschirm noch um einen *neue Anmeldung*-Button zu erweitern. Für Anwender, die noch bei keinem der CeWe-Kunden angemeldet sein, soll es auf die Weise dann möglich sein ein neues Benutzerkonto zu eröffnen, indem zunächst eine gültige E-Mail Adresse, ein Passwort und ein CeWe-Kunde ausgewählt wird. Ob es in diesem Fall auch möglich sein wird, ein kundenneutrales[5] Konto zu eröffnen, muss noch geklärt werden, die Möglichkeit die Anwendung in diese Richtung zu erweitern wird bei der Implementierung aber berücksichtigt. Zusätzlich soll im linken Bereich noch eine Tastaturschablone angegeben werden, auf der für den Benutzer zu sehen ist, wie er über die Tastatur seiner Fernbedienung den entsprechenden Text eingeben kann. Dies könnte, vergleichbar mit dem Verfahren bei Handys, dadurch geschehen, dass eine Taste auf der Fernbedienung mit mehreren Buchstaben belegt ist. Durch mehrmaliges Tippen der selben Taste, kann dann durch die verschiedene Buchstaben gewechselt werden. Ob dies aber in der Form technisch durchführbar ist, muss noch genauer untersucht werden. Sind die Angaben korrekt, gelangt der Benutzer durch Betätigung des *OK* Buttons[6] zum nächsten Fenster.

2.4.1.2 CeWe Color Partner wählen

Im Fall, dass der Benutzer mit der im Loginfenster eingetragenen E-Mail-Adresse bei mehreren Kunden registriert ist, wird in diesem Bildschirm (Abbildung 2.4) eine Wahl des Benutzers für einen der Kunden verlangt, mit dessen Archiv er sich verbinden möchte. Über die Fernbedienung kann die Auswahl erfolgen. Durch Betätigung des *OK* Buttons wird wiederum das nächste Fenster geöffnet.

2.4.1.3 Thumbnail Ansicht

Ist der Benutzer dann schließlich mit einem Archiv eines Kunden, in diesem Beispiel Schlecker, verbunden, wird die Thumbnailansicht geöffnet, siehe dazu Abbildung 2.5. In dieser Ansicht wird der Inhalt eines ausgewählten Albums in verkleinerter Form auf dem Bildschirm dargestellt. In der rechten oberen Ecke des Bildschirms soll ein Logo des Kunden abgebildet sein, wodurch die Anwendung zusätzlich einen leichten kundenspezifischen Aspekt

[5]Kundenneutral bedeutet: ohne explizit einen CeWe-Kunden, wie z. B. Schlecker, auszuwählen.

[6]Die Ok und Cancel Buttons sind in diesen prototypischen Screenshots noch nicht zu sehen, werden im fertigen Programm aber vorhanden sein.

Abbildung 2.3: Einloggen MG

Abbildung 2.4: CeWe Color Partner wählen

erhält. Am unteren Ende des Bildschirms sind drei Buttons zu sehen. Die Buttons *Zurück* und *Vor* dienen zum Navigieren in diesen Thumbnails, mit ihnen kann jeweils ein Bild vor- und zurückgesprungen werden. Durch Betätigung des *Menü* Buttons erscheint zum einen das Menü und die Albenansicht auf der linken Seite. Unter *Alben* werden die vorhandenen Alben und das derzeit Ausgewählte angezeigt. In dem Menü sollen dem Benutzer verschiedene Funktionen zur Verfügung gestellt werden. Dies können zum einen Funktionen sein wie *Dieses Foto bestellen*, die sich nur auf das gerade ausgewählte Foto beziehen, aber auch Funktionen, die unabhängig von einem bestimmten Foto sind, z. B. *Bestellung ausführen*. Dieses Menü und die Alben sollen aber beliebig ein- und ausblendbar sein, um dem Benutzer eine uneingeschränkte Sicht auf die Fotos zu gewährleisten. Sobald im Menü die Funktion *Vollbild* benutzt wird, gelangt der Benutzer zur Vollbildansicht.

Abbildung 2.5: Ansicht der Thumbnails

2.4.1.4 Vollbild Ansicht

In der Vollbildansicht (Abbildung 2.6) wird dem Benutzer das zuvor in der Thumbnailansicht gewählte Foto in Vollbild angezeigt. Diese Ansicht ähnelt in ihrer Gestaltung bewusst der Thumbnailansicht. Dem Anwender soll dadurch die Benutzung dieser Anwendung so leicht wie möglich gemacht werden. Die einzelnen Funktionen im Menü stimmen zu einem großen Teil auch

mit denen in der Thumbnailansicht überein. Die Funktion *zoomen* aber, die es ermöglicht das angezeigte Bild vergrößert darzustellen, wird nur in der Vollbildansicht verfügbar sein. Eine solche Funktion ist in der Thumbnailansicht unnötig. Durch die Buttons *Zurück* und *Vor* lässt sich auch in dieser Ansicht zwischen den einzelnen Bildern des Albums vor- und zurückschalten. Durch diese Gestaltung ist es dem Benutzer möglich, in beiden Ansichten nahezu alle Funktionen durchzuführen. Somit ist also kein lästiges Hin- und Herspringen zwischen verschiedenen Ansichten notwendig, um z. B. ein Bild in den Warenkorb zu legen.

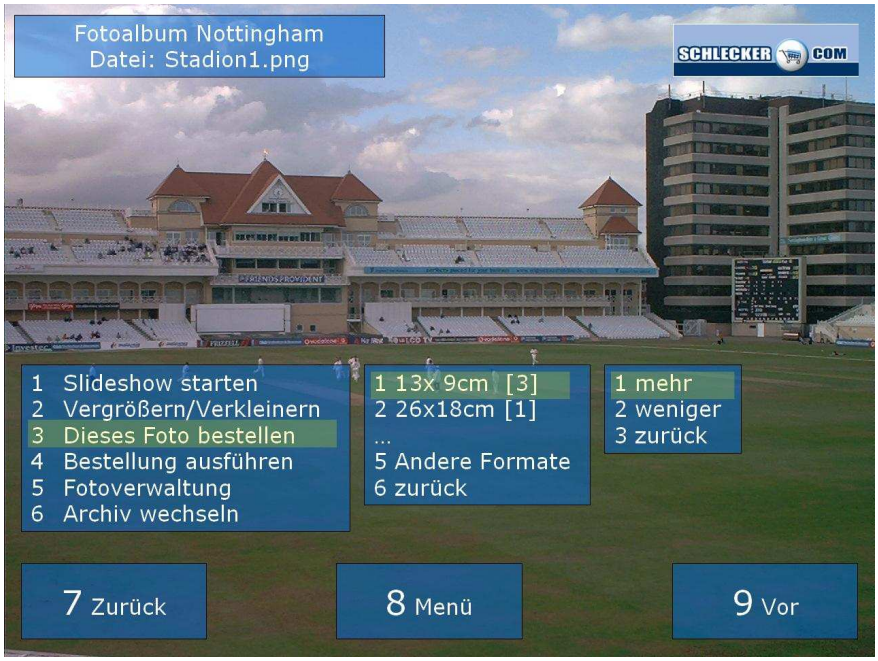

Abbildung 2.6: Ansicht des Vollbildes

2.4.2 Anwendungsfälle

Die hier im Folgenden aufgeführten Anwendungsfälle sind in die Bereiche Benutzerverwaltung, Freundeverwaltung, Albumverwaltung und Bildverwaltung unterteilt. Jeder Bereich soll dabei durch ein Anwendungsfalldiagramm beschrieben werden, das dann in einem folgenden Text erläutert werden soll.

2.4.2.1 Benutzerverwaltung

Im Diagramm der Abbildung 2.7 sind die Anwendungsfälle der Benutzerverwaltung aufgeführt. Der Benutzer kann seine *Benutzerdaten bearbeiten.*

Zu den Benutzerdaten gehören das Login und die Anschrift des Benutzers. Anschließend können die geänderten Daten über den Anwendungsfall *Benutzerdaten speichern* auf dem Server persistent abgelegt werden.

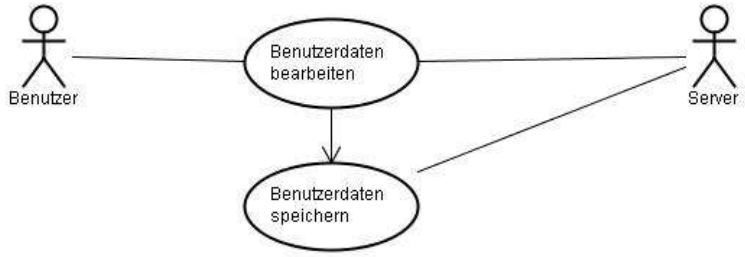

Abbildung 2.7: Anwendungsfälle der Benutzerverwaltung

2.4.2.2 Freundeverwaltung

In Abbildung 2.8 sind die Anwendungsfälle der Freundeverwaltung eingezeichnet. Der Benutzer kann seine *Liste der Freunde bearbeiten*. Dabei können die Freunde anhand ihres Logins zur Liste hinzugefügt bzw. entfernt werden. Danach kann die geänderte Liste, über den Anwendungsfall *Liste der Freunde speichern*, dem Server übergeben werden, um sie persistent zu machen.

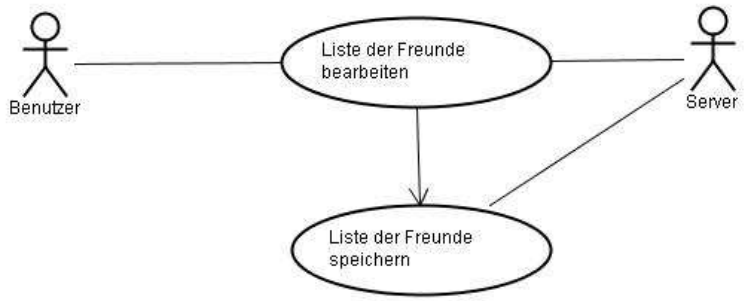

Abbildung 2.8: Anwendungsfälle der Freundeverwaltung

2.4.2.3 Albumverwaltung

Das Diagramm in Abbildung 2.9 zeigt die Anwendungsfälle der Albumverwaltung. Die Alben werden benötigt, um die Bilder des Benutzers zu strukturieren. Der Benutzer kann seine *Albumliste bearbeiten*, d. h. welche hinzufügen bzw. löschen. Dabei bekommt der Server die aktuelle Albumliste und speichert sie persistent. Außerdem können sowohl der Benutzer

als auch ein Freund des Benutzers ein *Album auswählen*. Der Freund bekommt dabei nur die für ihn freigegebenen Alben zur Auswahl. Von dem ausgewählten Album sendet der Server alle Bilder, bzw. für den Freund nur die freigegebenen Bilder, welche zunächst als Thumbnails angezeigt werden. Danach hat man über den Anwendungsfall *Bild auswählen* die Möglichkeit, aus der Thumbnailansicht ein Bild auswählen zu können. Außerdem besteht die Möglichkeit, zu dem ausgewählten Album ein *Bild hinzuzufügen*. Der Benutzer und der Freund können beide ihren lokal angelegten *Warenkorb bearbeiten* und ggf. über den Anwendungsfall *Druckauftrag annehmen*, den Warenkorb an den Server übermitteln, wodurch der Server einen Druckauftrag erhält. Zum Druckauftrag gehören Informationen zur Zahlungsart, Wahl der Zustellungsart und Preis des Auftrages. Im Folgenden wird beispielhaft auf den Anwendungsfall *Albumliste bearbeiten* eingegangen.

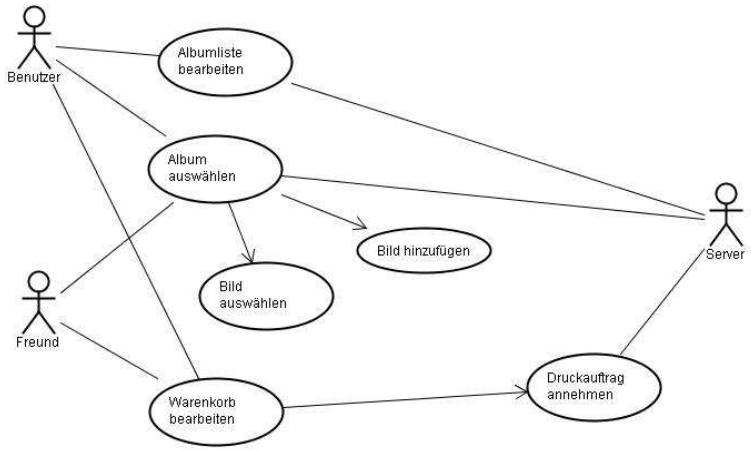

Abbildung 2.9: Anwendungsfälle der Albumverwaltung

Anwendungsfall Albumliste bearbeiten:

Zweck: die Albumliste soll bearbeitet werden

Akteure: Benutzer, Server

Ereignis: Benutzer will ein Album löschen

Vorbedingung: Benutzer muss eingeloggt sein, der Benutzer muss Besitzer des Albums sein

Relevante Eingabedaten: das zu löschende Album

Ausnahmen: Datenbankverbindung unterbrochen

Ergebnis: das zu löschende Album wird aus der Albumliste entfernt

Nachbedingung: die in dem Album enthaltenen Bilder werden gelöscht

Szenario:

1. Auswählen des zu löschenden Albums
2. Bestätigung des Löschvorgangs
3. Album wird aus der Albumliste entfernt
4. Albumliste in der Datenbank wird angepasst

2.4.2.4 Bild auswählen

Der Anwendungsfall *Bild auswählen* aus der Albumverwaltung wird in Abbildung 2.10 genauer beschrieben. Man hat über die Thumbnailansicht die Möglichkeit, sich ein *Bild im Vollbildmodus anzeigen* oder ein *Bild in den Warenkorb legen* zu lassen. Der Benutzer kann ein *Bild bearbeiten*, z. B. skalieren oder rotieren. Anschließend kann er das *Bild speichern*. Falls dabei der aktuelle Dateiname angegeben wird, ist das alte Bild überschrieben, andernfalls wird ein neues Bild angelegt. Des Weiteren kann der Benutzer seinen Freunden Bilder zugänglich machen, indem er den Anwendungsfall *Bild freigeben/sperren* ausführt. Danach kann der neue Freigabestatus über *Bild-Metadaten speichern* beim Server persistent gespeichert werden. Zudem kann der Benutzer ein *Bild löschen*. Dabei wird das Bild aus dem Album und der Datenbank entfernt. Im Folgenden werden beispielhaft drei Anwendungsfälle genauer betrachtet.

Anwendungsfall Bild bearbeiten:

Zweck: das Bild kann lokal verändert werden z. B. rotieren, skalieren

Akteure: Benutzer, Server

Ereignis: Benutzer will das Bild um 90° nach rechts rotieren lassen

Vorbedingung: Benutzer muss eingeloggt sein, der Benutzer muss Besitzer des Bildes sein

Relevante Eingabedaten: das zu bearbeitende Bild

Ausnahmen: der Benutzer ist nicht der Besitzer des Bildes

Ergebnis: alter Freigabestatus wird ersetzt, d. h. von nicht zugänglich auf zugänglich geändert bzw. umgekehrt

Nachbedingung: das Bild wurde lokal geändert, aber ist noch nicht persistent in der Datenbank abgespeichert worden

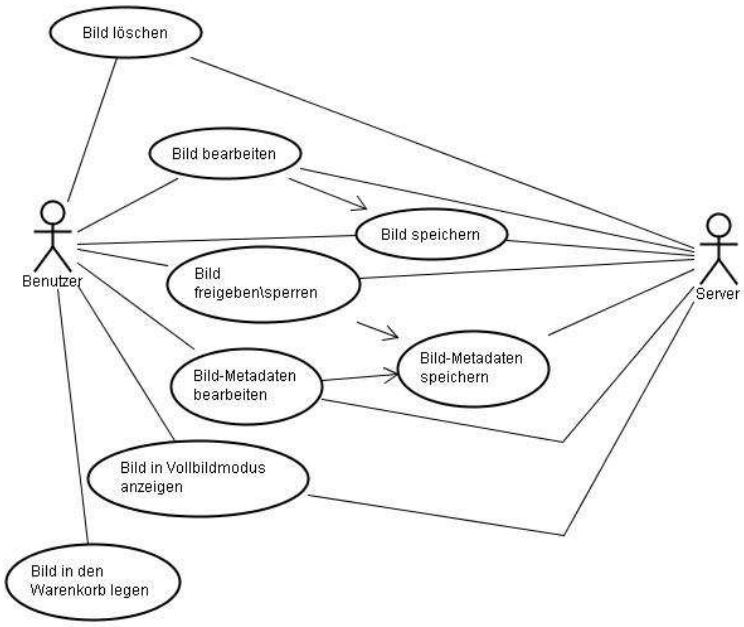

Abbildung 2.10: Anwendungsfälle eines ausgewählten Bildes

Szenario:

1. Auswahl des Bildes
2. Änderung des Freigabestatus

Anwendungsfall Bildfreigabe ändern:

Zweck: das Bild soll den Freunden zugänglich bzw. nicht mehr zugänglich sein

Akteure: Benutzer, Server

Ereignis: Benutzer will ein neues Bild seinen Freunden zeigen

Vorbedingung: Benutzer muss eingeloggt sein, der Benutzer muss Besitzer des Bildes sein

Relevante Eingabedaten: das zu bearbeitende Bild

Ausnahmen: Datenbankverbindung unterbrochen

Ergebnis: alter Freigabestatus wird ersetzt, d. h. von nicht zugänglich auf zugänglich geändert bzw. umgekehrt

Nachbedingung: alter Freigabestatus ist lokal geändert in den neuen Freigabestatus, aber noch nicht persistent in der Datenbank

Szenario:

1. Auswahl des Bildes
2. Änderung des Freigabestatus

Anwendungsfall Bild-Metadaten bearbeiten:

Zweck: Änderung der zu einem Bild abgelegten Daten

Akteure: Benutzer, Server

Ereignis: Benutzer will z. B. den Dateinamen eines seiner Bildes ändern

Vorbedingung: Benutzer muss eingeloggt sein, der Benutzer muss Besitzer des Bildes sein

Relevante Eingabedaten: das zu ersetzende Attribut samt neuen Wert

Ausnahmen: Datenbankverbindung unterbrochen

Ergebnis: alte bzw. noch nicht vorhandene Daten zum Bild werden durch die neuen ersetzt

Nachbedingung: alte Daten des Attributes sind gelöscht, aber noch nicht persistent in der Datenbank

Szenario:

1. Auswahl eines Attributes
2. Eingabe des neuen Wertes

2.5 Tipp2Win

Fußball ist in Deutschland Volkssport Nummer eins. Die EM 2004 in Portugal steht vor der Tür und die WM 2006 im eigenen Land ist auch nicht mehr weit. Oft bilden sich vor so einem Ereignis lokale Tippgruppen oder es entstehen Tippgemeinschaften im WWW. Die Möglichkeit, ein solch großes Ereignis vor dem heimischen Fernseher zu verfolgen, sich miteinander auszutauschen und eine Tippgemeinschaft zu bilden, ist erst durch die Einführung des MHP Standards möglich geworden. Tipp2Win soll eine einfach zu bedienende Anwendung sein, die genau das bietet, nämlich die Verfolgung aller Spiele live über das digitale Fernsehen (nebst Abrufen von Zusatzinformationen) und die Teilnahme an einer internationalen Tippgemeinschaft, mit deren Teilnehmern man sich jederzeit austauschen kann. Das folgende Szenario beschreibt die Möglichkeiten von Tipp2Win:

Alfred Zittermann kommt gestresst von der Arbeit nach Hause. Was hilft, um den grauen Alltag zu vergessen? König Fußball ist die Antwort. Die EM hat schon angefangen und Herr Zittermann hat sich rechtzeitig die Anwendung Tipp2Win über den Rückkanal seiner MHP Set-Top-Box heruntergeladen. Er startet das Programm, gibt seinen Benutzernamen und sein Passwort ein und landet prompt im Hauptmenü. Als erstes wechselt er in das Menü Spieletipps. Herr Zittermann sieht hier alle bereits getippten Spiele und die noch ausstehenden Vorrundenspiele aller Gruppen. Die Felder für die Finalspiele sind noch grau unterlegt, da die Begegnungen noch nicht feststehen. Als nächstes steht der Klassiker Deutschland - Holland an. Mit seiner Fernbedienung gelangt Herr Zittermann über das Menü zu den entsprechenden Eingabefeldern und trägt dort zwei Zahlen ein (eine für die getippten Tore für Deutschland, die andere für Holland). Nach einer Bestätigung wird der neue Tipp über den Rückkanal an den Datenbankserver geschickt und dort abgespeichert. „Puh, Glück gehabt“, denkt sich Herr Zittermann. Er hatte nur bis eine Stunde vor Spielbeginn Zeit, das Spiel noch zu tippen. Herr Zittermann wechselt in die jederzeit erreichbare Hilfefunktion und liest sich noch mal genau das Regelwerk durch, damit ihm so etwas nicht nochmal passiert. Nach dem Spiel wechselt Herr Zittermann in das Menü Rangliste. Er klickt auf den Button Sortieren *und sieht seinen Benutzernamen in der Rangliste farbig hervorgehoben. Dann beschließt Zittermann, sich noch ein wenig über das Spiel auszutauschen. Er wechselt in das Chatmenü, verbindet sich mit dem Chatserver und fachsimpelt eine halbe Stunde mit anderen Benutzern über das vorangegangene Spiel. Bevor Herr Zittermann zu Bett geht, entscheidet er sich noch einen* Avatar *in sein persönliches Benutzerprofil einzubinden. Dazu ruft er in dem Menü den Punkt Benutzerprofil auf und bindet, von z. B. auf einer SD-Karte ge-*

speicherten persönlichen Bildern, eines ein.

Im Folgenden werden die Benutzungsschnittstellen sowie die auftretenden Anwendungsfälle genauer beschrieben. Diese sind auf die Applikation Tipp2Win zugeschnitten. Es ist jedoch denkbar, das Xlet für andere Tippmöglichkeiten wie z. B. Bundesliga oder Championsleague anzupassen. Die GUI und die meisten anderen Funktionen wie Login, Benutzerprofil und die Datenbankanbindung könnte man durch einen generischen Klassenentwurf weiterverwenden.

2.5.1 Benutzungsschnittstellen

In den folgenden beiden Abschnitten werden die Bildschirmseiten Login und Hauptmenü vorgestellt, welche die Benutzungsschnittstelle der Anwendung Tipp2Win bilden sollen.

2.5.1.1 Login

Nach dem Starten der Anwendung (Starten des Xlets) gelangt der Benutzer zu der Anmeldeoberfläche (Abbildung 2.11), in dem sich der bereits registrierte Benutzer für das Tipp2Win Spiel einloggen kann. Sollte er sich noch nicht registriert haben, befinden sich Informationen über das Anmelde-/Registrierungsverfahren in der Hilfefunktion, die jederzeit durch einen entsprechenden Menüpunkt erreichbar ist. In der Hilfe bzw. dem Regelwerk (Teil der Hilfefunktion) befinden sich neben Informationen zur Registrierung weitere Informationen zum Tippspiel wie die Teilnahmebedingungen, die Spielregeln, die Gewinnverteilung. Durch Eingabe des registrierten Benutzernamen und des zugehörigen Passwortes in die entsprechenden Textfelder gelangt der Benutzer bei erfolgreichem Login in das Hauptmenü.

2.5.1.2 Hauptmenü

Nach dem erfolgreichen Login auf der Anmeldeoberfläche erscheint das Hauptmenü (Abbildung 2.12). Durch Betätigen der entsprechenden Tasten der Fernbedienung lassen sich die jeweiligen Untermenüs auswählen. Im Hauptmenü sollen folgende Untermenüs verfügbar sein: Benutzerprofil, Spieletipps, Rangliste, Statistiken, Regelwerk, Zurück, Chat und Vor. Mit dem Vor- bzw. Zurück-Button kann der Anwender zu der zuletzt bzw. zuvor angewählten Seite zurückkehren. Im Folgenden werden die einzelnen Menüs beschrieben.

Benutzerprofil: Hier gelangt der Anwender zu seinen persönlichen Daten und kann diese anpassen. Der Benutzer kann z. B. seine Adresse

Abbildung 2.11: Login-Menü

Abbildung 2.12: Hauptmenü

ändern, sein Login, sowie weitere persönliche Daten, die der Benutzer anderen Tippspielern zur Einsicht freigeben will (Lieblingsverein, Bild einfügen).

Spieletipps: Bei Aufruf dieses Menüpunktes erhält der Benutzer eine Ansicht aller bereits getippten Spiele (Abbildung 2.13) und noch ausstehenden Tipps. An dieser Stelle erfolgt auch die Eingabe der zu tippenden Spielergebnisse. Mit dem Vor- und Zurück-Button lässt sich zu den einzelnen Eingabefeldern für die Ergebnisse springen. Mit den auf jeder Fernbedienung entsprechenden Tasten (rote, grüne, gelbe und blaue Funktionstasten) lässt sich zwischen den einzelnen Tippfeldern hin und her springen sowie der Tipp bestätigen oder abbrechen. Die eingegebenen Daten werden an den Server versendet und dort in der Datenbank gespeichert. Um auch die Torjäger, fairste Mannschaft und Ähnliches tippen zu können, befinden sich oberhalb der Spieletipps die entsprechenden Buttons, über die in die jeweiligen Menüs gelangt werden kann. Die letzt genannten Tippmöglichkeiten sind separat von dem allgemeinen Tippspiel. Der Gewinn ist hier nicht gestaffelt, sondern es existieren nur die ersten drei Plätze.
Die Spieltage müssen immer bis zu bestimmten Deadlines (z. B. 0 Uhr des aktuellen Spieltages) getippt werden. Es werden alle Gruppenspiele und Finales (Achtelfinales, Viertelfinales, Halbfinales und Finale) getippt. Die Gewinnausschüttung ist hierbei gestaffelt und erfolgt nach einer Gewinnformel, in die Einsatz, Teilnehmerzahl und die Tippergebnisse einfließen. Dieser Teil der Anwendung soll aber, um eine einfache Änderung der Tippregeln vornehmen zu können, modular implementiert werden. Denkbar ist ein eigenes Modul für die Tippregeln, dass über eine wohl definierte Schnittstelle angesprochen und so durch ein Modul mit anderen Regeln ersetzt werden kann.

Rangliste: Dieses Menü enthält eine Übersicht der aktuellen Position in der Gesamtwertung. Diese setzt sich dabei aus der Summe der bisher erzielten Punkte zusammen, die nach jedem Spieltag aktualisiert und mittels einer Gewinnformel errechnet wird. Der eigene Benutzername wird farblich hervorgehoben und der Benutzer kann die Tabelle nach bestimmten Kriterien sortieren, wie z. B. alphabetisch, Höhe des erzielten Gewinns und Freunde.

Statistik: Bietet einen Zugriff auf technische Statistiken über das System (z. B. über die Benutzer) und vor allem über die aktuellen EM-Statistiken (Schiedsrichter, bisherige verteilte gelbe und rote Karten, Torjägerliste, Vorbereiter...).

Abbildung 2.13: Spieletipps

Hilfefunktion: Die Hilfefunktion ist von jedem anderen Menü aus erreichbar. Sie enthält die Registrierungsvoraussetzungen, AGB[7], Tippfristen, Gewinnverteilung, Wettmöglichkeiten und detaillierte Erläuterungen zu Tipp2Win.

Chat (optional): Ermöglicht die Kommunikation mit anderen Fußballfans, wobei hierzu eine Tastatur empfehlenswert ist, da ein Chat mittels Fernbedienung sehr umständlich wäre.

2.5.2 Anwendungsfälle

Im Folgenden werden typische Anwendungsfälle im Zusammenhang mit der Anwendung *Tipp2Win* vorgestellt. Die Anwendungsfälle werden in Bereiche zusammengefasst, wobei jeder Bereich durch ein Anwendungsfalldiagramm verdeutlicht wird.

2.5.2.1 Anmeldung

Das Diagramm in Abbildung 2.14 zeigt die beiden Anwendungsfälle, die auftreten können, wenn der Benutzer die Applikation Tipp2Win startet und die

[7] Allgemeine Geschäftsbedingungen

Anmeldungsoberfäche erscheint. Der Benutzer kann sich *einloggen* oder *registrieren*, falls er die Applikation zum ersten Mal ausführt. Für diesen Bereich soll repräsentativ der Anwendungsfall Login beschrieben werden.

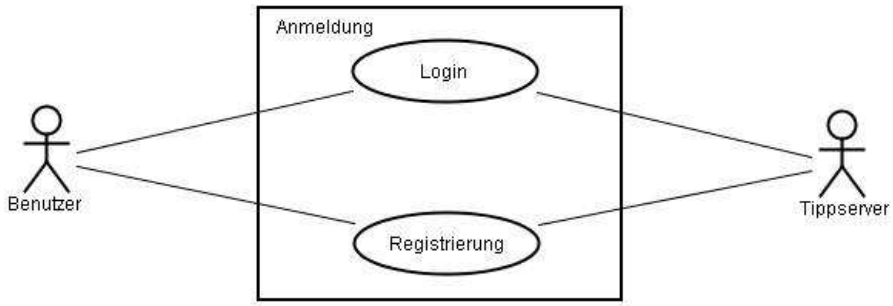

Abbildung 2.14: Anwendungsfälle Anmeldung

Anwendungsfall Login:

Zweck: Teilnahme am Tippspiel.

Akteure: Benutzer.

Ereignis: Benutzer will z. B. einen Tipp abgeben.

Vorbedingung: Benutzer muss angemeldet sein.

Relevante Eingabedaten: Benutzername und Passwort.

Ausnahmen: fehlerhafte Benutzereingabe, Verbindung zur Datenbank fehlgeschlagen.

Ergebnis: Benutzer ist erfolgreich eingeloggt.

Nachbedingung: keine.

Szenario:

1. Eingabe des Benutzernamens.
2. Eingabe des Passwortes.

2.5.2.2 Rangliste

Die Abbildung 2.15 beinhaltet alle Anwendungsfälle, die auftreten können, wenn der Benutzer den Menüpunkt *Rangliste* aktiviert. Der Benutzer kann die Rangliste *ansehen*, *aktualisieren* und *sortieren*. Zur Verdeutlichung wird der Anwendungsfall *Punktestand aktualisieren* exemplarisch beschrieben.

Anwendungsfall Punktestand aktualisieren:

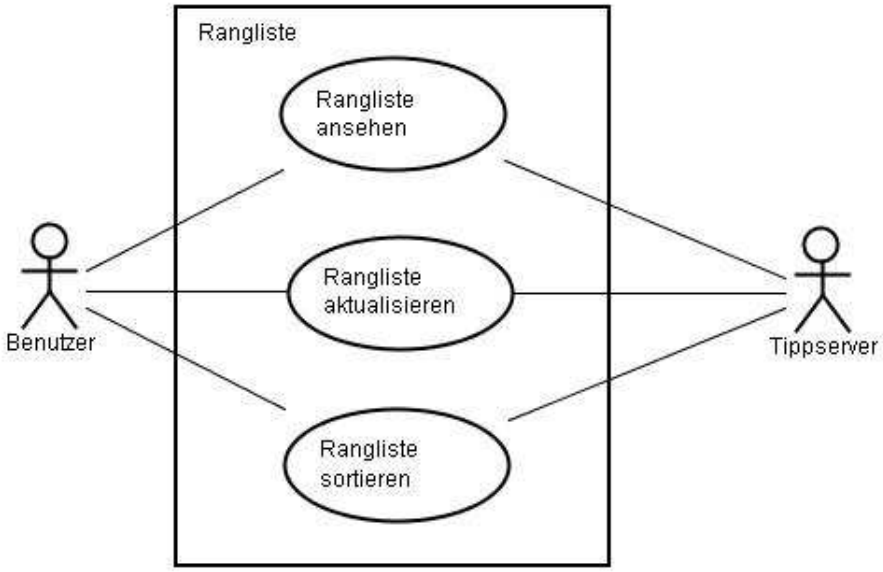

Abbildung 2.15: Anwendungsfälle Rangliste

Zweck: Anzeige des aktuellen Punktestandes.

Akteure: Benutzer, Server.

Ereignis: Benutzer will eine aktualisierte Version der Tabelle erhalten.

Vorbedingung: Benutzer muss eingeloggt sein.

Relevante Eingabedaten: keine.

Ausnahmen: Datenbankverbindung unterbrochen.

Ergebnis: Benutzer erhält eine aktuelle Ergebnistabelle.

Nachbedingung: keine.

Szenario:

1. Auswahl des Menüpunktes aktualisieren.
2. Anzeige der neuen Daten.

2.5.2.3 Benutzerprofil

Es existieren zwei Anwendungsfälle, die auftreten können, wenn der Benutzer den Menüpunkt *Profil* aktiviert. Der Benutzer kann sein eigenes oder die Profile von anderen Tippspielern *betrachten* oder er kann **sein** eigenes Profil *editieren* und *sortieren*. Der Anwendungsfall *Profil editieren* wird hier genau beschrieben. In Abbildung 2.16 ist das entsprechende Anwendungsfalldiagramm zu sehen.

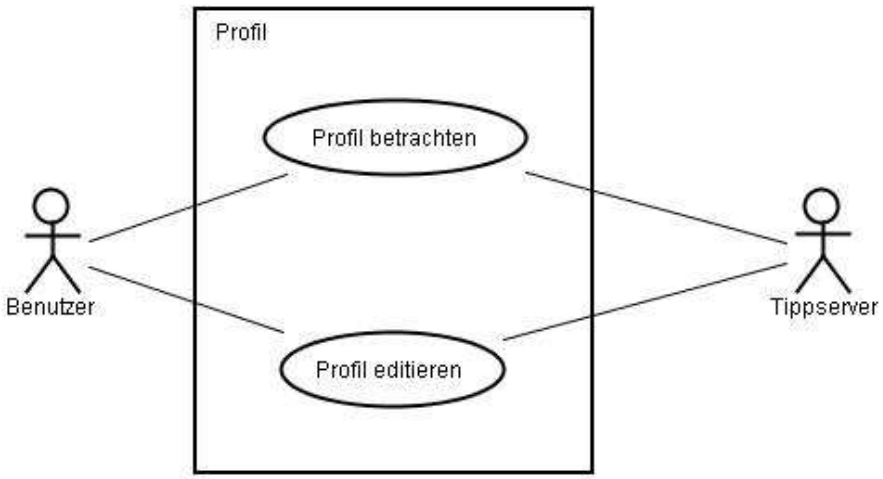

Abbildung 2.16: Anwendungsfälle Profil

Anwendungsfall Profil editieren:

Zweck: Editierung des individuellen Profils.

Akteure: Benutzer, Server.

Ereignis: Benutzer will sein Profil editieren.

Vorbedingung: Benutzer muss eingeloggt sein.

Relevante Eingabedaten: Auswahl des eigenen Profils.

Ausnahmen: keine.

Ergebnis: Benutzer editiert sein individuelles Profil.

Nachbedingung: Geänderte Profildaten werden in der Datenbank abgespeichert.

Szenario:

1. Auswahl des Menüs Benutzerprofile.
2. Auswahl des eigenen Profils.
3. Abspeicherung der Änderungen.

2.5.2.4 Spieletipps

Hier gibt es ebenfalls zwei Anwendungsfälle, wenn der Benutzer den Menüpunkt *Spieletipps* auswählt. Der Benutzer kann seine Spieletipps entweder *anzeigen* oder er kann sie *editieren*, das entsprechende Anwendungsfalldiagramm ist in Abbildung 2.17 dargestellt. Der Anwendungsfall *Spieletipps anzeigen* wird hier im Detail aufgeführt.

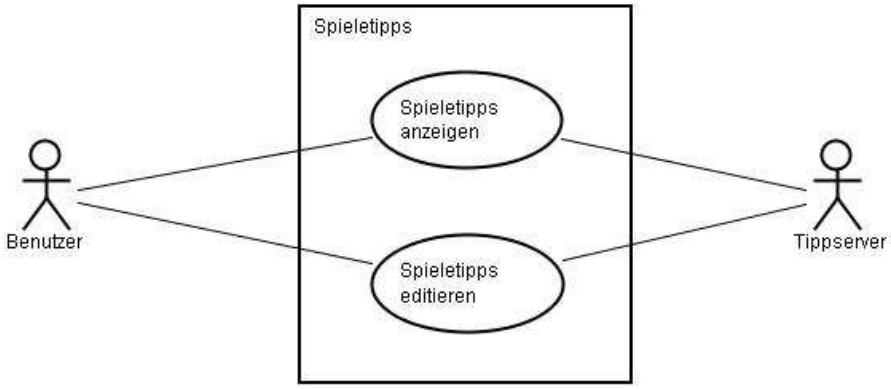

Abbildung 2.17: Anwendungsfälle Spieletipps

Anwendungsfall Spieletipps anzeigen:

Zweck: Betrachtung der bereits getippten Spiele.

Akteure: Benutzer, Server.

Ereignis: Benutzer will seine Spieletipps ansehen.

Vorbedingung: Benutzer muss eingeloggt sein.

Relevante Eingabedaten: keine.

Ausnahmen: Verbindung zur Datenbank fehlgeschlagen.

Ergebnis: Benutzer betrachtet seine Spieletipps.

Nachbedingung: keine.

Szenario:

1. Auswahl des Menüs Spieletipps.

2.5.2.5 Regelwerk, Hilfen, AGB

In Abbildung 2.18 ist das Anwendungsfalldiagramm zu *Regelwerk, Hilfe und AGB einsehen* dargestellt. Das Menü mit dem Regelwerk, der Hilfe und den AGB ist jederzeit von jedem anderen Menüpunkt aus erreichbar. Innerhalb des Menüs kann der Benutzer die gewünschten Informationen einsehen.

Anwendungsfall Regelwerk, Hilfe und AGB einsehen:

Zweck: Ansicht des/r gültigen Regelwerkes / AGB / Hilfe.

Akteure: Benutzer, Server.

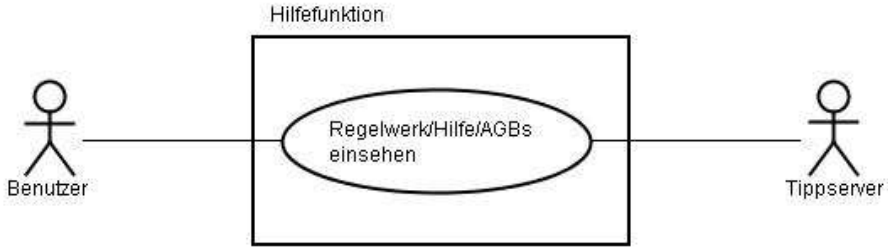

Abbildung 2.18: Anwendungsfälle Regelwerk, Hilfe, AGB

Ereignis: Benutzer will sich über Regelwerk / AGB / Hilfe informieren.

Vorbedingung: Benutzer muss eingeloggt sein.

Relevante Eingabedaten: keine.

Ausnahmen: Verbindung zur Datenbank ist unterbrochen.

Ergebnis: Anzeige der Geschäftsbedingungen und des Regelwerkes.

Nachbedingung: keine.

Szenario:

1. Auswahl des Menüpunktes Regelwerkes.
2. Anzeige des/r Regelwerkes / AGB / Hilfe.

2.5.2.6 Chat

Dieser Anwendungsfall ist optional, d. h. seine Implementierung wird erst in der Entwicklungsphase unter Absprache mit den Betreuern entschieden. Falls eine Implementierung erfolgt, wird es dem Benutzer ermöglicht, unter dem Menüpunkt *Chat* mit anderen Tippspielern zu chatten und seine Gedanken auszutauschen. Das entsprechende Anwendungsfalldiagramm ist in Abbildung 2.19 zu sehen.

Anwendungsfall Chat:

Zweck: Benutzer will mit anderen Tippspielern chatten.

Akteure: Benutzer, Server.

Ereignis: Benutzer möchte mit anderen Mitspielern kommunizieren.

Vorbedingung: Benutzer muss eingeloggt sein.

Relevante Eingabedaten: Nickname / Alias.

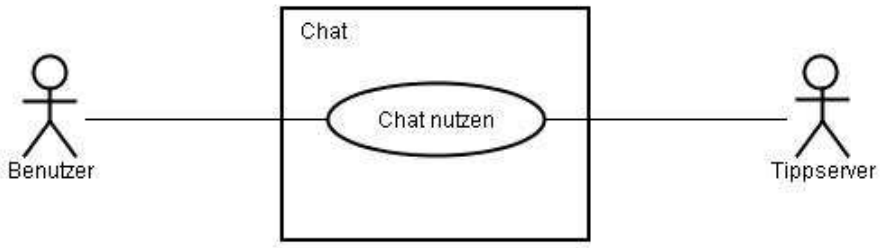

Abbildung 2.19: Anwendungsfall Chat

Ausnahmen: Verbindung zum Server ist unterbrochen. Server ist überlastet.

Ergebnis: Benutzer nimmt am Chat teil.

Nachbedingung: keine.

Szenario:

1. Auswahl des Menüpunktes Chat.
2. Eingabe des Nicknames
3. Anzeige des Chatfensters.

2.6 Electronic Program Guide

Zu Beginn des Fernsehens existierte eine überschaubare Anzahl an Fernsehprogrammen und der Zuschauer war nur mit wenigen Programminformationen konfrontiert. Mittlerweile ist die Programmanzahl um ein Vielfaches gestiegen und ein Ende dieser Entwicklung ist nicht abzusehen. Um dem Zuschauer einen besseren Überblick über die verschiedenen Programme zu bieten und ihm ein auf seine persönlichen Bedürfnisse abgestimmtes Programm zu liefern, wird im Rahmen der Projektgruppe ein EPG (Electronic Program Guide bzw. Elektronischer Programmführer) entwickelt.
Dabei handelt es sich um eine Art „erweiterte Programmzeitschrift“, die über den Fernseher in Verbindung mit einer Set-Top-Box und der MHP abrufbar ist. Der EPG orientiert sich an den Funktionen einer klassischen Programmzeitschrift, indem eine tabellarische Programmübersicht aller Rundfunkanstalten angezeigt werden kann. Zu allen Sendungen lassen sich dann detailliertere Informationen und multimediale Inhalte wie Trailer und Bilder abrufen. Darüber hinaus soll dem Zuschauer anhand seines Benutzerprofils ein individuelles Programm zusammengestellt werden, das seinen Vorlieben entspricht (wird im Folgenden *MeinEPG* genannt). Dieses Programm lässt sich vom Zuschauer durch Hinzufügen/Entfernen von Sendungen noch weiter anpassen und optional wird die Möglichkeit geboten, Sendungen aufzuzeichnen, sofern die Set-Top-Box über eine Festplatte verfügt. Der Benutzer kann in der Anwendung eine Liste anlegen, in der er die aufzuzeichnenden Sendungen eintragen kann. Dieser Programmteil wird im Folgenden als *PVR* (persönlicher Videorecorder) bezeichnet. Alle in der Liste eingetragenen Sendungen werden auf der Festplatte aufgezeichnet, eine umständliche Programmierung eines Videorecorders wird damit überflüssig. Natürlich wird dem Benutzer die Möglichkeit gegeben, diese Liste zu betrachten oder auch Einträge wieder zu entfernen.

Dazu stelle man sich folgendes beispielhafte Szenario vor:

Max Mustermann sitzt kurz vor Acht gemütlich vor dem Fernseher und zappt durch die zahlreichen Programme. Er ist auf der Suche nach einem guten Film, und da er keine Fernsehzeitschrift hat, ruft er auf jedem Programm den Videotext auf, um in Erfahrung zu bringen, welcher Film um 20:15 Uhr auf dem jeweiligen Sender anläuft. Da ihm die Filmtitel nicht viel sagen, ruft er vereinzelt die Videotextseiten mit den weiteren Informationen zum Film auf. Er ärgert sich, dass er diese Prozedur auf jedem Sender wiederholen muss und der Vorgang sehr zeitintensiv ist. Da fällt ihm ein, dass neuerdings ein senderübergreifender „Electronic Program Guide“ auf der Set-Top-Box verfügbar ist, bei dem er sich bereits mit seinem Benutzerprofil angemeldet hat, und

startet ihn. Sofort nach dem Einloggen sieht er im Hauptmenü den Bereich „MeinEPG“ mit seinem individuellen Programm und erkennt, dass um 20:15 Uhr der Film „Jagd auf Roter Oktober“ mit seinem Lieblingsschauspieler Sean Connery läuft. Er entscheidet sich, diesen Film anzuschauen. Gleichzeitig sieht er, dass zur selben Zeit eine Komödie, die seinem favorisierten Genre entspricht, läuft. Er wählt den Film aus, um weitere Informationen zu erhalten, und möchte am liebsten beide Filme gleichzeitig anschauen. Da das nicht möglich ist, markiert er den zweiten Film, der dadurch automatisch auf der Festplatte der Set-Top-Box aufgezeichnet wird, um ihn zu einem späteren Zeitpunkt anschauen zu können.

2.6.1 Benutzungsschnittstelle

Im Folgenden soll anhand einiger Screenshots die Benutzungsoberfläche vorgestellt werden.

2.6.1.1 Anmeldung

Nach Starten der Anwendung muss sich der Benutzer mit Login und einem Passwort im Anmeldebildschirm einloggen, siehe dazu Abbildung 2.20. Bisherige Login-Namen sollen bei Eingabe eines Zeichens automatisch ergänzt werden. Dadurch muss nicht bei jedem Anmeldevorgang der Benutzername vollständig neu eingegeben werden. Es muss lediglich noch das Passwort oder eine Pin-Nummer mit der Fernbedienung ergänzt werden. Neue Benutzer wählen die Schaltfläche *Neuanmeldung* und geben ihr Benutzerprofil ein.

2.6.1.2 Benutzerprofil

In diesem Bildschirm (Abbildung 2.21) kann ein neuer Benutzer sein Profil eingeben oder ein bereits registrierter Benutzer das Profil einsehen und verändern. Benutzername und Passwort bzw. PIN sind Pflichtangaben, die anderen Angaben sind optional und dienen der eigentlichen Personalisierung: man kann seine Lieblingsgenres auswählen, einige Begriffe in die Volltextsuche eingeben (beispielsweise Namen von Schauspielern oder Regisseuren) und die Reihenfolge der Programme in der Programmübersicht festlegen. Aus den Angaben Geburtsjahr, Geschlecht, Lieblingsgenres und den Stichworten der Volltextsuche können dem Anwender Programmvorschläge gemacht werden. Diese werden dann als *TagesTIPP* oder auch in *MeinEPG* angezeigt.

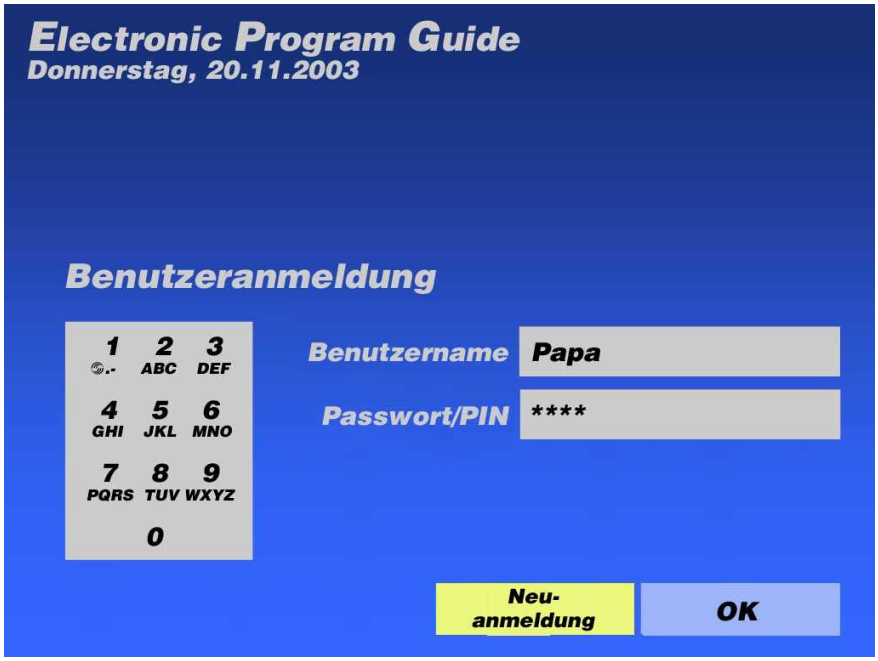

Abbildung 2.20: Login EPG

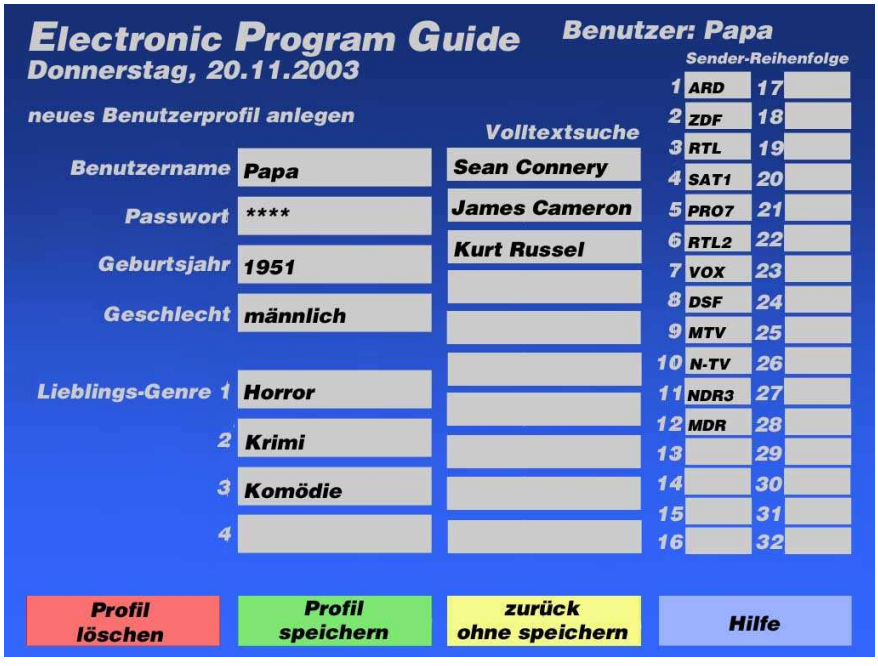

Abbildung 2.21: Benutzerprofil

2.6.1.3 Hauptmenü

Nach erfolgreicher Anmeldung erscheint das Hauptmenü, siehe dazu Abbildung 2.22. Hier werden dem Benutzer die tages- und zeitaktuellen Einträge aus *MeinEPG* angezeigt. Außerdem wird ein *TagesTIPP* angeboten, dieser soll optional personalisiert werden. Diese Einträge können direkt ausgewählt werden, um eine Detailansicht zu erhalten.
Aus dem Hauptmenü kann man in die Programmübersicht, in *MeinEPG*, in *PVR* und in die Hilfe gelangen. Der Wechsel in die Menüpunkte soll nach Möglichkeit über die vier farbigen Tasten auf der Fernbedienung erfolgen.

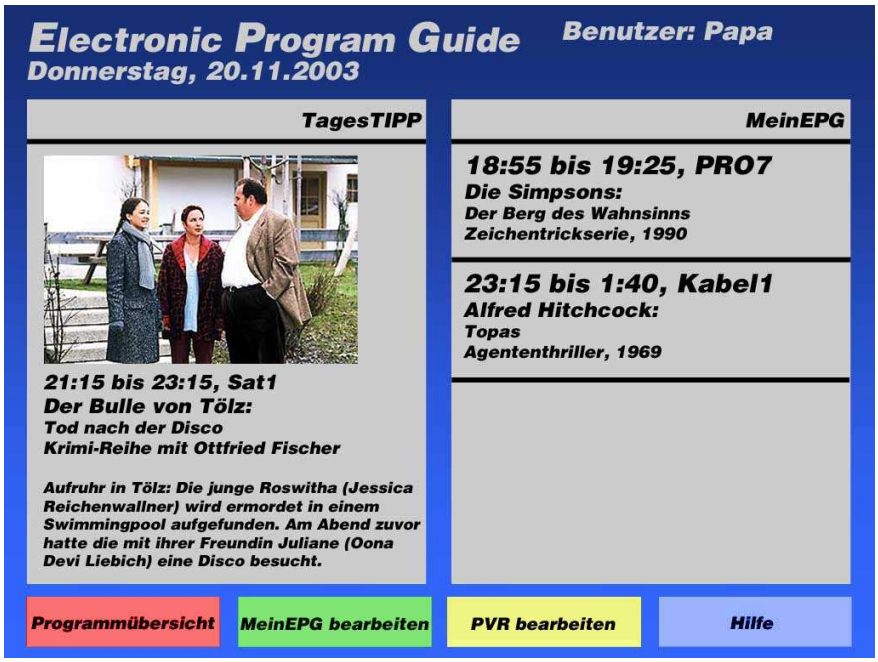

Abbildung 2.22: Hauptmenü EPG

2.6.1.4 Programmübersicht

Wechselt der Benutzer in die Programmübersicht (Abbildung 2.23), präsentiert sich ihm das aktuelle Fernsehprogramm in Senderblöcken zu je drei bis sechs Sendern. Die genaue Wahl der Anzahl der Senderblöcke wird, in Abhängigkeit der Lesbarkeit auf dem Fernsehbildschirm, noch zu einem späteren Zeitpunkt getroffen. Mit Hilfe von *vor*- und *zurück*-Buttons kann zwischen den Senderblöcken gewechselt werden. Jede Sendung wird durch Uhrzeit, Name und optional Genre in Tabellenform angezeigt (wie von der Fernsehzeitung gewohnt). Mit den Pfeiltasten kann jeder einzelne Eintrag ausgewählt werden,

um in die Detailansicht zu wechseln. Da das komplette Tagesprogramm nicht vollständig dargestellt werden kann, lässt sich über *früher*- und *später*- Buttons das gesamte Programm betrachten.
Aus der Programmübersicht kann der Benutzer mittels der verschiedenfarbigen Tasten in das Hauptmenü wechseln, sich die Hilfe anzeigen lassen und zwischen den Senderblöcken hin- und herspringen.

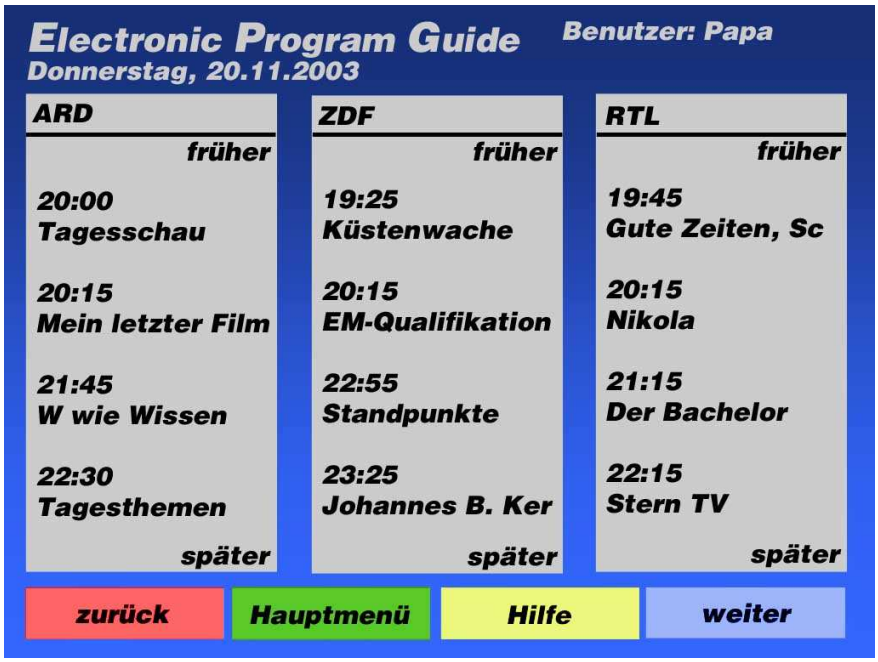

Abbildung 2.23: Programmübersicht EPG

2.6.1.5 Detailansicht

In dieser Ansicht werden dem Benutzer alle verfügbaren Informationen zur Sendung angezeigt (Abbildung 2.24):

- Zeiten (Beginn, Ende und Dauer).
- Inhalt oder Themen (Text).
- Originaltitel, Schauspieler, Moderatoren, Regie, Gäste, Genre, Erstausstrahlung, Wiederholung, Altersfreigabe, Erscheinungsjahr.
- Sonstige Zusatzinformationen wie Schwarz-Weiß, für Hörgeschädigte, Mehrkanalton.

- Es können Multimedia-Inhalte wie z. B. Bilder oder Trailer auf Anfrage angezeigt werden, diese werden dann über den Rückkanal vom Server geladen (optionale Funktionalität).

Welche Informationen tatsächlich dargestellt werden, hängt von der später eingesetzten Datenquelle ab. Sind die vorhandenen Informationen so umfangreich, dass sie nicht komplett auf dem Bildschirm dargestellt werden können, kann man mit den Pfeiltasten den Inhalt „scrollen“ (technische Realisierbarkeit vorausgesetzt).
Der Benutzer hat die Möglichkeit, die aktuelle ausgewählte Sendung über einen Button in *MeinEPG* hinzuzufügen oder mittels eines anderen farbigen Buttons die Sendung aufzuzeichnen. Das Aufzeichnen einer Sendung ist eine optionale Funktionalität, in diesem Fall muss die Set-Top-Box eine integrierte Festplatte besitzen. Auch hier kann die Hilfe über eine farbige Taste erreicht werden. Über den *Zurück*-Button gelangt man wieder zum vorherigen Menü.

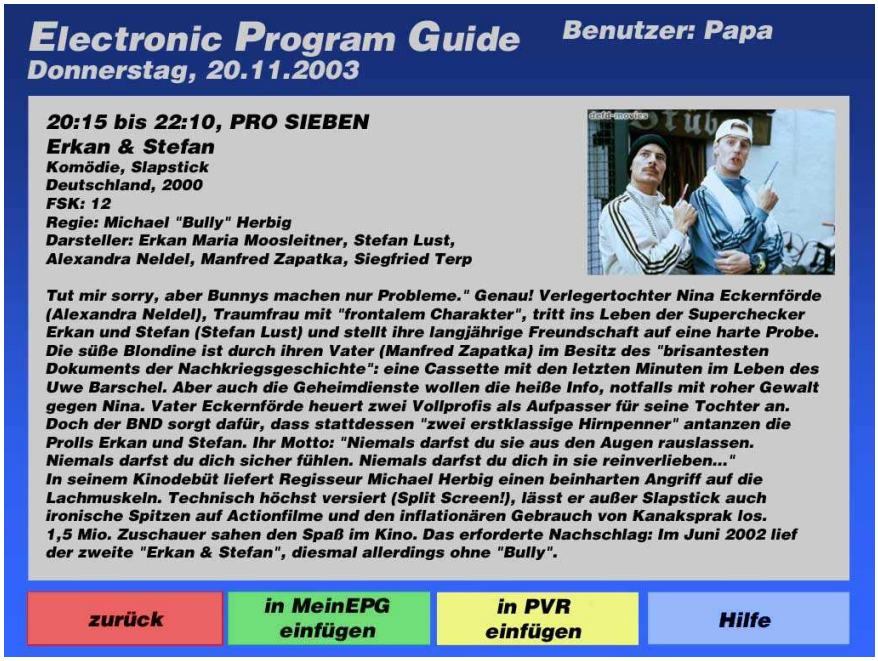

Abbildung 2.24: Detailansicht EPG

2.6.1.6 MeinEPG

Auf dieser Bildschirmseite kann man sich einen Überblick über seine persönlichen Programmvorlieben verschaffen (Abbildung 2.25). Die Ansicht der

Sendungen erfolgt tabellarisch wie in der Programmübersicht. Mit den Pfeiltasten lässt sich jede Sendung auswählen, um dann mit der *OK*-Taste in die entsprechende Detailansicht zu wechseln, den Eintrag über die farbigen Tasten zu entfernen oder in *PVR* aufzunehmen. Die gesamte Programmliste kann auch gelöscht werden. Überschneidungen werden in dieser Liste nicht weiter behandelt, zeitgleiche Sendungen werden untereinander angezeigt.

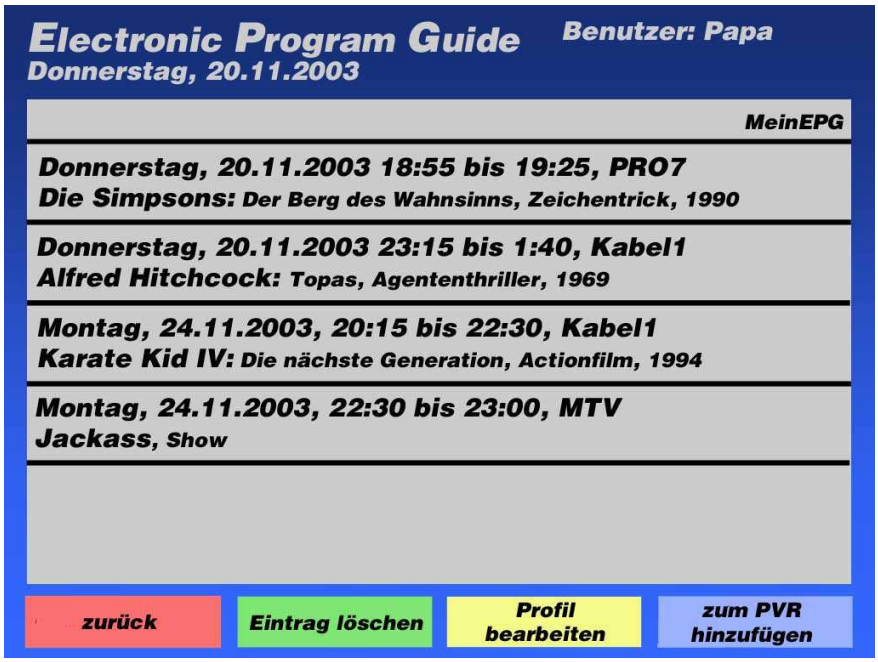

Abbildung 2.25: MeinEPG EPG

2.6.1.7 PVR (optional)

In dieser Liste werden die Sendungen angezeigt, die auf eine evtl. vorhandene Festplatte aufgenommen werden sollen. Funktionalität im wesentlichen siehe *MeinEPG*.

2.6.2 Anwendungsfälle

Im Folgenden werden die Anwendungsfälle näher betrachtet. Sie wurden anhand der Menüstruktur zusammengefasst. Es wird jeweils ein Anwendungsfall exemplarisch betrachtet.

2.6.2.1 Login

Im Login-Bereich (Abbildung 2.26) gibt es die Anwendungsfälle *Login* eines bereits registrierten Benutzers und die *Registrierung* eines neuen Benutzers. Im Folgenden wird der Anwendungsfall *Login* näher betrachtet.

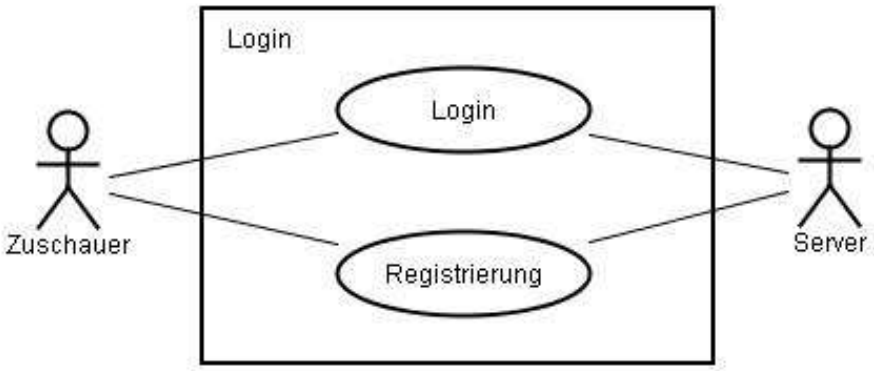

Abbildung 2.26: Anwendungsfälle EPG-Login

Anwendungsfall Login:

Zweck: ermöglicht das Anmelden eines bereits registrierten Benutzers.

Akteure: Zuschauer und Server.

Ereignis: Programm wurde gestartet.

Vorbedingung: Benutzer muss bereits registriert sein.

Relevante Eingabedaten: Benutzername und Passwort/PIN.

Ausnahmen: fehlerhafte Benutzereingabe, Verbindung zur Datenbank fehlgeschlagen.

Ergebnis: Benutzer ist erfolgreich eingeloggt.

Nachbedingung: keine.

Szenario:

1. Eingabe des Benutzernamens.
2. Eingabe des Passwortes/der PIN.

2.6.2.2 Hauptmenü

Im Hauptmenü gibt es die Anwendungsfälle *Programmübersicht ansehen*, *MeinEPG bearbeiten*, *PVR bearbeiten* und *Hilfe anzeigen* (Abbildung 2.27). Im Folgenden wird der Anwendungsfall *MeinEPG bearbeiten* näher betrachtet.

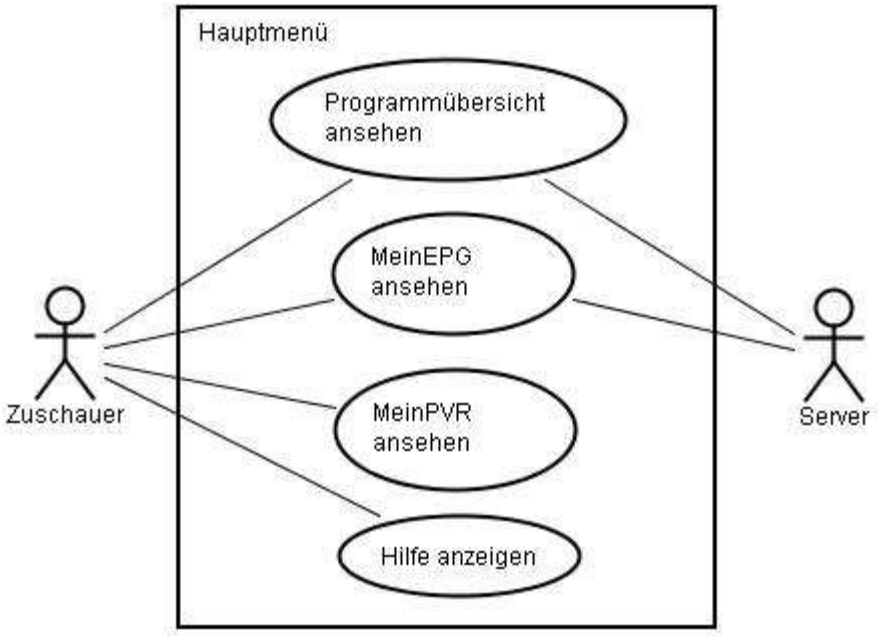

Abbildung 2.27: Anwendungsfälle im Bereich Hauptmenü

Anwendungsfall MeinEPG bearbeiten:

Zweck: ermöglicht, die eigenen Personalisierungsdaten anzuzeigen und zu verwalten.

Akteure: Zuschauer und Server.

Ereignis: Bereich *MeinEPG* wurde vom Zuschauer ausgewählt.

Vorbedingung: keine.

Relevante Eingabedaten: keine.

Ausnahmen: Verbindung zur Datenbank fehlgeschlagen.

Ergebnis: *MeinEPG* wird angezeigt.

Nachbedingung: keine.

Szenario:

1. Bereich *MeinEPG* wird ausgewählt.

2.6.2.3 Detailansicht

In der Detailansicht gibt es die Anwendungsfälle *in MeinEPG einfügen* und *in PVR einfügen*. Exemplarisch wird hier der Anwendungsfall *in MeinEPG*

einfügen betrachtet. Das Anwendungfalldiagramm für diesen Bereich ist in Abbildung 2.28 zu sehen.

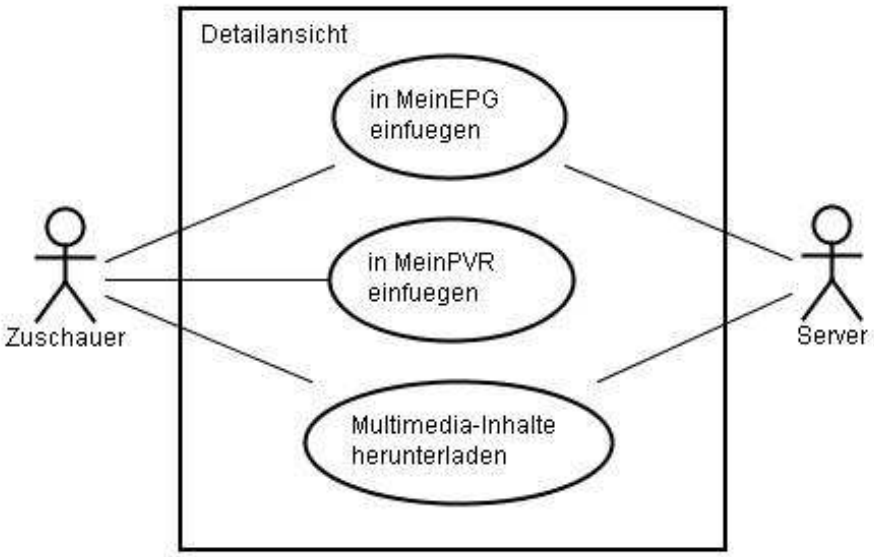

Abbildung 2.28: Anwendungsfälle Detailansicht

Anwendungsfall *in MeinEPG einfügen*:

Zweck: ermöglicht, eine Sendung in die Liste von *MeinEPG* einzufügen.

Akteure: Zuschauer und Server.

Ereignis: Button *in MeinEPG einfügen* wurde gedrückt.

Vorbedingung: Benutzer befindet sich in der Detailansicht.

Relevante Eingabedaten: keine.

Ausnahmen: Sendung befindet sich bereits in *MeinEPG*, Verbindung zur Datenbank fehlgeschlagen.

Ergebnis: Sendung befindet sich in der Liste von *MeinEPG*.

Nachbedingung: keine.

Szenario:

1. Drücken des Buttons *in MeinEPG einfügen*.
2. Bestätigung, ob Sendung wirklich hinzugefügt werden soll.

2.6.2.4 MeinEPG

In MeinEPG gibt es die Anwendungsfälle *Eintrag löschen*, *Profil bearbeiten* und *in PVR einfügen* (Abbildung 2.29). Der Anwendungsfall *Profil bearbeiten* wird nun näher betrachtet.

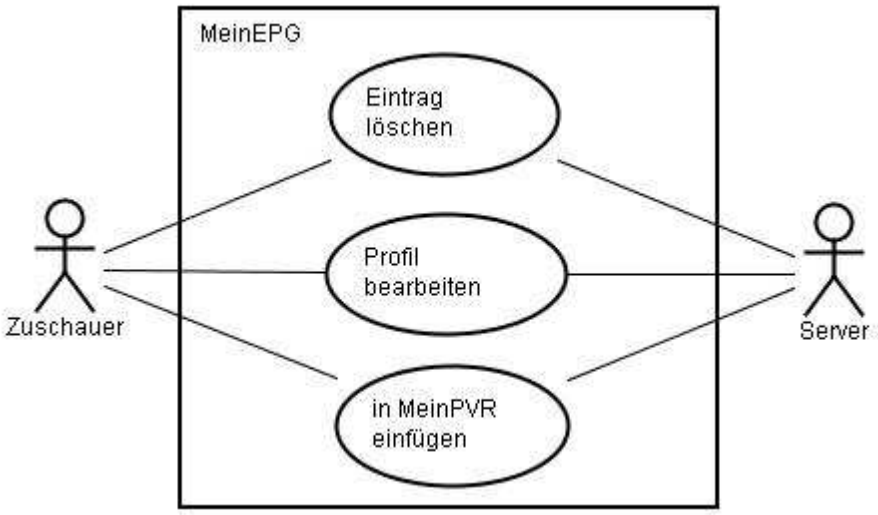

Abbildung 2.29: Anwendungsfälle im Bereich *MeinEPG*

Anwendungsfall *Profil bearbeiten* :

Zweck: ermöglicht, die Personalisierungsdaten eines bereits registrierten Anwenders anzusehen und zu bearbeiten.

Akteure: Zuschauer und Server.

Ereignis: Button *Profil bearbeiten* wurde gedrückt.

Vorbedingung: Benutzer muss sich im Menüpunkt *MeinEPG* befinden.

Relevante Eingabedaten: keine.

Ausnahmen: Verbindung zur Datenbank fehlgeschlagen.

Ergebnis: Personalisierungsdaten werden angezeigt.

Nachbedingung: keine.

Szenario:

1. Button *Profil bearbeiten* wird gedrückt.

2.6.2.5 PVR

In diesem Bereich gibt es nur den Anwendungsfall *Eintrag löschen* (Abbildung 2.30).

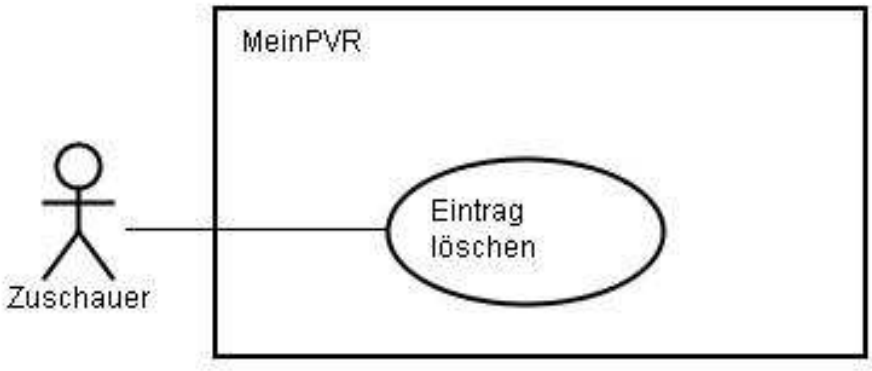

Abbildung 2.30: Anwendungsfälle im Bereich *PVR*

Anwendungsfall *Eintrag löschen* :

Zweck: ermöglicht, einen Eintrag aus der Liste der aufzuzeichnenden Sendungen zu löschen.

Akteure: Zuschauer.

Ereignis: Button *Eintrag löschen* wurde gedrückt.

Vorbedingung: Benutzer muss sich im Menüpunkt *PVR* befinden, ein Eintrag muss vorhanden sein.

Relevante Eingabedaten: keine.

Ausnahmen: keine.

Ergebnis: Eintrag wird aus der Liste der aufzuzeichnenden Sendungen entfernt.

Nachbedingung: keine.

Szenario:

1. Eintrag wird in der Liste ausgewählt.
2. Button *Eintrag löschen* wird gedrückt.

2.7 Abnahmekriterien

Im Folgenden werden die Abnahmekriterien für die Erstellung der Anwendungen, die die Projektgruppe erfüllen soll, näher beschrieben. Von den drei beschriebenen Modulen werden mindestens zwei ausgewählt und umgesetzt. Die realisierten Module sollen in Form eines Prototyps einsetzbar sein. Diese Einsetzbarkeit wird mit einer Vorführung der Funktionen demonstriert. Das System soll dazu im OFFIS installiert und die Anwendungen dort präsentiert werden.
Ein wesentliches Kriterium unseres Systems sollen die benutzungsfreundlichen Oberflächen bzw. Menüs bilden, die mit der Fernbedienung gesteuert werden können. Die Nutzung aller eingebauten Funktionen in den jeweiligen Menüs soll gewährleistet werden. Über dies ist die Einhaltung des Fertigstellungstermins sowie der funktionalen und nichtfunktionalen Anforderungen wichtig. Diese Funktionen werden im Laufe der Entwicklung getestet und bei Fehlverhalten analysiert und korrigiert. Ziel der Tests ist es, die Anwendungen auf Gebrauchstauglichkeit zu testen. Zudem soll die Dokumentation des Quellcodes mit JavaDoc erfolgen. Ein Zwischenbericht, der gegen Ende April veröffentlicht wird[8], soll Auskunft über den derzeitigen Stand der Entwicklung geben. Im Endbericht dagegen, der zum Abschluss der Projektgruppe erscheinen soll, wird ein Resümee gezogen und ein Ausblick auf mögliche Weiterentwicklungen der Anwendungen gegeben.

[8] siehe dazu Abschnitt 2.8

2.8 Vorgehensmodell und Zeitplan

Aufgrund der Verwendung neuer und unbekannter Technologie, wie sie die MHP darstellt, haben wir uns entschieden, zunächst erst eine der drei Anwendungen mit Hilfe des Prototypenmodells [Bal00] zu entwickeln. Durch die Kooperation mit CeWe Color ist die Entwicklung der Mediengalerie vorrangig, deshalb soll mit ihrer Erstellung begonnen werden. Anschließend lässt sich durch die hierbei gesammelten Erfahrungen der Aufwand für das weitere Vorgehen bzgl. der übrigen Anwendungen besser abschätzen.
Nachdem die Anforderungsdefinition bis zum 04.12.03 fertiggestellt sein soll, wird zuerst eine zweiwöchige Phase zur Einarbeitung in die Entwicklungsumgebung und zur Erstellung eines kleinen *Hello World*-Prototypen stattfinden. Hierbei sollen die technischen Möglichkeiten der MHP ergründet werden. Anschließend wird zum 04.02.04 nach einer Analysephase ein kompletter Entwurf der Mediengalerie erstellt. In der nächsten Phase wird bis zum 14.04.04 die Mediengalerie prototypisch als vertikaler Prototyp realisiert, der die primären Anforderungen abdecken wird. Parallel zur Implementierung werden Tests und eine erste Dokumentation des Codes durchgeführt. Nach Fertigstellung des Prototyps erfolgt ein Zwischenbericht, in dem die bis dahin erfolgte Dokumentation zusammengefasst, und das Ergebnis der Prototypenerstellung analysiert wird. Nach Analyse der Erfahrungen mit dem Prototypen wird in Abstimmung mit den Betreuern entschieden, ob im weiteren Verlauf beide oder nur eine der übrigen Anwendungen realisiert werden soll. Vorläufiger Abgabetermin für den Zwischenbericht soll der 28.04.04 sein. Zwischenzeitlich sollen dann schon erste Vorbereitungen für den Entwurf des Tippspiels/EPGs gemacht werden. Ab dem 29.04.04 findet dann eine parallele Entwicklung der Mediengalerie und des Tippspiels/EPGs statt. Bei der Mediengalerie soll ein Redesign und eine Erweiterung des Prototypen erfolgen, während ein Prototyp der anderen Anwendungen entwickelt wird. Die hauptsächliche Implementierungsphase der Mediengalerie soll dann gegen Anfang Juni abgeschlossen sein. So verbleibt im Anschluss noch ein genügend großer Zeitraum, in dem das Zusammenspiel alle Systemkomponenten überprüft werden kann. Hier sollen dann letzte Korrekturen von Fehlern im Code erfolgen und der Rest der Dokumentation fertig gestellt werden, so dass dann gegen Ende September das Programm inklusive Dokumentation abgabereif ist.
Die Prototypenerstellung des Tippspiels/EPGs erfolgt bis zum 14.07.04. Hierauf soll wiederum eine Analysephase folgen, deren Ergebnisse das weitere Vorgehen im Redesign beeinflussen. Das Redesign selbst soll dann im Zeitraum vom 15.07.04 bis zum 04.08.04 erfolgen. So bleibt auch hier ein Zeitraum von etwa einem Monat, in dem abschließende Tests, Fehlerkorrektur

und restliche Dokumentation beendet werden können. Ab dem 02.09.04 soll schließlich mit dem Endbericht begonnen werden, der dann das Projekt am 30.09.04 abschließt.

Nr	Datum	Ereignis
1	17.09.03 - 29.10.03	1.Seminarblock
2	30.10.03 - 03.12.03	Erstellen einer Anforderungsdefinition für drei Anwendungen auf der MHP
3	04.12.03 - 17.12.03	Einarbeitungsphase in die Entwicklungsumgebung
4	18.12.03 - 04.02.04	Entwurf der Mediengalerie
5	18.12.03 - 14.04.04	Dokumentation der Mediengalerie
6	05.02.04 - 14.04.04	Erstellung eines Prototyps der Mediengalerie
7	05.02.04 - 14.04.04	Test der Mediengalerie
8	01.03.04 - 02.03.04	Projektgruppenfahrt
9	Mitte März 2004	2.Seminarblock
10	15.04.04 - 28.04.04	Zwischenbericht
11	22.04.04 - 19.05.04	Entwurf des Tippspiels/EPGs
12	22.04.04 - 15.09.04	Dokumentation des Tippspiels/EPGs
13	29.04.04 - 09.06.04	Redesign der Mediengalerie und Erweiterung der Funktionalität
14	29.04.04 - 01.09.04	Dokumentation der Mediengalerie
15	20.05.04 - 14.07.04	Erstellung eines Prototyps des Tippspiels/EPGs
16	20.05.04 - 14.07.04	Test des Tippspiels/EPGs
17	10.06.04 - 01.09.04	Implementierung der Verbesserungen bzw. Erweiterungen der Mediengalerie
18	10.06.04 - 01.09.04	Test der Mediengalerie
19	15.07.04 - 04.08.04	Redesign des Tippspiels/EPGs und Erweiterung der Funktionalität
20	05.08.04 - 15.09.04	Implementierung der Verbesserungen bzw. Erweiterungen des Tippspiels/EPGs
21	05.08.04 - 15.09.04	Test des Tippspiels/EPGs
22	02.09.04 - 29.09.04	Endbericht
23	30.09.04	Ende der Projektgruppe

Tabelle 2.1: Liste der relevanten Termine

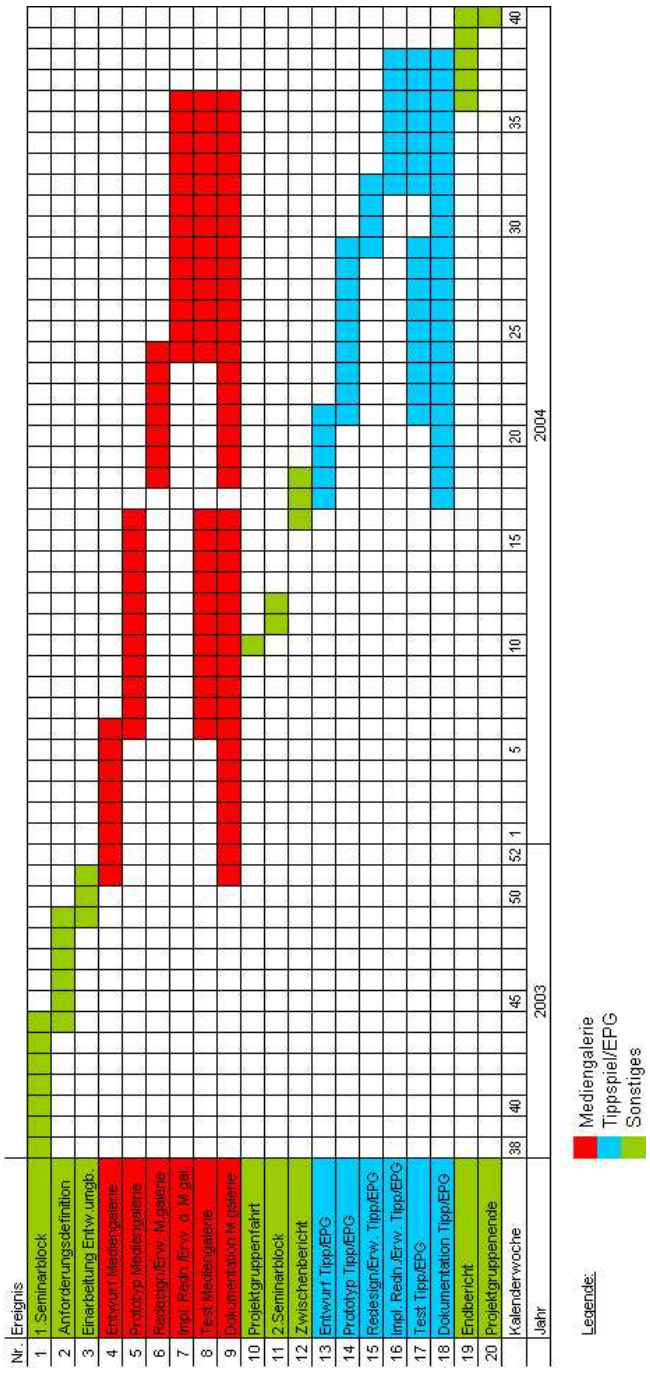

Abbildung 2.31: Zeitplan

Glossar

API: Eine *Application Programming Interface* (Dt.: Schnittstelle für die Anwendungsprogrammierung) ist eine definierte und dokumentierte Software-Schnittstelle zwischen zwei verschiedenen Programmen, meist zwischen einem Betriebssystem und einem Anwendungsprogramm.

Benutzerprofil: Datensammlung über einen Nutzer, mit deren Hilfe zum Beispiel dargebotene Inhalte auf den entsprechenden Nutzer angepasst werden können.

Broadcast: Beim Broadcasting werden Datenpakete ausgesendet an alle, die mit einem digitalen Rundfunkempfänger ausgestattet sind und nicht nur an einen bestimmten Empfänger [Zer03].

Client: Rechner oder Programm zum Abruf von Daten von einem Server über ein Computer-Netz.

Cookies: Kleine Informationsdateien, die von einer Homepage an den eigenen Web-Browser geschickt und von diesem auf der Festplatte des Anwenders gespeichert werden. Diese Informationen werden vom Browser bei einem erneuten Aufruf dieser Homepage zurückgeschickt. Dadurch können beispielsweise auf einigen Internet-Seiten eigene, individuelle Elemente erzeugt werden, die sofort erscheinen, sobald man diese Seite anwählt. Cookies gibt es auch auf der ↑MHP.

DAB: *DAB* ist die Abkürzung für *Digital Audio Broadcasting*. Es handelt sich dabei um ein standardisiertes Verfahren zur digitalen Übertragung von Radiosignalen. Neben einer verbesserten Klangqualität bietet die digitale Übertragungstechnik auch die Möglichkeit, zusätzliche Informationen zu übermitteln. Bei den Konsumenten hat sich DAB bisher nicht durchgesetzt.

DVB: *DVB* ist die Abkürzung für *Digital Video Broadcasting*. Hier handelt es sich allgemein um das Verfahren zur Verbreitung von Digitalem Fernsehen, welches vom DVB-Projekt entwickelt wurde. Verbesserte Übertragungstechniken sowie Fehlererkennungs- und -korrekturverfahren sorgen für eine bessere Bild- und Tonqualität (DVD-Qualität und Mehrkanalton). Aufgrund des Einsatzes von Datenreduktionsverfahren passen in einen analogen Übertragungskanal etwa vier digitale Programme.

DVB-C: *DVB-C* ist ein auf das Kabelfernsehen abgestimmter DVB-Standard, d. h. die Daten werden über das normale Kabelnetz verbreitet. Für den

Empfang benötigt der Zuschauer einen Kabelanschluss und eine DVB-C-taugliche Set-Top-Box.

DVB-S: *DVB-S* ist ein auf das Satellitenfernsehen abgestimmter DVB-Standard. Die Daten werden über die bestehenden Satelliten verbreitet, für den Empfang benötigt der Zuschauer einen Satelliten-Schüssel und eine DVB-S-taugliche Set-Top-Box.

DVB-T: *DVB-T* ist ein DVB-Standard, welcher die Verbreitung von Digitalem Fernsehen über terrestrische Sender ermöglicht. Für den DVB-T Empfang benötigt der Zuschauer eine Antenne (kleine Stabantenne, Zimmerantenne oder auch Dachantenne - je nach Signalstärke) und eine DVB-T-taugliche Set-Top-Box.

DSL: steht für *Digital Subscriber Line* und stellt ähnlich wie ein Modem eine Verbindung zum Internet oder anderen Diensten über das Telefonnetz her. Da beim DSL mehr Frequenzbereiche benutzt werden, steht eine wesentlich höhere Bandbreite zur Verfügung.

EPG: Der Begriff *EPG* steht für *electronic program guide* bzw. *elektronischer Programmführer*. Beim Digitalen Fernsehen werden neben den Video- und Audiodaten einige Informationen über das laufende Programm zum Zuschauer übertragen. Set-Top-Boxen verfügen in der Regel werksseitig über einen EPG, welcher die vorliegenden Informationen auf dem Bildschirm darstellt. Dies sind meistens kurze Informationen zum laufenden Programm oder zur jeweils nachfolgenden Sendung eines Senders. Der *EPG*, der von der Projektgruppe implementiert werden soll, geht über die beschriebene und übliche Funktionalität hinaus.

GUI: Englische Abkürzung für den Begriff *Graphical User Interface*. GUI bedeutet im Deutschen *Graphische Benutzungsoberfläche* und bezeichnet die visuelle Programm-Oberfläche, die aus Bildelementen besteht. Das GUI ist die *Benutzungsschnittstelle zwischen Anwender und Applikation*.

***Hello-World* -Prototyp:** Ein kleines Programm, welches den kurzen Satz *Hello World!* ausgibt. Es wird traditionell zu Testzwecken geschrieben, um sich mit der grundsätzlichen Struktur und den elementaren Möglichkeiten einer Programmiersprache vertraut zu machen.

Java-AWT: Das *Java Abstract Window Toolkit* ist eine umfangreiche Java-Klassenbibliothek zur Gestaltung von graphischen Benutzungsoberflächen.

Linux: Frei verfügbares Betriebssystem für PCs.

MPEG-2: Die Abkürzung *MPEG* steht für *motion picture experts group*. Dabei handelt es sich im Zusammenhang mit DVB um ein Datenkompressionsverfahren, welches bei der Verbreitung von Digitalem Fernsehen benutzt wird. Eingesetzt wird die zweite Norm dieses Verfahrens (MPEG-2), dabei wird die anfallende Datenmenge vor der Übertragung zum Zuschauer um durchschnittlich 96% reduziert. Das MPEG-2 - Verfahren ist daher eine wichtige Voraussetzung für das Digitale Fernsehen, denn ohne eine Datenkomprimierung würden die vorhandenen Übertragungskapazitäten nicht ausreichen.

Playout-Center: Es ist eine Funktionseinheit, in der Geräte zur Codierung, Multiplex-Bildung, Modulation und Satelliten-Ausstrahlung von digitalen Signalen enthalten sind. Die Sendeabwicklung und -überwachung der einzelnen Programme wird von Playout-Center vorgenommen[Zer03].

Proprietär: Herstellerspezifische, nicht-öffentliche Techniken und Verfahren.

PVR: *PVR* steht für *personal video recorder*. Dies ist ein Ersatz für den VHS-Recorder und nimmt TV-Sendungen auf eine Festplatte auf. Zu den üblichen Features gehört die Wahl der Aufnahmequalität, die Verwaltung der Aufnahmen und eine schnelle Rück- und Vorspulfunktion bei der Wiedergabe. Damit der Begriff *personal* an Bedeutung gewinnt, könnten *PVRs* in Zukunft Sendungen automatisch aufzeichnen, die den Zuschauer interessieren.

Redesign: Eine Phase im Entwicklungsprozess von Software, in der Entwurfsentscheidungen, evt. bis hin zu sehr grundsätzlichen, erneut überdacht, einer Prüfung unterzogen und evt. geändert werden.

Rückkanal: Es ist ein bidirektionaler Kanal, d. h. man kann sowohl Daten empfangen als auch senden. Innerhalb der ↑MHP dient es dazu, Informationen zum Diensteanbieter zurückzusenden und ermöglicht damit echte Interaktivität. Wird bei einer stationären ↑Set-Top-Box zumeist durch ein analoges Modem oder DSL, bei ↑DVB-T dagegen, damit die Mobilität nicht verloren geht, durch Umts oder GPRS, realisiert.

Workstation: Englische Bezeichnis für einen Rechner der meist zum Arbeiten eingesetzt wird.

SDK: Abkürzung für *Software Development Kit*. Umfassender Begriff für Entwicklungsumgebung, enthält meist alle zum Erstellen von Anwendungen notwendigen Programme und Bibliotheken.

Server: Rechner oder Programm, das in einem Netz zentrale Dienste zur Verfügung stellt.

Set-Top-Box: Zusatzgerät einschließlich Decoder zum Empfang digitaler Fernsehprogramme. Eine *Set-Top-Box* kann allen vorhandenen Fernsehgeräten vorgeschaltet werden, so dass sie auch für das digitale Fernsehen genutzt werden können [Ges03].

Schnittstelle: Mit *Schnittstelle* bezeichnet man sowohl den Verbindungspunkt zwischen zwei Hardware-Systemen (Geräten, Gerätekomponenten) als auch die Übergabestelle zwischen zwei Software-Systemen (zum Beispiel Anwendungsprogrammen, Datenübertragungsprotokollen, Datenbanken).

Trailer: Ein Kurzfilm, der den Versuch unternimmt, dem Zuschauer einen Eindruck vom eigentlichen Film zu vermitteln. Dieser wird häufig als Werbung für den Film verwendet.

Literaturverzeichnis

[Bal00] Helmut Balzert. *Lehrbuch der Softwaretechnik - Softwareentwicklung*. 2000. 1. Band.

[Bol04] Dietrich Boles. *Leitfaden zur Durchführung von Projektgruppen*. PDF-Datei, September 2002. URL http://www.informatik.uni-oldenburg.de/studium/leitfaeden/Leitfaden-PG.pdf - Letzter Zugriff: 14.02.04.

[Bre01] M. Brefort. Die Referenzimplementierung der Multimedia Home Platform. *Sonderdruck aus Fernseh und Kino-Technik*, 2001.

[DVB03] DVB Project Office. DVB - Digital Video Broadcasting - The DVB Project Office. http://www.dvb.org, 2003. letzter Zugriff: 27.Dezember 2003.

[ETS01] ETSI - European Telecommunications Standards Institute. http://www.etsi.org, 2001. letzer Zugriff: 27.Dezember 2003.

[Eur01] European Telecommunications Standards Institute. *Multimedia Home Platform (MHP) Specification 1.1*. Westedeutscher Verlag, November 2001.

[Eva98] Jean-Pierre Evain. *The Multimedia Home Platform - an overview*. EBU Technical Department, 1998.

[FKT00] FKTG - Fernseh- und Kinotechnische Gesellschaft e.V. *Fernsehen heute und morgen, Technik - Märkte - Strategien*, 3.Ausgabe, 2000.

[Fp03] Fernsehen-plus. Applikationen. http://www.mhp-forum.de, 2003. letzer Zugriff: 9.Dezember 2003.

[Ges03] Gesellschaft für Unterhaltungs- und Kommunikationselektronik. Das Lexikon mit allen wichtigen Stichworten der Consumer Electronics. http://www.gfu.de/pages/lexikon.html, 2003. letzer Zugriff: 9.Dezember 2003.

[Mul03] Multimedia Home Platform. DVB-MHP - Digital Video Broadcasting Multimedia Home Platform. http://www.mhp.org, 2003. letzter Zugriff: 27.Dezember 2003.

[SC01] Anthony Smith-Chaigneau. *DVB-MHP - A Snapshot - October 2001*. DVB Project Office, Oktober 2001.

[Sch03] Peter Schütz. Persönliches Fotoalbum über DVB-T -Empfang auf Settop-Boxen. Technical report, cewe digital GmbH, 2003.

[Sed01] R. Sedlmeyer. Multimedia Home Platform. *Sonderdruck aus Fernseh und Kino-Technik*, 2001.

[Ste03] Axel Steinbach. Diplomarbeit - Konzeption und Realisierung einer personalisierten Mediengalerie im Internet, 2003.

[Zer03] Frank Zervos. *Digitales Fernsehen in Deutschland*. Westedeutscher Verlag, 2003. 1. Auflage.

Unterschriften

Veranstalter:

Jun.Prof.Susanne Boll

Sabine Thieme

Ansgar Scherp

Teilnehmer:

Andre Dromowicz

Michael Hasseler

Sascha Heider

Jan Schoof

Sebastian Stockfleth

Daniel Thobe

Christian Wegener

Martin Willkomm

Carsten Zimmermann

Kapitel 3

Zwischenbericht

Thema dieses Dokumentes ist ein Zwischenbericht der Projektgruppe Show_Me_MHP, in dem der aktuelle Stand aufgezeigt wird. Dazu zählen die Änderungen an der Anforderungsdefinition, der grobe Entwurf und der aktuelle Zeitplan. Abgeschlossen wird die Projektgruppe mit einem Endbericht, in dem alle Arbeiten enthalten sein werden.

3.1 Kurzer überblick über die folgenden Kapitel

Nachdem in der Einleitung dem Leser grundlegende Einblicke in die Aufgabenstellung der Projektgruppe gegeben wurden, behandeln die nachfolgenden Kapitel folgende Thematiken:

Projektplanung: Analysiert den in der Anforderungsdefinition entworfenen Zeitplan, das bei der Implementierung verwendete Vorgehensmodell(e) und veranschaulicht die im Laufe der Projektgruppe zum Einsatz kommenden Entwicklungswerkzeuge.

Änderung an der Anforderungsdefinition: Geht auf alle Änderungen der Anforderungsdefinition ein. Dazu werden neue Screenshots der Mediengalerie beschrieben.

Entwurf: Beinhaltet die einzuhaltenen Code Conventions bei der Implementierung, die verwendete Architektur mit einer Schilderung der Programmschichten und Programmierparadigmen. Weiterhin erfolgt eine Veranschaulichung des statischen (Klassendigramme) und dynamischen (Sequenzdiagramme) Modells sowie die nötigen Nutzungsvoraussetzungen.

3.2 Projektplanung

Die Projektplanung dient dazu die im Zeitplan festgehaltenen Termine (für z.B. Anforderungsdefinition und Prototyp) zu überwachen und das verwendete Prozessmodell sowie den Werkzeugeinsatz zu dokumentieren.

3.2.1 Zeitplan

Die im Zeitplan festgehaltenen Termine konnten bisher ohne große Verzögerungen eingehalten werden. Die beiden Dokumente Anforderungsdefinition und speziell der Entwurf haben sich aufgrund ständiger Anpassungen an die Entwicklungsarbeiten, jeweils um wenige Wochen verschoben. Die folgende Tabelle

gibt Aufschluss über die bisher abgearbeiteten Aufgaben.

Nr	Datum	Ereignis
1	29.10.03 - 30.10.03	1.Seminarblock Powerpointvorträge
2	30.10.03 - 17.12.03	Erstellen einer Anforderungsdefinition für drei Anwendungen auf der MHP
3	04.12.03 - 17.12.03	Einarbeitungsphase in die Entwicklungsumgebung
4	18.12.03 - 27.03.04	Entwurf der Mediengalerie
5	18.12.03 - 14.04.04	Dokumentation der Mediengalerie
6	05.02.04 - 30.04.04	Erstellung eines Prototypen der Mediengalerie
7	05.02.04 - 30.04.04	Test der Mediengalerie
8	01.03.04 - 03.03.04	Projektgruppenfahrt
9	Anfang April 2004	2.Seminarblock
10	14.03.04 - Ende April	Zwischenbericht

Tabelle 3.1: Liste der relevanten Termine

Wie aus dem Vergleich mit dem Zeitplan aus der Anforderungsdefinition hervorgeht, wurden alle Termine außer obig genannter Ausnahme eingehalten. Die Projektgruppenfahrt wurde auf Wunsch der Projektgruppenteilnehmer um einen Tag verlängert. Der Zwischenbericht wird aufgrund momentan vorhandener personeller und zeitlicher Ressourcen um einen Monat nach vorne verschoben. Die Arbeiten an dem Prototypen sowie der Dokumentation und dem Test der Mediengalerie finden parallel statt und werden voraussichtlich bis zu den genannten Terminen abgeschlossen sein.

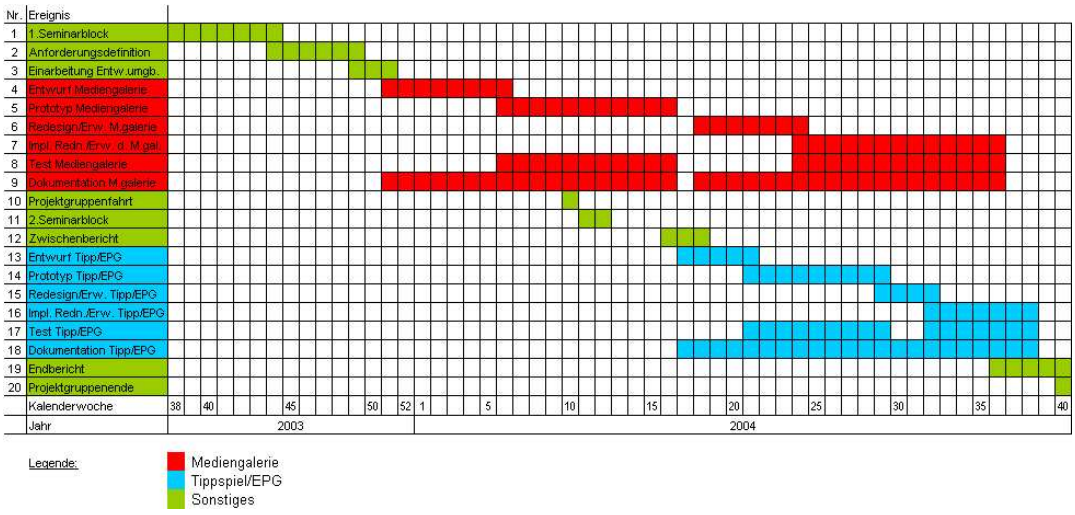

Abbildung 3.1: überarbeiteter Zeitplan

Im Vergleich zum vorherigen Zeitplan soll jetzt der Entwurf jeweils parallel zur Implementierung der Anwendungen durchgeführt werden, da sich im Verlauf der Programmierung des Prototyps herausstellte, dass der Entwurf häufig angepasst werden musste. Die zweite Windung wird nun farblich anders dargestellt, um zum einen die beiden Windungen deutlicher voneinander zu trennen, zum anderen aber auch um die Verteilung der Projektgruppenmitglieder auf die beiden Anwendungen zu verdeutlichen. Wie sich auch gezeigt hat, hat die Anforderungsdefinition mehr Zeit in Anspruch genommen als ursprünglich geplant und es gab keine klare Trennung zwischen der Einarbeitung in die Entwicklungsumgebung und Beginn der Implementierung des Prototyps.

3.2.2 Vorgehensmodell

Bei dem von uns favorisiertem Vorgehensmodell vermischen sich Elemente des klassischen *Spiralmodells*, des *Prototypenmodells* und des sogenannten *extreme Programming*. Die Entwicklung der prototypischen Mediengalerie soll in abgewandelter Form durch „Abarbeitung “ der einzelnen Windungen des Spiralmodells realisiert werden. Vorteilhaft ist hierbei die hohe Flexibilität, da für jede Windung einer Spirale ein eigenes Vorgehensmodell gewählt werden kann. Für die ersten Windungen der Spirale haben wir zunächst das Prototypenmodell vorgesehen, um erste Erfahrungen mit der neuen Entwicklungsumgebung zu sammeln. Für weitere Windungen ziehen wir unter anderem das Multimedia Vorgehensmodell in Betracht, was sich jedoch im Verlauf der Entwicklungen noch ändern kann falls sich ein anderes Vorgehensmodell als geeigneter erweist. Für die konkrete Implementierung im Detail ist das extreme Programming vorgesehen. Der Prototyp soll durch eine laufende Verfeinerung, wobei jede Verfeinerungsstufe einer Windung des Spiralmodells entspricht, immer weiter dem angepeilten Endprodukt angenähert werden. Die für jede Windung charakteristischen Schritte sollen dabei soweit wie möglich eingehalten werden:

1. Festlegung der Ziele und Randbedingungen für den Prototyp der jeweiligen Windung,
2. Evaluation der Alternativen hinsichtlich Zielen und Randbedingungen,
3. Auswahl der Alternative mit dem geringsten Restrisiko und Auswahl eines geeigneten Vorgehensmodells für die gewählte Alternative,
4. Review der ersten drei Schritte und Planung der nächsten Windung.

Als ein geeignetes Vorgehensmodell für die erste Windung halten wir das *pair programming* als wesentliches Merkmal des extreme Programming. Dies hat im wesentlichen drei Gründe:

- Fehlervermeidung,
- Förderung des Gruppengefühls und des gemeinsamen Austausches,
- Anpassung an Entwicklungsumgebung (es stehen nur zwei Referenzimplementierungen zur Verfügung)

Eine genaue Beschreibung der oben erwähnten Vorgehensmodelle ist in der Seminarausarbeitung *Prozesse zur Multimedia und Softwareentwicklung* unter Kapitel 2.2.1 und 2.3.2 von Sascha Heider zu finden. Eine Einteilung des Entwicklungsprozesses in Iterationsschritte, die den Windungen des Spiralmodells entsprechen, wurde vorgenommen, indem in der ersten Windung das Prototypenmodell verwendet wurde. Hier wurden erste Erfahrungen mit der Entwicklungsumgebung und dem Werkzeugeinsatz gesammelt. Die für jede Windung des Spiralmodells vorgesehenen Schritte zur Kontrolle der Arbeitsabläufe wurden aber nur teilweise befolgt. Die verschiedene Alternativen und Lösungsansätze für die Entwicklung eines Prototyps wurden diskutiert und es erfolgte eine Evaluation der Alternativen zum gewählten Vorgehensmodell. Jedoch die daran anschließenden Schritte wurden nicht konsequent umgesetzt. Es erfolgten keine Maßnahmen zur Reduktion der Risiken wie z.B. Machbarkeitsstudien. Die in der Einarbeitungsphase gesammelten Erfahrungen mündeten dann in dem Beginn der Entwicklungsarbeiten zum eigentlichen Prototyp der Mediengalerie. Dies kann man als Wechsel zu einer zweiten Windung des Spiralmodells betrachten. Jedoch fand das einem weiteren Iterationsschritt vorausgehende Review der ersten drei Schritte und die Planung der nächsten Windung kaum statt.
Die Konzepte des extreme Programming wurden dagegen wesentlich besser umgesetzt. Die Entwicklungsarbeiten werden durch einen Ansprechpartner des Kooperationspartners der Projektgruppe in Form von Herrn Schütz unterstützt. Durch ihn werden der Projektgruppe, während der gesamten Entwicklungsarbeit, Anregungen und Verbesserungsvorschläge mitgeteilt. Die Schnittstelle zu dem Kooperationspartner wird dabei durch einen Betreuer der Projektgruppe gebildet, die CeWe so fortwährend über den Fortschritt an den Entwicklungen der *personalisierten Mediengalerie* informiert. Die von CeWe die Datenbankanbindung und Verschlüsselung betreffenden Anforderungen, werden durch das der Projektgruppe zur Verfügung gestellte Dokument *XTCI-Schnittstelle* konkretisiert. Die Oberflächengestaltung ist zum

größten Teil der Kreativität der Projektgruppenteilnehmer überlassen. Jedoch richten wir uns bei der Anordnung von grafischen Elementen wie zum Beispiel den Eingabefeldern des Registrierungsbildschirms an schon bestehenden Anwendungen wie der Online Anmeldung für die Fotobestellung bei Schlecker aus. Eine Vorgabe ähnlich der XTCI-Schnittstelle liegt hier nicht vor. Auch die das extreme Programming und insbesondere das pair Programming charakterisierenden Arbeitsabläufe werden durch die Projektgruppenmitglieder eingehalten. Die Entwicklung, insbesondere der Oberflächen, aber auch die Datenbankanbindung betreffend, erfolgt immer durch ein Team aus 2-3 Projektgruppenmitgliedern. Dadurch erreichen wir eine höhere Fehlervermeidung, eine höhere Verständlichkeit des Codes und eine höhere Motivation der einzelnen Gruppenmitglieder.
Das für zukünftige Windungen der Spirale verwendete Vorgehensmodell setzt sich mit einer großen Wahrscheinlichkeit aus mehreren Modellen zusammen. Gründe dafür sind, dass auf immer neue Anforderungen eingegangen werden muss. Das pair Programming wird auf jeden Fall beibehalten werden, da es sich auf die Gruppenarbeit und den Fortgang der Entwicklungsarbeiten als positiv erwiesen hat. Zusätzlich ist es denkbar Elemente des Multimedia Vorgehensmodells zum Einsatz kommen, weil ein Hauptaugenmerk bei der Oberflächengestaltung liegt.

3.2.3 Werkzeugeinsatz

Ohne die Entwicklung unterstützende Werkzeuge ist ein professionelles Design von MHP-Anwendungen nicht möglich. Die Projektgruppe stützt sich auf eine Reihe von frei verfügbaren und handelsüblichen Entwicklungswerkzeugen, die im Folgenden mit ihren Funktionalitäten aufgelistet werden:

CVS: CVS dient dazu, mehreren Entwicklern zeitgleich die Arbeit an einem Projekt zu ermöglichen. Alle zu einem Projekt zugehörigen Dateien werden auf einem z.B. Linux basiertem Server archiviert. Der Zugriff auf Dateien erfolgt durch Befehle auf der Ebene der Kommandozeile oder durch Windows kompatible Programme wie WinCVS (im Falle der Projektgruppe). Es können sowohl einzelne Dateien als auch ganze Verzeichnisse hinzugefügt (*commit*) und ausgelesen (*update*) werden. Bei jeder Aktualisierung einer Datei im CVS Repository versieht das System die entsprechende Datei mit einer neuen Versionsnummer.

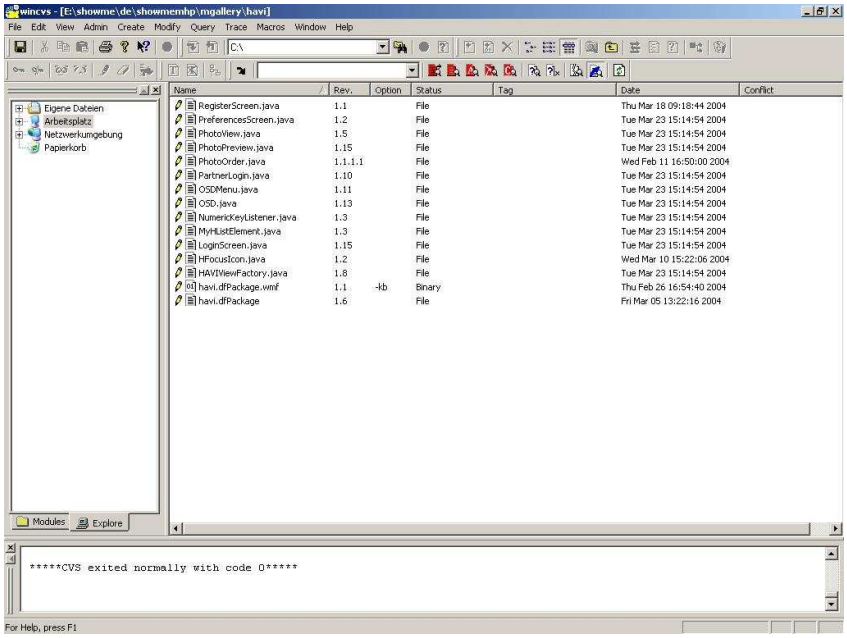

Abbildung 3.2: Screenshot CVS

FreePDF2: Erzeugt hoch auflösende (1200dpi) pdf Dateien aus fast jedem Dokumententyp.

Netbeans: Ist eine kostenlose integrierte Entwicklungsumgebung, die wichtige Codefragmente (z.B. Klammern) farbig darstellt, die Fehlererkennung durch Tooltips unterstützt, neue Klassen in den bestehenden einbindet und die Kompilierung einzelner, mehrerer oder ganzer Pakete zulässt.

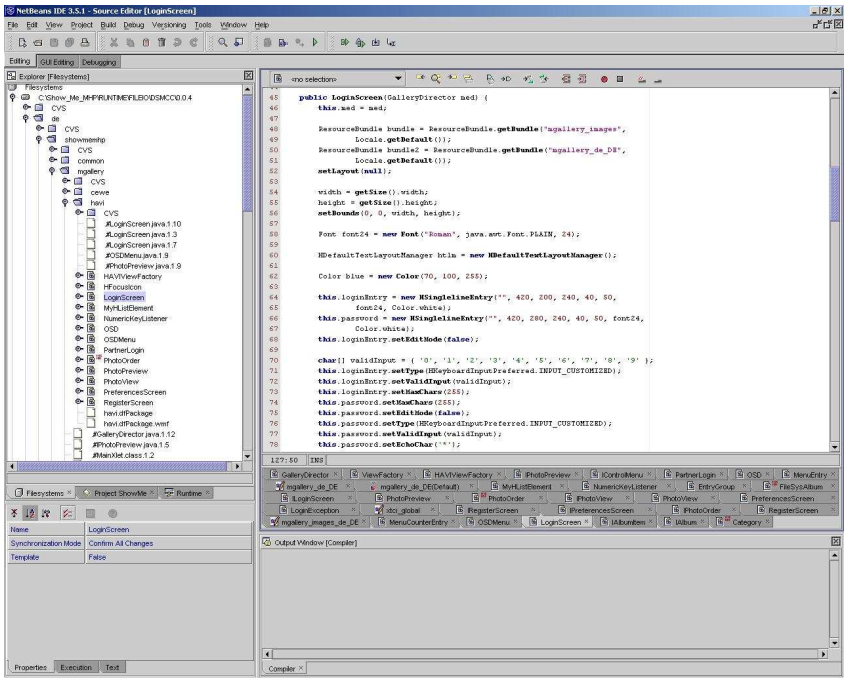

Abbildung 3.3: Screenshot Netbeans

Powerpoint: Dient zur Erstellung von Präsentationen. Beispiele hierfür sind die Vorstellung der einzelnen Seminarthemen durch die Projektgruppenmitglieder sowie die Vorträge während der Berlinfahrt beim Frauenhofer-Institut. Weiterhin wurden mit Hilfe von Powerpoint erste Studien zur Oberflächengestaltung vorgestellt.

TogetherJ: Grafisches Werkzeug zur Erstellung von Klassendiagrammen und Sequenzdiagrammen. Während der grafischen Erstellung von Klassendiagrammen wird durch TogetherJ zeitgleich Java Code erzeugt. Weiterhin ist ein sogenanntes *Roundshipping* möglich, wodurch selbst geschriebener Code dem Klassendiagramm hinzugefügt wird. Zusätzlich unterstützt es den Export von Diagrammen als GIF Dateien. So lassen sich Screenshots von den einzelnen Diagrammen in Dokumente wie die Anforderungsdefinition oder den Entwurf einbinden und erhöhen so deren übersichtlichkeit und das Verständnis des Gesamtkonzeptes.

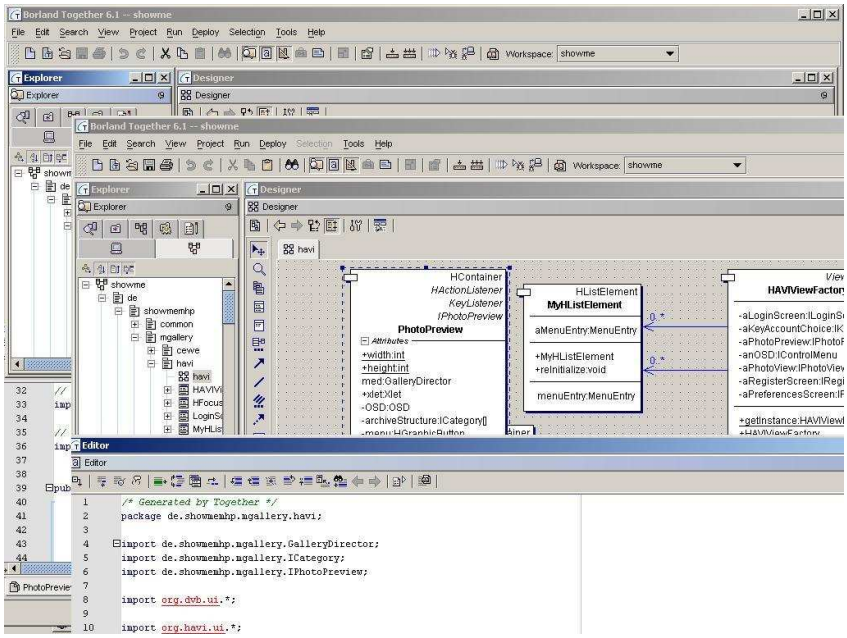

Abbildung 3.4: Screenshot TogetherJ

LaTeX: Textverarbeitungsprogramm von Knuth, welches das Schreiben von strukturierten und einheitlichen Dokumenten ermöglicht. Sämtliche abzuliefernden Dokumente (Anforderungsdefinition, Entwurf und Seminarausarbeitungen) der Projektgruppe werden mit diesem Programm erstellt. Unter anderem auch diesen Zwischenbericht, wie der geneigte Leser erkennen kann.

WinEdt: Texteditor der die Kompilierung und Anzeige von LaTeX Dokumenten gestattet sowie Adobe pdf Dokumente erzeugt.

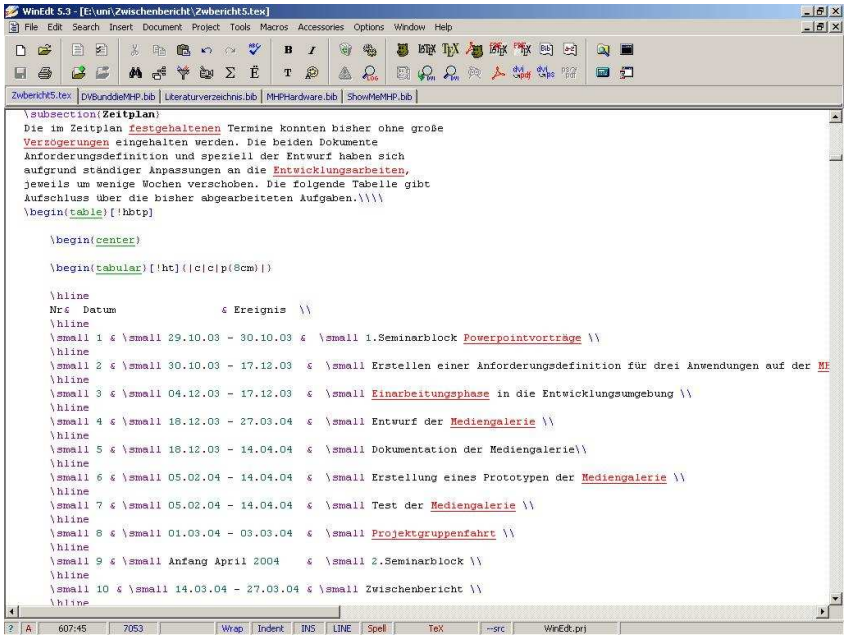

Abbildung 3.5: Screenshot WinEdt mit Latex Quellcode

Referenzimplementierung: Die beiden der Projektgruppe zur Verfügung stehenden Lizenzen sind die Voraussetzung für die Entwicklung von MHP Anwendungen, weil sie die nötigen MHP Klassen enthält und die Möglichkeit bietet, kompilierte MHP Xlets anzuzeigen.

Weitere Entwicklungsgwerkzeuge, die eventuell im Verlauf der Arbeit der Projektgruppe hinzukommen, finden sich dann im Abschlußbericht.

3.3 änderung an der Anforderungsdefinition

Die Anforderungsdefinition hat insbesondere was die Oberflächengestaltung betrifft einige wesentliche änderungen erfahren. Gründe hierfür sind die Rücksprache mit dem Kooperationspartner CeWe, externe Beratungen mit Mitarbeitern des OFFIS und Eindrücke, die auf der PG-Fahrt in Berlin gesammelt wurden. Im Folgenden werden wir nun die Modifikationen an den einzelnen Oberflächen beschreiben und einen neuen Screenshot als Vergleich beifügen. Zunächst beschreiben wir die änderungen, die alle Oberflächenmodule betreffen:

3.3.1 Allgemeines

änderungen die alle Oberflächenmodule betreffen sind folgende:

OSD-Menü: Komponente die zu jeder Oberfläche die zugehörige Menüfläche enthält. Der Funktionsaufruf ist dabei fest an den HauptmenüButton geknüpft.

Navigationsleiste: Wurde komplett überarbeitet. Anstatt von nummerierten Buttons verwenden wir nun farbige *GraphicButtons*, die den auf jeder Fernbedienung zu findenden Funktionstasten entsprechen. Der Farbcode für die Hilfefunktion (blau), und die Auswahl des OSD Menüs (rot) sind farblich strikt festgelegt, um dem Benutzer die Bedienung zu erleichtern.

Kopfleiste&Hintergrundbild: Wurden farblich und gestalterisch den Anforderungen an die Optik und die Navigationsleiste angepasst. Das sich in der Kopfleiste befindliche CeWe-Logo ist ebenfalls überarbeitet worden.

3.3.2 LoginScreen (Screenshot)

Tastenfeld: Besteht nun aus drei Reitern über die das große und kleine Alphabet bzw. die Sonderzeichen getrennt voneinander ausgewählt werden können. Die Eingabe erfolgt nun ähnlich des Tastencodes beim Handy über eine Eingabemaske, die bei Auswahl des entsprechenden Eingabefeldes (E-Mail und Passwort) geöffnet wird.

Neuanmeldung: Ist neu hinzugekommen. Wird über die grüne Funktionstaste der Fernbedienung aufgerufen.

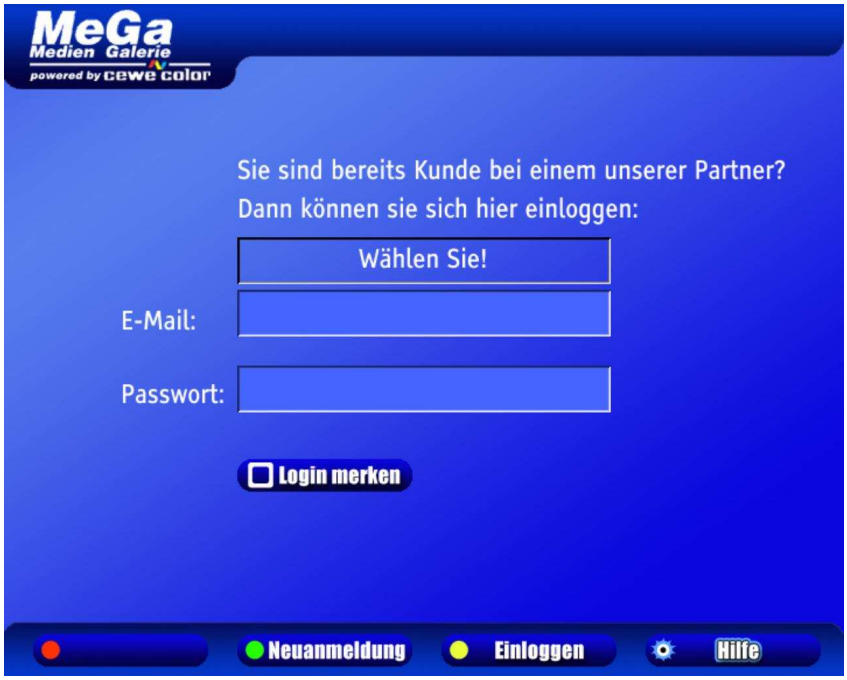

Abbildung 3.6: Screenshot LoginScreen

3.3.3 PartnerLogin (Screenshot)

Bei den änderungen im PartnerLogin handelt es sich im wesentlichen um die unter allgemeines aufgezeigten Gesichtspunkte.

Abbildung 3.7: Screenshot PartnerLoginScreen

3.3.4 Thumbnails (Screenshot)

Navigation: Die Auswahl der Thumbnails erfolgt nicht mehr über die Navigationsleiste, sondern über die Pfeiltasten der Fernbedienung.

Bildschirmaufteilung: Der Anteil der Thumbnails auf dem Bildschirm nimmt eine größere Fläche ein.

OSD-Menü: Die Auswahl der Alben erfolgt nun über das OSD Menü, welches über den Hauptmenü-Button aufgerufen wird.

Abbildung 3.8: Screenshot Thumbnailansicht

3.3.5 Vollbild (Screenshot)

Größe des Bildes: Nimmt nun nicht mehr den gesamten Bildschirm ein, da sonst die Menüführung verloren gehen könnte.

Abbildung 3.9: Screenshot Vollbildansicht

3.3.6 RegisterScreen (Screenshot)

Völlig neuer Screen, der den oben beschriebenen Oberflächen in Aufbau und Struktur gleicht.

Abbildung 3.10: Screenshot RegisterScreen

3.3.7 Ausblick (noch zu implementierende Oberflächen)

An noch zu realisierenden Oberflächen stehen noch die Hilfefunktion und der Warenkorb an.

3.4 Entwurf der Mediengalerie MeGa

3.4.1 Einleitung

3.4.1.1 Ausgangssituation und Zielsetzung

Die Projektgruppe hat sich entschieden zunächst mit der Implementierung der *personalisierten Mediengalerie* prototypisch zu beginnen. Aufbauend auf diesen Erfahrungen, insbesondere was die Oberflächengestaltung und die Anbindung zu den Servern zu CeWe Color betrifft, wird der Prototyp im Zusammenspiel mit dem Partner CeWe Color und Fachkräften[1]außerhalb der Projektgruppe überarbeitet und vervollständigt. Die Mediengalerie soll Funktionen bieten, die das Anschauen, Bearbeiten und Bestellen von Fotos bequem vom Wohnzimmer aus ermöglichen. Ein Hauptaugenmerk liegt vor allem bei der Einhaltung der Schnittstellen, die vom Kooperationspartner vorgegeben werden. So soll z. B. die Archivierung von Fotos auf hauseigenen Servern von CeWe realisiert werden. Parallel zu der Vervollständigung des Prototyps der *personalisierten Mediengalerie* beginnen dann die Arbeiten an einer der zu Auswahl stehenden Anwendungen *EPG* oder *personalisiertes Fußball Tippspiel*. Ob alle drei Anwendungen implementiert werden, hängt von dem Voranschreiten der Entwicklung an der *personalisierten Mediengalerie* und den daraus gewonnenen Erfahrungen ab. Grundfunktionalitäten des Tippspieles sollen dabei das Tippen der einzelnen Gruppenspiele bzw. Finals, ein individuelles Benutzerprofil und eine Ranglistenfunktion sein. Die Entwicklung eines weiteren EPGs könnte auf den ersten Blick überflüssig erscheinen, jedoch sind bisherige EPGs auf ein Programm beschränkt. Funktionen, die eine Anpassung des geplanten Programms an individuelle Bedürfnisse bieten und die alle Programme berücksichtigen, sind Neuheiten.

3.4.1.2 Aufbau des Entwurfs

Der hier vorgelegte Entwurf stellt die Arbeitsgrundlage für die Implementierung der Anwendungen Mediengalerie, EPG bzw. Tippspiel dar. In diesem ersten Kapitel wird ein erster überblick über die Vorgaben der Projektgruppe gegeben. Im Kapitel 4.2.2 werden die Konventionen für die Formatierung des Quellcodes festgelegt und eine Begründung gegeben aus welchem Grund dies von Bedeutung ist.
Die einzelnen Module der verschiedenen Anwendungen lassen sich in unterschiedliche Schichten einteilen. Dies sind die Dialogschicht, Anwendungungs-

[1]Beratung bei der Oberflächen gestaltung

logik (Mediator), Client Kommunikation, Server Kommunikation und die Datenbankschicht. Die näheren Beschreibungen sind in einem eigenen Kapitel 4.3.2 in diesem Entwurf untergebracht.
In den folgenden beiden Kapiteln werden die Module die implementiert werden sollen näher beschrieben. In dem Kapitel 4.3.3 werden die einzelnen Klassen der jeweiligen Anwendungen beschrieben, wobei gemeinsame Klassen zwischen den einzelnen Modulen in einem Unterkapitel betrachtet werden.
Das Kapitel 4.3.4 enthält die Sequenzdiagramme, in denen u.a. das dynamische Verhalten beim einloggen aufgezeigt wird.

3.4.2 Programmierrichtlinien

Es gibt einige gute Gründe sich beim Entwurf einer Software auch Konventionen für die Formatierung des Quellcodes zu überlegen:

- Der größte Teil des Arbeitsaufwandes (ca. 80 Prozent) im Lebenszyklus liegt in der Wartung und Pflege.
- Selten kann die Software während des gesamten Lebenszyklus nur vom Original-Autor gepflegt werden.
- Code Konventionen tragen erheblich zur Lesbarkeit des Codes bei und vereinfachen somit die Einarbeitung sowohl in fremden Code als auch in den eigenen Code, den man ein paar Wochen oder Monate nicht mehr angerührt hat.

In unserem Fall entwickeln wir unsere Software sogar im Team und jedes Mitglied ist motiviert so viel wie möglich zu lernen und den Quellcode des anderen möglichst schnell zu verstehen. Aus diesen Gründen haben wir uns auf die Einhaltung der weit verbreiteten Code Conventions von Sun geeinigt. Des weiteren sollen noch folgende Regeln eingehalten werden:

- Bezeichner, Methodennamen etc. sollen englisch sein.
- Kommentare sollen in Deutsch verfasst sein.
- Aufgrund besserer Lesbarkeit sollen Bezeichner innerhalb Methoden an der möglichst spätesten Stelle deklariert werden.
- Auf import-Anweisungen mit „*“ soll verzichtet werden. Jede Klasse soll also eine eigene import-Zeile bekommen.

3.4.3 Architektur

3.4.3.1 Programmschicht

In diesem Abschnitt wird zum besseren Verständnis eine Unterteilung der Software in mehrere Schichten vorgenommen. Dabei wird das gesamte Softwaresystem, also sowohl die Client-, als auch die Serverseite inklusive der Datenbank betrachtet. Der Client besteht somit aus

- einer Dialogschicht, die eine Schnittstelle zum Benutzer darstellt,
- einer Schicht welche die Anwendungslogik kapselt
- und einer Kommunikationsschicht, die einen Datenaustausch zwischen verschiedenen Servern ermöglicht.

Zwischen diesen drei Schichten wird mittels sogenannter Datenzugriffsklassen kommuniziert. Objekte dieser Klassen werden immer in der Kommunikationsschicht produziert und enthalten oder produzieren Daten, die sowohl aus internen als auch externen Datenquellen stammen können. So kann das Anwendungsprogramm also z.B. Daten aus dem Filesystem der Settop-Box oder über eine http-Verbindung beziehen. Client und Server müssen, genauso wie Server und Datenbank, über ein bestimmtes Protokoll miteinander kommunizieren und werden in jeweils einer Schicht dargestellt.

3.4.3.2 Mediatorprinzip und Module

Eine zweite Unterteilung der Client-Klassen erfolgt mittels Modulen. Es gibt lediglich zwei unterschiedliche Typen von Modulen. Zum einen gibt es Module der Dialogschicht und zum anderen Module der Kommunikationsschicht. Die Aufgaben der Module werden mittels Schnittstellen spezifiziert. Beide Typen von Modulen haben eine Referenz auf den Mediator. Der Mediator hat die Aufgabe die Anwendungslogik zu kapseln. Dies hat zur Folge, dass der Mediator sowohl den Verkehr von Datenzugriffsobjekten, als auch den Programmablauf kontrolliert. Nur über den Mediator können die Datenzugriffsobjekte also in die Module der Dialogschicht gelangen, in der sie für den Benutzer sichtbar und zugreifbar gemacht werden können.
Da offensichtlich eine klare Trennung zwischen Datenhaltung und Datendarstellung existiert, kann diese Architektur auch dem Model-View-Controller-Konzept entsprechen. Voraussetzung dafür ist, dass es eine Möglichkeit gibt, die gesamte Dialogschicht austauschen zu können. Bei der Definition von Modulen wurde daher darauf geachtet, dass Module eindeutig entweder der Kommunikations- oder der Dialogschicht zuzuordnen sind. Der Austausch

der gesamten Dialogschicht wird zudem durch eine AbstractFactory erleichtert. Eine abstraktes Moduldiagramm zeigt Abbildung 4.1:

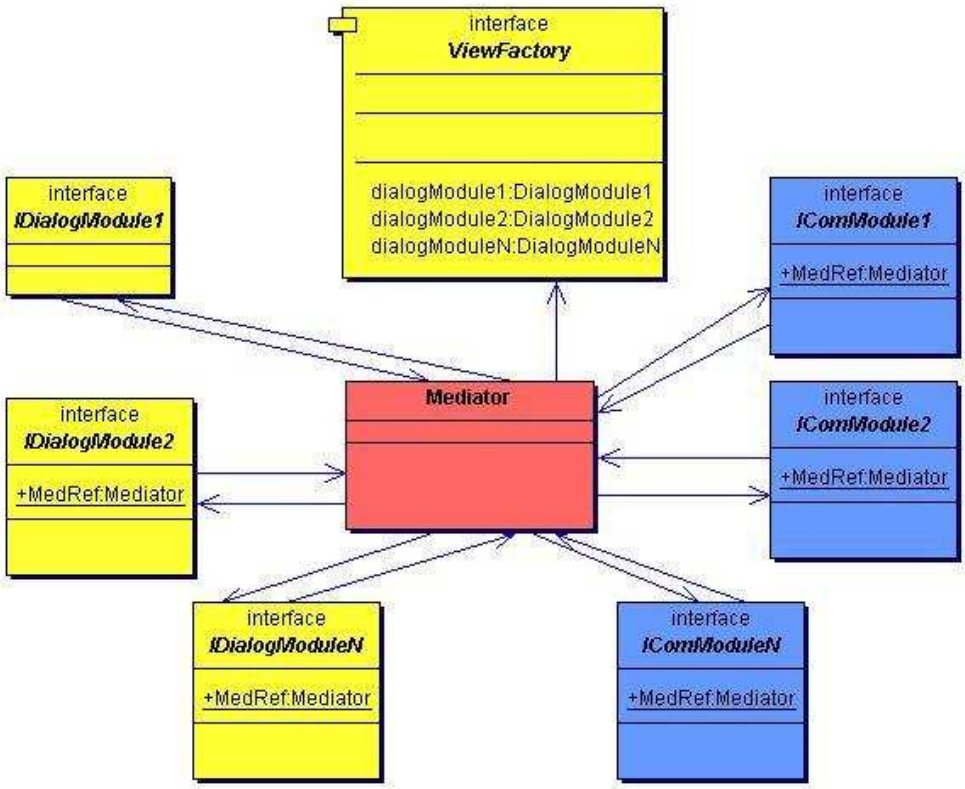

Abbildung 3.11: Mediator-Architektur

3.4.3.3 Pakete

Eine sinnvolle Paket-Hierarchie für die Aufteilung des Clients ergibt sich durch folgende Regelungen:

- Als Präfix dient de.showmemhp
- Die Anwendungen sollen in den Unterverzeichnissen mgallery, tipp2win und myepg ihren Platz haben. Allgemeine Klassen, die für alle drei Anwendungen von Bedeutung sind, befinden sich in einem Unterverzeichnis common.

- In den Verzeichnissen der Anwendungen sollen sich jeweils nur Klassen des Xlets, des Mediator sowie Modul-Schnittstellen und Datenzugriffsklassen befinden. Eine Ebene tiefer sollen sich, in einer logischen Unterteilung, Klassen zur Realisierung der Modul-Schnittstellen befinden.

Nach Anwendung dieser Regelungen könnte die Verzeichnisstruktur folgendermaßen aussehen:

```
./de/showmemhp/common de/showmemhp/common/havi
./de/showmemhp/common/confclient

./de/showmemhp/mgallery .de/showmemhp/mgallery/havi
./de/showmemhp/mgallery/cewe

./de/showmemhp/tipp2win de/showmemhp/tipp2win/havi

./de/showmemhp/myepg de/showmemhp/myepg/havi
```

3.4.3.4 Internationalisierung

Mehrsprachigkeit wird durch die Verwendung des I18N-Konzepts von Sun realisiert. Alle drei Anwendungen können dadurch auf ein gemeinsames und anwendungsspezifisches ResourceBundle zugreifen. Das Präfix für das ResourceBundle ist mit der Unterpaketbezeichnung der Anwendung, bzw. dem Unterpaket common identisch. Die ResourceBundle haben ihren Platz im Wurzelverzeichnis der Paketstruktur. Somit können zum Beispiel folgende Property-Dateien jeweils ein deutsches ResourceBundle in deutscher Sprache realisieren:

```
./common_de_DE.properties ./mgallery_de_DE.properties
./tipp2win_de_DE.properties ./myepg_de_DE.properties
```

Der Zugriff auf die ResourceBundles in den einzelnen Klassen erfolgt über den Aufruf statischer Methoden. Dies könnte folgendermaßen aussehen:

ResourceBundle bundle = ResourceBundle.getBundle
(”mgallery”,Locale.getDefault());

3.4.3.5 Gemeinsamkeiten der drei Anwendungen

Die Gemeinsamkeiten der zu realisierenden Anwendungen liegen vor allem bei der gemeinsamen Nutzung des *Mediator-Prinzips* zur Kapselung der Pro-

grammlogik und des MVC-Prinzips zur Unterteilung der Klassen in Aufgabenpakete bzw. zur Gewährleistung der Austauschbarkeit einzelner Programmschichten (z.B. der Dialogschicht für die Oberflächengestaltung) Die Funktionsweise des Mediator und des MVC-Prinzips wurden in Kapitel 4.2.3.2 hinreichend beschrieben. Auf der Modulebene gibt es ebenfalls Gemeinsamkeiten bei allen drei Anwendungen, welche ihre Ausprägungen in gemeinsamen Interfaces für Module der Dialogschicht und Kommunikationsschicht finden. Weiterhin planen wir alle drei Programme mehrsprachig zu implementieren, wofür wir die in Kapitel 4.2.3.4 beschriebenen *ResourceBundles* benutzen. Der programmtechnische Einsatz mit Interfaces für Datenhaltung bzw. GUI-Klassen sieht wie folgt aus:

Datenhaltung: Eine Arbeitsersparnis und Kapselung von Funktionen planen wir durch eine gemeinsame Datenhaltung der personalisierten Benutzerdaten zu erreichen. Die Daten sollen auf einem gemeinsamen Datenbankserver abgespeichert und Anfragen sowie Einträge von der gemeinsamen Datenbankklasse *ConfigurationClient* vorgenommen werden. Das entsprechende Interface *IConfigurationClient* stellt die Methodenrümpfe zur Verfügung, die dann in der Klasse ConfigurationClient (bzw. CeWe Client im Falle der Mediengalerie) in den entsprechenden Anwendung implementiert werden.

GUI-Klassen: Da alle Oberflächen für die MHP auf die HAVI [2] zurückgreifen, existiert für jedes Oberflächenmodul im Rahmen des MVC Konzeptes ein eigenes Interface. Referenzen dieser Interfaces für die Oberflächenmodule werden in der *ViewFactory* erzeugt. Ein konkretes, bei allen drei Anwendungen verwendetes Modul der Dialogschicht ist der *LoginScreen*. Das Interface stellt die Methoden und Konstanten zur Verfügung, die dann in den korrespondierenden *HAVI* Paketen zu Mediengalerie, EPG oder Tippspiel ihre spezifische Ausprägung finden.

Letztendlich wird sich bei allen drei Anwendungen eine Paketstruktur finden, in der die Implementationsdetails für gemeinsame Module, wie z.B. die GUI oder der Datenaustausch mit dem Kooperationspartner CeWe Color, gekapselt sind.

[2] Home Audio Video Interoperability

3.4.4 Detailliertes statisches Modell

3.4.4.1 übersicht der Klassen

Die folgenden Abbildungen stellen die Grundlage für die spätere Implementierung dar. Das Klassendiagramm wird zwecks besserer übersicht in die einzelnen Diagramme für *Model View Control*, *Datenzugriffsklassen für Alben* und *Datenzugriffsklassen für den Bestellvorgang* unterteilt.

Das Konzept des Model View Control wurde in Kapitel 4.3.2 Abschnitt 4.2.3.2 bereits hinreichend beschrieben. Die in der Abbildung 4.15 zu sehenden, farblich unterteilten Klassen, sind folgendermaßen zu interpretieren:

- rot markierte Klassen: beinhaltet den GalleryDirector, der in dem Model View Concept als *Controller* fungiert.
- gelb markierte Klassen: umfassen alle Klassen die für bildliche Visualisierung benötigt werden. Zu jeder Klasse existiert ein Interface, dass Konstanten und Methodenrümpfe zur Verfügung stellt und in der Klasse *ViewFactory* erzeugt wird. Die Implementierung der GUI Klassen sind in dem Paket *havi* gekapselt, um das Erweitern und Austauschen von Klassen einfach zu halten.
- blau markierte Klassen: bilden die *Modelkomponente* des Model View Controller Konzepts. Dazu gehören die vier Interfaces *IOnlinePrintClient*, *IConfigurationClient*, *IMediaClient* und *IMediaSource*, die über das Paket *cewe* auf die auf den Servern des Kooperationspartners CeWe liegenden Daten zugreifen können.

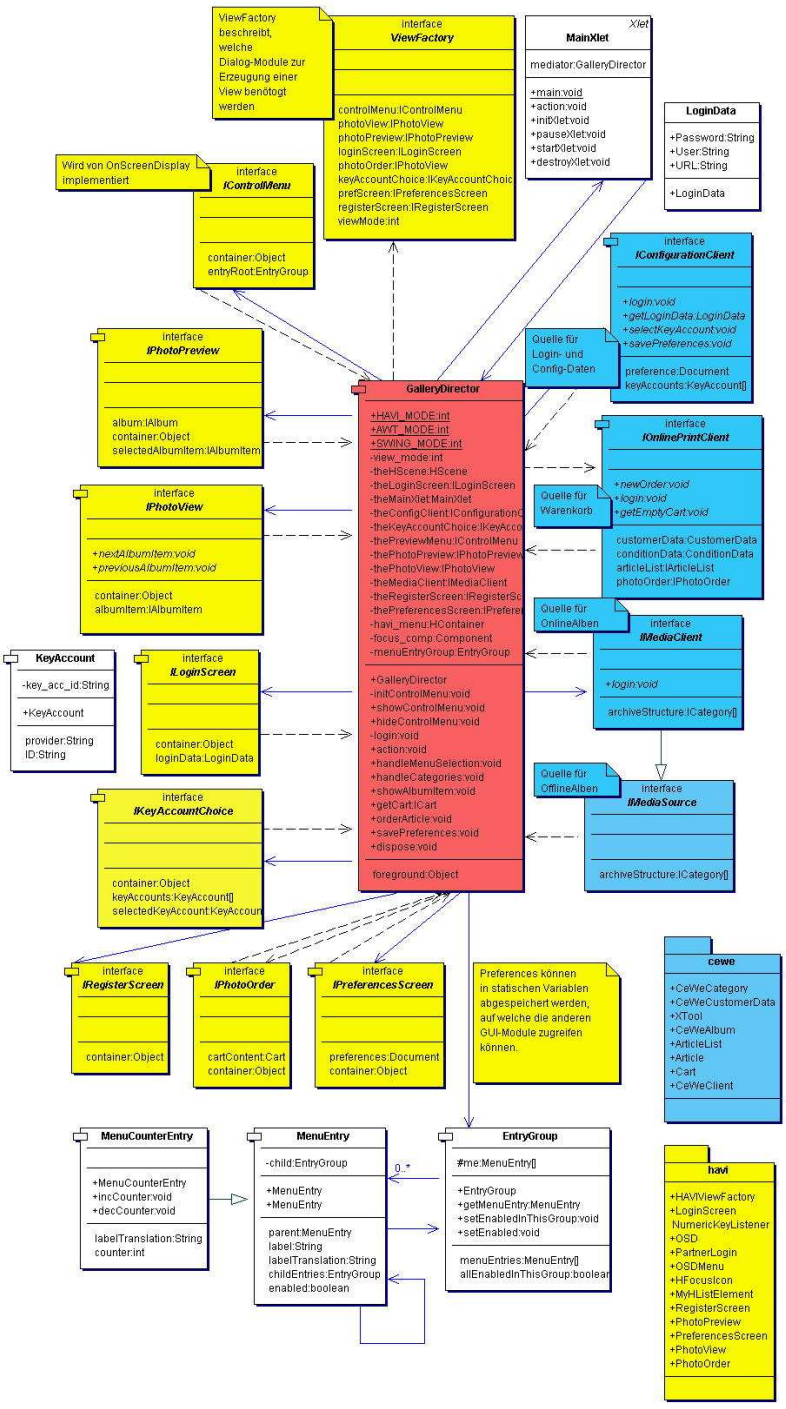

Abbildung 3.12: Model View Controller Diagramm

Die folgende Abbildung 4.6 zeigt die Datenzugriffsstrukturen für Online und Offline Alben.

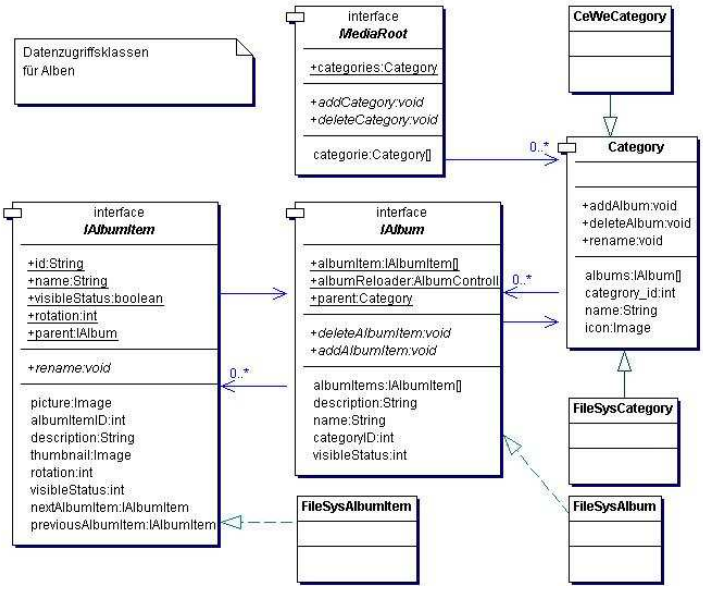

Abbildung 3.13: Datenzugriffsklassen für die Alben

Ein Zugriff erfolgt über eine Kategorie (Klasse *Category*), diese enthält bestimmte Alben (Interface *IAlbum*)in denen die entsprechenden Bilder (Interface *IAlbumItem*)zu finden sind.

Die abschließende Abbildung 4.7 des Klassendiagramms zeigt die Datenzugriffsstrukturen des Bestellvorganges.

Das Interface ICartContent enthält Methodenrümpfe die in der Klasse CartContent implementiert werden und dazu dienen Artikel dem Warenkorb hinzuzufügen oder zu entfernen. Einer Artikelliste (Klasse *IArticleList*) werden 1..n Artikel (Klasse *Article*) zugeordnet, die dann dem Warenkorb hinzugefügt werden können.

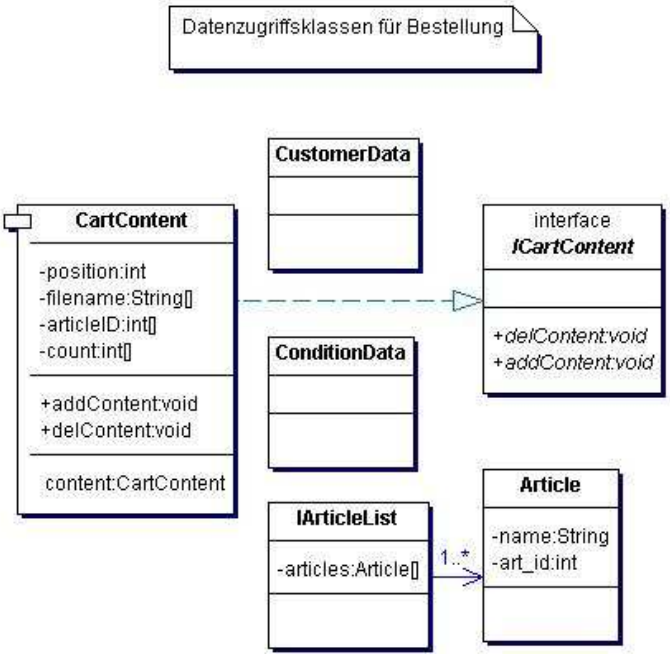

Abbildung 3.14: Datenzugriffsklassen für Bestellvorgang

3.4.4.2 Klassen der Dialogschicht

ViewFactory:

- Beschreibung: Das Interface ViewFactory wird in dem Paket HAVI in der Klasse HAVIViewFactory implementiert und dient als AbstractFactory. Mittels dieser Klasse soll es ermöglicht werden die Views einfach auszutauschen.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): HAVIViewFactory
- Attribute:
  - public static HAVIViewFactory getInstance()

  - private ILoginScreen aLoginScreen
  - private IKeyAccountChoice aKeyAccountChoice
  - private IPhotoPreview aPhotoPreview
  - private IControlMenu aOSD
  - private IPhotoView aPhotoView
  - private IRegisterScreen aRegisterScreen
  - private IPreferencesScreen aPreferencesScreen

  Die folgenden Methoden liefern jeweils die entsprechenden GUIs zurück:

- Methode public IConfigurationClient getConfigClient()
- Methode public IControlMenu getControlMenu()
- Methode public IPhotoPreview getPhotoPreview()
- Methode public ILoginScreen getLoginScreen()
- Methode public IPhotoView getPhotoOrder()
- Methode public IKeyAccountChoice getKeyAccountChoice()
- Methode public IPreferencesScreen getPrefScreen()
- Methode public IRegisterScreen getRegisterScreen();

IKeyAccountChoice:

- Beschreibung: Das Interface enthält Methodenrümpfe für die Implementierung des Auswahlmenüs in dem der CeWe Partner ausgewählt werden kann. Dieses Interface wird in dem Paket HAVI in der Klasse PartnerLogin implementiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): KeyAccountChoice
- Attribute:

  - private GalleryDirector mediator

- Methode public Object getContainer()

  Holt das aktuelle Containerobjekt.

- Methode public void setKeyAccounts(KeyAccount[] ka)

  übergibt die CeWe Partner, bei denen der Anwender einen Account besitzt.

- Methode public KeyAccount getSelectedKeyAccount()

  Gibt den im Auswahlmenü selektierten KeyAccount zurück.

LoginData:

- Beschreibung: Diese Klasse enthält die benötigten Login Daten wie das Passwort, den Benutzernamen und die URL.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Attribute.

- Oberklasse(n): keine
- abgeleitete Klasse(n): keine
- Attribute:
  - public String Password
  - public String User
  - public String URL

ILoginScreen:

- Beschreibung: Die GUI des Logins in dem der Benutzer sein Passwort und seine e-mail Adresse eingeben muss um in das Hauptmenü zu gelangen. Die Klasse LoginScreen befindet sich im Paket HAVI.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): LoginScreen
- Attribute:
  - private GalleryDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButton und andere Elemente auf einen Container angeordnet werden. Hier wird der Container für den LoginScreen zurückgegeben.
- Methode public LoginData getLoginData()

  Liefert als Rückgabewert die Login Daten

IPhotoOrder:

- Beschreibung: Die GUI für die Bestellung von Bildern wird in der Klasse PhotoOrder aus dem Paket HAVI implementiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): PhotoOrder
- Attribute:
  - private GalleryDirector mediator
- Methode Cart getCartContent()

  Liefert als Rückgabewert das Objekt Cart, welches den Warenkorb bildet.
- Methode Object getContainer()

  Hier wird der Container für die PhotoOrder zurückgegeben.

IPhotoPreview:

- Beschreibung: Die GUI für die Anzeige der Thumbnails (Hauptmenü) in dem auch auf Vollbild umgestellt werden kann bzw. weitere Menüpunkte ausgewählt werden können.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): PhotoPreview
- Attribute:
  - private GalleryDirector mediator
  - private OSD OSD
- Methode Object getContainer();

  Hier wird der Container für die PhotoPreview zurückgegeben.

- Methode void setMediaRoot(MediaRoot mediaRoot)

  Die Methode bekommt als Parameter ein Objekt der Klasse mediaRoot übergeben. Die Klasse MediaRoot repräsentiert die Archivstruktur und beinhaltet die Kategorien und Alben.

- Methode MediaRoot getMediaRoot()

  Liefert als Rückgabewert die Archivstruktur (MediaRoot).

- Methode public AlbumItem getSelectedAlbumItem()

  Liefert ein AlbumItem als Rückgabewert.

IPhotoView:

- Beschreibung: Die GUI für die Vollbildansicht des ausgewählten Bildes wird im PhotoView des Pakets HAVI implementiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n):
- abgeleitete Klasse(n): PhotoView

- Attribute:
  - private GalleryDirector mediator
  - private OSD OSD
  - private IAlbum Album
- Methode Object getContainer();

  Hier wird der Container für die PhotoView (Vollbildansicht) zurückgegeben.
- Methode public void setAlbumItem(IAlbumItem albumItem)

  Die Methode bekommt als Parameter ein albumItem (Bild) übergeben und dient dazu dieses zu setzen.
- Methode IAlbumItem nextAlbumItem()

  Liefert als Rückgabewert ein AlbumItem. Hier wird das nächste AlbumItem ermittelt und zurückgegeben. Dieses wird für die Slideshow benötigt, wie sie in den Sequenzdiagramm näher beschrieben wird.
- Methode IAlbumItem previousAlbumItem();

  Liefert als Rückgabewert ein AlbumItem. Hier wird das vorherige AlbumItem ermittelt und zurückgegeben. Diese wird ebenfalls für die Slideshow benötigt.

IPreferencesScreen:

- Beschreibung: Stellt eine Oberfläche zur Verfügung in der der Benutzer Konfigurationseinstellungen (z.B. Sprachauswahl) vornehmen kann. Das Objekt enthält statische Variablen in denen die Präferenzen abgespeichert werden, auf welche andere GUI-Module dann zugreifen können.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): PreferencesScreen
- Attribute:
  - private GalleryDirector mediator

- Methode public void setPreferences(Document pref)

  Hiermit wird das Objekt mit den zuletzt eingestellten Präferenzen initialisiert.

- Methode public Object getContainer();

  Hier wird der Container für den PreferencesScreen zurückgegeben.

IControlMenu:

- Beschreibung: Das ControlMenu wird in der Klasse OSD implementiert. Die Abkürzung OSD steht für OnScreenDisplay. Es dient als Auswahlmenü, welches über die Betätigung eines Buttons angezeigt wird und die entsprechenden Menüeinträge ausgewählt werden können.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): OSD
- Attribute:
  - private GalleryDirector mediator
- Methode public Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButton und andere Elemente auf einen Container angeordnet werden. Hier wird der Container für das OnScreenDisplay zurückgegeben.

- Methode public void setEntryRoot(EntryGroup items)

  Wird bei der Initialisierung des OSD Menüs aufgerufen und setzt die Menüeinträge, entsprechend der vorgegebenen Struktur, in das ControlMenu ein.

3.4.4.3 Klasse der Anwendungslogik (MainXlet u. Mediator)

GalleryDirector:

- Beschreibung: Der GalleryDirector funktioniert nach dem Mediator Pattern und beinhaltet die Programmlogik der Applikation *personalisierte Mediengalerie*. Er besitzt Referenzen von allen Klassen bzw. Interfaces und ermöglicht so eine Kommunikation der einzelnen Klassen untereinander.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): keine
- Beziehungen: enthält Referenzen zu allen Klassen und Interfaces.
- Attribute:
  - private HScene theHScene
  - private ILoginScreen theLoginScreen
  - private MainXlet theMainXlet
  - private IConfigurationClient theConfigClient
  - private IKeyAccountChoice theKeyAccountChoice
  - private IControlMenu theControlMenu
  - private IControlMenu thePreviewMenu
  - private IPhotoPreview thePhotoPreview
  - private IPhotoView thePhotoView
  - private IMediaClient theMediaClient
  - private IRegisterScreen theRegisterScreen
  - private IPreferencesScreen thePreferencesScreen
- Methode private void setForeground(Object container)

  Setzt den übergebenen Container in den Vordergrund, so dass dieser auf dem Bildschirm sichtbar wird.

- Methode public void showControlMenu(Object mod, java.awt.event.ActionEvent event)

  Anzeigen des OnScreenDisplays

- Methode public void hideControlMenu(Object mod)

  Ausblenden des OnScreenDisplays

- Methode public void handleMenuSelection(Object module, MenuEntry entry)

  Erkennen, welcher Menüeintrag ausgewählt wurde

- Methode public void showAlbumItem(Object module, IAlbumItem albumItem)

  Zum Anzeigen des AlbumItems (Bildes)

- Methode public void handleCategoryStructure(Object module, CategorieStructure[] catStruct)

  Für die Kategoriestruktur

- Methode public ICartContent getCart(Object mod)

  Für den Warenkorb

- Methode public void orderArticle(Object mod, IArticle article, int counts)

  Für die Bestellung von Artikeln

MainXlet:

- Beschreibung:Wird zum Starten der Applikation benötigt und initialisiert den GalleryDirector. Der Applikationsmanager kann über die Klasse MainXlet den Zustand des Programms festlegen. Die Zustände die das Xlet annehmen kann, sind: *initialisiert*, *gestartet*, *pausiert* und *zerstört*.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): Xlet
- abgeleitete Klasse(n): keine
- Attribute:
  - private XletContext context
  - private GalleryDirector med

- Methode public void initXlet(XletContext context)

  Initialisiert das Xlet. über den Parameter context kann das Xlet an den Applicationmanager seinen eigenen Status abfragen bzw. beeinflussen(Xlet kann sich z.B. selber pausieren oder zerstören).

- Methode public void startXlet()

  Startet das Xlet bzw. den GalleryDirector.

- Methode public void pauseXlet()

  Pausiert das Xlet.

- Methode public void destroyXlet(boolean unconditional)

  Zerstört das Xlet. über den Parameter unconditional kann das Xlet seine Zerstörung verhindern, sofern dieser den Wert *false* hat, andernfalls wird das Xlet zerstört.

3.4.4.4 Klassen der Client Kommunikation

IArticle:

- Beschreibung: Die Klasse Article ist für den Warenkorb vorgesehen. Es enthält den Artikelnamen und die Artikelnummer.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): Article
- Attribute:
  - private String name
  - private int art_id

ICartContent:

- Beschreibung: Enthält den Inhalt des Warenkorbes. Es können Artikel hinzugefügt und wieder entfernt werden.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): CartContent
- Attribute:
  - private int position
  - private int[] articleID
  - private String[] filename
  - private int[] count
- Methode public void addContent(int artID, String filename, int count)

  Fügt einen Artikel in den Warenkorb ein, hierzu werden die Artikel_id (als eindeutige Bezeichnung des Artikels), der filename und die Anzahl der zu bestellenden Artikel (count) als Parameter übergeben.

- Methode public void delContent(int position)

  Entfernt den markierten Artikel aus dem Warenkorb, dazu wird die Position des Artikels im Warenkorb als Parameter der Methode übergeben.

- Methode public CartContent getContent()

  Liefert den Warenkorbinhalt zurück.

ICategory:

- Beschreibung: Das Interface wird in den Klassen CeWeCategory und FileSysCategory implementiert. Enthälten die Funktionalitäten um ein Album in eine Kategorie hinzu zu fügen und auch wieder zu entfernen. Der Unterschied ist, dass es sich bei CeWeCategory um Online Alben und bei der FileSysCategory um die Offline Alben handelt.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): CeWeCategory, FileSysCategory
- Attribute:
  - private IAlbum[] albums
  - private int category_id
  - private String name
- Methode public IAlbum[] getAlbums()

  Liefert ein Array von Alben zurück.
- Methode public int getCategrory_id()

  Liefert eine Ganzzahl (Integer) der Kategorie zurück.
- Methode public String getName()

  Liefert den Namen der Kategorie zurück.
- Methode public void addAlbum(IAlbum album)

  Als Parameter wird ein Album übergeben, welches in die Kategorie eingefügt werden soll.
- Methode public void deleteAlbum(IAlbum album)

  Als Parameter wird ein Album übergeben, welches von der Kategorie entfernt werden soll.
- Methode public void rename(String s)

  Als Parameter wird ein String übergeben, der den neuen Namen der Kategorie enthalten soll.

IAlbum:

- Beschreibung: Beinhaltet die Funktionalitäten des Offline Albums wie das hinzufügen und entfernen von AlbumItems.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.
- Oberklasse(n): keine

- abgeleitete Klasse(n): FileSysAlbum
- Attribute:
  - private IAlbumItem[] albumItem
  - private AlbumController albumReloader
  - Category parent
- Methode public IAlbumItem[] getAlbumItems()
  Liefert ein Array von AlbumItems zurück.
- Methode public String getDescription()
  Liefert eine kurze Beschreibung der Alben.
- Methode public String getName()
  Liefert den Namen des Albums.
- Methode public int getCategoryID()
  Liefert die eindeutige Kategorie_id.
- Methode public boolean getVisibleStatus()
  Liefert den Sichtbarkeitsstatus des Albums, d.h. wer die Bilder betrachten darf (Freundeverwaltung).
- Methode void deleteAlbumItem();
  Methode zum entfernen eines AlbumItems.
- Methode void addAlbumItem();
  Methode zum hinzufügen eines AlbumItems.

IAlbumItem:

- Beschreibung: Beinhaltet die Funktionalitäten des Offline AlbumItems wie das hinzufügen und entfernen von AlbumItems, das drehen des Bildes, das nächste Bild und vorherige Bild können übergeben werden, welches für Funktion der Slideshow verwendet wird.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): FileSysAlbumItem
- Attribute:
  - protected String id
  - private AlbumController albumReloader
  - protected String name
  - protected boolean visibleStatus
  - protected int rotation
  - protected IAlbum parent
- Methode public Image getPicture()

  Liefert das Bild zurück.
- Methode public int getAlbumItemID()

  Liefert die eindeutige ID des AlbumItems zurück.
- Methode public String getDescription();

  Liefert kurze Beschreibung des AlbumItems.
- Methode public Image getThumbnail()

  Liefert das Bild in Thumbnail Ansicht.
- Methode public int getRotation()

  Funktion zur Rotation des Bildes.
- Methode public boolean getVisibleStatus()

  Liefert den Sichtbarkeitsstatus des Albums, d.h. wer die Bilder betrachten darf (Freundeverwaltung).
- Methode public void rename(String s)

  Der Parameter den die Methode übergeben bekommt, ist der neue Name des Bildes.
- Methode public IAlbumItem getNextAlbumItem()

  Liefert das nächste Bild zurück, wird für die Slideshow verwendet, in dem jeweils das nächste Bild aufgerufen wird und für einige Zeit am Bildschirm gezeigt wird.

- Methode public IAlbumItem getPreviousAlbumItem()

  Liefert das vorherige Bild zurück, wird ebenfalls für die Funktion der Slideshow verwendet.

IArticleList:

- Beschreibung: Enthält eine Liste (Array) von den Artikeln

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): Article
- Attribute:
  - private Article[] articles

IMediaRoot:

- Beschreibung: Das Interface wird in der Klasse MediaRoot implementiert und verwaltet die Archivstruktur. In Form eines Arrays werden die Kategorien abgespeichert, worin die Alben bzw. Bilder gespeichert sind.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): MediaRoot
- Attribute:
  - Category categories
- Methode public void addCategory(Category cat)

  Fügt eine Kategorie hinzu. Hierbei wird ein Parameter Category übergeben.

- Methode public void deleteCategory(Category cat)

  Diese Methode erhält als Parameter eine Kategorie, die gelöscht werden soll.

3.4.4.5 Klassen der Datenbankschicht u. Server Kommunikation

CeWeClient:

- Beschreibung: Implementiert die vier Interfaces *IConfigurationClient, IOnlinePrintClient, IMediaClient und IMediaSource*. Das Interface IConfigurationClient stellt Konstanten und Methodenrümpfe zur Verfügung, über die auf die kundenspezifischen Logindaten sowie Konfigurations-Dateien zugegriffen werden kann. IOnlinePrintClient ist für den Bestellvorgang zuständig. Dazu gehören Zugriffe auf die durch den Kunden spezifizierte ArticleList und den dazu passende ICartContent (Warenkorb mit Bildern oder Funprodukten). Der Zugriff auf Offline bzw. Online Alben werden durch Methodenrümpfe der Interfaces IMediaSource bzw. IMediaClient implementiert. Die Klasse *CeWeClient* selbst, stellt Methoden zum Laden der Bilder für die verschiedene Ansichtsmodi (Thumbnail, Vollbild) zur Verfügung.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): IConfigurationClient, IOnlinePrintClient und IMediaClient
- abgeleitete Klasse(n): keine
- Attribute:
  - private int sessionID
  - GalleryDirector mediator
- Methode public LoginData getLoginData(String ServerType)

  Methode die die Logindaten vom Server holt. Als Parameter wird der jeweilige Servertyp benötigt.

  IConfigurationClient

  - Methode public boolean login(LoginData login)
    Als Parameter werden die Logindaten übergeben, die dann auf ihre Richtigkeit überprüft werden.

- Methode public Document getPreference()
  Gibt die auf dem Server gespeicherten Konfigurationseinstellungen in Form eines Documentobjektes (XML Dokument von Xerxes)zurück.
- Methode CustomerData getCustomerData()
  Methode gibt CustomerData-Objekt zurück über das auf Daten, wie z.B. Kreditkartenummer zugegriffen werden kann.
- Methode IArticleList getArticleList()
  Gibt ein ArticleList-Objekt zurück.
- Methode public KeyAccount[] getKeyAccounts()
  Gibt Array zurück, dass alle wählbaren CeWe Partner (Fotopoint, Schlecker und Kruidvat) beinhaltet.
- Methode void selectKeyAccount(KeyAccount ka)
  Methode die den gewünschten CeWe Partner auswählt.

IMediaSource&IMediaClient

- Methode public Category[] getArchiveStructure()
  Gibt ein Array zurück, dass zu einer bestimmten Kategorie die entsprechenden Alben (aber noch ohne Bilder) enthält.

IOnlinePrintClient

- Methode void newOrder(Cart p0, DeliverData deliver)
  Methode die eine neue Bestellung implementiert. Als Parameter werden der aktuelle Warenkorb und die Lieferdaten übergeben.
- Methode void getEmptyCart()
  Liefert leeren Warenkorb zurück
- Methode public void login(LoginData login)
  Zum Anmelden auf dem Server.
- Methode IArticleList getArticleList()
  Methode gibt eine Liste mit allen Artikeln zurück die man bestellen kann.
- Methode IPhotoOrder getPhotoOrder()
  Liefert als Rückgabewert den Warenkorb zurück.

CeWeClient

  - Methode public IAlbum loadAlbumItems(IAlbum album)
    Bekommt als Parameter ein Album übergeben und gibt anhand des Albums die dazugehörigen Bilder zurück.
  - Methode public IAlbumItem loadPicture(IAlbumItem albumItem)
    Gibt zu einem ausgewählten Bild, das entsprechende Bild in Vollbildansicht zurück.
  - Methode public IAlbum loadThumbnails(IAlbum album)
    Liefert für ein Album alle zugehörigen Bilder als Thumbnails.

ICartContent:

- Beschreibung: Die Klasse beinhaltet die Funktionalitäten um den Warenkorb mit Artikeln zu füllen und etwas aus dem Warenkorb wieder zu entfernen.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): CartContent
- Attribute:
  - private int position
  - private int[] articleID
  - private String[] filename
  - private int[] count
- Methode public void addContent(IAlbumItem item, Article article, int count)

  In den Warenkorb werden die Artikel eingefügt, dazu wird der Methode das IAlbumItem (das Bild), der Artikel und die Anzahl (count) der zu bestellenden Artikel übergeben.

- Methode void delContent(int position)

  Um aus dem Warenkorb ein Artikel wieder zu entfernen wird als Parameter die Position des zu entfernendem Artikel im Warenkorb angegeben.

- Methode public Cart getContent()
  Gibt den Inhalt des Warenkorbes zurück.

3.4.5 Detailliertes dynamisches Modell

3.4.5.1 Sequenzdiagramme der Mediengalerie

Anmeldung:

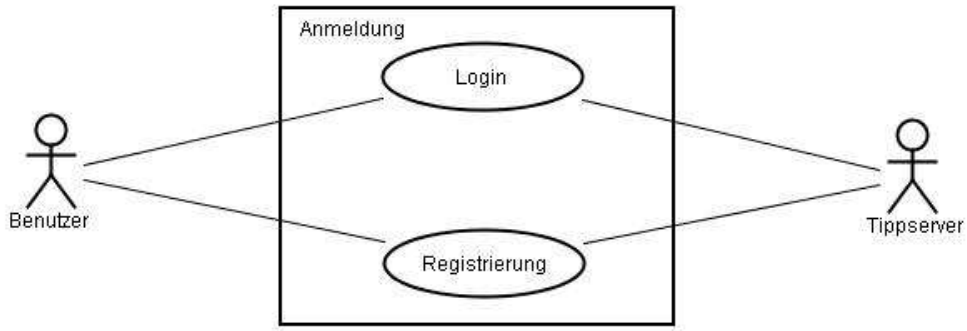

Abbildung 3.15: Sequenzdiagramm Anmeldung

Zuerst wird der *HScene* das Container Objekt des LoginScreens durch die Methode *getContainer* übergeben. Durch die Methode setForeground wird die HScene mit dem Container des LoginScreens in den Vordergrund gesetzt und somit auf dem Bildschirm sichtbar. Sobald der Benutzer sein *Login* und *Passwort* eingetragen sowie den *OKButton* betätigt hat, wird die Methode *action* aufgerufen. Durch die übergebenen Parameter *String* und *Objekt* kann der GalleryDirector (Mediator) den Methodenaufruf als Aufforderung zum Einloggen identifizieren und führt daraufhin die Methode getLoginData aus. Nachdem die Logindaten ausgelesen worden sind, beauftragt der GalleryDirector mit der Methode *login(LoginData)* den IConfigurationClient mit der Abgleichung der Logindaten in der Datenbank. Falls der Login korrekt ist, initialisiert der GalleryDirector das Auswahlmenü der CeWe Partner. Zunächst werden über die Methoden *getKeyAccount* und *setKeyAccount* die verfügbaren CeWe Partner dem Container Objekt *IKeyAccountChoice* zugewiesen. Daraufhin wird das Container Objekt mit der Methode *getContainer* der HScene übergeben und durch die Methode *setForeground* in den Vordergrund gesetzt. Durch die Methoden *getSelectedKeyAccount* bzw. *selectKeyAccount* werden die gewünschten CeWe Partner fokussiert und ausgewählt. Die kundenspezifischen Daten wie z.B. Liefermodalitäten oder eventuelle Gutscheinboni ordnet der GalleryDirector durch die Methoden *getCustomerDate&getConditionData* dem jeweiligen Benutzer zu. Ebenso werden die verfügbaren Artikel durch die Methode *getArticleList* geladen. Als letztes stellt der GalleryDirector mit Hilfe der Logindaten und der Methode *getArchiveStructure* eine Anfrage an die Datenbankzugriffsklasse *IMediaClient* die in einer Rückgabe der zugehörigen Kategoriestruktur resultiert.

Auswahl:

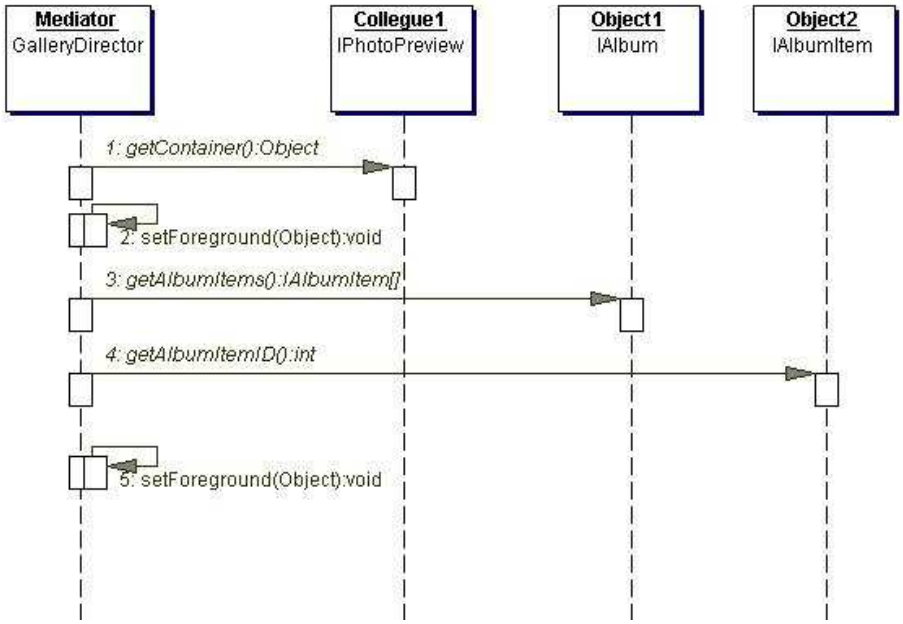

Abbildung 3.16: Sequenzdiagramm Auswahl

Zunächst bekommt die *HScene* das Container Objekt der Klasse PhotoPreview übergeben. Durch die Methode *setForeground* wird die HScene mit dem Container des PhotoPreview in den Vordergrund gesetzt und somit auf dem Bildschirm sichtbar. Die Auswahl der Alben erfolgt durch den Aufruf der Methoden *getAlbumItems* bzw. *getAlbumID*. Der daraus resultierende modifizierte Bildschirm wird durch die Methode *setForeground* neu angezeigt.

Bestellung:

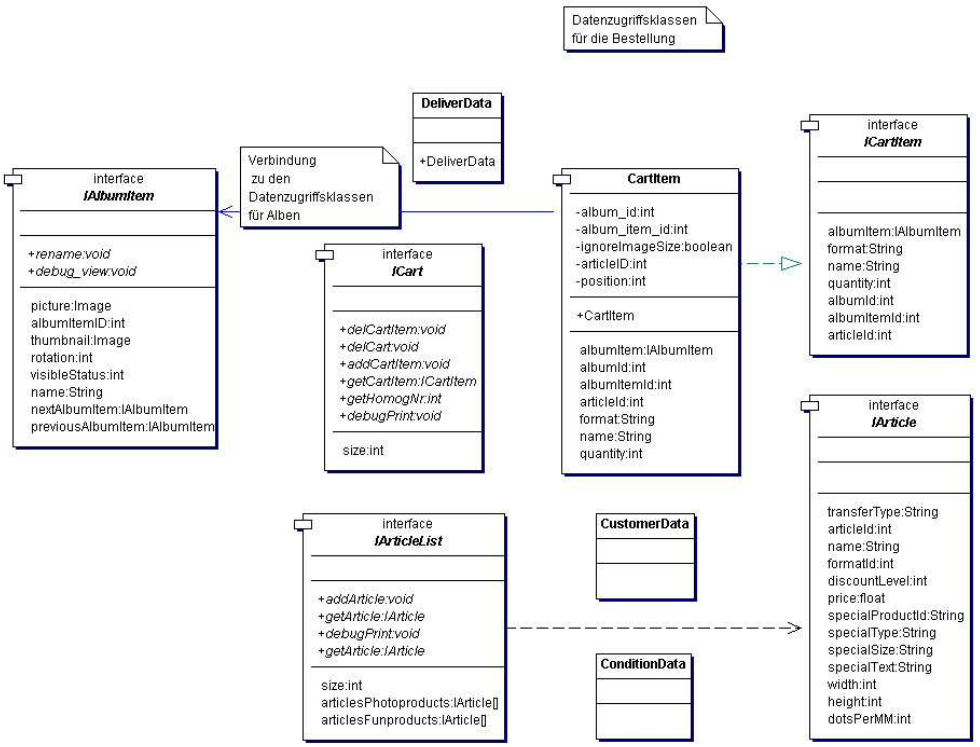

Abbildung 3.17: Sequenzdiagramm Bestellung

Zunächst weist der GalleryDirector der *HScene* durch Aufruf der Methode *getContainer* das Container Objekt des IPhotoOrderScreens zu. Durch die Methode *getContent* erhält das IPhotoOrder Objekt das Objekt Cart, in dem die Warenkorbfunktionalitäten implementiert sind. Nun setzt der GalleryDirector durch die Methode *setForeground* die HScene mit dem IPhotoOrder Objekt in den Vordergrund. Die nächste Aktion des GalleryDirectors wird durch die Betätigung des *Weiter Buttons* ausgelöst. Diese zieht den Aufruf der *action* Methode nach sich, die über die übergebenen Parameter *Objekt* und *String* den entsprechenden Button eindeutig identifiziert. Nun wird durch die Methoden *getContainer* und *setForeground* der *Bestellvorgang abschließen* Bildschirm der HScene zugewiesen und in den Vordergrund gesetzt. Durch den *Abschicken Button* wird der GalleryDirector beauftragt, den Bestellvorgang durch Aufruf der Methode *newOrder* einzuleiten. übergeben werden dieser Methode die Parameter *Cart(Warenkorb)* und *DeliverData(Lieferdaten)*, die an den IOnlinePrintClient übermittelt werden, der

dann die Verbindung mit dem CeWe Server herstellt.

Thumbnails:

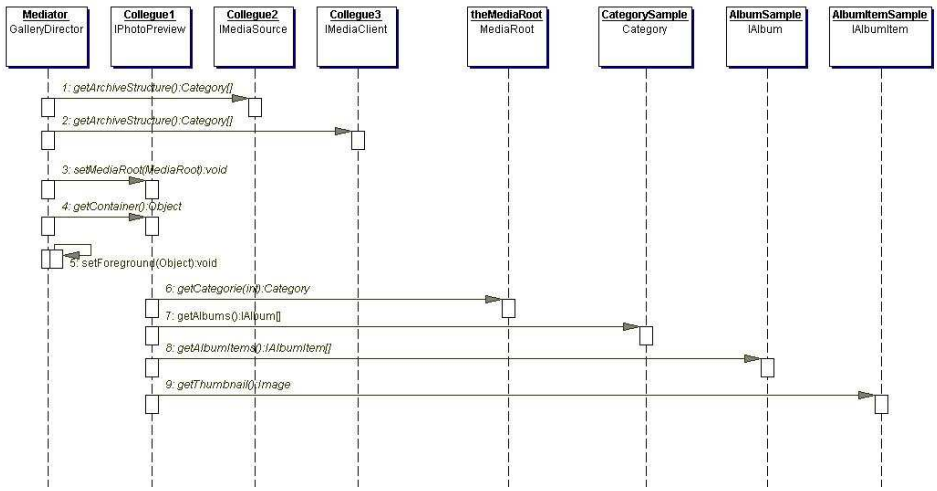

Abbildung 3.18: Sequenzdiagramm Thumbnailansicht

Zunächst holt der GalleryDirector mit den Methoden *getArchiveStructure* die Online (IMediaClient) bzw. Offline (IMediaSource) Kategorien des jeweiligen Benutzers. Danach baut der GalleryDirector durch die Methode *setMediaRoot* aus den einzelnen Kategorien eine baumartige Struktur auf. Anschließend weist der GalleryDirector der *HScene* durch Aufruf der Methode *getContainer* das Container Objekt des IPhotoPreviewScreens (Thumbnailsansicht) zu und setzt die HScene mit dem IPhotoPreview Objekt in den Vordergrund. Die Auswahl einer bestimmten Kategorie wird in der Thumbnailansicht durch Aufruf der Methode *getCategorie* realisiert. Die in der ausgewählten Kategorie verfügbaren Alben werden durch die Methode *getAlbums* der Thumbnailansicht zugewiesen. Die zu einem Album zugehörigen Bilder werden dann durch die Methoden *getAlbumItems&getThumbnail* geladen und angezeigt.

Vollbild:

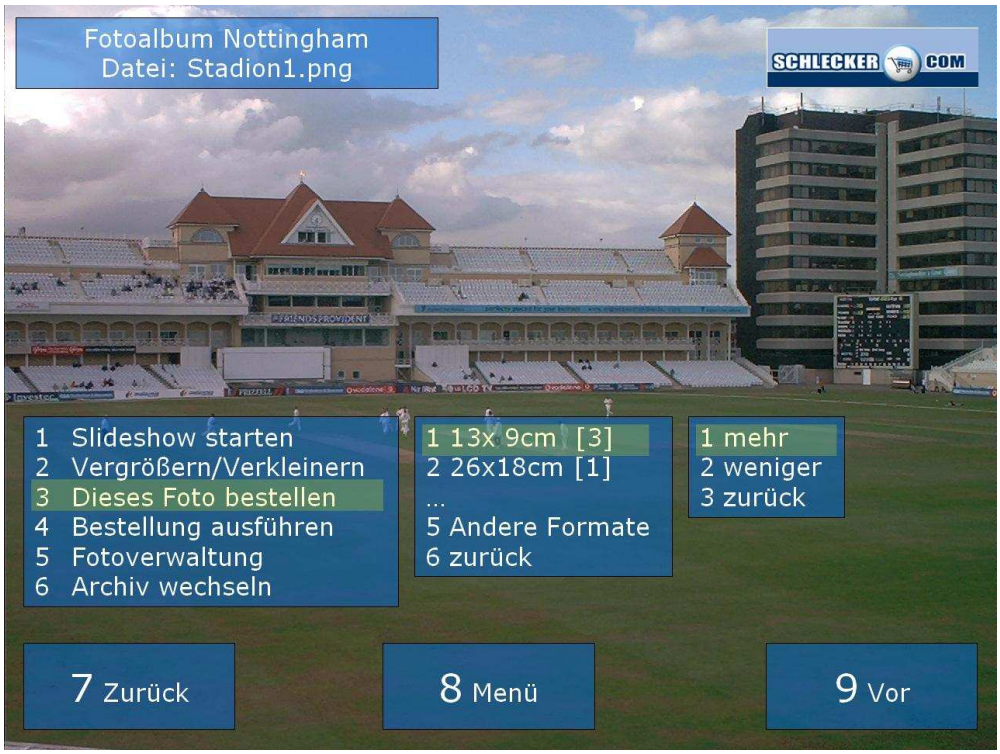

Abbildung 3.19: Sequenzdiagramm Vollbildansicht und Slideshow

Der IPhotoPreviewScreen beauftragt den GalleryDirector durch die Methode *showAlbumItems* ein ausgewähltes Bild als Vollbild anzuzeigen. Daraufhin ruft der GalleryDirector die Methode *setAlbumItems* des PhotoView auf. Die Methode *getPicture* liefert als Rückgabewert ein Image, welches dann als Vollbild angezeigt wird. Anschließend weist der GalleryDirector der *HScene* durch Aufruf der Methode *getContainer* das Container Objekt des IPhotoViewScreens (Vollbildansicht) zu und setzt die HScene mit dem IPhotoView Objekt in den Vordergrund.
Das Starten der Slideshow erfolgt durch Auswahl des Menüpunktes *Slideshow starten*, welches über den Hauptmenübutton zu erreichen ist, bzw. durch den Aufruf der Methode *action* durch den GalleryDirector. Durch die Methode *nextAlbumItem* beauftragt der GalleryDirector den IPhotoView das nächste Bild auszuwählen. Die Anzeige im Vollbildmodus erfolgt durch die Methoden *getNextAlbumItem&getPicture*.

Warenkorb:

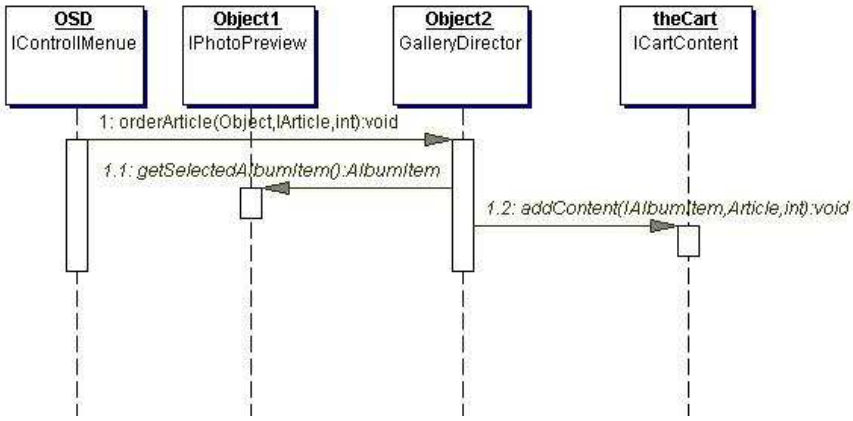

Abbildung 3.20: Sequenzdiagramm Warenkorb

Das im *OSD Menü* ausgewählte Bild wird durch die Methode *orderArticle* dem GalleryDirector gemeldet. Der GalleryDirector holt mit der Methode *getSelectedAlbumItem* das ausgewählte Foto (AlbumItem) und weist es durch die Methode *addContent* dem Warenkorb (CartContent) zu.

3.4.6 Nutzungsvoraussetzungen

Da alle drei Anwendungen auf dem heimischen Fernseher ablaufen und entweder terrestrisch, via Kabel oder Satellit auf die Set-Top-Box heruntergeladen werden, wird die Dokumentation für die Endbenutzer in Form einer Hilfefunktion als Teil des Programms beigefügt. Optional ist es auch denkbar, dass z.B. für die *personalisierte Mediengalerie* ein Handbuch auf den WWW-Seiten von CeWe Color im PDF-Format oder als WWW-taugliches HTML-Format zur Verfügung steht. So ist möglich, sich schon vor der Nutzung einer Anwendung mit der Funktionsweise vertraut zu machen. Die Gliederung in Kapitel und Formatierung werden weitestgehend identisch sein, so dass sich Benutzer die nur die Hilfefunktion des Programms oder die WWW-Dokumentation kennen problemlos auch in der anderen Form zurechtfinden. Ebenso sollen Abbildungen und Tabellen in der Online-Version genauso erscheinen wie in der Hilfefunktion des Programms selbst. Ein wichtiger Punkt stellen die zum Ausführen der Anwendung nötigen Systemvoraussetzungen, also Hardwarekomponenten wie Set-Top-Box oder MHP-fähiger Fernseher und verwendeter MHP-Standard dar. Diese Informationen müssen auf jeden Fall vor dem Herunterladen der Anwendung einsehbar sein. Andernfalls ist nicht auszuschließen, dass die Anwendungen auf einigen Set-Top-Boxen aufgrund z.B. eines nicht unterstützten MHP-Standards nicht auszuführen ist. Die Einsicht der Systemvoraussetzungen könnte auf den Webseiten der Hersteller der Anwendungen oder über den Videotext der Broadcaster ermöglicht werden. Der Programmstart erfolgt über das Auswahlmenü des Startbildschirms der nach dem Einschalten der Set-Top-Box erscheint. Denkbar ist, dass die Anwendung z.B. nach einer Werbung im Fernsehen zum Herunterladen angeboten wird.

Literaturverzeichnis

[Bal00] Helmut Balzert. *Lehrbuch der Softwaretechnik - Softwareentwicklung*. 2000. 1. Band.

[Bol04] Dietrich Boles. *Leitfaden zur Durchführung von Projektgruppen*. PDF-Datei, September 2002. URL http://www.informatik.uni-oldenburg.de/studium/leitfaeden/Leitfaden-PG.pdf - Letzter Zugriff: 14.02.04.

[Bre01] M. Brefort. Die Referenzimplementierung der Multimedia Home Platform. *Sonderdruck aus Fernseh und Kino-Technik*, 2001.

[DVB03] DVB Project Office. DVB - Digital Video Broadcasting - The DVB Project Office. http://www.dvb.org, 2003. letzter Zugriff: 27.Dezember 2003.

[ETS01] ETSI - European Telecommunications Standards Institute. http://www.etsi.org, 2001. letzer Zugriff: 27.Dezember 2003.

[Eur01] European Telecommunications Standards Institute. *Multimedia Home Platform (MHP) Specification 1.1*. Westedeutscher Verlag, November 2001.

[Eva98] Jean-Pierre Evain. *The Multimedia Home Platform - an overview*. EBU Technical Department, 1998.

[FKT00] FKTG - Fernseh- und Kinotechnische Gesellschaft e.V. *Fernsehen heute und morgen, Technik - Märkte - Strategien*, 3.Ausgabe, 2000.

[Fp03] Fernsehen-plus. Applikationen. http://www.mhp-forum.de, 2003. letzer Zugriff: 9.Dezember 2003.

[Ges03] Gesellschaft für Unterhaltungs- und Kommunikationselektronik. Das Lexikon mit allen wichtigen Stichworten der Consumer Electronics. http://www.gfu.de/pages/lexikon.html, 2003. letzer Zugriff: 9.Dezember 2003.

[Mul03] Multimedia Home Platform. DVB-MHP - Digital Video Broadcasting Multimedia Home Platform. http://www.mhp.org, 2003. letzter Zugriff: 27.Dezember 2003.

[SC01] Anthony Smith-Chaigneau. *DVB-MHP - A Snapshot - October 2001*. DVB Project Office, Oktober 2001.

[Sch03] Peter Schütz. Persönliches Fotoalbum über DVB-T -Empfang auf Settop-Boxen. Technical report, cewe digital GmbH, 2003.

[Sed01] R. Sedlmeyer. Multimedia Home Platform. *Sonderdruck aus Fernseh und Kino-Technik*, 2001.

[Ste03] Axel Steinbach. Diplomarbeit - Konzeption und Realisierung einer personalisierten Mediengalerie im Internet, 2003.

[Zer03] Frank Zervos. *Digitales Fernsehen in Deutschland*. Westedeutscher Verlag, 2003. 1. Auflage.

Kapitel 4

Abschlussbericht

4.1 Kurzer Überblick über die folgenden Kapitel

Dieses Dokument enthält den überarbeiteten Entwurf der Mediengalerie MeGa, den Entwurf für das Tippspiel Tipp2Win, das technische Handbuch für die Wartung der Anwendungen, die beiden Handbücher für Tipp2Win und MeGa sowie eine Beschreibung des common-Paketes, welches die gemeinsamen Klassen der beiden Anwendungen beinhaltet. Die Entwürfe stellen die Grundlage für die Implementierung dar. Das technische Handbuch enthält u.a., welche Probleme bei der Nutzung der Box zustande kommen (siehe *Unterschiede zwischen Referenzimplementierung und der Galaxis-Box* und Informationen zur Wartung bzw. Installationshinweise der Applikationen. Desweiteren gibt es ein Kapitel Ausblick auf mögliche Weiterentwicklungen und einen Erfahrungsbericht unserer Projektgruppe.

4.2 Überarbeiteter Entwurf der Mediengalerie MeGa

4.2.1 Einleitung

4.2.1.1 Ausgangssituation und Zielsetzung

Die Projektgruppe hat sich entschieden zunächst mit der Implementierung der *personalisierten Mediengalerie* prototypisch zu beginnen. Aufbauend auf diesen Erfahrungen, insbesondere was die Oberflächengestaltung und die Anbindung zu den Servern zu CeWe Color betrifft, wird der Prototyp im Zusammenspiel mit dem Partner CeWe Color und Fachkräften[1]außerhalb der Projektgruppe überarbeitet und vervollständigt. Die Mediengalerie soll Funktionen bieten, die das Anschauen, Bearbeiten und Bestellen von Fotos bequem vom Wohnzimmer aus ermöglichen. Ein Hauptaugenmerk liegt vor allem bei der Einhaltung der Schnittstellen, die vom Kooperationspartner vorgegeben werden. So soll z. B. die Archivierung von Fotos auf hauseigenen Servern von CeWe realisiert werden. Parallel zu der Vervollständigung des Prototyps der *personalisierten Mediengalerie* beginnen dann die Arbeiten an dem *personalisierten Fußball Tippspiel Tipp2Win*. Grundfunktionalitäten des Tippspieles sollen dabei das Tippen der einzelnen Gruppenspiele bzw. Finals, ein individuelles Benutzerprofil und eine Ranglistenfunktion sein.

4.2.1.2 Aufbau des Entwurfs

Der hier vorgelegte Entwurf stellt die Arbeitsgrundlage für die Implementierung der Anwendung Mediengalerie bzw. Tippspiel dar. In diesem ersten Kapitel wird ein erster Überblick über die Vorgaben der Projektgruppe gegeben. Im Kapitel 4.2.2 werden die Konventionen für die Formatierung des Quellcodes festgelegt und eine Begründung gegeben aus welchem Grund dies von Bedeutung ist.
Die einzelnen Module der verschiedenen Anwendungen lassen sich in unterschiedliche Schichten einteilen. Dies sind die Dialogschicht, Anwendungslogik (Mediator), Client Kommunikation, Server Kommunikation und die Datenbankschicht. Die näheren Beschreibungen sind in einem eigenen Kapitel 4.3.2 in diesem Entwurf untergebracht.
In den folgenden beiden Kapiteln werden die Module die implementiert werden sollen näher beschrieben. In dem Kapitel 4.3.3 werden die einzelnen Klassen der jeweiligen Anwendungen beschrieben.

[1]Beratung bei der Oberflächen gestaltung

Das Kapitel 4.3.4 enthält die Sequenzdiagramme, in denen u.a. das dynamische Verhalten beim einloggen aufgezeigt wird.

4.2.2 Programmierrichtlinien

Es gibt einige gute Gründe sich beim Entwurf einer Software auch Konventionen für die Formatierung des Quellcodes zu überlegen:

- Der größte Teil des Arbeitsaufwandes (ca. 80 Prozent) im Lebenszyklus liegt in der Wartung und Pflege.
- Selten kann die Software während des gesamten Lebenszyklus nur vom Original-Autor gepflegt werden.
- Code Konventionen tragen erheblich zur Lesbarkeit des Codes bei und vereinfachen somit die Einarbeitung sowohl in fremden Code als auch in den eigenen Code, den man ein paar Wochen oder Monate nicht mehr angerührt hat.

In unserem Fall entwickeln wir unsere Software sogar im Team und jedes Mitglied ist motiviert so viel wie möglich zu lernen und den Quellcode des anderen möglichst schnell zu verstehen. Aus diesen Gründen haben wir uns auf die Einhaltung der weit verbreiteten Code Conventions von Sun geeinigt. Des weiteren sollen noch folgende Regeln eingehalten werden:

- Bezeichner, Methodennamen etc. sollen englisch sein.
- Kommentare sollen in Deutsch verfasst sein.
- Aufgrund besserer Lesbarkeit sollen Bezeichner innerhalb Methoden an der möglichst spätesten Stelle deklariert werden.

4.2.3 Architektur

4.2.3.1 Programmschicht

In diesem Abschnitt wird zum besseren Verständnis eine Unterteilung der Software in mehrere Schichten vorgenommen. Dabei wird das gesamte Softwaresystem, also sowohl die Client-, als auch die Serverseite inklusive der Datenbank betrachtet. Der Client besteht somit aus

- einer Dialogschicht, die eine Schnittstelle zum Benutzer darstellt,
- einer Schicht welche die Anwendungslogik kapselt

- und einer Kommunikationsschicht, die einen Datenaustausch zwischen verschiedenen Servern ermöglicht.

Zwischen diesen drei Schichten wird mittels sogenannter Datenzugriffsklassen kommuniziert. Objekte dieser Klassen werden immer in der Kommunikationsschicht produziert und enthalten oder produzieren Daten, die sowohl aus internen als auch externen Datenquellen stammen können. So kann das Anwendungsprogramm also z.B. Daten aus dem Filesystem der Settop-Box oder über eine http-Verbindung beziehen. Client und Server müssen, genauso wie Server und Datenbank, über ein bestimmtes Protokoll miteinander kommunizieren und werden in jeweils einer Schicht dargestellt.

4.2.3.2 Mediatorprinzip und Module

Eine zweite Unterteilung der Client-Klassen erfolgt mittels Modulen. Es gibt lediglich zwei unterschiedliche Typen von Modulen. Zum einen gibt es Module der Dialogschicht und zum anderen Module der Kommunkationsschicht. Die Aufgaben der Module werden mittels Schnittstellen spezifiziert. Beide Typen von Modulen haben eine Referenz auf den Mediator. Der Mediator hat die Aufgabe die Anwendungslogik zu kapseln. Dies hat zur Folge, dass der Mediator sowohl den Verkehr von Datenzugriffsobjekten, als auch den Programmablauf kontrolliert. Nur über den Mediator können die Datenzugriffsobjekte in die Module der Dialogschicht gelangen, in der sie für den Benutzer sichtbar und zugreifbar gemacht werden können.
Da offensichtlich eine klare Trennung zwischen Datenhaltung und Datendarstellung existiert, kann diese Architektur auch dem Model-View-Controller-Konzept entsprechen. Voraussetzung dafür ist, dass es eine Möglichkeit gibt, die gesamte Dialogschicht austauschen zu können. Bei der Definition von Modulen wurde daher darauf geachtet, dass Module eindeutig entweder der Kommunikations- oder der Dialogschicht zuzuordnen sind. Der Austausch der gesamten Dialogschicht wird zudem durch eine AbstractFactory erleichtert. Ein abstraktes Moduldiagramm zeigt Abbildung 4.1:

4.2.3.3 Pakete

Eine sinnvolle Paket-Hierarchie für die Aufteilung des Clients ergibt sich durch folgende Regelungen:

- Als Präfix dient de.showmemhp
- Die Anwendungen sollen in den Unterverzeichnissen mgallery und tipp2win ihren Platz haben. Allgemeine Klassen, die für beide Anwendungen von

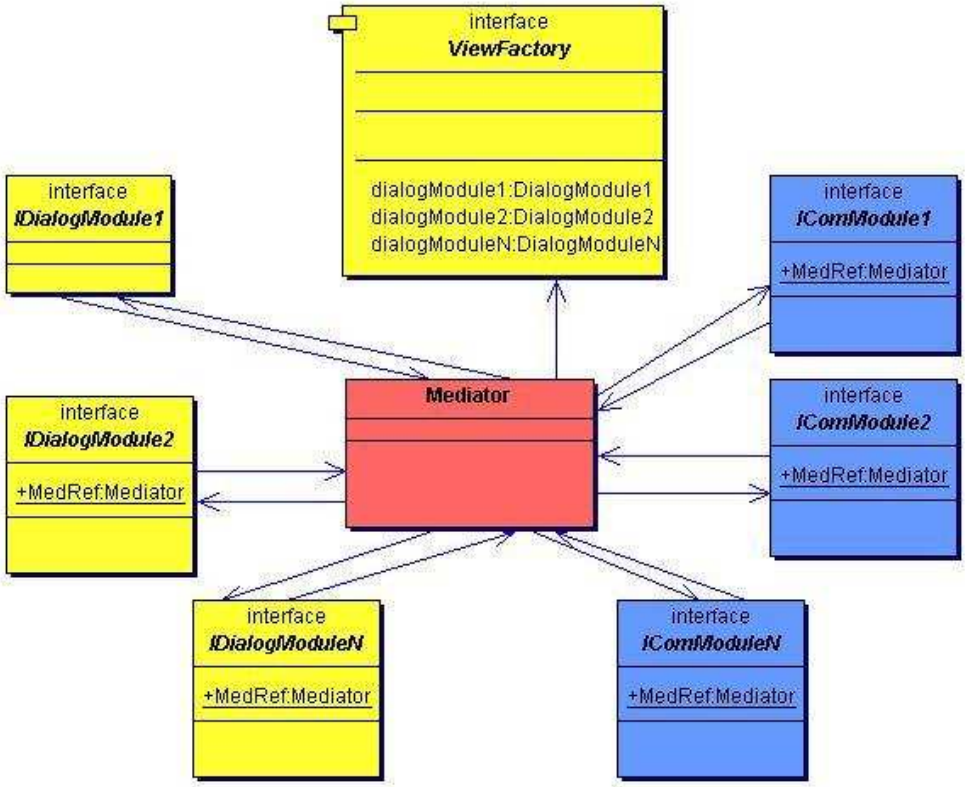

Abbildung 4.1: Mediator-Architektur

Bedeutung sind, befinden sich in einem Unterzeichnisverzeichnis common.

- In den Verzeichnissen der Anwendungen sollen sich jeweils nur Klassen des Xlets, des Mediators sowie Modul-Schnittstellen und Datenzugriffsklassen befinden. Eine Ebene tiefer sollen sich in einer logischen Unterteilung, Klassen zur Realisierung der Modul-Schnittstellen befinden.

Nach Anwendung dieser Regelungen könnte die Verzeichnisstruktur folgendermaßen aussehen:

```
./de/showmemhp/common ./de/showmemhp/common/havi

./de/showmemhp/mgallery ./de/showmemhp/mgallery/havi
./de/showmemhp/mgallery/cewe

./de/showmemhp/tipp2win ./de/showmemhp/tipp2win/havi
```

4.2.3.4 Internationalisierung

Mehrsprachigkeit wird durch die Verwendung des I18N-Konzeptes von Sun realisiert. Alle drei Anwendungen können dadurch auf ein gemeinsames und anwendungsspezifisches ResourceBundle zugreifen. Das Präfix für das ResourceBundle ist mit der Unterpaketbezeichnung der Anwendung, bzw. dem Unterpaket common identisch. Die ResourceBundle haben ihren Platz im Wurzelverzeichnis der Paketstruktur. Somit können zum Beispiel folgende Property-Dateien jeweils ein deutsches ResourceBundle in deutscher Sprache realisieren:

```
./mgallery.properties ./mgallery_help.properties
./mgallery_images.properties

./tipp2win_de_DE.properties ./tipp2win_help.properties
./tipp2win_images.properties
```

Der Zugriff auf die ResourceBundles in den einzelnen Klassen erfolgt über den Aufruf statischer Methoden. Dies könnte folgendermaßen aussehen:

ResourceBundle bundle = ResourceBundle.getBundle
("mgallery",Locale.getDefault());

4.2.3.5 Gemeinsamkeiten der beiden Anwendungen

Die Gemeinsamkeiten der zu realisierenden Anwendungen liegen vor allem bei der gemeinsamen Nutzung des *Mediator-Prinzips* zur Kapselung der Programmlogik und des MVC-Prinzips zur Unterteilung der Klassen in Aufgabenpakete bzw. zur Gewährleistung der Austauschbarkeit einzelner Programmschichten (z.B. der Dialogschicht für die Oberflächengestaltung) Die Funktionsweise des Mediator und des MVC-Prinzips wurden in Kapitel 4.2.3.2 hinreichend beschrieben. Auf der Modulebene gibt es ebenfalls Gemeinsamkeiten, welche ihre Ausprägungen in gemeinsamen Interfaces für Module der Dialogschicht und Kommunikationsschicht finden. Weiterhin planen wir mehrsprachig zu implementieren, wofür wir die in Kapitel 4.2.3.4 beschriebenen *ResourceBundles* benutzen. Der programmtechnische Einsatz mit Interfaces für Datenhaltung bzw. GUI-Klassen sieht wie folgt aus:

Datenhaltung: Eine Arbeitsersparnis und Kapselung von Funktionen planen wir durch eine gemeinsame Datenhaltung der personalisierten Benutzerdaten zu erreichen. Die Daten sollen auf einem gemeinsamen Datenbankserver abgespeichert und Anfragen sowie Einträge von der

gemeinsamen Datenbankklasse *ConfigurationClient*vorgenommen werden. Das entsprechende Interface *IConfigurationClient* stellt die Methodenrümpfe zur Verfügung, die dann in der Klasse ConfigurationClient (bzw. CeWe Client im Falle der Mediengalerie) in den entsprechenden Anwendung implementiert werden.

GUI-Klassen: Da alle Oberflächen für die MHP auf die HAVI [2] zurückgreifen, existiert für jedes Oberflächenmodul im Rahmen des MVC Konzeptes ein eigenes Interface. Referenzen dieser Interfaces für die Oberflächenmodule werden in der *ViewFactory* erzeugt. Ein konkretes, bei beiden Anwendungen verwendetes Modul der Dialogschicht ist der *LoginScreen*. Das Interface stellt die Methoden und Konstanten zur Verfügung, die dann in den korrespondierenden *HAVI* Paketen zu Mediengalerie und dem Tippspiel ihre spezifische Ausprägung finden.

Letztendlich wird sich bei beiden Anwendungen eine Paketstruktur finden, in der die Implementationsdetails für gemeinsame Module, wie z.B. die GUI oder der Datenaustausch mit dem Kooperationspartner CeWe Color, gekapselt sind.

[2]Home Audio Video Interoperability

4.2.4 Detailliertes statisches Modell

4.2.4.1 Übersicht der Klassen

Die folgenden Abbildungen stellen die Grundlage für die spätere Implementierung der personalisierten Mediengalerie dar. Die zu verwendeten Klassen werden gemäß dem Prinzip der Widerverwendbarkeit in die beiden Pakete *mgallery* und *common* unterteilt. Folgende Abbildung zeigt die Struktur der Pakete.

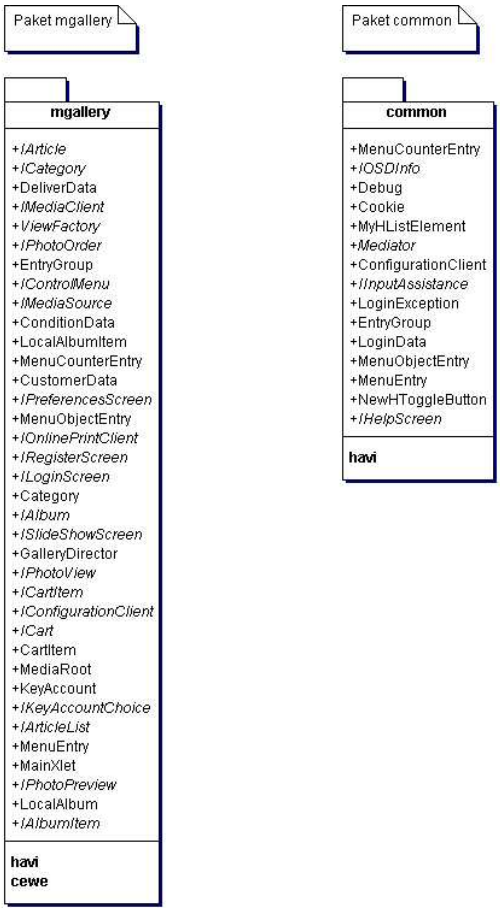

Abbildung 4.2: Paketstruktur

Die Klassen des Pakets *common* beinhalten mehrheitlich Interfaces und Klassen mit Funktionalitäten und Methodenrümpfen die von mehreren auf der MHP basierenden Anwendungen implementiert werden können. Alle Klassen des Pakets *mgallery* stellen ausschließlich Funktionalitäten für die Mediengalerie zur Verfügung.

4.2.4.2 Paket common

Das Paket *common* weist die, für das Konzept des Model View Control typischen und in Kapitel 4.3.2 Abschnitt 4.2.3.2 bereits hinreichend beschriebene dreiteilige Struktur auf.

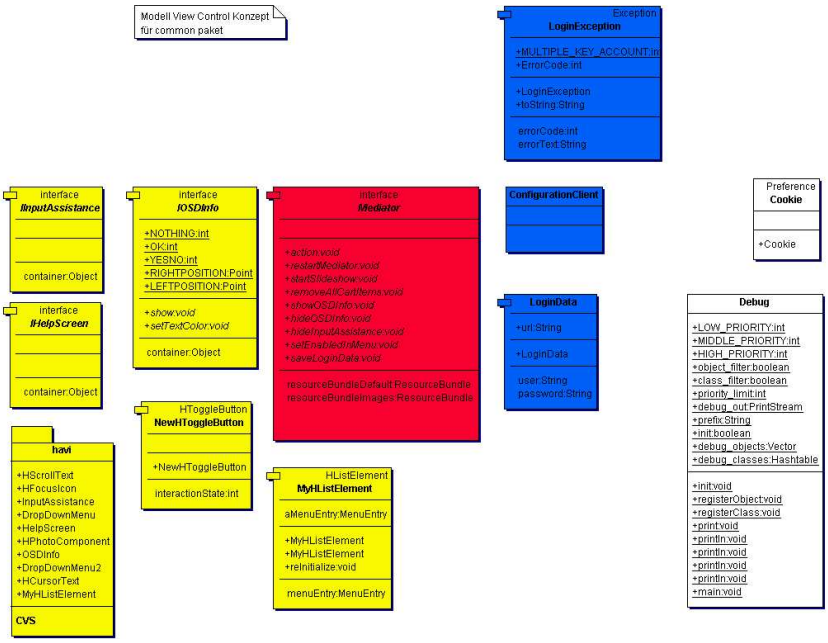

Abbildung 4.3: Model View Control Diagramm common Paket

Die in der Abbildung 4.3 zu sehenden, farblich unterteilten Klassen, sind folgendermaßen zu interpretieren:

- rot markierte Klassen: beinhaltet den GalleryDirector, der in dem Model View Concept als *Controller* fungiert.
- gelb markierte Klassen: umfassen alle Klassen die für bildliche Visualisierung benötigt werden. Zu jeder Klasse existiert ein Interface, das Konstanten und Methodenrümpfe zur Verfügung stellt und in der Klasse *HAVIViewFactory* erzeugt wird. Die Implementierung der GUI Klassen sind in dem Paket *havi* gekapselt, um das Erweitern und Austauschen von Klassen einfach zu halten.
- blau markierte Klassen: bilden die *Modelkomponente* des Model View Controller Konzepts. Dazu gehören die vier Interfaces *IOnlinePrintClient*, *IConfigurationClient*, *IMediaClient* und *IMediaSource*, die über das Paket *cewe* auf die auf den Servern des Kooperationspartners CeWe liegenden Daten zugreifen können.

- weiß markierte Klassen: gehören nicht zum Model View Control Konzept und beinhalten Funktionalitäten die die Speicherung von Einstellungen, die Ausgabe von Fehlermeldungen und im Falle der nächsten Abbildung, die Erzeugung von Einträgen im OnScreenDisplay ermöglichen.

Die im folgenden Diagramm zusammengefassten Klassen der Gruppe *Menuentry* dienen zur Erzeugung von Einträgen im Menü einer jeden auf diesen Klassen des common Pakets basierenden Anwendung.

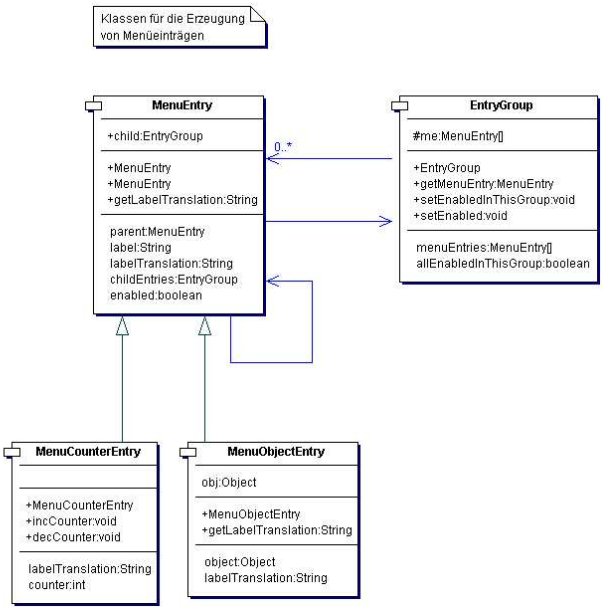

Abbildung 4.4: Model View Control Diagramm

4.2.4.2.1 Paket mgallery Das Klassendiagramm des Pakets Mediengalerie wird ebenfalls in die einzelnen Diagramme für *Model View Control* und die *Datenzugriffsklassen für Alben* und *Datenzugriffsklassen für den Bestellvorgang* unterteilt.

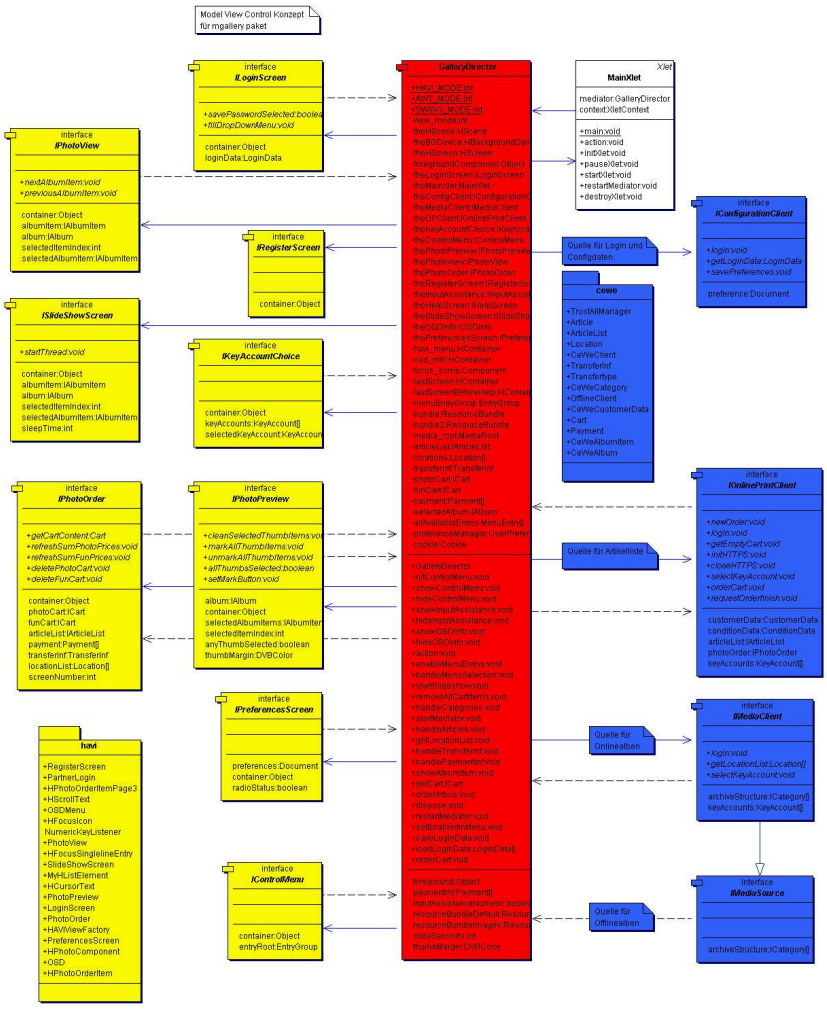

Abbildung 4.5: Model View Control Diagramm

Die beiden Pakete *HAVI* und *CeWe* beinhalten die notwendigen Klassen für die GUI der Oberfläche der Anwendung bzw. für die Kommunikation mit den Servern des Kooperationspartners CeWe Color.
Die folgenden Abbildungen 4.6 und 4.7 zeigen die Datenzugriffsstrukturen für Online und Offline Alben sowie den Bestellvorgang. Anhand der Klasse *CartItem* und dem Interface *IAlbumItem* wird die Verbindung zwischen Datenzugriffsstrukturen für Alben und Bestellvorgang deutlich.

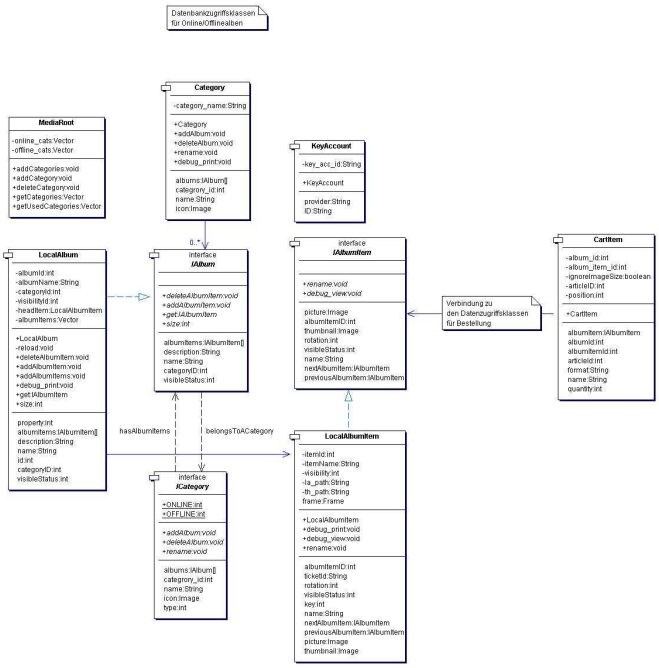

Abbildung 4.6: Datenzugriffsklassen für die Alben

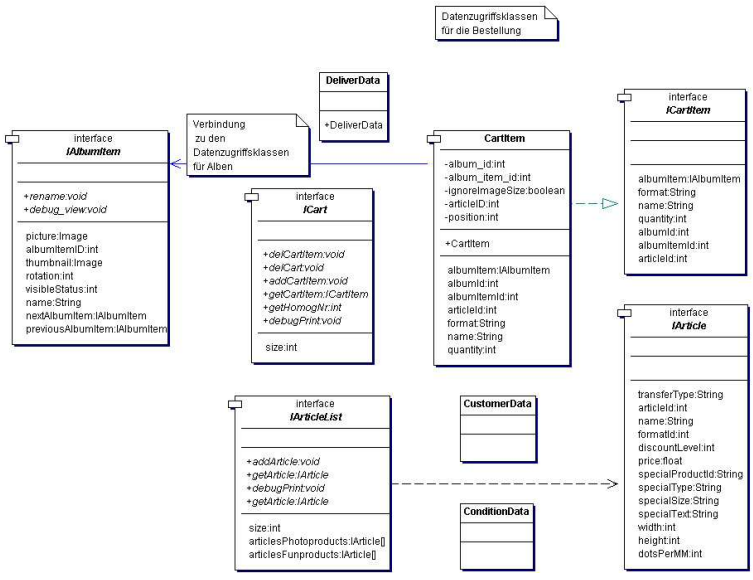

Abbildung 4.7: Datenzugriffsklassen für Bestellvorgang

Ein Zugriff erfolgt über eine Kategorie (Klasse *Category*), diese enthält bestimmte Alben (Interface *IAlbum*)in denen die entsprechenden Bilder (In-

terface *IAlbumItem*)zu finden sind. Das Interface ICartContent enthält Methodenrümpfe die in der Klasse Cart implementiert werden und dazu dienen Artikel dem Warenkorb hinzuzufügen oder zu entfernen. Einer Artikelliste (Klasse *IArticleList*) werden 1..n Artikel (Klasse *Article*) zugeordnet, die dann dem Warenkorb hinzugefügt werden kann. Objekte der Klasse *IAlbumItem* sind als Attribute in der Klasse *CartItem* zu finden. Die durch die Klassen der Datenzugriffstrukturen für Online und Offline Alben erzeugten Objekte stellen also Grundlage für jeden Bestellvorgang dar.

4.2.4.3 Klassen der Dialogschicht

4.2.4.3.1 IControlMenu

- Beschreibung: Dieses Interface wird in der Klasse OSD implementiert und stellt u.a. Methoden zum öffnen und schließen des OSD-Menüs zur Verfügung.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): OSD

Methoden:

- public Object getContainer()
  Liefert Container-Objekt zurück
- public void setEntryRoot(de.showmemhp.mgallery.EntryGroup items)
  Setzt die Pfade für die Menüeinträge

4.2.4.3.2 IKeyAccountChoice

- Beschreibung: Interface für den PartnerLoginScreen, welcher die Auswahl des Photopartners realisiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): PartnerLogin

Methoden:

- public Object getContainer()
  Liefert Container-Objekt zurück
- public void setKeyAccounts(de.showmemhp.mgallery.KeyAccount[] ka)
  Setzt alle PhotoPartner bei denen der Benutzer registriert ist
- public KeyAccount getSelectedKeyAccount()
  Liefert den ausgewählten Photopartner zurück

4.2.4.3.3 ILoginScreen

- Beschreibung: Interface für den LoginScreen, welcher die Anmeldung realisiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): LoginScreen

Methoden:

- public Object getContainer()
  Liefert das Container-Objekt zurück
- public LoginData getLoginData()
  Liefert die Login-Daten zurück
- public boolean savePasswordSelected()
  Abfrage, ob das Passwort gespeichert werden soll
- public void fillDropDownMenu(LoginData[] loginData)
  DropDown-Menü für die E-Mail wird mit Einträgen gefüllt

4.2.4.3.4 IPhotoOrder

- Beschreibung: Interface für den PhotoOrder, welcher die Funktionen des Warenkorb realisiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): PhotoOrder

Methoden:

- public Cart getCartContent()
  Liefert den Inhalt des Warenkorbes zurück

- public Object getContainer()

  Liefert das Container-Objekt zurück

- public void setPhotoCart(ICart pCart)

  Setzt den Warenkorb für die Fotobestellung

- public void setFunCart(ICart fCart)

  Setzt den Warenkorb für die Funproduktbestellung

- public void setArticleList(IArticleList aList)

  Setzt die Artikelliste

- public void setPayment(Payment[] pay)

  Setzen der Bezahlart (Kreditkarte, Rechnung etc.)

- public void refreshSumPhotoPrices()

  Neu berechnen der Fotopreise, wird aufgerufen sobald die Menge oder das Format geändert wird

- public void refreshSumFunPrices()

  Neu berechnen der Funproduktpreise, wird aufgerufen sobald die Menge oder das Format geändert wird

- public void setTransferInf(TransferInf tInf)

  Setzt die Transfer-Information

- public void setLocationList(Location[] locs)

  Setzt die Liste der Filialen, die für die Filialsuche verwendet werden

- public void deletePhotoCart()

  Löschen des kompletten Warenkorbes für die Fotobestellung

- public void deleteFunCart()

  Löschen des kompletten Warenkorbes für die Fotobestellung

- public int getScreenNumber()

  Liefert die Bildschirmnummer zurück

4.2.4.3.5 IPhotoPreview

- Beschreibung: Interface für den PhotoPreview, welcher die Funktionen der Miniaturansicht bereitstellt.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): PhotoPreview

Methoden:

- public void setAlbum(IAlbum cat)

  Setzen des Albums

- public IAlbum getAlbum()

  Liefert das Album zurück

- public Object getContainer()

  Liefert das Container-Objekt zurück

- public IAlbumItem[] getSelectedAlbumItems()

  Liefert ein Array mit den ausgewählten Fotos in der Miniaturansicht zurück

- public int getSelectedItemIndex()

  Liefert den Index des ausgewählten Fotos zurück

- public void setSelectedItemIndex(int position)

  Setzt das Foto auf markiert

- public void cleanSelectedThumbItems()

  Das markierte Foto wird wieder auf unselektiert gesetzt

- public void markAllThumbItems()

  Alle Thumbnails in dem Album werden markiert

- public void unmarkAllThumbItems()

  Alle Thumbnails in dem Album werden demarkiert

- public boolean isAnyThumbSelected()
  Abfrage, ob irgendein Thumbnail in dem Album markiert ist
- public boolean allThumbsSelected()
  Abfrage, ob alle Thumbnails ausgewählt sind
- public void setThumbMargin(DVBColor color)
  Setzt die Markierungsfarbe der Thumbnails

4.2.4.3.6 IPhotoView

- Beschreibung: Interface für den PhotoView, welcher die Funktionen der Vollbildansicht realisiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): PhotoView

Methoden:

- public Object getContainer()
  Liefert das Container-Objekt zurück
- public void setAlbumItem(IAlbumItem albumItem)
  Setzt die Fotos
- public void setAlbum(IAlbum album)
  Setzen des Albums
- public void setSelectedItemIndex(int position)
  Setzen des Index für das ausgewählte Bild
- public int getSelectedItemIndex()
  Liefert den Index des ausgewählten Fotos zurück
- public IAlbumItem getSelectedAlbumItem()
  Liefert das ausgewählte Foto zurück
- public IAlbum getAlbum()
  Liefert das Album zurück

4.2.4.3.7 IPreferencesScreen

- Beschreibung: Interface für den PreferencesScreen, welcher die Einstellungen des Benutzers realisiert. Dort können u.a. die Geschwindigkeit der Slideshow und die Sprache ausgewählt werden.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): PreferencesScreen

Methoden:

- public void setPreferences(Document pref)

  Setzen der Einstellungen

- public Object getContainer()

  Liefert das Container-Objekt zurück

- public boolean getRadioStatus()

  Liefert zurück, ob der RadioButton aktiviert oder deaktiviert ist (d.h. welche Anrede verwendet wird, ob Herr oder Frau)

4.2.4.3.8 IRegisterScreen

- Beschreibung: Interface für den RegisterScreen, welcher die Daten des Benutzers erfasst (u.a. für die Neuanmeldung).

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): RegisterScreen

Methoden:

- public Object getContainer()

  Liefert das Container-Objekt zurück

4.2.4.3.9 ISlideShowScreen

- Beschreibung: Interface für den SlideShowScreen, welcher die Bilder eines Albums in Vollbildansicht automatisch weiterschaltet (ähnlich einer Diashow).

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): SlideShowScreen

Methoden:

- public Object getContainer()
  Liefert das Container-Objekt zurück
- public void setAlbumItem(IAlbumItem albumItem)
  Setzt das Foto
- public void setAlbum(IAlbum album)
  Setzen des Albums
- public void setSelectedItemIndex(int position)
  Setzt den Index des ausgewählten Fotos
- public int getSelectedItemIndex()
  Liefert den Index des ausgewählten Fotos zurück
- public IAlbumItem getSelectedAlbumItem()
  Liefert das ausgewählte Foto zurück
- public IAlbum getAlbum()
  Liefert das Album zurück
- public void startThread()
  Thread für die Slideshow wird gestartet
- public void setSleepTime(int milliseconds)
  Zeitintervall für die Slideshow wird festgelegt (Die Zeit, wann das nächste Bild angezeigt werde soll)

4.2.4.3.10 ViewFactory

- Beschreibung: Das Interface ViewFactory wird in dem Paket HAVI in der Klasse HAVIViewFactory implementiert und dient als AbstarctFactory. Mittels dieser Klasse soll es ermöglicht werden die Views einfach auszutauschen.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): HAVIViewFactory

Methoden:

- IControlMenu getControlMenu()
  Liefert Control-Menü zurück
- public IPhotoView getPhotoView()
  Liefert PhotoView (Vollbildansicht) zurück
- public IPhotoPreview getPhotoPreview()
  Liefert PhotoPreview (Miniaturansicht) zurück
- public IPhotoOrder getPhotoOrder()
  Liefert PhotoOrder (Warenkorb) zurück
- public ILoginScreen getLoginScreen()
  Liefert LoginScreen (Anmeldung) zurück
- public IKeyAccountChoice getKeyAccountChoice()
  Liefert KeyAccountChoice (Photopartnerauswahl) zurück
- public IPreferencesScreen getPrefScreen()
  Liefert PreferencesScreen (Einstellungen) zurück
- public IRegisterScreen getRegisterScreen()
  Liefert RegisterScreen (Neuanmeldung) zurück

- public IInputAssistance getInputAssistance()
  Liefert InputAssistance (Eingabehilfe) zurück
- public ISlideShowScreen getSlideShowScreen()
  Liefert SlideShowScreen (Slideshow) zurück
- public IHelpScreen getHelpScreen()
  Liefert HelpScreen (Hilfe) zurück
- public IOSDInfo getOSDInfo()
  Liefert OSDInfo (Info Bildschirm) zurück
- public int getViewMode()
  Liefert View Modus zurück

4.2.4.4 Klasse der Anwendungslogik (MainXlet&Mediator)

4.2.4.4.1 GalleryDirector

- Beschreibung: Der GalleryDirector funktioniert nach dem Mediator Pattern und beinhaltet die Programmlogik der Applikation *personalisierte Mediengalerie*. Er besitzt Referenzen von allen Klassen bzw. Interfaces und ermöglicht so eine Kommunikation der einzelnen Klassen untereinander.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): keine
- Beziehungen: enthält Referenzen zu allen Klassen und Interfaces.

Methoden:

- private void initControlMenu()
  Initialisieren des OSD-Menüs
- private void setForeground(Object container)
  Setzt den jeweils übergebenen Container in den Vordergrund

- public void showControlMenu(Object button, java.awt.event.KeyEvent event)

  Zeigt das Control-Menü an

- public void hideControlMenu(Object mod)

  Schließen des OSD-Menüs

- public void showInputAssistance(Object mod, java.awt.AWTEvent event)

  Anzeigen der Eingabehilfe

- public void hideInputAssistance(Object mod)

  Ausblenden der Eingabehilfe

- public void showOSDInfo(Object mod, String command)

  Anzeige des OSD-Info (Zeigt Fehlermeldungen an bzw. gibt dem Benutzer Hinweise für die Benutzung des Programms)

- public void hideOSDInfo(Object mod)

  Ausblenden des OSD-Menüs

- public void action(Object module, String actionCommand)

  Kommunikation mit den anderen Klassen

- public void enableMenuEntrys()

  Sichtbarkeit der Menüeinträge werden festgelegt. Z.B. sind zu Anfang alle Einträge des OSD-Menüs (Hauptmenü) solange nicht sichtbar, bis ein Album ausgewählt wurde.

- public void handleMenuSelection(Object module, MenuEntry entry)

  Regelt Auswahl von Menüeinträgen

- public void startSlideshow()

  Startet die Slideshow

- public void removeAllCartItems()

  Leeren des Warenkorbes

- public void handleCategories(Object module, ICategory[] cat)

  Handhabung der Kategorien

- public void startMediator()
  Starten des Mediators
- public void handleArticles(Object module, IArticleList artList)
  Handhabung der Artikel
- public void getLocationList(Object module, String loc)
  Liste für die Filialen
- public void handleTransferInf(Object module, TransferInf tInf)
  TransferInformationen
- public void handlePaymentInf(Object module, Payment[] pay)
  Zahlungsmöglichkeiten (Kreditkarte, Rechnung etc.)
- public void showAlbumItem(Object module, IAlbumItem albumItem)
  Anzeige des Bildes
- public void dispose(Xlet xlet)
  Zerstören des Xlet
- public void setInputAssistanceNumeric(boolean numeric)
  Eingabehilfe für ausschließlich numerische Eingabe
- public ResourceBundle getResourceBundleDefault()
  Liefert das Property Datei (default)
- public ResourceBundle getResourceBundleImages()
  Liefert die Property Datei für die Images
- public void restartMediator(Object mod)
  Neustarten des Mediators
- public void setEnabledInMenu(Object obj, String label, boolean enabled)
  Setzen der Sichtbarkeit der Menüeinträge des OSD-Menüs
- public void saveLoginData(LoginData loginData)
  Speichern der LoginDaten

- public void setSlideSeconds(int millis)

  Zeitintervall für die Slideshow wird festgelegt

- public void setThumbMargin(DVBColor color)

  Farbe für die Markierung der Thumbnails wird festgelegt

- public void orderCart(String loc_id, String transfer_id, String couponid, String coupon_artid, String payment_id, String title, ICart cart)

  Warenkorb Bestellung

4.2.4.4.2 MainXlet

- Beschreibung: Wird zum Starten der Applikation benötigt und initialisiert den GalleryDirector. Der Applikationsmanager kann über die Klasse MainXlet den Zustand des Programms festlegen. Die Zustände die das Xlet annehmen kann, sind: *initialisiert*, *gestartet*, *pausiert* und *zerstört*.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): Xlet

- abgeleitete Klasse(n): keine

Methoden:

- public void initXlet(XletContext context)

  Initialisiert das Xlet. Über den Parameter context kann das Xlet an den Applicationmanager seinen eigenen Status abfragen bzw. beeinflussen(Xlet kann sich z.B. selber pausieren oder zerstören).

- public void startXlet()

  Startet das Xlet bzw. den GalleryDirector.

- public void pauseXlet()

  Pausiert das Xlet.

- public void destroyXlet(boolean unconditional)

  Zerstört das Xlet. Über den Parameter unconditional kann das Xlet seine Zerstörung verhindern, sofern dieser den Wert *false* hat, andernfalls wird das Xlet zerstört.

4.2.4.5 Klassen der Client Kommunikation

4.2.4.5.1 IAlbum

- Beschreibung: Dieses Interface wird in der Klasse CeWeAlbum implementiert und liefert für die Alben u.a. den Name des Albums, eine Beschreibung etc.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): CeWeAlbum

Methoden:

- public IAlbumItem[] getAlbumItems()
  Liefert ein Array von Alben
- public String getDescription()
  Liefert die Beschreibung für ein Album
- public String getName()
  Liefert den Namen des Albums
- public int getCategoryID()
  Liefert die KategorieID eines Albums
- public int getVisibleStatus()
  Liefert den Sichtbarkeitsstatus
- public void deleteAlbumItem()
  Zum löschen eines Fotos im Album
- public void addAlbumItem(IAlbumItem albumItem)
  Einfügen eines Fotos in ein Album
- public IAlbumItem get(int i)
- public int size()
  Liefert die Größe des Albums

4.2.4.5.2 IAlbumItem

- Beschreibung: Dieses Interface wird in der Klasse CeWeAlbumItem implementiert und liefert das Bild bzw. Thumbnails, den Namen des Bildes etc.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): CeWeAlbumItem

Methoden:

- public Image getPicture()
  Liefert das Bild zurück
- public int getAlbumItemID()
  Liefert die ID des Bildes zurück
- public Image getThumbnail()
  Liefert das Bild in Thumbnailgröße zurück
- public int getRotation()
  Liefert die Rotation des Bildes zurück (siehe XTCI-Spezifikation)
- public int getVisibleStatus()
  Liefert Sichbarkeitsstatus zurück
- public void rename(String s)
  Umbenennen des Bildes
- public String getName()
  Liefert den Namen des Bildes zurück
- public IAlbumItem getNextAlbumItem()
  Liefert das nächste Bild zurück
- public IAlbumItem getPreviousAlbumItem()
  Liefert das vorherige Bild zurück

4.2.4.5.3 IArticle

- Beschreibung: Dieses Interface wird in der Klasse Article implementiert und liefert Informationen über die Artikel, wie die ArtikelID, FormatID, den Preis etc zurück

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): Article

Methoden:

- public String getTransferType()

  Liefert den Transfer-Typ zurück

- public int getArticleId()

  Liefert die ArtikelID zurück

- public String getName()

  Liefert den Namen des Artikels zurück

- public int getFormatId()

  Liefert die FormatID des Artikels zurück

- public int getDiscountLevel()

  Liefert den Discount Level des Artikels zurück (werden viele Artikel einer Sorte bestellt, kann der Preis gesenkt werden)

- public float getPrice()

  Liefert den Preis des Artikels zurück

- public String getSpecialProductId()

  Liefert die ProduktID des Artikels zurück

- public String getSpecialType()

  Liefert den Spezialtyp des Artikels zurück

- public String getSpecialSize()
  Liefert die Spezialgröße des Artikels zurück
- public String getSpecialText()
  Liefert den Spezialtext zurück
- public int getWidth()
  Liefert Breite des Artikels
- public int getHeight()
  Liefert Höhe des Artikels
- public int getDotsPerMM()

4.2.4.5.4 IArticleList

- Beschreibung: Dieses Interface wird in der Klasse ArticleList implementiert und fügt Artikel zu einer Artikelliste zusammen

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): ArticleList

Methoden:

- public void addArticle(IArticle a)
  Fügt einen Artikel einer Liste hinzu
- public int getSize()
  Liefert die Größe der Artikelliste
- public IArticle getArticle(int nr)
  Liefert einen bestimmten Artikel aus der Artikelliste zurück
- public IArticle[] getArticlesPhotoproducts()
  Liefert die Artikelliste für die Fotoprodukte zurück

- public IArticle[] getArticlesFunproducts()
  Liefert die Artikelliste für die Funprodukte zurück
- public double getPrice(String format, String transfer_type, int discount)
  Liefert den Preis für den Artikel zurück

4.2.4.5.5 ICartItem

- Beschreibung: Dieses Interface wird in der Klasse CartItem implementiert und ist ein Artikel, der in den Warenkorb gelegt werden kann (dieser enthält u.a. das Format und die Menge des Artikels der bestellt werden soll)

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): CartItem

Methoden:

- public IAlbumItem getAlbumItem()
  Liefert das zu bestellende Bild zurück
- public String getFormat()
  Liefert das Format des zu bestellenden Bildes zurück
- public void setFormat(String format)
  Legt das Format des Bildes fest
- public String getName()
  Liefert Name des zu bestellenden Produktes
- public int getQuantity()
  Liefert die Menge, die von diesem Produkt bestellt werden soll
- public int getAlbumId()
  Liefert die AlbumID zurück

- public int getAlbumItemId()
  Liefert die BildID zurück
- public int getArticleId()
  Liefert die ArtikelID zurück
- public void setQuantity(int nr)
  Setzen der Bestellmenge

4.2.4.5.6 ICart

- Beschreibung: Dieses Interface wird in der Klasse Cart implementiert und bietet Methoden, um den Warenkorb bzw. einzelne Artikel aus dem Warenkorb zu löschen, sowie Artikel den Warenkorb hinzu zu fügen.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): Cart

Methoden:

- public void delCartItem(int position)
  Löschen eines einzelnen Artikels aus dem Warenkorb
- public void delCart()
  Leeren des kompletten Warenkorbes
- public void addCartItem(ICartItem cartItem)
  Hinzufügen eines Artikels in den Warenkorb
- public int getSize()
  Liefert die Größe des Warenkorbes zurück
- public ICartItem getCartItem(int nr)
  Liefert einen bestimmten Artikel aus dem Warenkorb zurück
- public int getHomogNr(String format)
  Liefert die Anzahl homogener Bestellung zurück (wichtig für die Preisberechnung da es dort einige Rabatte gibt)

4.2.4.5.7 ICategory

- Beschreibung: Dieses Interface wird in der Klasse CeWeCategory implementiert und bietet Methoden die u.a. Alben in Kategorien einteilt.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): CeWeCategory

Methoden:

- public IAlbum[] getAlbums()
  Liefert ein Array von Alben
- public int getCategrory_id()
  Liefert die KategorieID zurück
- public String getName()
  Liefert den Namen der Kategorie zurück
- public void addAlbum(IAlbum album)
  Fügt ein Album einer Kategorie hinzu
- public void deleteAlbum(IAlbum album)
  Album aus einer Kategorie entfernen
- public void rename(String s)
  Umbenennen einer Kategorie
- public int getType()
  Liefert den Typ zurück (online oder offline)

4.2.4.6 Klassen der Datenbankschicht&Server Kommunikation

4.2.4.6.1 CeweClient

- Beschreibung: Diese Klasse bildet den CeweClient der mit den Servern von CeWe über XML-Dokumente kommuniziert. Als Antwort liefert der CeWe-Server ebenfalls XML-Dokumente, die in verschiedenen Methoden geparst werden und die Werte als Objekte abspeichert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): IOnlinePrintClient, IMediaClient
- abgeleitete Klasse(n): keine

Methoden:

- public static String replace(String template, String[] values, String dtd_title)
- private static String encode(String s)
  Methode zum kodieren
- public static String decode(String s)
  Methode zum dekodieren
- private String processFile(String filename)
- protected XMLElement getContent(BufferedReader br)
  Liefert das XML-Element
- public void requestXTCI2(String client_name, String client_version, String protocol_version)
  XTCI2-Request (Anfrage) wird ausgeführt
- protected void handleXTCI2(XMLElement cont)
  XTCI2-Request (Antwort) wird verarbeitet
- public void requestKALogin(String account, String pwd)
  KALogin-Request wird ausgeführt (Anfrage für Photopartner)
- protected void handleKALogin(XMLElement cont)
  KALogin-Request wird verarbeitet

- public void requestSession(String KeyAccount, String client_name, String client_version, String protocol_version)

  Session-Request wird ausgeführt

- protected void handleSession(XMLElement cont)

  Session-Request wird verarbeitet

- protected void requestLogin(String ocsid, String account, String pwd)

  Login-Request wird ausgeführt

- protected void requestCategory(String ocsid)

  Anfrage für die Kategorie wird ausgeführt

- protected void handleCategory(XMLElement cont)

  Kategorie-Request wird verarbeitet

- protected void requestAlbumList(String ocsid)

  AlbumList-Request wird ausgeführt

- protected void handleAlbumList(XMLElement cont)

  AlbumList-Request wir verarbeitet

- protected void requestAlbumItemList(String ocsid, String albumId)

  AlbumItemList-Request wird ausgeführt

- protected void handleAlbumItemList(XMLElement cont)

  AlbumItemList wird verarbeitet

- protected void requestArticleList(String ocsid)

  ArticleList-Request wird ausgeführt

- protected void handleArticleList(XMLElement cont)

  ArticleList wird verarbeitet

- protected void requestTransferList(String ocsid)

  TransferList-Request wird ausgeführt

- protected void handleTransferList(XMLElement cont)

  TransferList wird verarbeitet

- protected void requestPaymentInfo(String ocsid)

  PaymentInfo-Request wird ausgeführt

- protected void handlePaymentInfo(XMLElement cont)

  PaymentInfo wird verarbeitet

- public void orderCart(String loc_id, String transfer_id, String couponid, String coupon_artid, String payment_id, String title, ICart cart)

  Bestellvorgang wird vorbereitet

- protected void requestOrderdeclaration_FC(String ocsid, String loc_id, String transfer_id, String couponid, String coupon_artid, String payment_id, String title, ICart cart)

  Orderdeclaration-FC-Request wird ausgeführt (für Selbstabholung)

- protected void handleOrderdeclaration_FC(XMLElement cont)

  Orderdeclaration-FC wird verarbeitet

- private String createOrderPositions(ICart cart)

  Warenkorbposition wird erzeugt

- protected void requestLocationList(String ocsid, String plz_or_city)

  LocationList-Request wird ausgeführt (Anfrage für die Filialliste)

- protected void handleLocationList(XMLElement cont)

  LocationList (Filialliste) wird verarbeitet

- private void putAlbumsInCategorys()

  Album wird in eine bestimmte Kategorie gesetzt

- protected void loadAlbumItems()

  Laden der Bilder

- public static String getGetQuery(String path, String host, String parameter)

  GET-Methode wird ausgeführt

- public static String getPostQuery(String xml_part, String xtci_path, String host)

  POST-Methode wird ausgeführt

- protected void initHTTP()
  Initialisieren der HTTP-Verbindung
- protected void initHTTPS()
  Initialisieren der HTTPS-Verbindung
- protected void closeHTTP()
  Schließen der HTTP-Verbindung
- protected void closeHTTPS()
  Schließen der HTTPS-Verbindung
- protected BufferedReader sendMessage(PrintWriter out, String message)
  Senden der Nachricht
- public KeyAccount[] getKeyAccounts()
  Liefert die Liste der Photopartner des Benutzers zurück
- public ICategory[] getArchiveStructure()
  Liefert die Kategoriestruktur zurück
- public IArticleList getArticleList()
  Liefert die Artikelliste zurück
- public Location[] getLocationList(String loc)
  Liefert die LocationList (Filialliste) zurück

4.2.4.6.2 IConfigurationClient

- Beschreibung: Dieses Interface soll in der Klasse CeweClient implementiert werden und soll Konstanten und Methodenrümpfe zur Verfügung stellen, über die auf die kundenspezifischen Logindaten sowie Konfigurations-Dateien zugegriffen werden kann.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): CeweClient (vorgesehen)

Methoden:

- public void login(LoginData login)
  Login mit den Logindaten wird ausgeführt
- public LoginData getLoginData(String ServerType)
  Liefert die Logindaten zurück
- public Document getPreference()
  Liefert die Einstellungen zurück
- public void savePreferences(Document prefs)
  Speichert die Einstellungen ab

4.2.4.6.3 IOnlinePrintClient

- Beschreibung: IOnlinePrintClient ist für den Bestellvorgang zuständig. Dazu gehören Zugriffe auf die durch den Kunden spezifizierte ArticleList und den dazu passende ICart (Warenkorb mit Bildern oder Funprodukten).

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): CeweClient

Methoden:

- public void login(LoginData login)
  Login wird mit Logindaten ausgeführt
- public void getEmptyCart()
  Liefert einen leeren Warenkorb zurück
- public IArticleList getArticleList()
  Liefert die Artikelliste zurück
- public IPhotoOrder getPhotoOrder()
  Liefert das Warenkorb-Objekt zurück

- public void selectKeyAccount(KeyAccount ka)

  Setzt den ausgewählten Photopartner

- public KeyAccount[] getKeyAccounts()

  Liefert eine Liste der Photopartner de Benutzers

- public void orderCart(String loc_id, String transfer_id, String couponid, String coupon_artid, String payment_id, String title, ICart cart)

  Bestellvorgang wird eingeleitet

4.2.4.6.4 IMediaClient

- Beschreibung: Der Zugriff auf Offline bzw. Online Alben werden durch Methodenrümpfe der Interfaces IMediaSource bzw. IMediaClient implementiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine

- abgeleitete Klasse(n): CeweClient

Methoden:

- public void login(LoginData login)

  Login mit Logindaten wird ausgeführt

- public ICategory[] getArchiveStructure()

  Liefert die Kategoriestruktur zurück

- public Location[] getLocationList(String loc)

  Liefert die LocationList (Filialliste) zurück

- public void selectKeyAccount(KeyAccount ka)

  Setzt den ausgewählten Photopartner

- public KeyAccount[] getKeyAccounts()

  Liefert Liste der Photopartner des Benutzers

4.2.4.6.5 IMediaSource

- Beschreibung: Der Zugriff auf Offline bzw. Online Alben werden durch Methodenrümpfe der Interfaces IMediaSource bzw. IMediaClient implementiert.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): CeweClient (vorgesehen)

Methoden:

- public ICategory[] getArchiveStructure()
  Liefert die Kategoriestruktur zurück

4.2.5 Detailliertes dynamisches Modell

4.2.5.1 Sequenzdiagramme der Mediengalerie

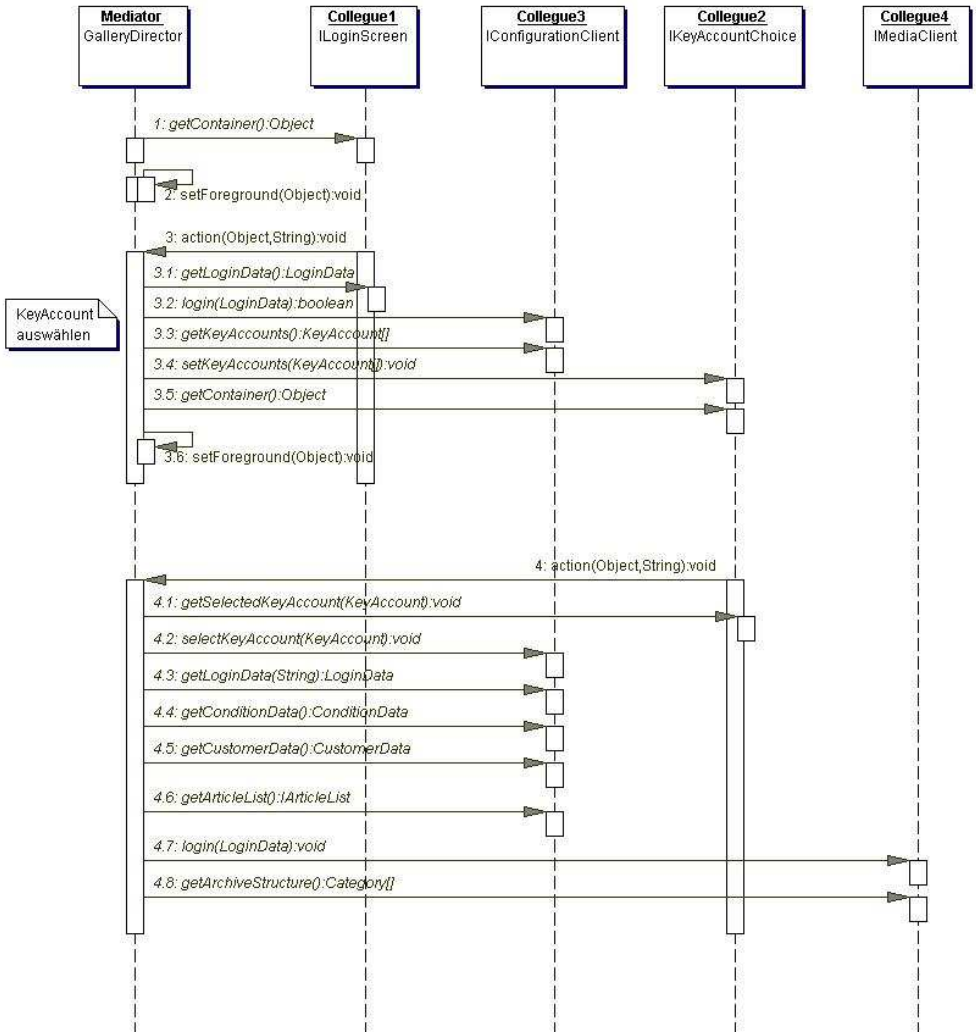

Abbildung 4.8: Sequenzdiagramm Anmeldung

4.2.5.1.1 Anmeldung

Zuerst wird der *HScene* des Container-Objekts des LoginScreens durch die Methode *getContainer* übergeben. Durch die Methode setForeground wird die HScene mit dem Container des LoginScreens in den Vordergrund gesetzt und somit auf dem Bildschirm sichtbar. Sobald der Benutzer sein *Login* und *Passwort* eingetragen sowie den *OKButton* betätigt hat, wird die Methode *action* aufgerufen. Durch die übergebenen Parameter *String* und *Objekt* kann der GalleryDirector (Mediator) den Methodenaufruf als Aufforderung zum Einloggen identifizieren und führt daraufhin die Methode getLoginData aus. Nachdem die Logindaten ausgelesen worden sind, beauftragt der GalleryDirector mit der Methode *login(LoginData)* den IConfigurationClient mit der Abgleichung der Logindaten in der Datenbank. Falls der Login korrekt ist, initialisiert der GalleryDirector das Auswahlmenü der CeWe Partner. Zunächst werden über die Methoden *getKeyAccount* und *setKeyAccount* die verfübaren CeWe Partner dem Container Objekt *IKeyAccountChoice* zugewiesen. Daraufhin wird das Container Objekt mit der Methode *getContainer* der HScene übergeben und durch die Methode *setForeground* in den Vordergrund gesetzt. Durch die Methoden *getSelectedKeyAccount* bzw. *selectKeyAccount* werden die gewünschten CeWe Partner fokusiert und ausgewählt. Die kundenspezifischen Daten wie z.B. Liefermodalitäten oder eventuelle Gutscheinboni ordnet der GalleryDirector durch die Methoden *getCustomerDate&getConditionData* dem jeweiligen Benutzer zu. Ebenso werden die verfügbaren Artikel durch die Methode *getArticleList* geladen. Als letztes stellt der GalleryDirector mit Hilfe der Logindaten und der Methode *getArchiveStructure* eine Anfrage an die Datenbankzugriffsklasse *IMediaClient* die in einer Rückgabe der zugehörigen Kategoriestruktur resultiert.

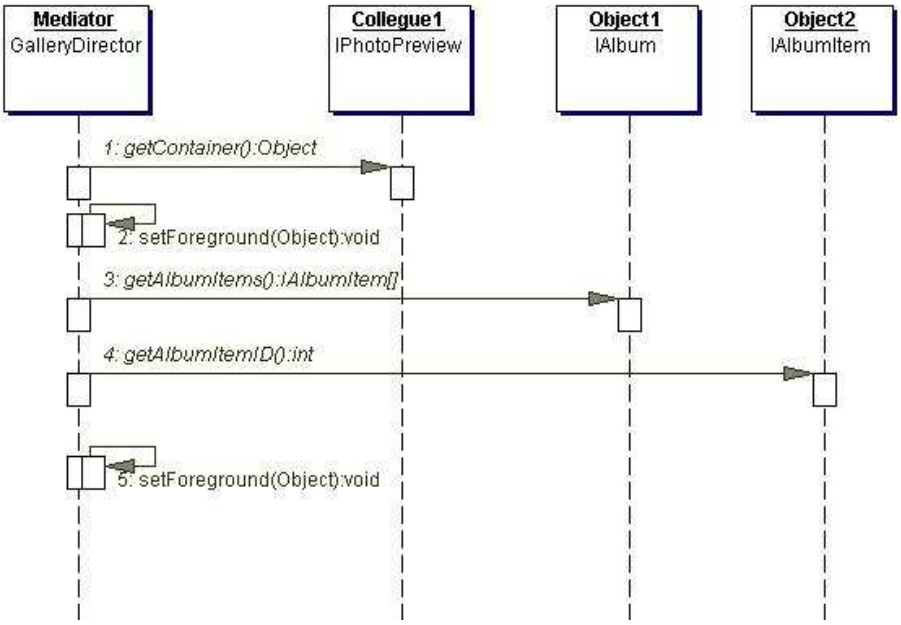

Abbildung 4.9: Sequenzdiagramm Auswahl

4.2.5.1.2 Auswahl

Zunächst bekommt die *HScene* das Container Objekt der Klasse PhotoPreview übergeben. Durch die Methode *setForeground* wird die HScene mit dem Container des PhotoPreview in den Vordergrund gesetzt und somit auf dem Bildschirm sichtbar. Die Auswahl der Alben erfolgt durch den Aufruf der Methoden *getAlbumItems* bzw. *getAlbumID*. Der daraus resultierende modifizierte Bildschirm wird durch die Methode *setForeground* neu angezeigt.

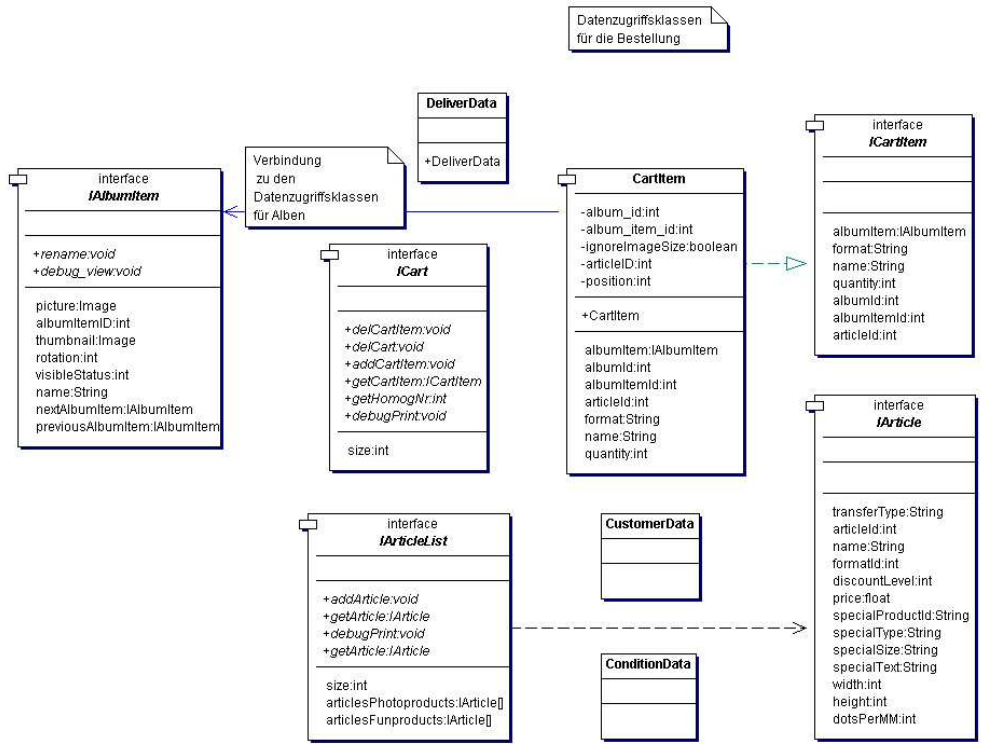

Abbildung 4.10: Sequenzdiagramm Bestellung

4.2.5.1.3 Bestellung

Zunächst weist der GalleryDirector der *HScene* durch Aufruf der Methode *getContainer* das Container Objekt des IPhotoOrderScreens zu. Durch die Methode *getContent* erhält das IPhotoOrder Objekt das Objekt Cart, in dem die Warenkorbfunktionalitäten implementiert sind. Nun setzt der GalleryDirector durch die Methode *setForeground* die HScene mit dem IPhotoOrder Objekt in den Vordergrund. Die nächste Aktion des GalleryDirectors wird durch die Betätigung des *Weiter Buttons* ausgelöst. Diese zieht den Aufruf der *action* Methode nach sich, die über die übergebenen Parameter *Objekt* und *String* den entsprechenden Button eindeutig identifiziert. Nun wird durch die Methoden *getContainer* und *setForeground* der *Bestellvorgang abschließen* Bildschirm der HScene zugewiesen und in den Vordergrund gesetzt. Durch den *Abschicken Button* wird der GalleryDirector beauftragt, den Bestellvorgang durch Aufruf der Methode *newOrder* einzuleiten. Übergeben werden dieser Methode die Parameter *Cart(Warenkorb)* und *DeliverData(Lieferdaten)*, die an den IOnlinePrintClient übermittelt werden, der dann die Verbindung mit dem CeWe Server herstellt.

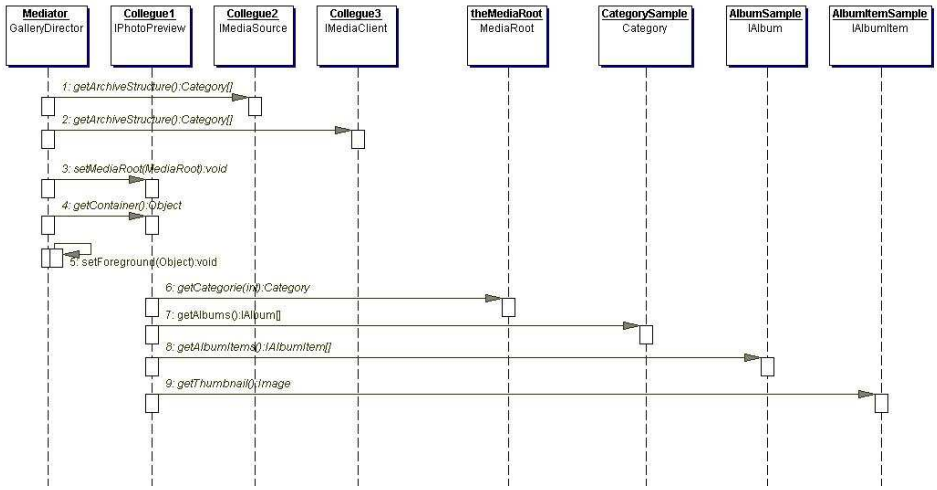

Abbildung 4.11: Sequenzdiagramm Thumbnailansicht

4.2.5.1.4 Thumbnails

Zunächst holt der GalleryDirector mit den Methoden *getArchiveStructure* die Online (IMediaClient) bzw. Offline (IMediaSource) Kategorien des jeweiligen Benutzers. Danach baut der GalleryDirector durch die Methode *setMediaRoot* aus den einzelnen Kategorien eine baumartige Struktur auf. Anschließend weist der GalleryDirector der *HScene* durch Aufruf der Methode *getContainer* das Container Objekt des IPhotoPreviewScreens (Thumbnailsansicht) zu und setzt die HScene mit dem IPhotoPreview Objekt in den Vordergrund. Die Auswahl einer bestimmten Kategorie wird in der Thumbnailansicht durch Aufruf der Methode *getCategorie* realisiert. Die in der ausgewählten Kategorie verfügbaren Alben werden durch die Methode *getAlbums* der Thumbnailansicht zugewiesen. Die zu einem Album zugehörigen Bilder werden dann durch die Methoden *getAlbumItems&getThumbnail* geladen und angezeigt.

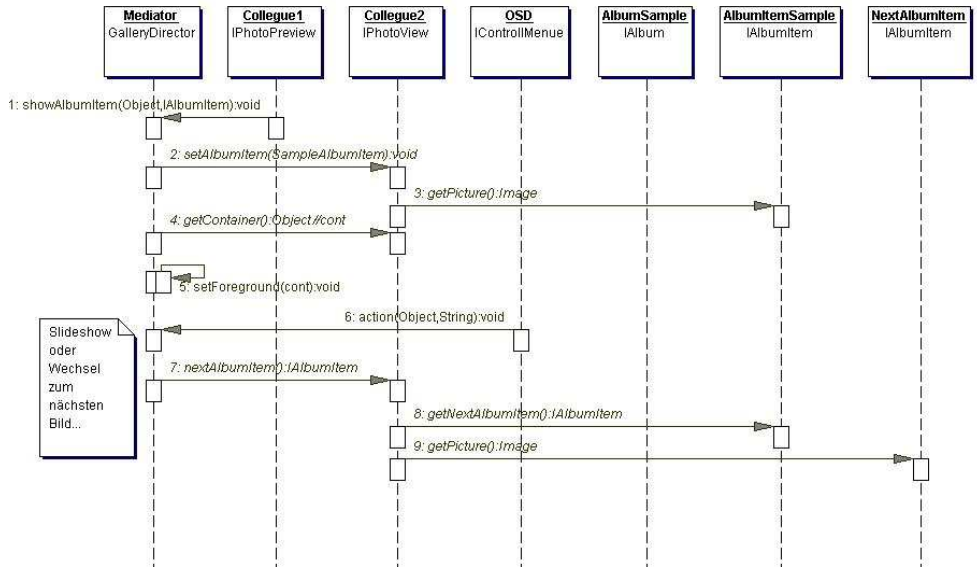

Abbildung 4.12: Sequenzdiagramm Vollbildansicht und Slideshow

4.2.5.1.5 Vollbild

Der IPhotoPreviewScreen beauftragt den GalleryDirector durch die Methode *showAlbumItems* ein ausgewähltes Bild als Vollbild anzuzeigen. Daraufhin ruft der GalleryDirector die Methode *setAlbumItems* des PhotoView auf. Die Methode *getPicture* liefert als Rückgabewert ein Image, welches dann als Vollbild angezeigt wird. Anschließend weist der GalleryDirector der *HScene* durch Aufruf der Methode *getContainer* das Container Objekt des IPhotoViewScreens (Vollbildansicht) zu und setzt die HScene mit dem IPhotoView Objekt in den Vordergrund.
Das Starten der Slideshow erfolgt durch Auswahl des Menüpunktes *Slideshow starten*, welches über den Hauptmenübutton zu erreichen ist, bzw. durch den Aufruf der Methode *action* durch den GalleryDirector.

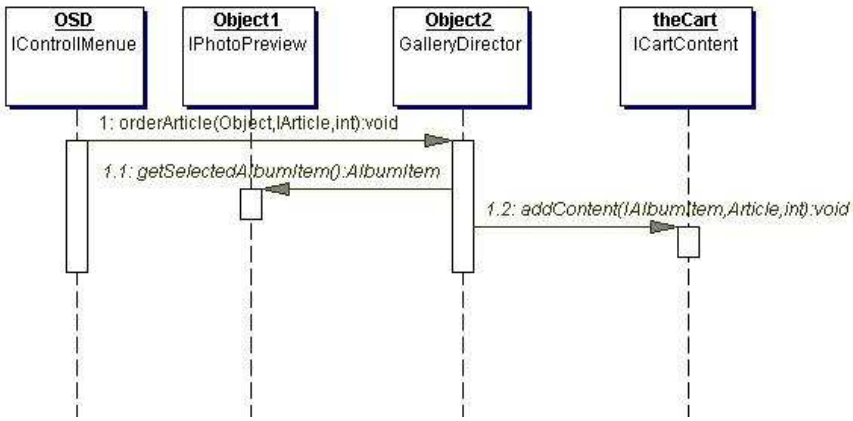

Abbildung 4.13: Sequenzdiagramm Warenkorb

4.2.5.1.6 Warenkorb

Das im *OSD Menü* ausgewählte Bild wird durch die Methode *orderArticle* dem GalleryDirector gemeldet. Der GalleryDirector holt mit der Methode *getSelectedAlbumItem* das ausgewählte Foto (AlbumItem) und weist es durch die Methode *addContent* dem Warenkorb (CartContent) zu.

4.2.6 Nutzungsvoraussetzungen

Da beide Anwendungen auf dem heimischen Fernseher ablaufen und entweder terrestrisch, via Kabel oder Satellit auf die Set-Top-Box heruntergeladen werden, wird die Dokumentation für die Endbenutzer in Form einer Hilfefunktion als Teil des Programms beigefügt. Optional ist es auch denkbar, dass z.B. für die *personalisierte Mediengalerie* ein Handbuch auf den WWW-Seiten von CeWe Color im PDF-Format oder als WWW-taugliches HTML-Format zur Verfügung steht. So ist möglich, sich schon vor der Nutzung einer Anwendung mit der Funktionsweise vertraut zu machen. Die Gliederung in Kapitel und Formatierung werden weitestgehend identisch sein, so dass sich Benutzer die nur die Hilfefunktion des Programms oder die WWW-Dokumentation kennen problemlos auch in der anderen Form zurechtfinden. Ebenso sollen Abbildungen und Tabellen in der Online-Version genauso erscheinen wie in der Hilfefunktion des Programms selbst. Ein wichtiger Punkt stellen die zum Ausführen der Anwendung nötigen Systemvoraussetzungen, also Hardwarekomponenten wie Set-Top-Box oder MHP-fähiger Fernseher und verwendeter MHP-Standard dar. Diese Informationen müssen auf jeden Fall vor dem Herunterladen der Anwendung einsehbar sein. Andernfalls ist nicht auszuschließen, dass die Anwendungen auf einigen Set-Top-Boxen aufgrund z.B.

eines nicht unterstützten MHP-Standards nicht auszuführen ist. Die Einsicht der Systemvoraussetzungen könnte auf den Webseiten der Hersteller der Anwendungen oder über den Videotext der Broadcaster ermöglicht werden. Der Programmstart erfolgt über das Auswahlmenü des Startbildschirms der nach dem Einschalten der Set-Top-Box erscheint. Denkbar ist, dass die Anwendung z.B. nach einer Werbung im Fernsehen zum Herunterladen angeboten wird.

4.3 Entwurf des Tippspiels Tipp2Win

4.3.1 Einleitung

4.3.1.1 Ausgangssituation und Zielsetzung

Die Projektgruppe beginnt nun parallel zur Entwicklung der Mediengalerie mit den Arbeiten an der zweiten Anwendung, dem Fußball-Tippspiel „Tipp2Win“. Die Entwicklung der zweiten alternativen Anwendung (Electronic Program Guide) wird nicht weiter fortgeführt. Gründe hierfür sind zum einen der zeitliche Aufwand und zum anderen die Tatsache, dass es schon zahlreiche Implementierungen eines EPGs gibt.
Über Tipp2Win soll es möglich sein, zu Fußballspielen der Europameisterschaft, Weltmeisterschaft oder auch der Bundesliga verschiedenste Tipps abzugeben (Spielergebnis, Torschützenkönig, etc.). Dies soll im Wettbewerb mit anderen Nutzern der Anwendung geschehen. Da diese Anwendung ebenfalls personalisiert ist, muss sich der Nutzer vor der Abgabe von Tipps anmelden.

4.3.1.2 Aufbau des Entwurfs

Der hier vorgelegte Entwurf stellt die Arbeitsgrundlage für die Implementierung des Tippspieles dar. In diesem ersten Kapitel wird ein erster Überblick über die Vorgaben der Projektgruppe in Hinblick auf das Tippspiel gegeben. Die einzelnen Module der verschiedenen Anwendungen lassen sich in unterschiedliche Schichten einteilen. Dies sind die Dialogschicht, Anwendungungslogik (Mediator), Client Kommunikation, Server Kommunikation und die Datenbankschicht. Die näheren Beschreibungen sind in einem eigenen Kapitel 4.3.2 in diesem Entwurf untergebracht.
In den folgenden beiden Kapiteln werden die Module, die implementiert werden sollen, näher beschrieben. In dem Kapitel 4.3.3 werden die einzelnen Klassen beschrieben.
Das Kapitel 4.3.4 enthält die Sequenzdiagramme, in denen u.a. das dynamische Verhalten beim Einloggen aufgezeigt wird.

4.3.2 Architektur

4.3.2.1 Programmschichten

In diesem Abschnitt wird zum besseren Verständnis eine Unterteilung der Software in mehrere Schichten vorgenommen. Dabei wird das gesamte Softwaresystem, also sowohl die Client-, als auch die Serverseite inklusive der Datenbank betrachtet. Der Client besteht somit aus

- einer Dialogschicht, die eine Schnittstelle zum Benutzer darstellt,
- einer Schicht welche die Anwendungslogik kapselt
- und einer Kommunikationsschicht, die einen Datenaustausch mit dem Server ermöglicht.

4.3.2.2 Client Server Architektur

Tipp2Win ist eine Applikation, bei der sehr große Mengen an Daten verwaltet werden müssen. Bei einem Fußball-Tippspiel sind das zum einen alle Daten zu den aktuellen Ergebnissen der Spiele, zum anderen aber auch die Tippabgaben, personenbezogenen Daten und die Punkte der Teilnehmer des Tippspiels. Für die Verwaltung einer so großen Datenmenge muss eine Datenbank eingesetzt werden. Da diese Datenbank Daten zu vielen Teilnehmern verwaltet, die alle mit diesen Daten versorgt werden müssen, soll in Tipp2Win eine Client-Server Architektur eingesetzt werden (siehe Abbildung 4.14). Die Daten werden dabei in einer zentralen Datenbank gehalten, während jedem Teilnehmer eine Client-Software zur Verfügung gestellt wird. Die Client-Software bietet die Möglichkeit, die für den jeweiligen Benutzer relevanten Daten von der Datenbank zu laden und über die GUI des Clients zur Anzeige zu bringen. Neue Tipps, die vom Benutzer abgegeben werden, können ebenfalls über die GUI entgegengenommen werden. Für die Administratoren des Tippspiels soll zusätzlich ein Administrations-Programm, das sogenannte Tipp2Win-Admintool erstellt werden. Dieses Tool soll die Administration der Datenbank erleichtern und wird ebenfalls in Form eines Clients realisiert.
Die Datenbank selbst wird hinter dem Datenbankmodul *(DBModul)* gekapselt. Das Datenbankmodul bildet die Schnittstelle der Datenbank. Für Anfragen werden geeignete Methoden bereitgestellt, die den Zugriff auf die Datenbank ermöglichen. Diese Methoden generieren die entsprechenden SQL-Statements, mit denen dann die Datenbank angesprochen wird. Die Schnittstelle zu den Clients wird durch Servlets gebildet. Der *InfoServer* bildet dabei die Schnittstelle für die Tipp2Win-Clients und der *AdminServer* die Schnittstelle für die Admin-Tools. Über die *POST*-Methode können Anfragen an die Servlets gestellt werden. Die Servlets werden auf einem geeigneten Webserver, wie z.B. dem Tomcat-Server ausgeführt und besitzen über das *(DB-Modul)* Zugang zur Datenbank. Die Gegenstücke zu den Servlets bilden auf Seite der Clients der *InfoConnector* und der *AdminConnector*, die jeweils durch ein Modul in den Clients gebildet werden. Die Anfragen, die von den Benutzern der Clients indirekt erstellt werden, werden von den Connector-Modulen in entsprechende XML-Files transformiert. Die XML-Files werden

dann über das Internet durch Aufruf der *POST*-Methoden der Servlets, an diese gesendet. In den Servlets werden die XML-Files geparst und daraufhin die entsprechenden Methoden des *(DBModuls)* aufgerufen, welche schließlich die SQL-Anfragen generieren, die an die Datenbank übermittelt werden. Die Antworten werden von den Servlets wiederum in XML-Files umgewandelt, die dann Clientseitig von den Connector-Modulen geparst und über die GUI angezeigt werden.

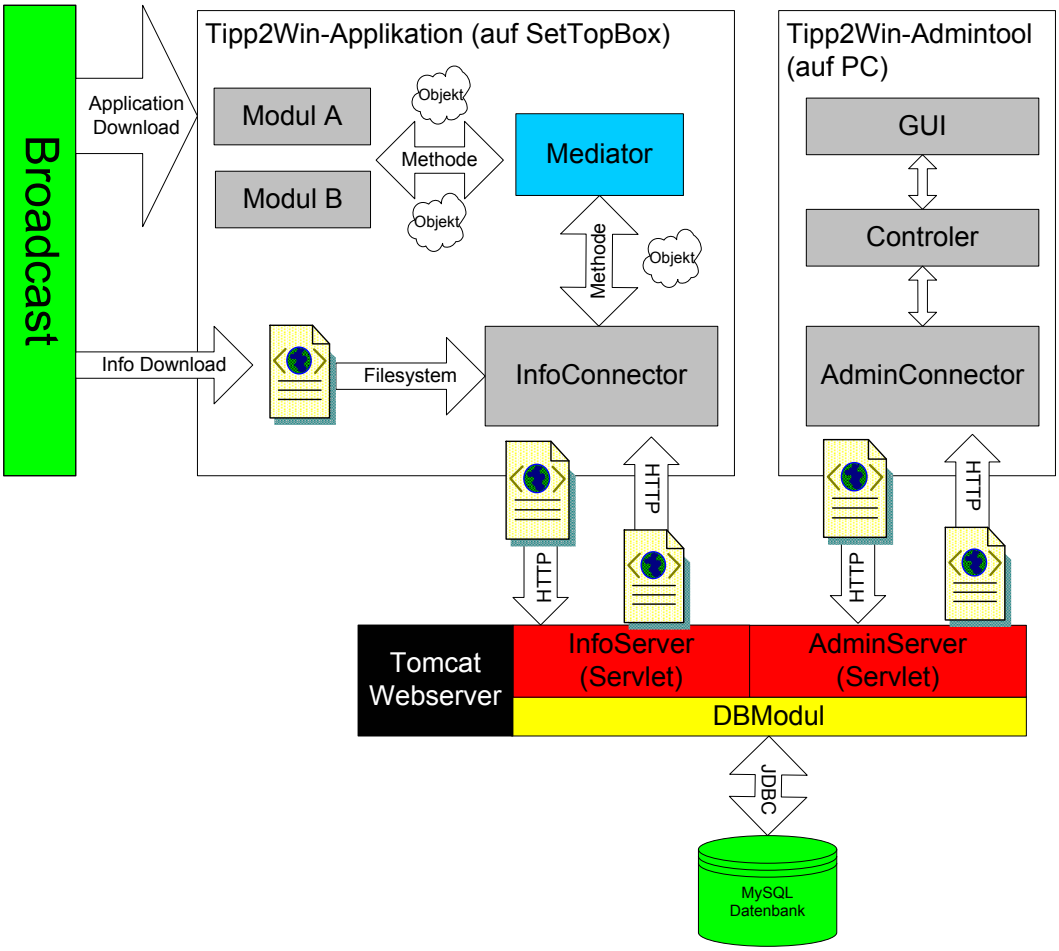

Abbildung 4.14: Client-Server Architektur

Die komplette Kommunikation zwischen Clients und Server erfolgt also in Form von XML-Files. XML-Files bieten diverse Vorteile, z.B. bilden sie eine offene Schnittstelle, die nicht programmiersprachenabhängig ist. Außerdem können sie unter Verwendung von DTDs oder Schemas auf ihre korrekte Struktur hin untersucht werden.

Zusätzlich ist in der Abbildung 4.14 noch der Broadcast zu sehen. Über den Broadcast wird den Tippspielern die Client-Software Bereitgestellt. Da viele Daten, wie z.B. die Begegnungen, Ergebnisse und Punktetabelle, die vom Tipp2Win-Client verwendet werden für alle Tippspieler von Interesse sind, ist es Sinnvoll diese zusätzlich auch über den Broadcast anzubieten. Die Tipp2Win-Applictation überprüft deshalb, ob die gewünschten Informationen auch auf dem Broadcast vorhanden sind, ist dies der Fall, werden sie von dort geladen. Der Vorteil liegt zum einen darin, dass der Tippspieler die Daten nicht über den möglicherweise kostenpflichtigen Rückkanal laden muss, zum anderen wird aber auch einer Überlastung der Verbindung zur Datenbank entgegengewirkt. Gerade nach einem beendeten Fußballspiel ist die Wahrscheinlichkeit groß, dass sehr viele Tippspieler ihren aktuellen Punktestand im Tippspiel überprüfen wollen. Werden zu diesem Zeitpunkt alle Daten, die nicht benutzerspezifisch sind, über den Broadcast übertragen, senkt das die Belastung des Datenbankservers erheblich. Um das Format, in dem die Daten vom Connector entgegengenommen werden einheitlich zu halten, werden die Informationen, die über den Broadcast kommen, ebenfalls in Form von XML-Files zur Verfügung gestellt.

4.3.2.3 Mediatorkonzept des Tipp2Win-Clients

Wie in Abbildung 4.14 zu erkennen ist, soll wie schon bei *MeGa* der Fall, auch der Tipp2Win-Client durch ein Mediatorkonzept realisiert werden. Da dieses Konzept schon im Entwurf zur Mediengalerie im Abschnitt: „Mediatorprinzip und Module“ erläutert wird, soll an dieser Stelle lediglich darauf verwiesen werden.

4.3.3 Detailliertes statisches Modell

4.3.3.1 Übersicht der Klassen

Die folgenden Abbildungen stellen die Grundlage für die spätere Implementierung dar.

Das Konzept des Model View Control wurde in Kapitel 4.3.2 Abschnitt 4.2.3.2 bereits hinreichend beschrieben. Die in der Abbildung 4.15 zu sehenden, farblich unterteilten Klassen, sind folgendermaßen zu interpretieren:

- rot markierte Klassen: beinhaltet den Tipp2WinDirector, der in dem Model View Controller Prinzip als *Controller* fungiert.
- gelb markierte Klassen: umfassen alle Klassen, die für die Visualisierung benötigt werden. Zu jeder Klasse existiert ein Interface, das Konstanten

und Methodenrümpfe zur Verfügung stellt und in der Klasse *ViewFactory* erzeugt wird. Die Implementierung der GUI Klassen sind in dem Paket *havi* gekapselt, um das Erweitern und Austauschen von Klassen einfach zu halten.

- blau markierte Klassen: bilden die *Modelkomponente* des Model View Controller Konzepts.

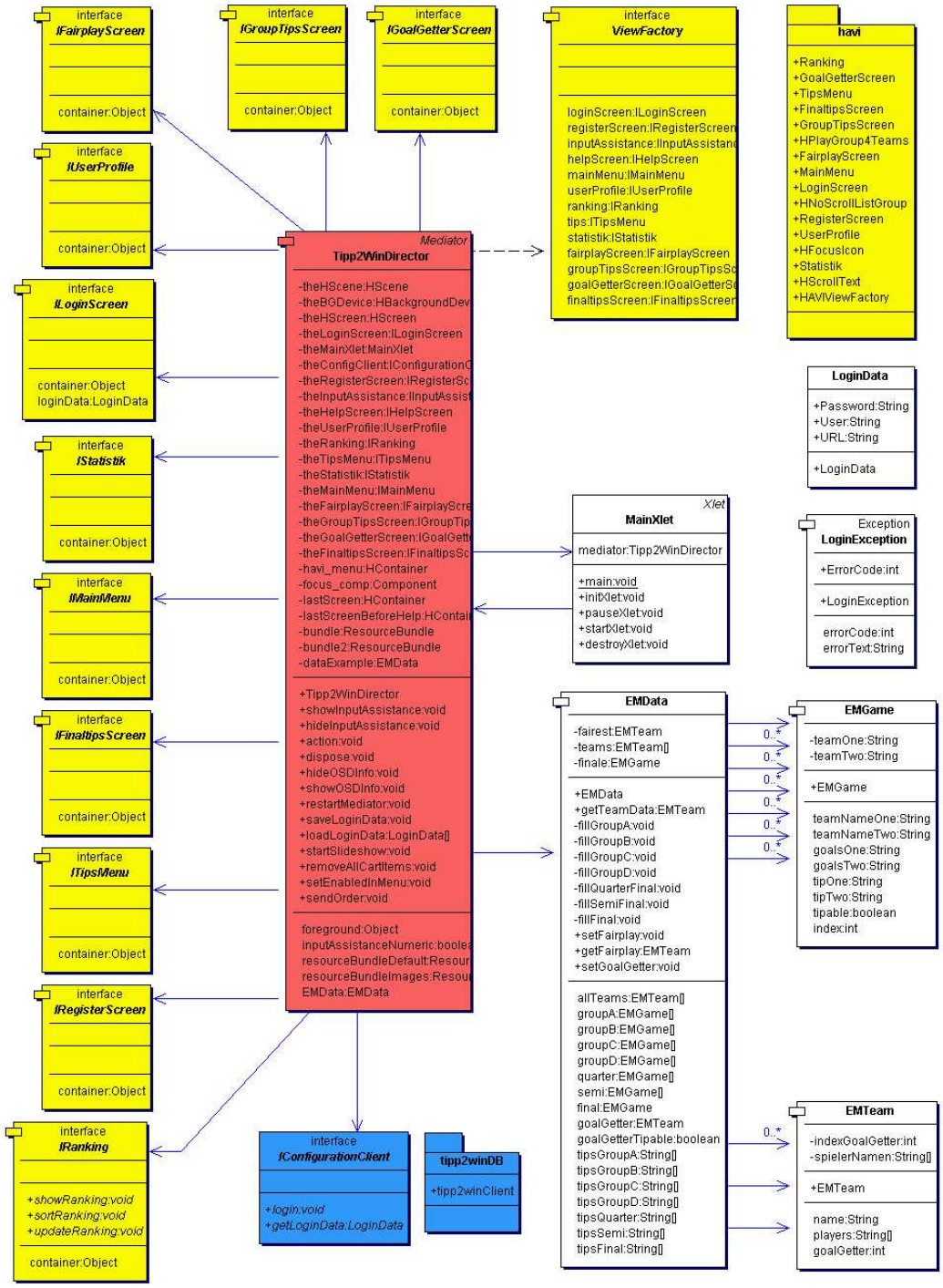

Abbildung 4.15: Model View Control Diagramm

Die beiden Pakete *HAVI* und *Tipp2WinDB* beinhalten die notwendigen

Klassen für die GUI der Anwendung bzw. für die Kommunikation mit dem Datenbank-Server.
Die Klassen *EMData*, *EMTeam* und *EMGame* stellen die Datenzugriffsklassen für Spielinformationen, etc. des Tippspiels dar. Die Klassen *LoginData* und *LoginException* sind die Datentransportklassen bzw. die Klassen zur Fehlerübermittung.

4.3.3.2 Klassen der Dialogschicht

4.3.3.2.1 ViewFactory

- Beschreibung: Das Interface ViewFactory wird in dem Paket HAVI in der Klasse HAVIViewFactory implementiert und dient als AbstractFactory. Mittels dieser Klasse soll es ermöglicht werden, die Views einfach auszutauschen.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): HAVIViewFactory
- Attribute:
  - public static HAVIViewFactory getInstance()
  - private ILoginScreen aLoginScreen
  - private IMainMenu aMainMenu
  - private IRegisterScreen aRegisterScreen
  - private IPreferencesScreen aPreferencesScreen
  - private IHelpScreen aHelpScreen
  - private IInputAssistance aInputAssistance
  - private ITipsMenu aTipsMenu
  - private IFairplayScreen aFairplayScreen
  - private IGoalGetterScreen aGoalGetterScreen
  - private IGroupTipsScreen aGroupTipsScreen
  - private IFinalTipScreen aFinalTipsScreen
  - private IRanking aRanking

  Die folgenden Methoden liefern jeweils die entsprechenden GUIs zurück:

- Methode public IMainMenu getLoginScreen()
- Methode public ILoginScreen getLoginScreen()
- Methode public IHelpScreen getHelpScreen()

- Methode public IInputAssistance getInputAssistance()
- Methode public ITipsMenu getTipsMenu()
- Methode public IFairplayScreen getFairplayScreen()
- Methode public IGoalGetterScreen getGoalGetterScreen()
- Methode public IGroupTipsScreen getGroupTipsScreen()
- Methode public IFinalTipsScreen getFinalTipsScreen()
- Methode public IPreferencesScreen getPrefScreen()
- Methode public IRegisterScreen getRegisterScreen()
- Methode public IRanking getRanking()

4.3.3.2.2 ILoginScreen

- Beschreibung: Die GUI des Logins, in dem der Benutzer sein Login-Namen (E-Mail Adresse oder Nickname) und sein Passwort eingeben muss, um in das Hauptmenü zu gelangen. Die Klasse LoginScreen befindet sich im Paket HAVI.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): LoginScreen
- Attribute:
  - private Tipp2WinDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButton und andere Elemente auf einen Container angeordnet werden. Hier wird der Container für den LoginScreen zurückgegeben.

- Methode public LoginData getLoginData()

  Liefert als Rückgabewert die Login Daten

4.3.3.2.3 IMainMenu

- Beschreibung: Die GUI des Hauptmenüs, in der der Benutzer zu allen wichtigen Punkten der Anwendung gelangen kann. Die Klasse MainMenu befindet sich im Paket HAVI.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): MainMenu
- Attribute:
  - private Tipp2WinDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den MainMenu - Screen zurückgegeben.

4.3.3.2.4 IRegisterScreen

- Beschreibung: Die GUI der Neuanmeldung, wo ein neuer Benutzer seine persönlichen Angaben macht. Der IRegisterScreen wird durch die Klasse RegisterScreen implementiert, welche sich im Paket HAVI befindet.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): RegisterScreen
- Attribute:
  - private Tipp2WinDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den RegisterScreen zurückgegeben.

4.3.3.2.5 IPreferencesScreen

- Beschreibung: Stellt eine Oberfläche zur Verfügung in der der Benutzer Konfigurationseinstellungen (z.B. Sprachauswahl) vornehmen kann. Das Objekt enthält statische Variablen in denen die Präferenzen abgespeichert werden, auf welche andere GUI-Module dann zugreifen können.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): PreferencesScreen
- Attribute:
  - private Tipp2WinDirector mediator
- Methode public void setPreferences(Document pref)

  Hiermit wird das Objekt mit den zuletzt eingestellten Präferenzen initialisiert.

- Methode public Object getContainer();

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den PreferencesScreen zurückgegeben.

4.3.3.2.6 IHelpScreen

- Beschreibung: Die GUI der Hilfe, wo dem Benutzer zu den einzelnen Programmteilen Hilfetexte angezeigt werden. Der IHelpScreen wird durch die Klasse HelpScreen implementiert, welche sich im Paket HAVI befindet.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): HelpScreen
- Attribute:
  - private Tipp2WinDirector mediator

- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den HelpScreen zurückgegeben.

4.3.3.2.7 IInputAssistance

- Beschreibung: Die InputAssistance ermöglicht die Eingabe von Texten via Zahlenfeld der Fernbedienung. Das Interface IInputAssistance wird durch die Klasse InputAssistance implementiert, welche sich im Paket HAVI befindet.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): InputAssistance
- Attribute:
  - private Tipp2WinDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den InputAssistance - Screen zurückgegeben.

4.3.3.2.8 ITipsMenu

- Beschreibung: Die Implementation dieses Interfaces stellt ein Menü dar, welches die Auswahl eines speziellen Tipps ermöglicht. Die Klasse befindet sich im Paket HAVI.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): TipsMenu
- Attribute:

  - private Tipp2WinDirector mediator

- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den TipsMenu - Screen zurückgegeben.

4.3.3.2.9 IFairplayScreen

- Beschreibung: Die Implementation dieses Interfaces stellt einen Screen dar, welcher die Eingabe einer Mannschaft als fairste Mannschaft über ein Drop-Down-Menü ermöglicht. Die Klasse befindet sich im Paket HAVI.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine

- abgeleitete Klasse(n): FairplayScreen

- Attribute:

  - private Tipp2WinDirector mediator

- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den Fairplay - Screen zurückgegeben.

4.3.3.2.10 IGoalGetterScreen

- Beschreibung: Die Implementation dieses Interfaces stellt einen Screen dar, welcher es ermöglicht, über die Auswahl einer Mannschaft und dann eines Spielers dieser Mannschaft den Torschützenkönig zu tippen. Die Klasse befindet sich im Paket HAVI.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine

- abgeleitete Klasse(n): GoalGetterScreen

- Attribute:
  - private Tipp2WinDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den GoalGetter - Screen zurückgegeben.

4.3.3.2.11 IGroupTipsScreen

- Beschreibung: Die Implementation dieses Interfaces stellt einen Screen dar, in welchem Ergebnistipps zu den Gruppenspielen abgegeben werden können. Die Ergebnisse bereits gelaufener Spiele können hier ebenfalls eingesehen werden. Die Klasse befindet sich im Paket HAVI.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): GroupTipsScreen
- Attribute:
  - private Tipp2WinDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den GroupTips - Screen zurückgegeben.

4.3.3.2.12 IFinalTipsScreen

- Beschreibung: Die Implementation dieses Interfaces stellt einen Screen dar, in dem die Ergebnisse von Finalspielen (Viertelfinal-, Halbfinal- und Finalspiel) getippt werden können. Auch hier können die Ergebnisse von bereits gelaufenen Spielen betrachtet werden. Die Klasse befindet sich im Paket HAVI.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): FinalTipsScreen
- Attribute:
  - private Tipp2WinDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den FinalTips - Screen zurückgegeben.

4.3.3.2.13 IRanking

- Beschreibung: Hier werden die aktuellen Ranglisten der Mitspieler angezeigt. Das Interface IRanking wird durch die Klasse Ranking implementiert, welche sich im Paket HAVI befindet.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.
- Oberklasse(n): keine
- abgeleitete Klasse(n): Ranking
- Attribute:
  - private Tipp2WinDirector mediator
- Methode Object getContainer()

  Existiert für alle GUI Klassen, da die Elemente wie die HButtons und andere Elemente in einem Container zurückgegeben werden. Hier wird der Container für den Ranking - Screen zurückgegeben.

4.3.3.3 Klasse der Anwendungslogik (MainXlet&Mediator)

4.3.3.3.1 Tipp2WinDirector

- Beschreibung: Der Tipp2WinDirector funktioniert nach dem Mediator Pattern und beinhaltet die Programmlogik der Applikation *personalisiertes Fußball-Tippspiel Tipp2Win*. Er besitzt Referenzen von allen Klassen bzw. Interfaces und ermöglicht so eine Kommunikation der einzelnen Klassen untereinander.

Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): keine
- Beziehungen: enthält Referenzen zu allen Klassen und Interfaces.
- Attribute:
  - private HScene theHScene
  - private ILoginScreen theLoginScreen
  - private MainXlet theMainXlet
  - private IConfigurationClient theConfigClient
  - private IRegisterScreen theRegisterScreen
  - private IPreferencesScreen thePreferencesScreen
  - private ITipsMenu theTipsMenu
  - private IFairplayScreen theFairplayScreen
  - private IGoalGetterScreen theGoalGetterScreen
  - private IGroupTipsScreen theGroupTipsScreen
  - private IFinalTipScreen theFinalTipsScreen
  - private IRanking theRanking
  - private IMainMenu theMainMenu
  - private IHelpScreen theHelpScreen
  - private IInputAssistance theInputAssistance
- Methode private void setForeground(Object container)

  Setzt den übergebenen Container in den Vordergrund, so dass dieser auf dem Bildschirm sichtbar wird.
- Methode public void showInputAssistance(Object mod, java.awt.AWTEvent event)

  Anzeigen der Eingabehilfe
- Methode public void hideInputAssistance(Object mod)

  Ausblenden der Eingabehilfe

4.3.3.3.2 MainXlet

- Beschreibung:Wird zum Starten der Applikation benötigt und initialisiert den Tipp2WinDirector. Der Applikationsmanager kann über die Klasse MainXlet den Zustand des Programms festlegen. Die Zustände, die das Xlet annehmen kann, sind: *initialisiert*, *gestartet*, *pausiert* und *zerstört*.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen, Attribute und Methodenrümpfe.

- Oberklasse(n): Xlet
- abgeleitete Klasse(n): keine
- Attribute:
  - private XletContext context
  - private Tipp2WinDirector med
- Methode public void initXlet(XletContext context)

  Initialisiert das Xlet. Über den Parameter context kann das Xlet beim Applicationmanager seinen eigenen Status abfragen bzw. beeinflussen(Xlet kann sich z.B. selber pausieren oder zerstören).

- Methode public void startXlet()

  Startet das Xlet bzw. den Tipp2WinDirector.

- Methode public void pauseXlet()

  Pausiert das Xlet.

- Methode public void destroyXlet(boolean unconditional)

  Zerstört das Xlet. Über den Parameter unconditional kann das Xlet seine Zerstörung verhindern, sofern dieser den Wert *false* hat, andernfalls wird das Xlet zerstört.

4.3.3.4 Klassen der Client Kommunikation

4.3.3.4.1 IConfigurationClient

- Beschreibung: Hier werden die aktuellen Ranglisten der Mitspieler angezeigt. Das Interface IConfigurationClient wird durch die Klasse ConfigurationClient implementiert, welche sich im Paket Tipp2Win befindet.

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): ConfigurationClient
- Attribute:
  - private Tipp2WinDirector mediator

4.3.3.4.2 EMData

- Beschreibung: Diese Klasse dient zur Speicherung und Abfrage aller durch das Tippspiel benötigten Daten, welche rein auf die Tipps selber bezogen sind (z.B. Mannschaftsnamen, Spielergebnisse, etc.).

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): keine
- Attribute:
  - private boolean goalGetterTipable
  - private EMTeam fairest
  - private EMTeam goalGetter
  - private EMTeam[] teams
  - private EMGame[] groupA
  - private EMGame[] groupB
  - private EMGame[] groupC
  - private EMGame[] groupD
  - private EMGame[] quarter
  - private EMGame[] semi

  - private EMGame finale

- Methode public EMTeam getTeamData(String name)
- Methode public EMTeam[] getAllTeams()
- Methode public EMGame[] getGroupA()
- Methode public EMGame[] getGroupB()
- Methode public EMGame[] getGroupC()
- Methode public EMGame[] getGroupD()
- Methode public EMGame[] getQuarter()
- Methode public EMGame[] getSemi()
- Methode public EMGame getFinal()
- Methode private void fillGroupA()
- Methode private void fillGroupB()
- Methode private void fillGroupC()
- Methode private void fillGroupD()
- Methode private void fillQuarterFinal()
- Methode private void fillSemiFinal()
- Methode private void fillFinal()
- Methode public void setFairplay(String teamname)
- Methode public EMTeam getFairplay()
- Methode public EMTeam getGoalGetter()
- Methode public void setGoalGetter(String teamname, int playerIndex)
- Methode public boolean isGoalGetterTipable()
- Methode public void setTipsGroupA(String[] tips)
- Methode public void setTipsGroupB(String[] tips)
- Methode public void setTipsGroupC(String[] tips)

- Methode public void setTipsGroupD(String[] tips)
- Methode public void setTipsQuarter(String[] tips)
- Methode public void setTipsSemi(String[] tips)
- Methode public void setTipsFinal(String[] tips)

4.3.3.4.3 EMTeam

- Beschreibung: Diese Klasse dient zur Speicherung für die Informationen zu einem speziellen Team (zum Beispiel Mannschaftsname und Spielernamen).

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): keine
- Attribute:
  - private int indexGoalGetter
  - private String name
  - private String[] spielerNamen
- Methode public String getName()
- Methode public String[] getPlayers()
- Methode public void setGoalGetter(int index)

4.3.3.4.4 EMGame

- Beschreibung: Diese Klasse dient zur Speicherung für die Informationen zu einem speziellen EM-Spiel (beteiligte Mannschaften, Ergebnis, etc.).

  Die nun folgende Aufstellung beschreibt alle Oberklassen, abgeleiteten Klassen und Methodenrümpfe.

- Oberklasse(n): keine
- abgeleitete Klasse(n): keine

- Attribute:
  - private int index
  - private String teamOne
  - private String teamTwo
  - private String goalsOne
  - private String goalsTwo
  - private String tipOne
  - private String tipTwo
  - private boolean tipable
- public EMGame(int index, String teamOne, String teamTwo, String goalsOne, String goalsTwo, String tipOne, String tipTwo, boolean tipable)
- Methode public String getTeamNameOne()
- Methode public String getTeamNameTwo()
- Methode public String getGoalsOne()
- Methode public String getGoalsTwo()
- Methode public String getTipOne()
- Methode public void setTipOne(String value)
- Methode public String getTipTwo()
- Methode public void setTipTwo(String value)
- Methode public void setTipable(boolean value)
- Methode public int getIndex()

4.3.3.5 Aufbau der Datenbank

4.3.3.5.1 Entwurf des Datenbankschemas In diesem Kapitel wird ein Datenbankschema für das Fußball-Tippspiel Tipp2Win vorgestellt. Die Datenbank soll nicht den Anspruch einer Statistik-Datenbank erheben, bei der minutiös alle möglichen Fakten abgelegt werden. Stattdessen beschränken wir die Datenbank darauf, die Daten zu speichern, die das Tippspiel zur Auswertung der Tippmöglichkeiten benötigt. Die Daten sollen aber auf lange

Sicht noch in der Datenbank persistent sein und nicht nur die der aktuellen EM, sondern alle EM´s seit Inbetriebnahme der Datenbank.

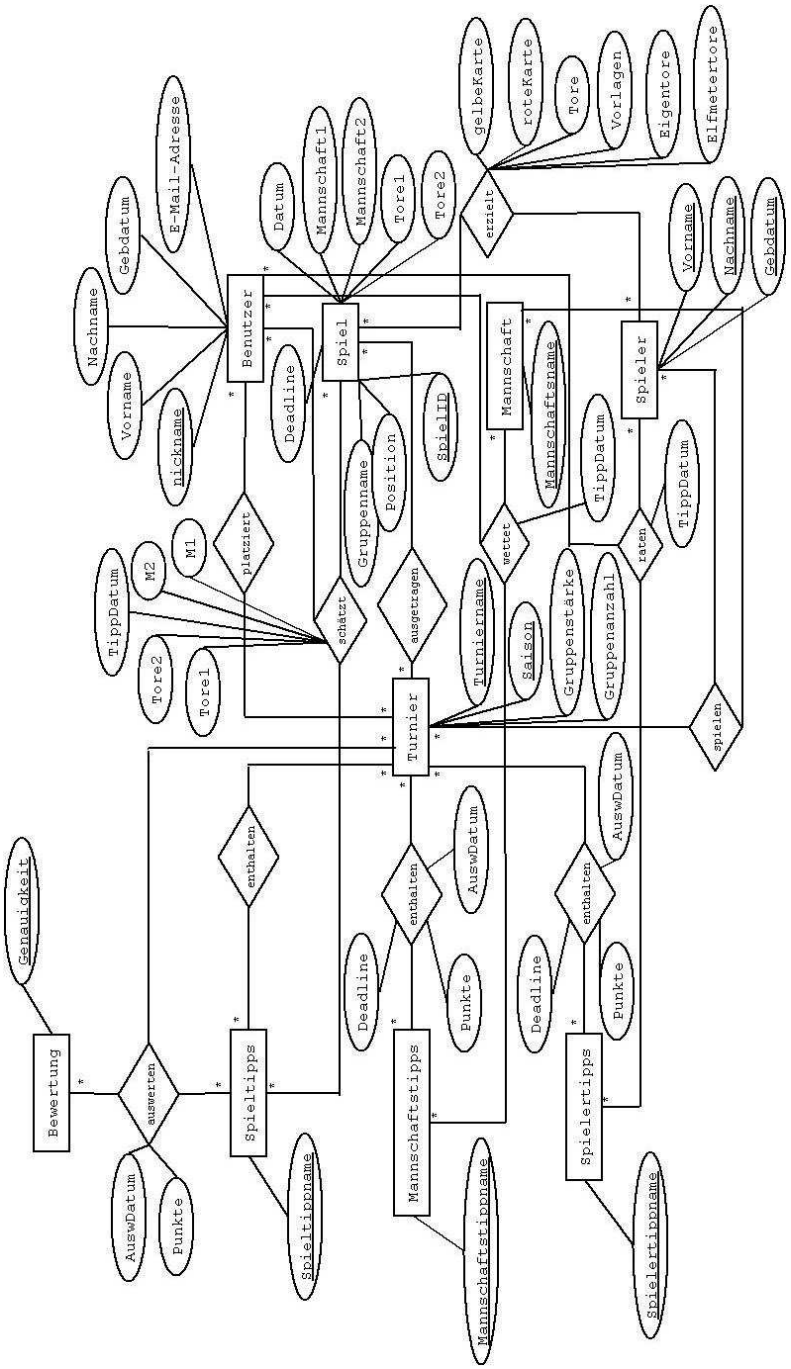

Abbildung 4.16: E-R-Diagramm zum Tipp2Win

E-R-Diagramm
Als ersten Schritt modellieren wir die Datenbank auf konzeptioneller Ebene in Form eines E-R-Diagramms. Die Abbildung 4.16 zeigt das E-R-Diagramm. Darin enthalten sind 9 Entitäten. Die Entität Benutzer besitzt alle wichtigen Informationen zum Benutzer.
In der Mitte befindet sich die Entität Turnier, welche die Attribute Turniername, Saison, Gruppenstärke und Gruppenanzahl besitzt. Die Datenbank soll in der Lage sein nicht nur Daten einer EM zu verwalten, sondern auch eine WM oder einer Bundesliga. Deshalb wurde zur Parametrisierung die Entität Turnier eingeführt, in der alle wichtigen Größen angegeben werden können, z. B. Gruppenanzahl.
In der Abb. 4.16 sind links 3 Entitäten angeordnet, welche sämtliche Möglichkeiten der Tippabgabe enthalten. Es gibt Spiel-, Mannschafts- und Spielertipps. Eine Unterscheidung der Tipps in die drei Entitäten haben wir vorgenommen, da die jeweilige spezielle Entität nur mit dem korrespondierenden der Entitäten Spiel, Mannschaft und Spieler über eine Relation verbunden werden muss.
Rechts gibt es die Entitäten Spiel, Mannschaft und Spieler. Diese drei Entitäten enthalten die fußballbezogenen Daten, die im Verlauf der Turniere anfallen.
Als letzte zu beschreibende Entität bleibt noch Bewertung. Diese gibt Differenzierungen an, in wie weit der Tipp mit dem realen Ergebnis übereinstimmt. Bei Mannschafts- und Spielertipps gibt es bei der Auswertung entweder ein richtig oder falsch. Aber die Spielergebnisse sollen nicht nur richtig oder falsch getippt werden können, sondern auch tendenziell oder richtige Differenz.
Im E-R-Diagramm sind 11 Relationen enthalten. Aus dem letzten Absatz geht hervor, dass Bewertung nur mit der Entität Spieltipps verbunden sein muss. Diese Relation haben wir auswerten genannt, die eine dritte Verbindung zu Turnier besitzt, da die Punktevergabe von Turnier zu Turnier nicht konstant sein muss.
Die Relation enthalten taucht dreimal im Diagramm auf und gibt an welche Tipps bei einem Turnier vorhanden sind. Das Auswertungsdatum legt fest, ab wann das Ergebnis bekannt ist und somit erstmalig ausgewertet werden darf. Andernfalls könnte der Benutzer bei schwankenden Zwischenergebnissen mal mehr oder weniger Punkte haben.
Die vom Benutzer abgegebenen Tipps werden in den Relationen schätzt, wettet und raten gespeichert.

Dann gibt es noch die vier Relationen spielen, ausgetragen, erzielt und platziert. Über die Relation spielen bekommt man die Spieler einer Mannschaft bei einem Turnier heraus. Ausgetragen ordnet die Spiele einem Tur-

nier zu. In der Relation erzielt werden alle relevanten Daten abgelegt, die ein Spieler bei einem Spiel erlangt. Platziert enthält die tippenden Benutzer bei einem Turnier.

Entwurf des logischen Datenbankschemas

Im ersten Schritt werden alle Entitäten in Tabellen umgewandelt. Die Attribute, die den Primärschlüssel der Tabellen bilden, werden in der Spalte Feld fett und kursiv hervorgehoben.

Tabelle 4.1: Struktur der Tabelle Benutzer

Feld	Typ	Null	Standard
Nickname	varchar(10)	Nein	
Vorname	varchar(20)	Ja	NULL
Nachname	varchar(30)	Ja	NULL
GebDatum	date	Ja	NULL
E-Mail-Adresse	varchar(60)	Nein	

Tabelle 4.2: Struktur der Tabelle Turnier

Feld	Typ	Null	Standard
Turniername	varchar(20)	Nein	
Saison	varchar(10)	Nein	0
Gruppenanzahl	int(2)	Ja	NULL
Gruppenstärke	int(2)	Ja	NULL

Tabelle 4.3: Struktur der Tabelle Spielertipps

Feld	Typ	Null	Standard
Spielertippname	varchar(30)	Nein	

Tabelle 4.4: Struktur der Tabelle Mannschaftstipps

Feld	Typ	Null	Standard
Mannschaftstippname	varchar(30)	Nein	

Tabelle 4.5: Struktur der Tabelle Spieltipps

Feld	Typ	Null	Standard
Spieltippname	varchar(30)	Nein	

Tabelle 4.6: Struktur der Tabelle Spiel

Feld	Typ	Null	Standard
Spiel_ID	int(10)	Nein	
Position	int(2)	Ja	NULL
Gruppenname	varchar(30)	Nein	
Datum	datetime	Nein	0000-00-00 00:00:00
Spieldeadline	datetime	Nein	0000-00-00 00:00:00
Mannschaft1	varchar(40)	Nein	
Mannschaft2	varchar(40)	Nein	
Tore1	int(2)	Nein	0
Tore2	int(2)	Nein	0

Zum Wertebereich von Gruppenname gehören beispielsweise A, B, C, D, aber auch Viertelfinale, Halbfinale, Finale und Bundesliga.

Tabelle 4.7: Struktur der Tabelle Mannschaft

Feld	Typ	Null	Standard
Mannschaftsname	varchar(40)	Nein	

Tabelle 4.8: Struktur der Tabelle Spieler

Feld	Typ	Null	Standard
Vorname	varchar(20)	Nein	
Nachname	varchar(30)	Nein	
GebDatum	date	Nein	0000-00-00

Tabelle 4.9: Struktur der Tabelle Bewertung

Feld	Typ	Null	Standard
Genauigkeit	varchar(12)	Nein	

Der Wertebereich des Attributes Genauigkeit ist absteigend bisher Korrekt, Differenz, Tendenz und Falsch.

Im zweiten Schritt werden die Tabellen zu den Relationen erzeugt. Hier kommt es uns zu Gute, dass wir gerade die Primärschlüssel der Entitäten festgelegt haben. Denn die Primärschlüssel der Relationen sind die Kombinierung der Primärschlüssel, der von ihnen verbundenen Entitäten.
Die Relation enthalten taucht dreimal im E-R-Diagramm auf. Aber da die Tabelle nur einmal erzeugt werden kann, werden die Relationen zu enthaltenSpieltipp, enthaltenMtipp und enthaltenSpielertipp umbenannt.

Tabelle 4.10: Struktur der Tabelle auswerten

Feld	Typ	Null	Standard
AuswDatum	datetime	Nein	0000-00-00 00:00:00
Punkte	int(4)	Ja	0
GenauigkeitInz	varchar(12)	Nein	
SpieltippnameInz	varchar(30)	Nein	
TurniernameInz	varchar(20)	Nein	
SaisonInz	varchar(10)	Nein	

Tabelle 4.11: Struktur der Tabelle enthaltenSpieltipp

Feld	Typ	Null	Standard
SpieltippnameInz	varchar(30)	Nein	
TurniernameInz	varchar(20)	Nein	
SaisonInz	varchar(10)	Nein	

Tabelle 4.12: Struktur der Tabelle enthaltenMtipp

Feld	Typ	Null	Standard
Deadline	datetime	Nein	0000-00-00 00:00:00
AuswDatum	datetime	Nein	0000-00-00 00:00:00
Punkte	int(4)	Ja	0
MannschaftstippnameInz	varchar(30)	Nein	
TurniernameInz	varchar(20)	Nein	
SaisonInz	varchar(10)	Nein	

Tabelle 4.13: Struktur der Tabelle enthaltenSpielertipp

Feld	Typ	Null	Standard
Deadline	datetime	Nein	0000-00-00 00:00:00
AuswDatum	datetime	Nein	0000-00-00 00:00:00
Punkte	int(4)	Ja	0
SpielertippnameInz	varchar(30)	Nein	
TurniernameInz	varchar(20)	Nein	
SaisonInz	varchar(10)	Nein	

Tabelle 4.14: Struktur der Tabelle schaetzt

Feld	Typ	Null	Standard
TippDatum	datetime	Nein	0000-00-00 00:00:00
Mannschaft1	varchar(40)	Nein	
Mannschaft2	varchar(40)	Nein	
Tore1	int(2)	Ja	0
Tore2	int(2)	Ja	0
NicknameInz	varchar(10)	Nein	
SpieltippnameInz	varchar(30)	Nein	
SpielIDInz	int(10)	Nein	0

Tabelle 4.15: Struktur der Tabelle wettet

Feld	Typ	Null	Standard
TippDatum	datetime	Nein	0000-00-00 00:00:00
NicknameInz	varchar(10)	Nein	
MannschaftstippnameInz	varchar(30)	Nein	
MannschaftsnameInz	varchar(40)	Nein	

Tabelle 4.16: Struktur der Tabelle raten

Feld	Typ	Null	Standard
TippDatum	datetime	Nein	0000-00-00 00:00:00
NicknameInz	varchar(10)	Nein	
SpielertippnameInz	varchar(30)	Nein	
VornameInz	varchar(20)	Nein	
NachnameInz	varchar(30)	Nein	
GebDatumInz	date	Nein	0000-00-00

Tabelle 4.17: Struktur der Tabelle spielen

Feld	Typ	Null	Standard
VornameInz	varchar(20)	Nein	
NachnameInz	varchar(30)	Nein	
GebDatumInz	date	Nein	0000-00-00
MannschaftsnameInz	varchar(40)	Nein	
TurniernameInz	varchar(20)	Nein	
SaisonInz	varchar(10)	Nein	

Tabelle 4.18: Struktur der Tabelle ausgetragen

Feld	Typ	Null	Standard
SpielIDInz	int(10)	Nein	0
TurniernameInz	varchar(20)	Nein	
SaisonInz	varchar(10)	Nein	

Tabelle 4.19: Struktur der Tabelle erzielt

Feld	Typ	Null	Standard
Tore	int(2)	Nein	0
Vorlagen	int(2)	Nein	0
Eigentore	int(2)	Nein	0
Elfmetertore	int(2)	Nein	0
GelbeKarten	int(2)	Nein	0
RoteKarten	int(2)	Nein	0
VornameInz	varchar(20)	Nein	
NachnameInz	varchar(30)	Nein	
GebDatumInz	date	Nein	0000-00-00
SpielIDInz	int(10)	Nein	0

Tabelle 4.20: Struktur der Tabelle platziert

Feld	Typ	Null	Standard
NicknameInz	varchar(10)	Nein	
TurniernameInz	varchar(20)	Nein	
SaisonInz	varchar(10)	Nein	

Überprüfung der Normalform

Die Entitäten und Relationen erfüllen alle die Boyce-Codd-Normalform. Da diese als die ideale Normalform angesehen wird, bedarf das Datenbankschema keiner Änderungen mehr. Desweiteren garantiert uns die Normalform, dass keine Update-, Insert- und Delete-Anomalien auftreten, sowie die Redundanz minimal ist.

4.3.4 Detailliertes dynamisches Modell

4.3.4.1 Sequenzdiagramme des Tippspieles

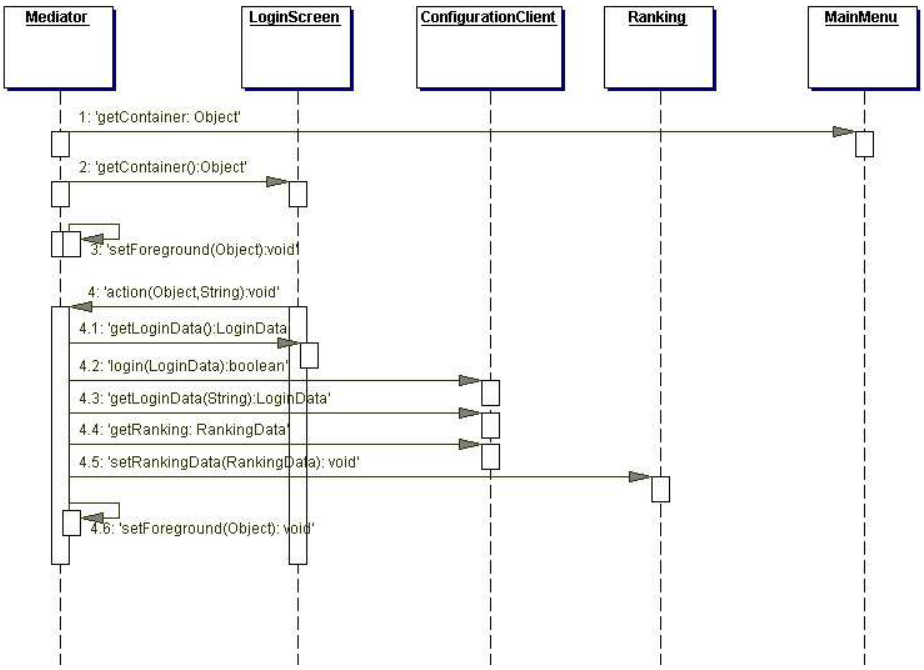

Abbildung 4.17: Sequenzdiagramm Anmeldung

4.3.4.1.1 Anmeldung

Der Mediator ruft bei jeder GUI-Klasse die Methode *getContainer()* auf (hier auf zwei Klassen reduziert) und erhält eine entsprechende Referenz zurück.
Danach wird über die Methode *setForeground()* der Login-Screen angezeigt, indem er in den Vordergrund gesetzt wird. Dieser ruft nach einer Benutzereingabe die *action()*-Methode im Mediator auf, was dazu führt, dass sich dieser die eingegebenen Daten vom Login-Screen holt.
Die erhaltenen Daten werden dann vom Mediator an den *ConfigurationClient* gesandt, der diese dann mit den Daten in der Datenbank abgleicht.
Darauf folgt ein Abruf der Ranking-Daten (Ranglisten der Mitspieler, etc.) durch den *ConfigurationClient*, angestoßen durch den Mediator.
Sobald dann alle benötigten Daten vorhanden sind, wird das *MainMenu* in den Vordergrund gesetzt, womit der Login-Vorgang beendet ist.

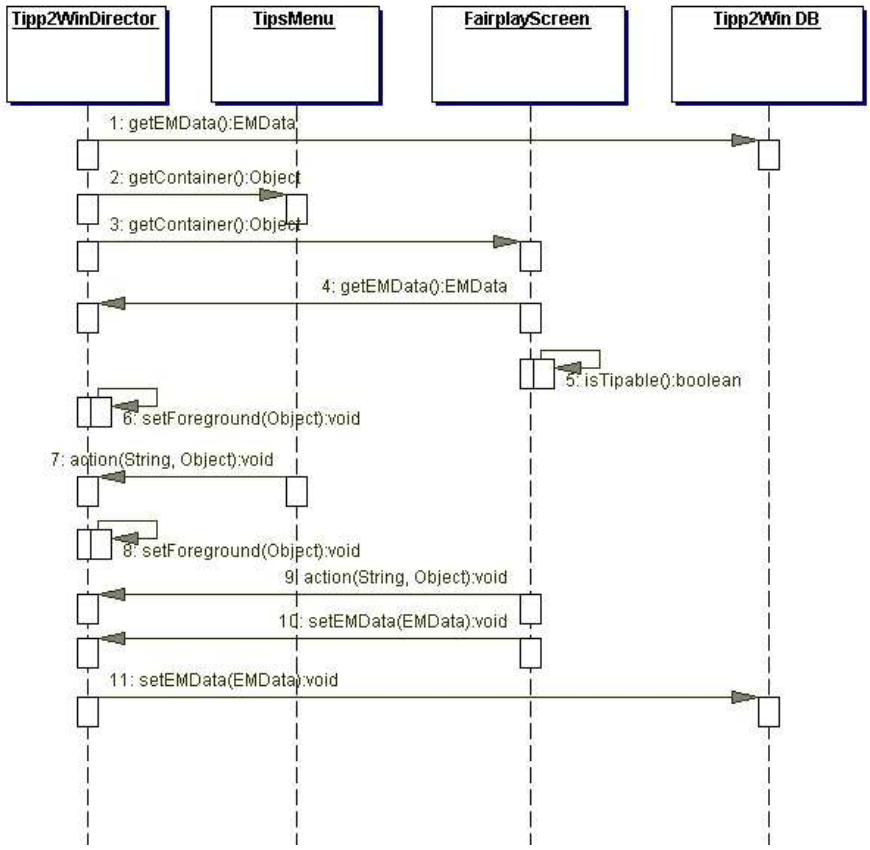

Abbildung 4.18: Sequenzdiagramm Tipp abgeben

4.3.4.1.2 Tipp abgeben

Zunächst wird mit Hilfe der Methode *getEMData()* ein EMData-Objekt aus den Daten der Datenbank generiert. Anschließend werden mittels der Methode *getContainer()* die GUI-Objekte des TipsMenu und des Fairplay-Screens beschafft. Der Fairplay-Screen fragt nun vom Mediator die EM-Daten ab (*getEMData()*) und bestimmt anhand dieser, ob der Tipp überhaupt noch abgegeben werden kann. Danach wird das TipsMenu in den Vordergrund gesetzt. Dort wird nun der Menüpunkt Fairplay ausgewählt, was dazu führt, dass nun der Fairplay-Screen in den Vordergrund gestellt wird.
Nun kann der Benutzer einen Tipp abgeben. Wenn er diesen bestätigt, wird der Tipp in das EMData-Objekt eingetragen und über die *setEMData()*-Methode zurück an den Tipp2WinDirector gesandt. Über eine weitere *setEMData()*-Methode wird dieses Objekt nun an die Datenbank übermittelt.

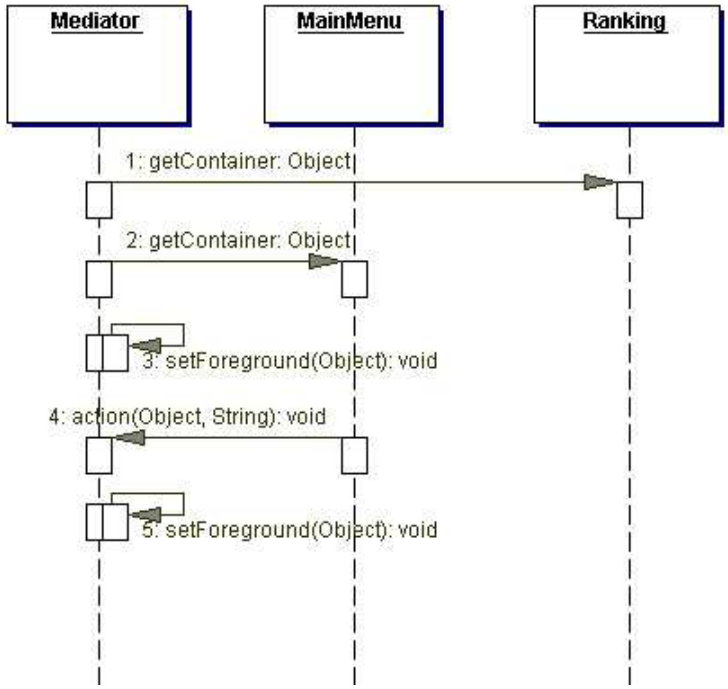

Abbildung 4.19: Sequenzdiagramm Rangliste anzeigen

4.3.4.1.3 Rangliste anzeigen

Der Mediator bekommt hier wieder über die Methode *getContainer()* die Instanzen von *MainMenu* und *Ranking*. Ersteres wird wieder im Mediator in den Vordergrund gesetzt. Wird nun der Menüpunkt „Rangliste “anzeigen ausgewählt, so wird die Instanz der Klasse *Ranking* in den Vordergrund gesetzt. Diese zeigt dann die Daten der Ranglisten an, welche beim Login bereits geladen wurden.

4.4 Gemeinsam genutzte Klassen (common-Paket)

In diesem Kapitel werden Klassen beschrieben, die sowohl für die Mediengalerie MeGa, als auch für das Tippspiel Tipp2Win verwendet werden bzw. Verwendung finden könnten. Weitere Informationen zu den Klassen und deren Methoden befinden sich in den Javadocs.

4.4.1 Unterteilung in Komponenten/Schichten

Die gemeinsamen Klassen lassen sich in drei Gruppen unterteilen: Allgemeine Klassen, selbstentwickelte HAVI-Komponenten und gemeinsam genutzte Klassen der Dialogschicht.

Allgemeine Klassen: Dazu zählen *Cookie*, *Debug*, *EntryGroup*, *MenuEntry*, *MyHListElement* und *HPhotoComponent*. Der Cookie ist zum abspeichern der E-Mail-Adresse und des Passwortes vorgesehen. Die Debug-Klasse dient dem Ausgeben von (Fehler-)Text auf dem Bildschirm. Diese soll Informationen liefern, an welcher Stelle des Programms ein Fehler auftaucht, dazu wird als Parameter ein Schriftzug übergeben. Optional können auch die Priorität der Nachricht und das Modul, welches die (Fehler-)Meldung verursacht angegeben werden. Die HPhotoComponent bildet eine Fotokomponente, d.h. es wird ein Bild in einem bestimmten Format ausgegeben. Die EntryGroup, MenuEntry und MyHListElement bilden Klassen, welche Menüeinträge für das OSD-Menü erzeugt und zu einer Gruppe zusammenfassen.

Selbstentwickelte HAVI-Komponenten: Dazu gehören *DropDownMenu und DropDownMenu2*, *HCursorText*, *HFocusIcon* und *HScrollText*. Die beiden DropDown-Menüs unterscheiden sich nur geringfügig, wobei das DropDownMenu2 aufgrund besserer Handhabung verwendet werden sollte. Sie bestehen aus einem HFocusSinglelineEntry und einer

HListGroup. Es handelt sich hierbei um ein klassisches DropDown-Menü. In ihm können Einträge aufgenommen, entfernt und ausgewählt werden. Der HCursorText wird für die Eingabehilfe verwendet. Der HFocusIcon ist ein fokussiertes Icon, welches von einem HIcon abgeleitet wurde. Der Vorteil liegt darin, dass die neu entwickelte HAVI-Komponente auf dem Fernseher besser zu erkennen ist. HScrollText implementiert das Durchblättern von Text in einem HMultilineEntry. In diesem wird der erklärende Text in der Hilfe angezeigt.

Dialogschicht: Dazu zählen *IHelpScreen*, *IInputAssistance* und *IOSDInfo*.

4.4.2 IHelpScreen

- Beschreibung: Interface für den HelpScreen, welcher die Hilfe realisiert.
- Oberklasse(n): keine
- Abgeleitete Klasse(n): HelpScreen

Methoden:

- public Object getContainer()

  Liefert Container-Objekt zurück

4.4.3 IInputAssistance

- Beschreibung: Interface für die InputAssistance (Eingabehilfe), welcher die Eingabe von Text und Zeichen realisiert.
- Oberklasse(n): keine
- Abgeleitete Klasse(n): InputAssistance

Methoden:

- public Object getContainer()

  Liefert Container-Objekt zurück

4.4.4 IOSDInfo

- Beschreibung: Interface für das OSDInfo, welcher Informationen in einem Fenster anzeigt.
- Oberklasse(n): keine
- abgeleitete Klasse(n): OSDInfo

Methoden:

- public Object getContainer()

  Liefert Container-Objekt zurück
- public void show(String info, int displayType)

  Anzeigen des OSDInfo-Fensters
- public void setTextColor(int r, int g, int b)

  Farbe des Schriftzuges im OSDInfo-Fenster wird mittels RGB-Wert festgelegt

4.5 Technisches Handbuch

In diesem Kapitel soll erklärt werden, welche Schritte bzw. Einstellungen notwendig sind, um die Mediengalerie zum einen auf der Referenzimplementierung, zum anderen auf der Set-Top-Box von Galaxis ausführen zu können.

4.5.1 Kompilieren des Quellcodes

Da die Mediengalerie aus einer Vielzahl an Quellcode-Dateien besteht, ist es hilfreich die Kompilierung durch ein Build-Tool zu erleichtern. Ein solches Werkzeug ist *Ant* von der 'Apache Software Foundation', welches im Rahmen der Projektgruppe zum Einsatz kam. Erhältlich ist *Ant* auf den Webseiten des Herstellers: *http://ant.apache.org/*. Es handelt sich dabei um mehrere Pakete von Klassenbibliotheken, welche unter Verwendung einer speziellen XML-Datei, namens 'build.xml' genutzt werden können. *Ant* ermöglicht es alle Quellcode-Dateien auf einmal zu kompilieren und Distributionen für unterschiedliche Einsatzumgebungen (z.B. Set-Top-Box) zu erstellen. Die verschiedenen Befehle zum Bearbeiten XML-Datei können bei Bedarf in der Online-Dokumentation des Herstellers nachgelesen werden. *Ant* lässt sich sowohl auf der Ebene der Kommandozeile, sowie auch eingebunden in Entwicklungsumgebungen wie zum Beispiel *NetBeans* verwenden.

```
C:\Show_Me_MHP\antroot>dir
 Volume in Laufwerk C: hat keine Bezeichnung.
 Volumeseriennummer: A8C6-1AE1

 Verzeichnis von C:\Show_Me_MHP\antroot

07.10.2004  15:23    <DIR>          .
07.10.2004  15:23    <DIR>          ..
07.10.2004  15:15    <DIR>          build
07.10.2004  15:22             5.074 build.xml
07.10.2004  14:28    <DIR>          lib
07.10.2004  15:01    <DIR>          src
               1 Datei(en)          5.074 Bytes
               5 Verzeichnis(se), 21.386.665.984 Bytes frei

C:\Show_Me_MHP\antroot>
```

Abbildung 4.20: Verzeichnisstruktur des Verzeichnisses antroot

Für beide Methoden werden zunächst die Quelldateien der beiden ShowMeMHP-Applikationen benötigt. Die aktuellen Sourcen können vom CVS unter dem Modulnamen antroot abgerufen werden. Der Stand zum Ende der Projektgruppe wurde zudem auf der Abschluss-DVD im Verzeichnis antroot festgehalten. Die Sourcen liegen in beiden Fällen in einer Verzeichnisstuktur gemäß Abbildung 4.20 vor.

Wie man sieht, beinhaltet das Verzeichnis antroot eine XML-Datei und folgende Unterverzeichnisse:

lib: Dieses Verzeichnis beinhaltet die gesamte Klassenbibliothek der Multimedia Home Platform. *Ant* benötigt diese, um Applikationen, welche auf der Basis der *MHP* entwickelt wurden kompilieren zu können.

src: Das Verzeichnis 'src' beinhaltet sämtliche Quellcode-Dateien der Mediengalerie. Diese sind die Quelldateien für die Kompilierung durch *Ant*.

build: 'build' dient als Zielverzeichnis des Kompiliervorganges von *Ant* und wird, falls noch nicht vorhanden, beim Ausführen von ant angelegt.

4.5.1.1 Kompilieren auf Kommandozeilenebene

Nach der korrekten Installation von *Ant*, kann das Kommandozeilenwerkzeug folgendermaßen aus dem antroot-Verzeichnis aufgerufen werden: :

ant *target*

Der Parameter *target* gibt an, welche Aufgabe ausgeführt wird. Nachstehende Liste gibt, mit einer kurzen Beschreibung, sämtliche für die Mediengalerie und das Tippspiel relevanten Werte für den ***target***-Parameter an:

- **dist-mhpri:** Erzeugt eine Distribution aller sich im 'src'-Verzeichnis befindlichen Quelldateien aller vorhandenen Anwendungen für die Referenzimplementierung und speichert sie im Zielverzeichnis '0.0.x'.
- **tipp2win-dist-mhpri:** Erzeugt eine Distribution des Tippspiels für die Referenzimplementierung und speichert die ausführbaren Dateien im Zielverzeichnis '0.0.x_only_tipp2win'.
- **mega-dist-mhpri:** Erzeugt eine Distribution der Mediengalerie für die Referenzimplementierung und speichert die ausführbaren Dateien im Zielverzeichnis '0.0.x_only_mega'.
- **dist-stb:** Erzeugt eine Distribution aller vorhandenen Anwendungen für die Set-Top-Box und speichert sie im Zielverzeichnis 'all_dist_stb'.
- **mega-dist-stb:** Erzeugt eine Distribution der Mediengalerie für die Set-Top-Box und speichert die ausführbaren Dateien im Zielverzeichnis 'mega_dist_stb'.

- **tipp2win-dist-stb:** Erzeugt eine Distribution des Tippspiels für die Set-Top-Box und speichert die ausführbaren Dateien im Zielverzeichnis 'tipp2win_dist_stb'.
- **javadoc:** Generiert ein angepasstes Java-Doc aus allen Quelldateien.
- **clear:** Mit diesem Kommandozeilenparameter werden alle Verzeichnisse außer den in Abbildung 4.20 zu sehenden, gelöscht und das 'build'-Verzeichnis wird geleert.

Die Parameter sind in der XML-Datei verankert und können editiert werden. Beispielsweise kann der Name des 'build'-Verzeichnisses geändert werden oder es können neue Aufgaben definiert werden.

4.5.1.2 Kompilieren mit NetBeans

Um die Quelldateien der Mediengalerie (oder anderen Anwendungen) mit Hilfe einer Entwicklungsumgebung wie NetBeans *Ant* erfolgreich kompilieren zu können, sind folgende Schritte notwendig.

Mounten: Als erstes muss der antroot-Verzeichnis, also z.B. C:\Show_Me_MHP\antroot, im Explorer-Fenster des Editing Modus von NetBeans gemounted werden. Dazu müssen nach dem Start von NetBeans nachstehende Schritte ausgeführt werden:

- Auswahl des Reiters *Editing*, welcher unter der Toolbar im Hauptfenster von NetBeans zu finden ist
- Anklicken des Icons *Filesystem* im Explorer-Fenster des Editing-Modus
- Nach einem weiteren Rechtsklick auf *Filesystem* erscheint ein Menü.

Abbildung 4.21: Auswahl des zu mountenen Verzeichnisses

Wie auf Abbildung 4.21 zu sehen ist, enthält das Untermenü *Mount* einen Eintrag *Local Filesystem*, der auszuwählen ist. Nun lässt sich der gewünschte Pfad in der Verzeichnisstruktur des jeweiligen Betriebssystems auswählen.

Kompilieren: Nun kann das eigentliche Kompilieren beginnen. Dazu ist ein Rechtsklick auf die, nun im gemounteten Verzeichnis des Explorer Fensters von NetBeans zu sehende, 'build'-XML Datei nötig. Jetzt lassen sich im Menü *Run Target* sämtliche in der 'build'-XML Datei festgelegten *Parameter* ausführen.

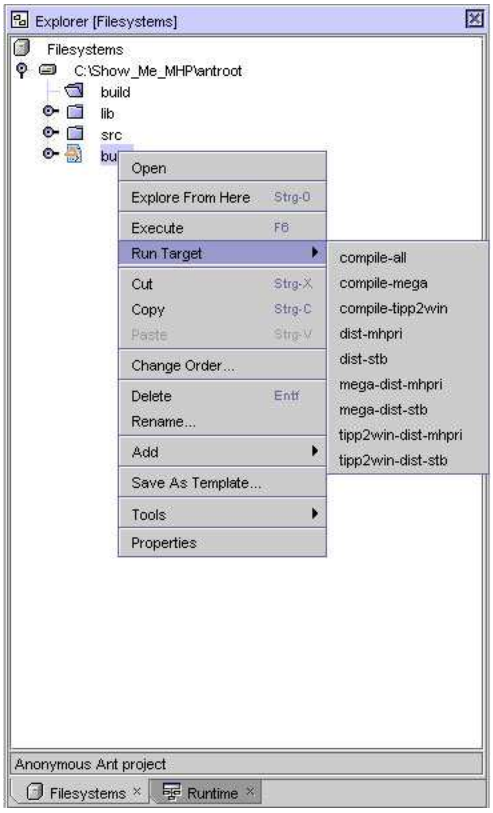

Abbildung 4.22: Auswahl der möglichen Parameter für *Ant*

Nach der Auswahl eines *Parameters* werden die Zieldateien, also die ausführbaren Dateien für beispielsweise eine Distribution der Mediengalerie für die Set-Top-Box (entspräche dem *Paramter* mega-dist-stb) in dem zugehörigen Zielverzeichnis gespeichert.

Das gemountete Verzeichnis bleibt nach einem Neustart von NetBeans erhalten, so dass kein erneutes *mounten* erforderlich ist.

4.5.2 Nutzung der MeGa auf der IRT Referenzimplementierung

Für den Fall, dass keine Set-Top-Box zur Verfügung steht, besteht auch die Möglichkeit, die Mediengalerie mit Hilfe der Referenzimplementierung des IRT auf einem PC zu Testen.

4.5.2.1 Anforderungen

Um die Mediengalerie auf dem PC zu Testen, müssen folgende Anforderungen erfüllt sein:

- Ein PC
- Die Referenzimplementierung des IRT
- Eine kompilierte Version der Mediengalerie MeGa
- Eine bestehende Internetverbindung über einen DSL-Router oder einen anderen PC, alternativ kann auch ein DSL-Modem verwendet werden.

4.5.2.2 Installation der Referenzimplementierung

Um die Referenzimplementierung auf einem Windows-PC zu installieren, muss das beiliegende Setup-Programm ausgeführt werden. Dieses Programm erklärt dann alle weiteren Schritte, die für die Installation notwendig sind. „C:/Programme/IrtMhpRI/“ ist der Standardinstallationspfad. In dem Ordner: „/IrtMhpRI/windows/mhpri/runtime/obfuscated/“ befinden sich die Jar-Dateien: mhpri.jar, boot.jar, platform.jar, die die Klassen enthalten, die zum Ausführen des EPGs der Referenzimplementierung benötigt werden. Falls das Setup-Programm diese Dateien nicht automatisch in den Klassenpfad („CLASSPATH“) eingefügt hat, muss dies manuell nachgeholt werden. Dazu muss im Startmenü von Windows unter *Einstellungen* die *Systemsteuerung* geöffnet werden. In der *Systemsteuerung* muss der Eintrag *System* ausgewählt werden, der die Systemeigenschaften öffnet. Der Reiter *Erweitert* enthält einen Button *Umgebungsvariablen* dessen Betätigung das gleichnamige Fenster öffnet. In diesem Fenster *Umgebungsvariablen* kann nun der Eintrag *CLASSPATH* entweder angepasst oder neu angelegt werden. In den *CLASSPATH* müssen dann die kompletten Pfade zu den Jar-Dateien angegeben werden (in diesem Fall z.B. „C:/Programme/IrtMhpRI/windows/-mhpri/runtime/obfuscated/mhpri.jar“), mehrere Einträge werden dabei durch „;“ getrennt.

4.5.2.3 Vorbereiten der Programmdaten

Um die Mediengalerie MeGa über die Referenzimplementierung zu starten, muss sie zunächst in kompilierter Form (nur .class-Dateien, Bilder, Property-Dateien und die XML-Templates .xml-Dateien) vorliegen. Diese kompilierte Version muss wie in der in Abbildung 4.27 dargestellten Verzeichnisstruktur abgelegt sein. Wichtig hierbei ist, dass sich die Programmdaten in einem Verzeichnis mit dem Namen „0.0.x“ befinden, wobei 'x' eine beliebige natürliche Zahl ist. Um die Anwendung in den EPG der Referenzimplementierung aufzunehmen, muss dieses Verzeichnis in die Konfigurationsdatei „Applications.cfg“ eingetragen werden. Ein Beispiel für einen solchen Eintrag ist in Abbildung 4.23 dargestellt. Die Datei befindet sich im Verzeichnis *runtime/cfg/*.

```
...
###### SHOW_ME_MHP
FirstXletExample;702 = Test: FirstXletExample;dvb://0.0.2
HaviExample;706 = Test: HaviExample;dvb://0.0.2
de.showmemhp.mgallery.MainXlet;703 = MeGa;dvb://0.0.4
de.showmemhp.tipp2win.MainXlet;704 = Tipp2Win;dvb://0.0.4
...
```

Abbildung 4.23: Ausschnitt aus der Datei „Applications.cfg“

Der Aufbau der einzelnen Zeilen erfolgt nach folgendem Schema: Zunächst wird der Name des Xlets angegeben (hier: de.showmemhp.mgallery.MainXlet). Es folgt eine eindeutige Nummer (zum Beispiel 703), ein Gleichheitszeichen und der Name der Anwendung (Mega). Dieser Name wird dann auch im Menü des EPG verwendet. Abgeschlossen wird eine Zeile mit der Angabe des Verzeichnisses, in dem sich die Anwendungsdaten befinden (hier: 0.0.4).

4.5.2.4 Programmstart der Mediengalerie

Um die Anwendung nach der Einrichtung zu starten, muss die Datei *runtime/mhp.bat* gestartet werden.
Im darauf erscheinenden Menü, dem EPG der Referenzimplementierung, wird in der Spalte 'Applications' ein Eintrag für die Mediengalerie, mit dem in der Konfigurationsdatei angegebenen Namen, angezeigt. Durch den Aufruf dieses Menüelementes wird die Mediengalerie gestartet.

4.5.3 Inbetriebnahme der Fernbedienung für die Referenzimplementierung

Die Mediengalerie kann auch innerhalb der Referenzimplementierung mit einer Fernbedienung bedient und getestet werden. Um die Fernbedienung nutzen zu können, sind folgende Installationsschritte nötig:

1. Zunächst muss das Verzeichnis *mhpcontrol* mit allen Unterverzeichnissen von der DVD auf die Festplatte kopiert werden, z.B. `C:\mhpcontrol`.

2. Nun muss das Programm *WinLirc*, das sich auf der DVD befindet, installiert werden.

3. Anschließend muss der Infrarot-Empfänger an eine freie serielle Schnittstelle des PC angeschlossen werden und so positioniert werden, dass er die Signale der Fernbedienung empfangen kann.

Abbildung 4.24: Konfiguration des Programm *WinLirc*

4. Nun muss das Programm *WinLirc* gestartet werden und unter dem Button *Reconfigure* sowohl der richtige COM-Port (also die serielle Schnittstelle) des Empfängers als auch unter *Config* der Pfad zu der passenden Konfigurationsdatei der Fernbedienung angegeben werden[3] (siehe Abbildung 4.24). Die im Verzeichnis mhpcontrol befindliche Konfigurationsdatei `SBC_RP420.cfg` ist eine angepasste Datei für die Fernbedienung einer Phillips DSR 5600 Set-Top-Box.

[3]Eine umfangreiche Sammlung an Konfigurationsdateien findet sich unter http://lirc.sourceforge.net/remotes/

5. Anschließend kann das Fenster mit *OK* geschlossen werden und im Hauptfenster des Programms die richtige Fernbedienung im Drop-Down-Menü ausgewählt werden (siehe Abbildung 4.25).

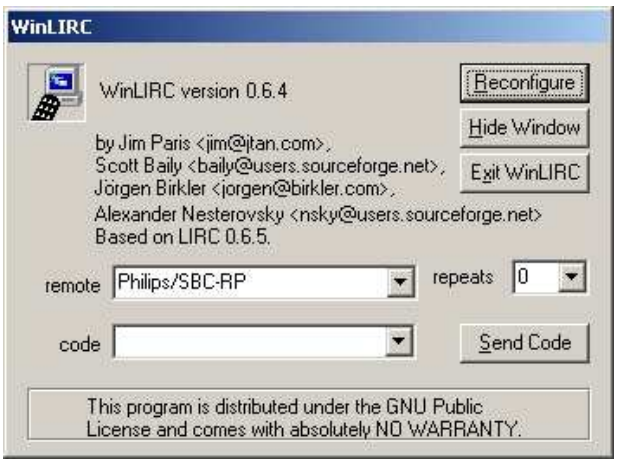

Abbildung 4.25: Auswählen der Fernbedienung

6. Jetzt muss die Datei `run.bat` im Verzeichnis `mhpcontrol` editiert werden. Zum einen muss, wenn die Variable `java_home` noch nicht gesetzt wurde (erfolgt normalerweise beim Installieren eines Java JRE) der Pfad auf die Java-Umgebung gesetzt werden. Zum anderen muss die Variable `working_dir` auf den Pfad gesetzt werden, in dem sich die Datei `run.bat` befindet (siehe Abbildung 4.26).

Abbildung 4.26: Variablen in der Datei `run.bat`

7. Abschließend muss die Datei `run.bat` ausgeführt werden.

Nach diesen Schritten kann die IRT-Referenzimplementierung mit der Fernbedienung am PC bedient werden.

4.5.4 Nutzung der MeGa auf der Galaxis Set-Top-Box

Dieses Kapitel beschreibt die notwendigen Maßnahmen, um die Mediengalerie auf einer Set-Top-Box (wie z.B. die „LinuxTV “ von Galaxis) ausführen zu können.

4.5.4.1 Anforderungen

Um die Mediengalerie nutzen zu können, müssen folgende Dinge zur Verfügung stehen:

- Eine Galaxis Set-Top-Box mit Fernbedienung
- Ein USB-Speichermedium
- Ein PC mit USB-Anschluss
- Eine kompilierte Version der Mediengalerie MeGa
- Eine bestehende Internetverbindung über einen DSL-Router oder einen anderen PC, alternativ kann auch ein DSL-Modem verwendet werden.

4.5.4.2 Vorbereiten der Programmdaten

Für das Testen der MeGa auf der Galaxis Set-Top-Box muss als erstes eine Kopie der Mediengalerie mit allen benötigten Dateien auf den USB-Stick kopiert werden. Dazu werden die Dateien der Mediengalerie in ein Unterverzeichnis eines USB-Sticks entpackt bzw. kopiert, z.B. `MeGa`. Es ist darauf zu achten, die MeGa-Dateien nicht direkt ins Hauptverzeichnis des USB-Sticks zu kopieren, und die Ordnerstruktur beizubehalten. Unter Windows kann die Konsistenz überprüft werden, indem man sich die auf dem USB-Stick vorhandene Dateistruktur mit dem Datei-Explorer anzeigen lässt (siehe Abbildung 4.27). Als Laufwerksbezeichnung ist hier `F:` gewählt. Diese Laufwerksbezeichnung des USB-Mediums kann abweichen.

Weiterhin kann überprüft werden, ob unter anderem folgende Dateien auf dem Stick vorhanden sind:

```
F:\MeGa\mhp-resident-app.xml F:\MeGa\mgallery.properties
```

Wichtig ist die Datei `mhp-resident-app.xml` (siehe Abbildung 4.28), ohne die die Galaxis-Box das Programm nicht erkennt. Diese Datei muss sich, wie oben angegeben, im Stammverzeichnis der Programmdaten befinden. Entscheidend sind hier die Elemente: default_name und initial_class. Im Element „default_name“ muss der Name des Programms angegeben werden

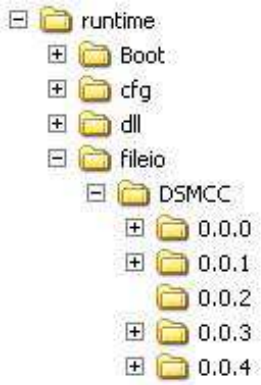

Abbildung 4.27: Ordnerstrukturen

(„MeGa“). In „initial_class“ muss die Hauptklasse mit richtiger Paketstruktur angegeben werden „de.showmemhp.mgallery.MainXlet“). Die Hauptklasse des Programms ist die Klasse, die die Schnittstelle (Interface) Xlet implementiert. Die anderen Einträge können wie in der Abbildung beibehalten werden.

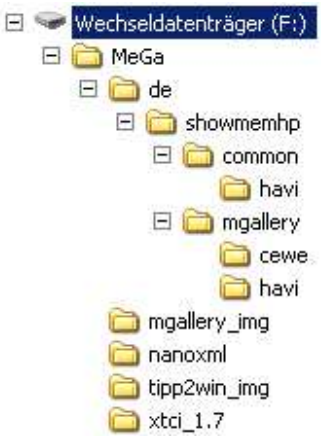

Abbildung 4.28: mhp-resident-app.xml

Wenn alle Dateien auf den Stick kopiert wurden, kann der USB-Stick vom PC entfernt werden.

4.5.4.3 Benötigte Voreinstellungen auf der Settopbox

Um die Mediengalerie auf der Box nutzen zu können, muss zuvor sichergestellt werden, dass eine funktionierende Internetverbindung vorhanden ist.

Diese kann zum einen über einen ISDN/DSL-Router erfolgen, zum anderen auch über jeden PC mit Internetzugang, der über eine freie Netzwerkkarte verfügt und der Box eine Internetverbindung bereitstellen kann.

- *Verbindung über einen Router:* Zuerst muss die Box mit der auf der Rückseite befindlichen RJ45-Dose an einen Router angeschlossen werden, dies kann entweder ein ISDN bzw. DSL-Router oder ein PC mit freier Netzwerkkarte und Internetverbindung sein. Darüber hinaus muss in den Optionen der Galaxis-Settopbox die IP-Adresse des Routers als Gateway und DNS-Nameserver eingetragen werden. Diese Einstellungen können vorgenommen werden, indem mit der Navigatortaste in das Hauptmenü gewechselt wird und dort der Punkt 7 - *Einstellungen* angewählt wird. Im nächsten Menü ist Punkt 7.8 - *Systemeinstellungen* zu wählen, nun muss man weiter wechseln zum Punkt 7.8.4 -*Netzwerkkonfiguration*.

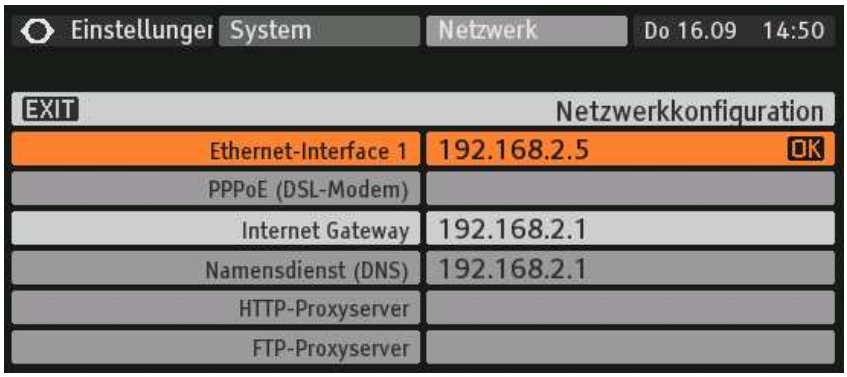

Abbildung 4.29: Einstellen der IP-Adressen in der Galaxis-Box

Im Feld *Ethernet Interface 1* muss eine in Ihrem Netz freie lokale IP-Adresse eingetragen werden. In den Feldern *Internet Gateway* und *Namesdienst (DNS)* wird die IP-Adresse Ihres Routers eingetragen. Im Beispiel oben erhält die Box die Adresse 192.168.2.5, der Router hat die Adresse 192.168.2.1.Abschließend sollten die vorgenommen Änderungen gesichert werden und das Menü geschlossen werden. Wenn jetzt die Internetverbindung des Routers aktiviert wird, sollte die Settopbox eine Verbindung herstellen können.

- *Verbindung über ein DSL-Modem:* Falls die Box direkt an ein DSL-Modem angeschlossen werden soll, muss die Settopbox direkt mit dem Modem verbunden werden. Nun muss mit der Navigatortaste das Hauptmenü geöffnet werden Punkt 7 - *Einstellungen* gewählt werden. Im

nächsten Menü wählt man dann den Punkt 7.8 - *Systemeinstellungen*, und wechselt weiter zum Punkt 7.8.4 -*Netzwerkkonfiguration*. Jetzt muss der Punkt *PPPoE (DSL-Modem)* ausgewählt werden.

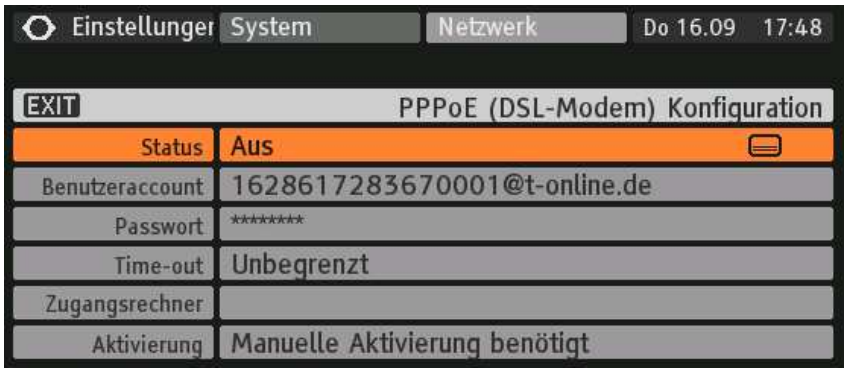

Abbildung 4.30: DSL-Verbindung konfigurieren

In dem Menü *PPPoE (DSL-Modem) Konfiguration* werden der Benutzername und das Passwort für den Internetzugang eingetragen, und die Eingaben mit der blauen Taste bestätigt. Jetzt erscheint der Schriftzug *jedes Modem* im Feld *PPPoE (DSL-Modem)* sowie ein gelber Button *anstellen*. Wenn jetzt die gelbe Taste auf der Fernbedienung betätigt wird, wird die Settopbox versuchen, selbst eine Internetverbindung über die DSL-Leitung herzustellen. An gleicher Stelle kann später die Verbindung auch wieder getrennt werden, sobald die Mediengalerie MeGa nicht weiter benutzt werden sollte.

4.5.4.4 Programmstart der Mediengalerie

Zunächst sollte sichergestellt werden dass die Box ausgeschaltet ist und sich nicht im Standby Modus befindet. Nun wird der USB-Stick an den USB-Port auf der Rückseite der Set-Top-Box angeschlossen. Das Gerät kann jetzt eingeschaltet werden. Nun Muss gewartet werden, bis die Box vollständig hochgefahren ist. Weiterhin sollte sichergestellt werden, dass die Galaxis-Box über eine aktive Internetverbindung verfügt (siehe auch Punkt 4.5.4.3). Nun kann mit der Navigatortaste das Hauptmenü der Set-Top-Box geöffnet werden und dort der Menüpunkt 4 - *MHP* aufgerufen werden. Zum Programmstart der Mediengalerie muss der Unterpunkt *[EXT.RESIDENT] Mediengalerie MeGa* ausgewählt werden.

4.5.5 Unterschiede zwischen der Referenzimplementierung und der Set-Top-Box von Galaxis

Die beiden Anwendungen MeGa und Tipp2Win wurden zunächst nur mit Hilfe der Referenzimplementierung des IRT entwickelt und getestet. Erst in der späten Entwicklungsphase hatten wir die Möglichkeit, die Anwendungen auf einer Set-Top-Box zu testen. Es handelte sich hierbei um das Gerät „LinuxTV“ des Herstellers Galaxis. Bei dem ersten erfolgreichen Start der Mediengalerie stellten wir fest, dass unsere Anwendung von der Set-Top-Box anders als von der Referenzimplementierung dargestellt wurde. Hier sind hauptsächlich abweichende Darstellungen einiger Komponenten zu nennen:

- HSinglelineEntries haben auf der Referenzimplementierung einen schattierten Rand, auf der Set-Top-Box ist kein Rand vorhanden. Um die Komponenten dennoch zu Umranden, wurde eine neue Klasse HFocusSinglelineEntry implementiert, in der die paint()-Methode überschrieben wurde.

- Der Fokus von HAVI-Komponenten hatte auf der Set-Top-Box ein anderes Aussehen. Auch der Fokus wurde mit Hilfe der Klasse HFocusSinglelineEntry angepasst.

- Selbst erstellte Graphik-Elemente im PNG-Format (wie beispielsweise Buttons), welche transparente Bereiche haben sollten, wurden nicht richtig angezeigt: sie waren nicht transparent zum Hintergrund, sondern das aktuelle Fernsehbild schien in diesen Bereichen durch. Dieses Problem konnte nur dadurch gelöst werden, alle mit Photoshop erstellten Komponenten zunächst als GIF-Dateien abzuspeichern und erst anschließend in PNG-Dateien zu konvertieren.

- Die Schriftgrößen scheinen nicht identisch zu sein: auf der Set-Top-Box werden einige Texte (z.B. HStaticText) nicht vollständig angezeigt. Die Texte werden durch die andere Größe vermutlich zu lang, so dass sie dann teilweise abgeschnitten werden (z.B.: aus der Zeichenkette „Passwort:“ wird dann „Passw...“).

- Beim Testen der InputAssistance auf der Referenzimplementierung trat folgendes Problem in Erscheinung: mit jeder Betätigung einer beliebigen Farbtaste wurde ein Fragezeichen in die HSinglelineEntry eingetragen, welches allerdings nicht sichtbar war. Dieses „sonderbare“ Problem wurde nun so gelöst, dass mit jeder Betätigung einer Farbtaste wieder ein Zeichen (hier das Fragezeichen) gelöscht wird. Beim Testen mit

der Set-Top-Box stellten wir dann fest, dass kein Fragezeichen in die HSinglelineEntry eingetragen wird. Die vorherige Problemlösung wurde nun zum Problem, da mit jedem Drücken einer Farbtaste ein Zeichen gelöscht wurde, welches natürlich nicht entfernt werden sollte. Dieses „neue“ Problem wurde nun wie folgt gelöst: eine Variable *referenceImplementation* wurde in die Eingabehilfe integriert. Hat diese boolesche Variable den Wert true, so wird das letzte Zeichen gelöscht, steht sie auf false, wird beim Drücken einer Farbtaste kein Zeichen gelöscht.

4.5.6 Vorhandene Fakes in der Mediengalerie MeGa

Da ein Paar Funktionen des CeWe-Servers zurzeit noch nicht implementiert sind und um das Testen zu erleichtern, wurden in die Mediengalerie MeGa einige Fakes eingebaut. Diese werden nun im Folgenden aufgezählt:

- Um beim Testen die Anmeldung bei der Mediengalerie zu beschleunigen, werden in der Methode *loginKAChoice* der Klasse *CeWeClient* die Partner, bei denen der Kunde mit dem angegebenen Loginnamen und Passwort angemeldet ist, auf die beiden Test-KeyAccounts 499 und 2166 gesetzt. Dies geschieht mit der auf *false* gesetzten Variablen *mitKaLogin*, die später im laufenden Betrieb auf *true* gesetzt werden muss.

- Für die Foto- und Funproduktebestellung existieren jeweils eigene KeyAccounts. Bisher liefert der CeWe-Server bei dem *kaindependantlogin*-Request nur den KeyAccount für die Fotobestellung zurück. Daher wird in der Methode *handleKALogin* der Klasse *CeWeClient* für die Funproduktebestellung derselbe KeyAccount wie für die Fotobestellung eingetragen.

- Die Neuanmeldung eines Kunden erfolgt bisher nur bei dem Test-KeyAccount 2166. Dies ist in der Methode *addUser* der Klasse *CeWeClient* hart kodiert. Im späteren laufenden Betrieb muss bei der Neuanmeldung eine Liste mit Partnern zur Verfügung stehen, in der ein gewünschter Partner ausgewählt wird und dann der entsprechende KeyAccount bei der Neuanmeldung verwendet wird.

- Im Warenkorb wird bei der Qualitätsberechnung der Bilder deren Originalgröße benötigt. Bei dem *albumitemlist*-Request an den CeWe-Server ist bereits in dessen DTD vorgesehen, dass der Request u.a. die Originalgröße der Bilder zurückliefert. Zurzeit liefert der CeWe-Server allerdings noch keine Größenangaben zurück, so dass diese in der Methode

checkQuality der Klasse *HPhotoOrderItem* fest auf „1500x1000“ eingestellt sind.

4.5.7 ResourceBundles

Die Verwendung der ResourceBundles ist im Kapitel 4.2.3.4 beschrieben.

4.6 Handbuch Mediengalerie MeGa

4.6.1 Allgemeines

4.6.1.1 Einleitung

Die Mediengalerie MeGa, bietet dem Anwender die Möglichkeit Bilder mit der Familie oder Freunden auf dem Fernseher anzusehen. Zudem ist es möglich nicht nur die Bilder anzusehen, sondern auch über den Fernseher zu bestellen. Durch viele Funktionalitäten wie der Slideshow kann der Fernsehabend zu einer wunderschönen Diashow mit der Familie oder Freunden am Fernseher werden. Hierzu sind eine digitale MHP-fähige Settopbox mit Rückkanal sowie ein Fernseher erforderlich. Dem Benutzer wird eine leicht bedienbare Benutzeroberfläche bereitgestellt. Dieses Handbuch soll den Einstieg in die Bedienung des Programms erleichtern bzw. als Nachschlagwerk dienen.

4.6.1.2 Aufbau

Dieses Handbuch ist wie folgt aufgebaut:

- allgemeine Informationen zur Bedienung mit der Fernbedienung
- Beschreibung der einzelnen Seiten des Programms MeGa. Zur Illustration werden einige Screenshots der Beschreibung hinzugefügt.

4.6.1.3 Bedienung

Die Bedienung des Programms erfolgt mit der Fernbedienung der Set-Top-Box. Zur Navigation innerhalb einer Seite werden die Pfeiltasten benutzt, um zu den einzelnen Eingabefelder zu gelangen. Durch drücken der OK-Taste können die Eingabefelder angewählt bzw. Aktionen bestätigt werden. Viele Funktionen lassen sich über die Farbtasten (rot, grün, gelb und blau) der Fernbedienung aufrufen. In der unteren Leiste ist die Belegung der Farbtasten für die jeweilige Seite dargestellt. Welche Funktion diese Farbtaste hat, steht neben der farblichen Markierung. Die blaue Taste ist einheitlich auf jeder Seite für die *Hilfe* reserviert. Die rote Taste ist hauptsächlich für das *Hauptmenü* bestimmt, in dem viele Funktionen aufgerufen werden können. Näheres zu den Funktionalitäten findet sich in dem folgendem Kapitel. Im Kapitel *Programmstart der Mediengalerie* wird beschrieben wie die Anwendung gestartet wird.

4.6.2 Beschreibung der einzelnen Bildschirme

4.6.2.1 Anmeldung

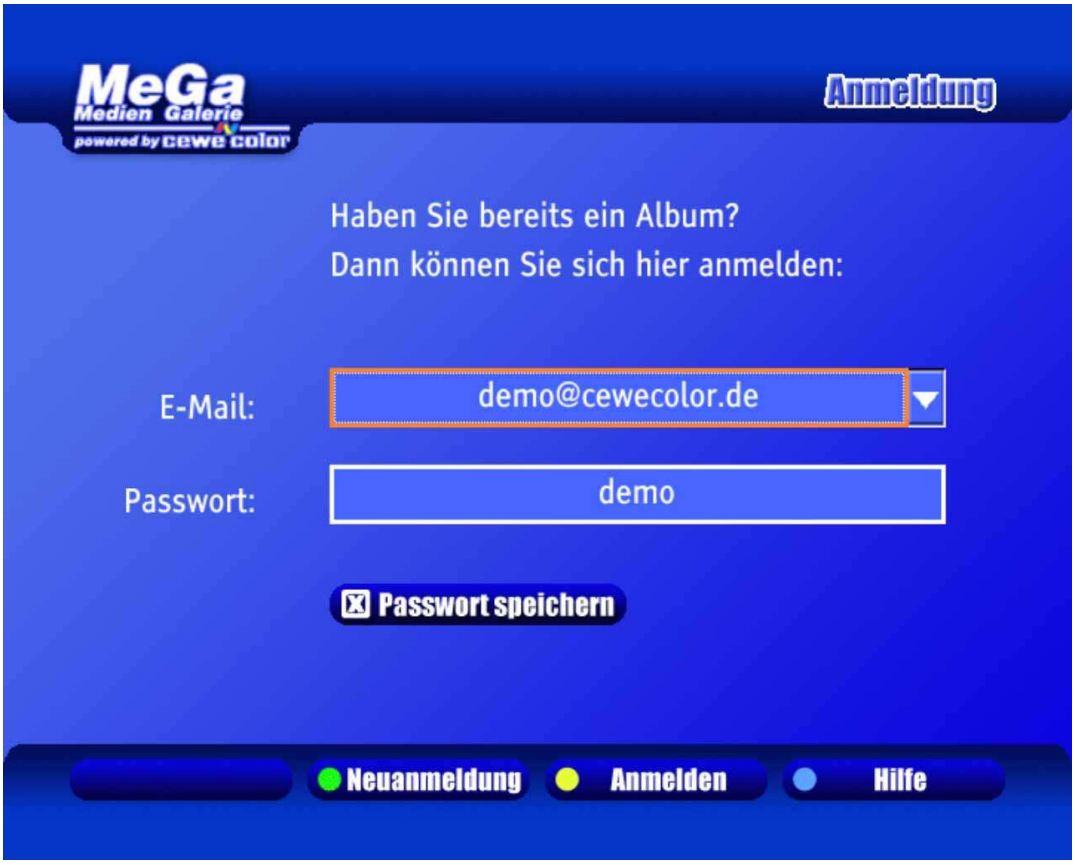

Abbildung 4.31: Anmeldung

Nachdem das Programm gestartet und geladen wurde, gelangen Sie zunächst zur Anmeldung. Bevor Sie diese Anwendung nutzen können, müssen Sie sich einmalig registriert haben (*Neuanmeldung* - grüne Farbtaste) und sich vor jeder Benutzung mit Ihrer E-Mail-Adresse und Ihrem Passwort anmelden. Falls Sie bereits registriert sind und bei einem unserer Kunden ein Album besitzen, können Sie sich anmelden. Dazu gehen Sie zunächst auf das Feld *E-Mail* und drücken die OK-Taste. Es erscheint eine Liste mit bereits verwendeten E-Mail-Adressen, aus der Sie ihre Adresse auswählen können. Befindet sich Ihre Adresse noch nicht in dieser Liste oder Sie melden sich zum ersten mal an, so wählen Sie den Eintrag *E-Mail eintragen* und bestätigen mit der OK-Taste. Sie gelangen dann in die *Eingabehilfe*, wo Sie eine neue E-Mail-Adresse eintragen können. Als nächstes müssen Sie noch Ihr Passwort

eingeben. Gehen Sie auf das Feld *Passwort* und drücken Sie die OK-Taste. Sie gelangen in die *Eingabehilfe* und können Ihr Passwort eingeben. Es besteht auch die Möglichkeit das Passwort abzuspeichern, um es nicht jedes Mal eintragen zu müssen. Dazu geben Sie wie beschrieben E-Mail-Adresse und Passwort ein und gehen auf das Feld *Passwort speichern*. Durch Drücken der OK-Taste können Sie diese Option aktivieren bzw. deaktivieren. Wenn Sie Ihre E-Mail-Adresse und Ihr Passwort eingegeben haben, drücken Sie die gelbe Taste auf Ihrer Fernbedienung, um den Anmeldevorgang zu starten.

4.6.2.2 Eingabehilfe

Abbildung 4.32: Eingabehilfe

Die Eingabehilfe dient dem Eingeben von Texten mit Hilfe der Fernbedienung, vergleichbar mit dem Schreiben einer SMS-Nachricht mit einem Mobiltelefon. Mit den Nummerntasten 0-9 Ihrer Fernbedienung können Sie einzelne Zeichen eingeben. Die Zuordnung der Zeichen zu den Nummern-

tasten der Fernbedienung ist der Grafik zu entnehmen. Beispiel: um eines kleines c einzugeben, müssen zunächst die Kleinbuchstaben aktiviert werden (mit der grünen Farbtaste kann zwischen Klein- und Großbuchstaben, sowie Sonderzeichen gewechselt werden). Bei dreimaliger Betätigung der Taste 2 erscheint dann der Buchstabe c. Sollten Sie sich vertippt haben, können Sie über die rote Farbtaste das letzte Zeichen löschen. Über die gelbe Farbtaste bestätigen Sie die Eingabe des Wortes bzw. der Zeichenkette und dieses erscheint in dem Feld in dem die Eingabehilfe aufgerufen wurde.

4.6.2.3 Neuanmeldung

Abbildung 4.33: Neuanmeldung

Falls Sie bisher noch nicht registriert sind, können Sie dieses in der Neuanmeldung vornehmen. In drei Schritten ist es möglich sich ganz einfach anzumelden. Die mit Sternchen (*) gekennzeichneten Felder müssen Sie eintragen, um sich registrieren zu lassen, die anderen Felder sind optional. Im ersten

Schritt wählen Sie den Photopartner bei dem Sie sich anmelden möchten aus, tragen Ihre *E-Mail-Adresse* und ihr *Passwort* ein. Über die gelbe Taste der Fernbedienung gelangen Sie zu dem zweiten Schritt: *Anrede*, *Vorname*, *Nachname* und *Geburtsdatum* müssen Sie hierzu eingeben. Erneut gelangen Sie mit der gelben Taste zum dritten und letzten Schritt. Hier tragen Sie Ihre Adressdaten ein. Bestätigen können Sie ihre Registrierung über die gelbe Taste. Mit der roten Taste (*Abbrechen*) können Sie jederzeit die Registrierung abbrechen und gelangen zurück zur Anmeldung.

4.6.2.4 Auswahl des Photo Partners

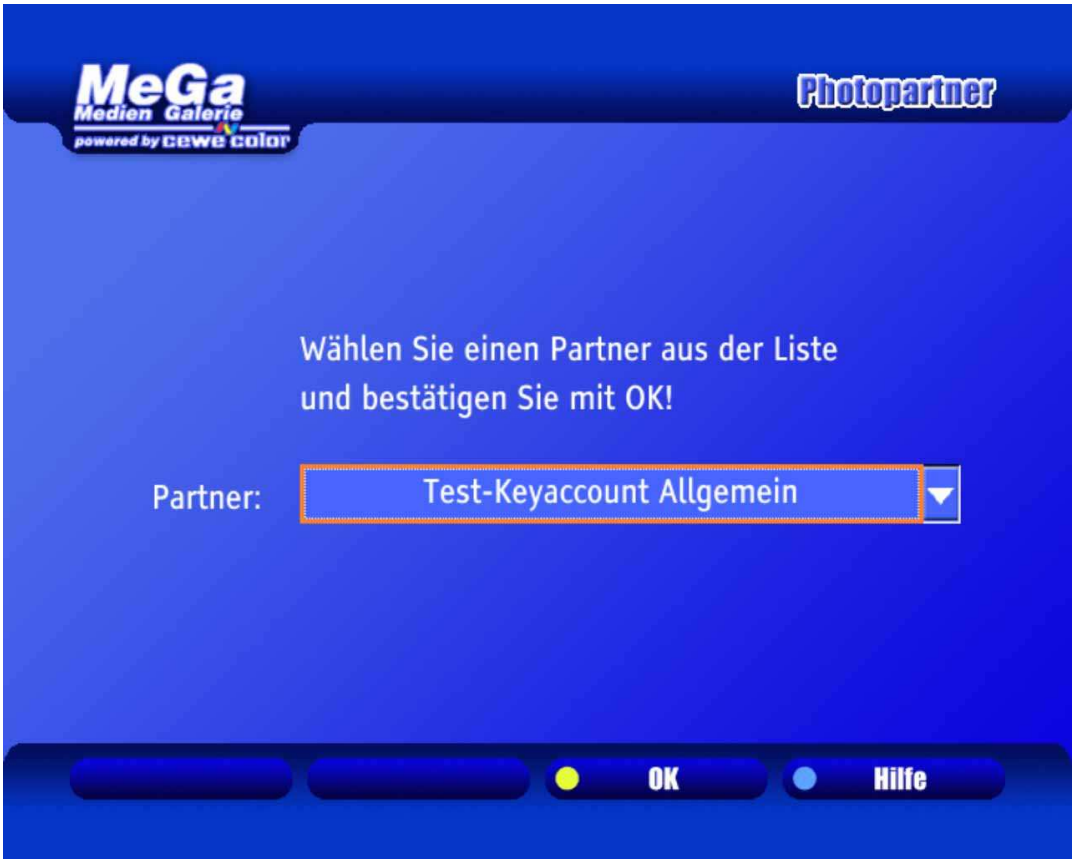

Abbildung 4.34: Auswahl des Photo Partners

Nach erfolgreicher Anmeldung wird man aufgefordert seinen Photopartner auszuwählen. Hierzu wird die OK-Taste gedrückt, um die Liste der Photopartner (Fotopoint, Kruidvat und Schlecker) anzeigen zu lassen. Hier wählen Sie ihren Photopartner aus, in dem Sie mit den Pfeiltasten auf Ihren Partner

gehen, und mit der OK-Taste auswählen. Mit der gelben Taste bestätigen Sie ihre Auswahl und gelangen somit in die Miniaturansicht, in der Sie alle Funktionalitäten dieses Programms nutzen können (u.a. Slideshow, Bestellung von Bildern, Auswahl eines Albums, Vollbildansicht der Bilder etc.).

4.6.2.5 Miniaturansicht

Abbildung 4.35: Miniaturansicht

In der Miniaturansicht können sämtliche Funktionen wie die *Vollbildansicht*, *Diashow*, *Bestellung* von Bildern vorgenommen werden. Über die rote Taste können Sie das *Hauptmenü* öffnen. Hier erscheinen, die verfügbaren Funktionen. Zunächst kann man sich nur *Abmelden* bzw. ein *Album auswählen*, welches auf dem Bildschirm angezeigt werden soll (siehe *Album auswählen*). Im *Hauptmenü* (rote Taste),können nach Auswahl eines Albums weitere Funktionen ausgeführt werden: *Vollbildansicht* (Bilder des ausgewählten Albums können in größerer Ansicht angesehen werden), *Album markieren* (alle

Bilder des Albums werden markiert, dieses ist für das bestellen eines ganzen Albums erforderlich), *Diashow* (die Bilder des Albums werden in bestimmten Zeitintervallen im Vollbild angezeigt), *Alben* (neues Album kann ausgewählt werden), *bestellen* (siehe Kapitel *Fotobestellung/Funproduktbestellung*) und *Einstellungen* (hier kann die Sprache, die Geschwindigkeit der Diashow und die Markierungsfarbe in der Miniaturansicht eingestellt werden).

4.6.2.6 Album auswählen

Abbildung 4.36: Album Auswahl

Mit Betätigen der OK-Taste auf dem Menüpunkt *Alben...* öffnet sich eine Liste die die Alben in verschiedene Kategorien gliedert. Mit der Pfeiltaste wählen Sie die gewünschte Kategorie aus und bestätigen mit der OK-Taste. Nun erscheint der Albumname den Sie wieder auswählen und mit der OK-Taste bestätigen. Es werden die Bilder des ausgewählten Albums geladen und auf dem Bildschirm angezeigt.

4.6.2.7 Fotobestellung/Funproduktbestellung

Abbildung 4.37: Fotobestellung - Auswahl des Formats

Um Bilder zu bestellen markieren Sie die gewünschten Bilder mit der gelben Farbtaste der Fernbedienung. Möchten Sie ein komplettes Album bestellen, können Sie auf dem Menüpunkt *Album markieren* drücken (OK-Taste) und das gesamte Album wird markiert. Nun gehen Sie auf den Menüpunkt *Bestellen* im Hauptmenü (rote Farbtaste). Es erscheint eine weitere Liste, in der Sie wählen können, ob Sie die markierten Bilder als *Funprodukte* (Tasse,

T-Shirt, Mousepad oder Puzzle) oder als *Fotos* bestellen möchten. Wählen Sie den Menüpunkt *Fotos* können Sie das Format der Fotos festlegen, dieses kann im Warenkorb noch verändert werden. Wählen Sie den Menüpunkt *Funprodukte* können Sie den gewünschten Artikel auswählen, der ebenfalls im Warenkorb noch verändert werden kann.

Abbildung 4.38: Fotobestellung - Fotos in Warenkorb abgelegt

Ist das Format bzw. Funproduktartikel ausgewählt, erscheint ein Informationsfenster, welches anzeigt wieviele Bilder bzw. Funproduktartikel in den Warenkorb gelegt worden sind und stellt die Frage, ob Sie die Bestellung jetzt ausführen möchten. Wenn Sie *Ja* wählen gelangen Sie in den *Warenkorb*, sollten Sie *Nein* wählen gelangen Sie wieder in die Miniaturansicht. Dort können Sie weitere Fotos bzw. Funproduktartikel in den Warenkorb legen. Um zum Warenkorb zu gelangen Drücken Sie die rote Farbtaste (Hauptmenü) und wählen den Menüpunkt *Warenkorb* aus. Näheres zum Warenkorb finden Sie im Kapitel *Warenkorb* der in drei Schritten die Bestellung abschließt.

4.6.2.8 Vollbildansicht

Abbildung 4.39: Vollbildansicht

In der Vollbildansicht können die Bilder des ausgewählten Albums einzeln angesehen werden. Über die Pfeiltasten nach links und rechts gelangen Sie somit zu dem vorherigen bzw. nächsten Bild. Mit der roten Farbtaste der Fernbedienung öffnet sich das Hauptmenü über welches Sie zurück in die *Miniaturansicht* gelangen können, ein neues Album öffnen bzw. über den Menüpunkt *Bestellen...* das jeweils vorliegende Bild in den Warenkorb (für den Bestellvorgang) legen können. Nachdem Sie ein bzw. mehrere Fotos in den Warenkorb gelegt haben, erreichen Sie diesen über das Hauptmenü (rote Taste) und dem Menüpunkt *Warenkorb*.

4.6.2.9 Diashow

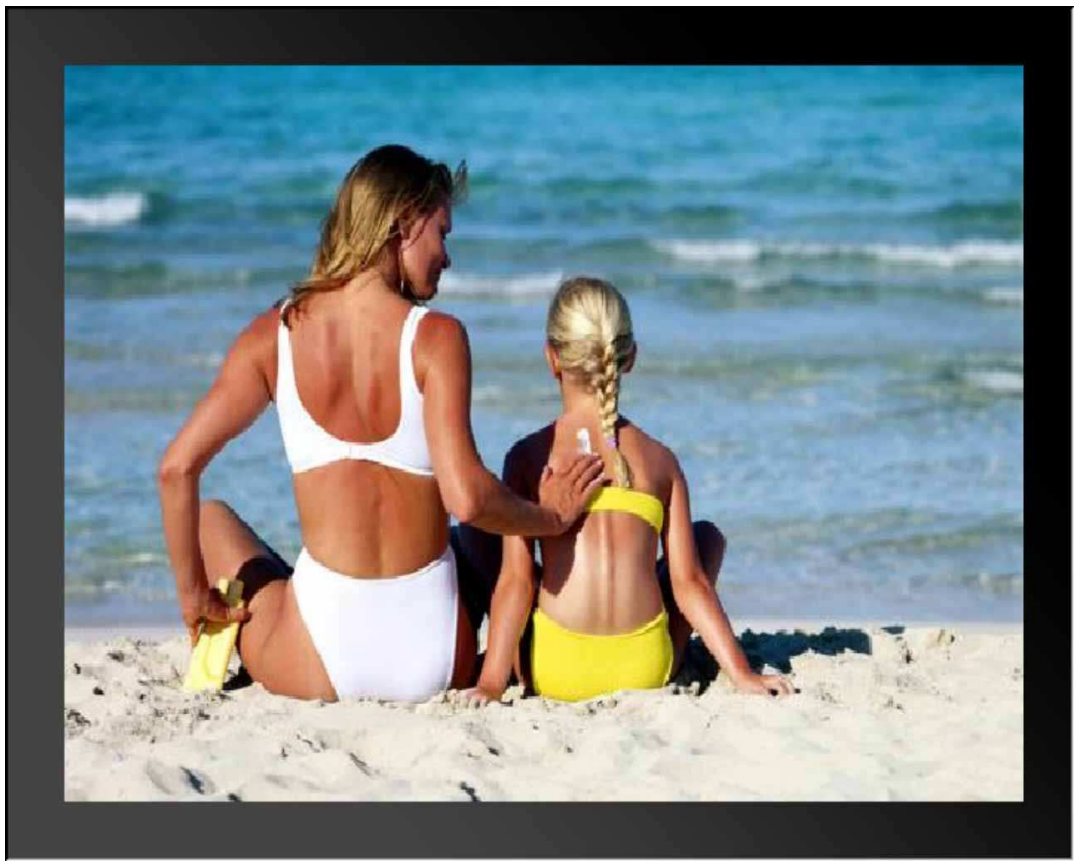

Abbildung 4.40: Diashow

Die Diashow (Bilder des zuvor ausgewählten Albums erscheinen nun im Vollbildmodus und es wird automatisch in einem bestimmten Zeittakt zum nächsten Bild übergegangen) erreichen Sie nach Auswahl eines Albums über das Hauptmenü (rote Taste) und dem Menüpunkt *Diashow*. Es erscheint eine Information, dass Sie *OK* drücken müssen, um die Slideshow zu starten. Beendet wird diese durch Drücken einer beliebigen Taste.

4.6.2.10 Einstellungen

Abbildung 4.41: Einstellungen

An dieser Stelle können die Sprache eingestellt werden (zurzeit ist nur Deutsch möglich), die Geschwindigkeit der Diashow (Zeittakt in dem die Bilder wechseln), die Markierungsfarbe in der Miniaturansicht und die Aktivierung bzw. Deaktivierung des Informationsfensters (erscheint u.a. vor der Diashow, welches Informationen über die Bedienung liefert).

4.6.2.11 Warenkorb - Schritt1: Fotobestellung

Abbildung 4.42: Warenkorb - Schritt1: Fotobestellung

Der Warenkorb gliedert sich in mehrere Schritte. Der erste Schritt ist die Fotobestellung, in der die Formate und die Mengenangaben eingegeben werden können. Im zweiten Schritt ist die Funproduktbestellung, in dem die Artikel und die Mengenangaben eingegeben werden können. Sollten nur Fotos bzw. nur Funprodukte bestellt werden, wird ein Schritt übersprungen. Im dritten und letzten Schritt wird die Bestellung abgeschickt. Über die Pfeiltasten der Fernbedienung können Sie die Felder *Format* und *Menge* anwählen (OK-Taste). Beim Format erscheint eine Liste in der Sie das Format des jeweiligen Bildes festlegen können. Eine Ampelfunktion (rot bis grün) zeigt Ihnen dabei an, wie sich die Formatänderung auf die Qualität des Fotos auswirken wird. Grün bedeutet, dass die Qualität des Fotos gut sein wird, rot dagegen das die Qualität nicht empfehlenswert sein würde. Bei dem Feld Menge geben Sie die Anzahl der zu bestellenden Fotos über das Zahlenfeld

der Fernbedienung ein. Der Preis wird bei jeder Änderung automatisch aktualisiert. In dem unteren Balken können Sie auch noch auswählen, ob Sie die bestellten Fotos in der Filiale abholen möchten (*Filialabholung*) oder lieber den Postweg wählen (*Postversand*).

4.6.2.12 Warenkorb - Schritt2: Funproduktbestellung

Abbildung 4.43: Warenkorb - Schritt1: Funproduktbestellung

Über die Pfeiltasten der Fernbedienung können Sie die Felder *Format* und *Menge* anwählen (OK-Taste). Beim Format erscheint eine Liste in der Sie den gewünschten Artikel (Bierkrug, Baseballcap, Mousepad, Puzzle, Tasse, Teddy oder T-Shirt) festlegen können. Bei dem Feld Menge geben Sie die Anzahl der zu bestellenden Artikel über das Zahlenfeld der Fernbedienung ein. Der Preis wird bei jeder Änderung automatisch aktualisiert. In dem unteren Balken können Sie auswählen, ob Sie die bestellten Artikel in der Filiale abholen möchten (*Filialabholung*) oder lieber den Postweg wählen (*Postversand*).

4.6.2.13 Warenkorb - Schritt3: Bestellung abschicken

Im dritten Schritt wird die Bestellung abgeschickt. Im Fall der Filialabholung wird die Filiale ausgewählt, an dem die Bestellung abgeholt werden soll. Im Falle des Postversands muss die Anschrift des Kunden eingegeben bzw. kontrolliert und die Zahlungsmodalitäten (Kreditkartenzahlung, Lastschriftverfahren oder Rechnung) festgelegt werden. Über die grüne Taste gelangen Sie zurück zu dem ersten Bestellschritt, falls Sie Änderungen vornehmen möchten. Mittels der roten Taste (im ersten bzw. zweiten Schritt) können Sie in die Vollbild- bzw. Miniaturansicht wechseln. Zudem ist es auch möglich den Warenkorb zu leeren.

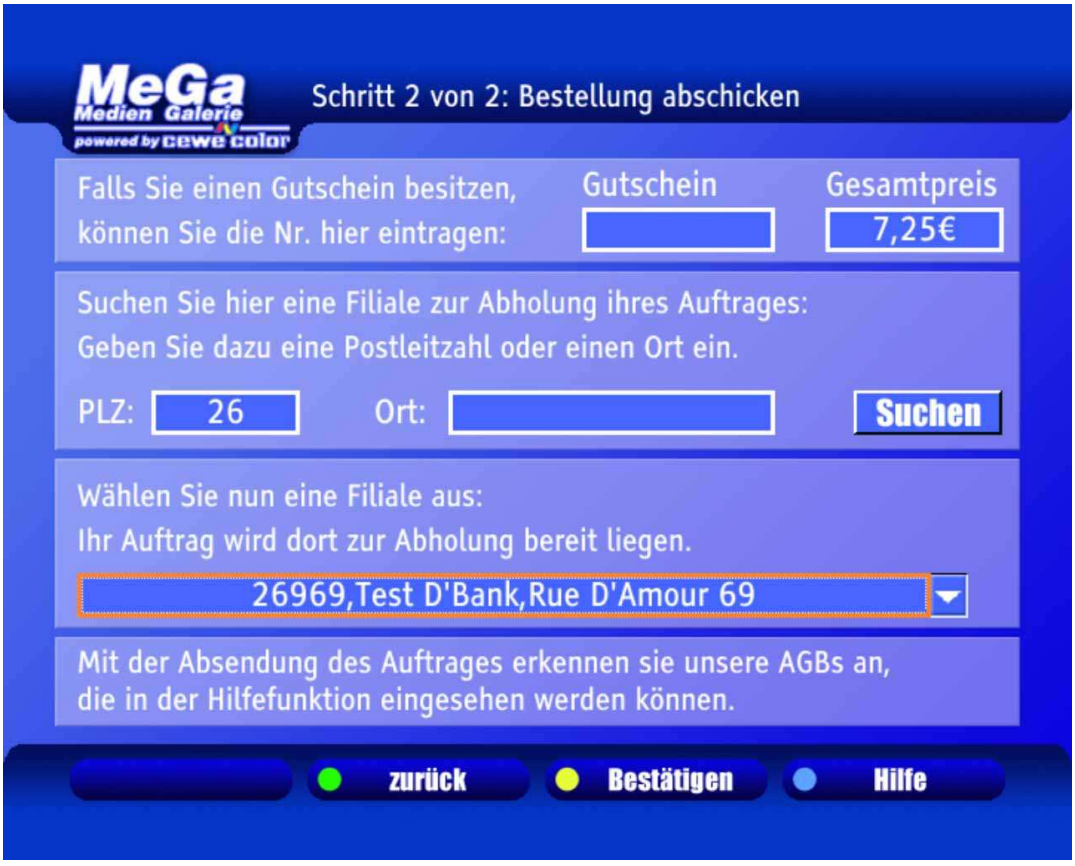

Abbildung 4.44: Warenkorb - Schritt3: Bestellung abschicken (Filialabholung)

4.6.2.13.1 Filialabholung Über das Eingabefeld PLZ und/oder Ort können Sie die nächst gelegene Filiale suchen lassen (dazu Drücken Sie nachdem PLZ

und/oder Ort angegeben wurde den *Suchen*-Button). In der Liste können Sie nun eine der Filialen aussuchen, bei der Sie die Bestellung abholen möchten. Sollte keine Filiale gefunden worden sein, erscheint diese Information in einem Fenster. Überprüfen Sie in diesem Fall nochmal ihre Eingaben. Nachdem Sie dann eine Filiale ausgewählt haben, können Sie mit der gelben Farbtaste die Bestellung abschicken. Wenn diese erfolgreich ausgeführt wurde, erscheint eine Bestätigung.

Abbildung 4.45: Warenkorb - Schritt3: Bestellung abschicken (Postversand)

4.6.2.13.2 Postversand Zunächst wird die Bezahlungsart (Kreditkartenzahlung, Lastschriftverfahren oder Rechnung) festgelegt. In den nächsten Feldern folgt die Anschrift des Kunden, die auf Richtigkeit kontrolliert werden muss. Falls *Rechnung* als Zahlungsmodalität ausgewählt wurde, wird die Rechnung an diese Anschrift geschickt. Sollte das *Lastschriftverfahren* bzw. *Kreditkartenzahlung* ausgewählt worden sein, müssen die Eingabefelder für

das Lastschriftverfahren bzw. Kreditkartenzahlung ausgefüllt werden. Nachdem alle Daten eingetragen wurden, kann die Bestellung abgeschickt werden. Mit der gelben Taste bestätigen Sie ihre Bestellung und erhalten bei erfolgreichem Bestellvorgang eine Bestätigung.

4.6.2.14 Hilfe

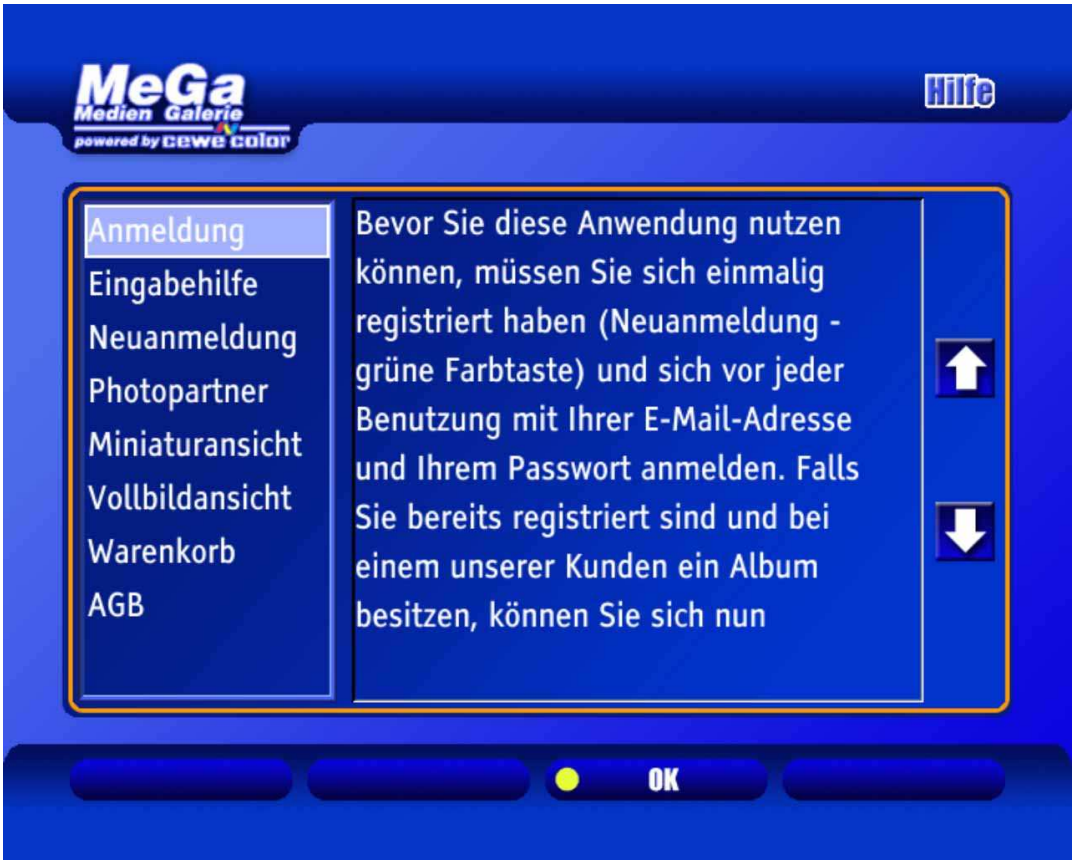

Abbildung 4.46: Hilfe

Die Hilfe erreichen Sie auf jeder Seite über die blaue Taste. In der linken Spalte befindet sich das Inhaltsverzeichnis und in dem rechten Feld befindet sich der Hilfetext zu der jeweiligen Seite, den Sie ausgewählt haben. Mit der Pfeil-nach-unten- bzw. Pfeil-nach-oben-Taste können Sie den Hilfetext durchblättern. Zurück gelangen Sie über die gelbe Farbtaste der Fernbedienung.

4.7 Handbuch Tippspiel Tipp2Win

4.7.1 Allgemeines

4.7.1.1 Einleitung

Mit dem Übergang vom analogen zum digitalen Fernsehen und der damit verbundenen Entwicklung von digitalen Fernsehempfängern, den sogenannten *Settopboxen*, eröffnet sich eine neue Möglichkeit Programme einem breiten Publikum zugänglich zu machen. Es wird kein kompletter PC mehr benötigt. Ein MHP-fähiger Digitalempfänger in Verbindung mit einem Fernseher ist vollkommen ausreichend.
Da viele dieser *Settopboxen* über ein Modem oder auch einen Netzwerkanschluss verfügen, ist ebenfalls die Möglichkeit dazu gegeben, mehrere Boxen miteinander kommunizieren zu lassen. Eine der offensichtlichsten Anwendungsmöglichkeiten ist hier ein Spiel anzubieten, dass es einer großen Gemeinschaft von Nutzern ermöglicht, eine spannende Zeit vor dem Fernseher zu verbringen.
Dieser Gedanke wurde mit **Tipp2Win** aufgegriffen. Der Fernsehzuschauer kann dem ohnehin schon hohen Nervenkitzel einer Fußball-Europameisterschaft nun einen weiteren spannenden Aspekt hinzufügen. Er kann zusammen mit anderen in einer Art Wettgemeinschaft Tipps abgeben und sich so mit weiteren Zuschauern, welche ebenfalls im Besitz von **Tipp2Win** sind, im Wettbewerb messen.
Die Tipps die abgegeben werden können, beschränken sich nicht nur auf Spielergebnisse, sondern umfassen auch andere Aspekte, wie z.B. die Tippmöglichkeit eines Torschützenkönigs.

4.7.1.2 Aufbau

Dieses Handbuch ist wie folgt aufgebaut:

- allgemeine Informationen zur Bedienung mit der Fernbedienung
- Beschreibung der einzelnen Bildschirme. Zur Illustration werden einige Screenshots der Beschreibung hinzugefügt.

4.7.1.3 Bedienung

Die Bedienung des Programms erfolgt mit der Fernbedienung der Settopbox. Die Pfeiltasten werden benutzt, um zu den einzelnen Eingabefeldern zu gelangen, die dann mit der OK-Taste angewählt werden können. Die hauptsächliche Bedienung erfolgt über die Farbtasten (rot, grün, gelb und

blau) der Fernbedienung. In der unteren Leiste jedes Bildschirmes ist die Belegung der Farbtasten für den jeweiligen Bildschirm dargestellt. Die blaue Taste ist einheitlich in jedem Bildschirm für die *Hilfe* reserviert. Näheres zu den Funktionalitäten findet sich in dem folgenden Kapitel.

4.7.2 Beschreibung der einzelnen Bildschirme

4.7.2.1 Anmeldung

Bevor Sie diese Anwendung nutzen können, müssen Sie sich einmalig registriert haben (*Neuanmeldung* - grüne Farbtaste) und sich vor jeder Benutzung mit Ihrer E-Mail-Adresse und Ihrem Passwort anmelden. Falls Sie bereits registriert sind können Sie dies nun machen. Dazu gehen Sie zunächst auf das Feld *E-Mail* und drücken die OK-Taste. Es erscheint eine Liste mit bereits verwendeten E-Mail-Adressen, aus der Sie ihre Adresse auswählen können. Befindet sich Ihre Adresse noch nicht in dieser Liste oder melden Sie sich zum ersten Mal an, so wählen Sie den Eintrag *E-Mail eintragen* und bestätigen mit der OK-Taste. Sie gelangen dann in die *Eingabehilfe*, wo Sie eine neue E-Mail-Adresse eintragen können. Als nächstes müssen Sie noch Ihr Passwort eingeben. Gehen Sie auf das Feld *Passwort* und drücken Sie die OK-Taste. Sie gelangen dann ebenfalls in die *Eingabehilfe* und können Ihr Passwort eingeben. Es besteht auch die Möglichkeit das Passwort abzuspeichern, um es nicht jedes Mal eintragen zu müssen. Dazu geben Sie wie beschrieben E-Mail-Adresse und Passwort ein und gehen auf das Feld *Passwort speichern*. Durch Drücken der OK-Taste können Sie diese Option aktivieren bzw. deaktivieren. Wenn Sie Ihre E-Mail-Adresse und Ihr Passwort eingegeben haben, drücken Sie die gelbe Taste auf Ihrer Fernbedienung, um den Anmeldevorgang zu starten.

Abbildung 4.47: Anmeldung

4.7.2.2 Eingabehilfe

Die Eingabehilfe dient dem Eingeben von Texten mit Hilfe der Fernbedienung, vergleichbar mit dem Schreiben einer SMS-Nachricht mit einem Mobiltelefon. Mit den Nummerntasten 0-9 Ihrer Fernbedienung können Sie einzelne Zeichen eingeben. Die Zuordnung der Zeichen zu den Nummerntasten der Fernbedienung ist der Grafik auf der linken Seite zu entnehmen. Beispiel: um eines kleines 'c' einzugeben, müssen zunächst die Kleinbuchstaben aktiviert werden (mit der grünen Farbtaste kann zwischen Klein- und Großbuchstaben, sowie Sonderzeichen gewechselt werden). Bei dreimaliger Betätigung der Taste 2 erscheint dann der Buchstabe 'c'. Sollten Sie sich vertippt haben, können Sie über die rote Farbtaste das letzte Zeichen löschen. Über die gelbe Farbtaste bestätigen Sie die Eingabe des Wortes bzw. der Zeichenkette und dieses erscheint in dem Feld in dem die Eingabehilfe aufgerufen wurde.

Abbildung 4.48: Eingabehilfe

4.7.2.3 Neuanmeldung

Falls Sie bisher noch nicht registriert sind, können Sie dieses in der Neuanmeldung durchführen. In zwei Schritten ist es möglich sich ganz einfach anzumelden. Tragen Sie zu diesem Zweck einfach Ihre persönlichen Daten in die dafür vorgesehenen Felder ein. Sie können anhand der Zahlen am oberen Bildschirmrand leicht erkennen, in welchem Schritt Sie sich gerade befinden. In den zweiten Schritt gelangen Sie, indem Sie, wenn sie sich in einem Eingabefeld befinden, die rechte Pfeiltaste auf ihrer Fernbedienung drücken. Bestätigen können Sie ihre Registrierung über die gelbe Taste. Mit der roten Taste (*Hauptmenü*) können Sie jederzeit die Registrierung abbrechen und gelangen zurück zur Anmeldung.

Abbildung 4.49: Neuanmeldung

4.7.2.4 Hauptmenü

Nach erfolgreicher Anmeldung werden Sie das Hauptmenü von **Tipp2Win** vor sich sehen. Dieses bietet Ihnen vier Auswahlmöglichkeiten.

- **Tipps:** Über diesen Menüpunkt gelangt man in das Tipps-Menü. (Dazu später mehr)
- **Rangliste:** Hier haben Sie die Möglichkeit Ihren aktuellen Punktestand einzusehen und ihn mit dem der anderen Spieler zu vergleichen.
- **Statistik:** Hinter diesem Menüpunkt verbirgt sich der Teil von **Tipp2-Win**, in dem Sie Statistiken zu den bereits gelaufenen Europameisterschaften abrufen können.
- **Profil:** Hier kann das Profil (Ihre persönlichen Daten und Einstellungen) bearbeitet werden.

Sie können einen dieser Punkte auswählen, indem Sie ihn mittels der Links- und Rechts-Pfeiltasten Ihrer Fernbedienung markieren und dann die Ok-Taste ebenfalls auf Ihrer Fernbedienung drücken. Über ein Drücken der roten Taste gelangen Sie wieder zum Anmeldebildschirm zurück.

Abbildung 4.50: Hauptmenü

4.7.2.5 Rangliste

In der Rangliste können Sie den Punktestand sämtlicher Mitspieler einsehen und die sich daraus ergebende Platzierung. Innerhalb der Liste können Sie nach oben bzw. unten scrollen, wenn sie sich auf dem ersten oder letzten Eintrag befinden und dann die obere bzw. untere Pfeiltaste auf Ihrer Fernbedienung betätigen. Wenn Sie einen der Nicknames aus der Liste auswählen (*Ok-Taste*), werden im rechten Bereich des Bildschirms weitere Informationen zu diesem Mitspieler angezeigt. Durch diese können sie ebenfalls mittels der Pfeile am rechten Rand hindurchscrollen. Dies geschieht durch ein Auswählen des jeweiligen Pfeiles gefolgt von einem Druck auf die Taste '*Ok*'.

Abbildung 4.51: Rangliste

4.7.2.6 Statistik

In diesem Abschnitt können Sie die für **Tipp2Win** verfügbaren statistischen Informationen bezüglich der Europameisterschaft abrufen. Dazu wählen Sie zunächst den entsprechenden Bereich über das Drop-Down-Menü im oberen Bereich des Bildschirms aus. Der dazugehörige Text wird dann in dem Textfeld direkt unter dem Menü angezeigt. Durch diesen Text können Sie über die sich rechts davon befindenden Pfeiltasten hindurchscrollen.

Abbildung 4.52: Statistik

4.7.2.7 Profil

Hier können Sie ihr persönliches Benutzerprofil bearbeiten. Mit den Hoch- und Runter-Pfeiltasten auf ihrer Fernbedienung können sie die einzelnen Textfelder auswählen, in welche sie dann nach einem Drücken der Enter-Taste Ihrer Fernbedienung einen Text eintragen können. Über ein Drücken der Links- bzw. Rechts-Pfeiltaste gelangen sie auf die zweite oder wieder auf die erste Seite der Profilbearbeitung. Mit der gelben Taste Ihrer Fernbedienung können Sie Ihre Änderungen übernehmen. Sie können sie jedoch auch verwerfen, indem Sie mittels der roten Taste ins Hauptmenü zurückkehren.

Abbildung 4.53: Profil

4.7.2.8 Tippsmenü

In dieses Auswahlmenü können Sie gelangen, indem Sie im Hauptmenü den Punkt *Tipps* auswählen. Die Navigation funktioniert auf dieselbe Art und Weise wie im Hauptmenü, bloß dass Sie hier über die rote Taste zum Hauptmenü zurückgelangen. Die Punkte die hier zur Auswahl stehen sind:

- **Fairplay:** Hier können Sie die Ihrer Meinung nach fairste Mannschaft tippen.
- **Torschütze:** Hier kann der Spieler getippt werden, der während der Europameisterschaft die meisten Tore erzielen wird.
- **Gruppentipps:** In diesem Abschnitt können Sie die Ergebnisse der Gruppenspiele bei der Europameisterschaft tippen.
- **Finaltipps:** Hier können ebenfalls Spielergebnisse getippt werden. Nur handelt es sich hier um die Spiele im Viertel-, Halb- und Finalspiel.

Abbildung 4.54: Tippsmenü

4.7.2.9 Fairplaytipp

Hier haben Sie die Möglichkeit auf die Nationalmannschaft zu tippen, die Ihrer Meinung nach die wenigsten gelben und roten Karten erhalten wird. Sprich: *Die Mannschaft, die am fairsten spielt.*
Sie können die Mannschaft über das Drop-Down-Menü in der Mitte auswählen. Sind Sie mit Ihrer Wahl einverstanden, so können Sie diesen Tipp über einen Druck auf die gelbe Taste (*Bestätigen*) verbindlich abgeben.
Vorsicht: Abgegebene Tipps können nicht mehr verändert werden!

Abbildung 4.55: Fairplay-Tipp

4.7.2.10 Torschützentipp

In diesem Bildschirm können Sie den Torschützenkönig bestimmen, also jenen Spieler, welcher im Verlauf der Europameisterschaft die meisten Tore erzielen wird. Dazu gehen Sie wie folgt vor.
Zunächst wählen Sie im linken Drop-Down-Menü die Nationalmannschaft aus, aus der der Spieler stammen soll und danach wählen Sie ihn im rechten Menü direkt aus.
Auch hier bestätigen Sie Ihren Tipp wieder mit der gelben Taste auf Ihrer Fernbedienung.
Vorsicht: Abgegebene Tipps können nicht mehr verändert werden!

Abbildung 4.56: Torschützentipp

4.7.2.11 Gruppentipps

In diesem Abschnitt von **Tipp2Win** können Sie Tipps darüber abgeben, wie ein Spiel in den einzelnen Gruppen ausgehen wird.
Zu diesem Zweck wählen Sie über das Drop-Down-Menü die Gruppe aus, welche angezeigt werden soll. Nun können sie in die Texteingabefelder Ihren Tipp eingeben. Die Beurteilung der Tipps erfolgt nach drei Kriterien:

- **Korrektheit des Ergebnisses:** Das Ergebnis wurde komplett richtig getippt.
- **Richtige Tordifferenz:** Der Unterschied an Toren bei Spielergebnis und Tipp ist der gleiche.
- **Richtige Tendenz des Tipps:** Der Tipp stimmt mit dem Spielergebnis in der grundsätzlichen Richtung überein.

Mittels der gelben Taste bestätigen Sie auch hier wieder Ihren Tipp. Wichtig ist hierbei, dass immer nur die aktuell sichtbaren Spiele getippt werden. Leere Eingabefelder werden bei der Bestätigung eines Tipps als '0' gewertet. Sollten Spiele bereits gelaufen sein, so sind im Bereich rechts neben den Eingabefeldern die Ergebnisse eingetragen. Natürlich ist es in diesem Fall nicht mehr möglich noch einen Tipp abzugeben.
Vorsicht: Abgegebene Tipps können nicht mehr verändert werden!

Abbildung 4.57: Gruppentipps

4.7.2.12 Finaltipps

In diesem Abschnitt von **Tipp2Win** können Sie tippen, wie die Spiele in den Finalrunden ausgehen werden.
Über das Drop-Down-Menü können Sie das entsprechende Finale auswählen und dann, sobald dies geschehen ist, Ihre Tipps abgeben. Genau wie bei den Gruppentipps werden auch hier die Ergebnisse bereits stattgefundener Spiele rechts neben den Eingabefeldern angezeigt. Die Bewertung der Tipps erfolgt auch hier in der gleichen Art und Weise, wie bei den Gruppentipps.
Bei der Bestätigung über die gelbe Taste gilt auch hier, dass immer nur die aktuell sichtbaren Spiele getippt werden. Leere Eingabefelder werden bei der Bestätigung eines Tipps als '0' gewertet.
Vorsicht: Abgegebene Tipps können nicht mehr verändert werden!

Abbildung 4.58: Finaltipps

4.7.2.13 Hilfe

Die Hilfe erreichen Sie in jedem Bildschirm über die blaue Taste. In der linken Spalte befindet sich das Inhaltsverzeichnis und in dem rechten Feld befindet sich der Hilfetext zu dem jeweiligen Bildschirm, den Sie ausgewählt haben. Mit der Pfeil-nach-unten- bzw. Pfeil-nach-oben-Taste neben dem Text können Sie jenen durchblättern. Zurück gelangen Sie über die gelbe Farbtaste der Fernbedienung.

Abbildung 4.59: Hilfe

4.8 Mögliche Weiterentwicklungen

4.8.1 Mediengalerie

In dem Jahr Projektgruppenarbeit wurden die vom Projektpartner CeWe Color gewünschte Features wie das Anzeigen der Bilder und Alben, das Vorhandensein einer Diashow und die Möglichkeit Fotos und Funprodukte zu bestellen, umgesetzt. Im Folgenden werden weitere optionale und sinnvolle Erweiterungen der Mediengalerie vorgestellt:

Anlegen eines Offline-Albums: Dem Benutzer soll über einen Eintrag im Hauptmenü ermöglicht werden diverse Alben anzulegen und über Untermenüs mit Bildern aus dem Fotoarchiv zu füllen. Diese sollen lokal auf einem USB-Stick oder anderem Speichermedium abgelegt werden können. Ein anderes Speichermedium kann dabei auch eine resident in der Box vorhandene Festplatte sein, die jedoch noch kein Standardbauteil von Set-Top-Boxen ist.

Lokale Bilder bestellen: Da es in Zukunft auch vermehrt Set-Top-Boxen auf dem Markt geben wird, die über eine Festplatte, ein CD/DVD-Laufwerk oder zumindest über einen USB-Port verfügen, soll es in Zukunft auch möglich sein, Bilder, die sich lokal auf der Set-Top-Box befinden, zu bestellen. Hierfür ist der sogenannte *orderdeclaration_request* zuständig. Im *CeWe-Client* ist diese Anfrage deshalb bereits implementiert. Im Fall einer späteren Weiterentwicklung der Mediengalerie, ist also die Anbindung an den CeWe-Server über den *CeWe-Client* bereits realisiert. Eine Möglichkeit das Feature: „Bestellung von lokalen Bildern“ fertig zu stellen, wird im Folgenden skizziert:
Um die Bilder auf der Festplatte, der CD/DVD oder dem USB-Stick zu lokalisieren, ist normalerweise eine Art Dateiexplorer notwendig. Da die Bedienung eines solchen Explorers möglicherweise zu kompliziert für den einfachen Fernsehzuschauer ist, wäre es auch möglich einen Pfad zu einem bestimmten Ordner festzulegen, in dem sich die Bilder befinden müssen. Ein solcher Pfad könnte z.B. als zusätzliche Option im Preferences-Screen einstellbar sein. Dem Hauptmenü müssten dann noch zwei Elemente hinzugefügt werden, mit denen die Bilder entweder in den Foto-Warenkorb oder in den Funartikel-Warenkorb geladen würden. Dies hieße, dass automatisch alle Bilder, die sich in dem spezifizierten Ordner befinden, in den Warenkorb gelegt werden. Das Löschen von nicht erwünschten Bildern oder das Ändern der Formate lässt sich auch noch im Warenkorb-Bildschirm tätigen. Zusätzlich zu den Informationen, die bereits für die normale Bestellung aus dem

Archiv benötigt werden, müssen für den *orderdeclaration_request* aber noch andere Angaben, wie z.B. die *crc* und die *filesize* (siehe dazu die XTCI-Beschreibung) ermittelt werden. Diese müssen jedem *CartItem* zusätzlich in Form eines *OrderdeclarationData*-Objektes übergeben werden. Dies erfolgt durch Aufruf der Methode *setOrderdeclarationData*, die sich im *CartItem*-Objekt befindet. Zur Überlieferung der Bestelldaten an den CeWe-Server muss im *CeWeClient* nur noch die bereits implementierte Methode *orderCartOrderdeclaration* mit den richtigen Parametern (abhängig von Postversand oder Filialabholung) aufgerufen werden. Damit wäre das Bestellen von lokalen Bildern abgeschlossen.

Erweiterte Albenverwaltung: Bislang bietet die Mediengalerie noch keine Möglichkeit, Bilder innerhalb von Alben zu verschieben, oder Alben anzulegen und zu löschen. Diese Funktionalitäten können durch geringen programmiertechnischen Aufwand ins Hauptmenü integriert werden.

Freigeben von Alben/Bildern für Freunde: Wenn ein Benutzer seine Bilder Freunden oder Bekannten zeigen möchte, ohne Sie persönlich aufzusuchen, kann diese Möglichkeit über die Freigabe von bestimmten Alben umgesetzt werden. Dazu notwendig ist eine Freundeverwaltung, in der man nach Nutzern bzw. E-Mailadressen sucht und dort einzelne Benutzer als befreundet kennzeichnet.

Zusätzliche Einstellmöglichkeiten: Weitere Einstellungsmöglichkeiten wären zum Beispiel ein Dateimenü für die Auswahl alternativer Skins, ein Regler für die Lautstärke von Musik und An/Ausschalter für diverse Features wie einen Assistenten (der einem Hilfestellung während des Programms gibt) oder die Tooltips (Informationsanzeige, was u.a. bei Drücken eines Buttons passiert).

Filter/Manipulationsfunktionen: Über Filter wird es dem Benutzer ermöglicht, die Bilder individuell umzugestalten. Beispiele für solche Filter sind:

- Drehen, spiegeln
- Strecken, zerren
- Ändern der Auflösung
- Manipulieren der Farben/des Kontrastes
- Gaußscher Weichzeichner

- Mosaik

Zusätzliche Statusanzeigen: Über weitere Statusanzeigen kann dem Anwender mehr Feedback über den aktuellen Status des Programms mitgeteilt werden. Sinnvoll sind diese unter anderem bei dem Anmeldevorgang, dem Laden der Bilder, der Slideshow, bei dem Verschieben von Bildern oder dem Anwenden von Filtern. Denkbar sind auch an und ausschaltbare Tooltips, die unerfahrenen Anwendern den Umgang mit der personalisierten Mediengalerie erleichtern.

Assistent: Die Aufgabe des Assistenten ist es, unerfahrene Benutzer in der Bedienung des Programms zu unterstützen. Angelehnt an bekannte Assistenten aus der Windows-Welt ist der virtuelle Assistent der Mediengalerie intuitiv verständlich und lässt sich je nach Bedarf deaktivieren bzw. Intensität der Hilfestellung anpassen.

Versenden von Bildern als Postkarte oder E-Mail: Alle Bilder, die der Anwender in seinen Online oder Offline-Alben gespeichert hat können als Postkarte oder E-Mail verschickt werden.

Sound/Musikuntermalung: Alle Aktionen des Benutzers sollen durch eine passende Musik begleitet werden. Das schließt sowohl Musikstücke ein, als auch Soundeffekte, die bei der Betätigung von Buttons und dem Abschließen von Aktionen (z.B. dem Bestellvorgang) gespielt werden. Sound und Musikoptionen können im Einstellungsmenü jederzeit an bzw. abgeschaltet sowie in ihrer Lautstärke reguliert werden.

Optimierung der Fotoanzeige: Zurzeit werden Bilder als JPG-Datei geladen, von der CPU dekodiert und im Application-Layer angezeigt. Als vorteilhaft könnte sich die Anzeige von Fotos im Video-Layer herausstellen, da diese Grafikebene auf der Settop-Box durch einen speziellen MPEG-Video-Chip unterstützt wird. Die JPEG-Dateien müssten dabei möglicherweise als MPEG-Standbild kodiert werden. Vorteile sind auf einigen Settop-Boxen nicht nur bezüglich Anzeigegeschwindigkeit, sondern auch bei der Farbtiefe zu erwarten.

4.8.2 Tipp2Win

Die Implementierung der zweiten Anwendung *Tipp2win* wurde mit einem Oberflächenprototyp mit Navigationsfähigkeit durch die einzelnen Bildschirme abgeschlossen. Ebenso existiert eine auf MySQL basierende Datenbank, für die bereits ein auf mehrere Anwendungen portables Schema entwickelt

wurde. Ein Datenaustausch zwischen Modulen der *Model Ebene* und der *View Ebene* findet noch nicht statt. Alle Anzeigen sind zurzeit statisch und hart kodiert. Weiterführende Arbeiten an der Anwendung *Tipp2win* zielen somit in die Richtung einer vollständigen Implementierung und einer Verfeinerung des Oberflächendesigns sowie dem Hinzufügen weiterer Features ab.

Ausdehnung der Tippmöglichkeiten auf diverse Sportarten: Dazu könnten folgende Mannschaftssportarten zählen: Eishockey, Basketball oder Handball zählen. Diese werden auch in einem Turniermodus absolviert, so dass die Tippmöglichkeiten wie Torschützenkönig, fairste Mannschaft und Gruppentipps nur leicht verändert werden müssten. Denkbar ist auch eine Erweiterung auf Einzelsportarten wie Formel 1, Tennis oder Tischtennis. Hier müsste die Oberfläche allerdings angepasst bzw. erneuert werden. Getippt werden könnten u.a. die Startaufstellung und der Sieger.

Erweiterte statistische Funktionen: Denkbar sind archivierte Statistiken über beispielsweise alle Spielergebnisse eines Turniers oder den jeweiligen Torschützenkönig.

4.9 Erfahrungsbericht der Projektgruppe

4.9.1 Allgemeines

Zu Beginn der Projektgruppe hatten wir alle nur eine vage Vermutung, was uns in den folgenden zwölf Monaten erwarten sollte. Es fielen Stichworte wie MHP-Anwendungen erstellen, Seminararbeiten, Fußballturnier und eine PG-Fahrt. Die ersten Wochen wurden zur Vorbereitung der Seminararbeiten genutzt. So wurde eine Basis für die weitere PG-Arbeit geschaffen, indem jeder Teilnehmer sein Themengebiet den anderen vorstellte. Dies war auch wichtig, denn es lagen nach einiger Zeit wichtige Entscheidungen an, z. B. welche Software zur Implementierung verwendet oder welche Hardware angeschafft werden sollte. Solche Dinge haben wir in unseren wöchentlichen PG-Treffen diskutiert, bei denen die Betreuer(innen) beratend zur Seite standen. Diese Treffen wurden von den PG-Teilnehmern, im wöchentlichen Wechsel, geleitet und protokolliert. In einer der ersten Sitzungen vergaben wir Ämter, so dass für wichtige Dinge immer ein Ansprechpartner vorhanden war. Zu den Ämtern gehörten: Administrator, Webmaster, Dokumentenwart, Kümmerer, Urlaubswart und Kassenwart. Der Administrator hatte bald die Aufgabe das Multimedia-Labor mit der nötigen Software auszustatten. Für die Erstellung

und die Aktualisierung der PG-Homepage war der Webmaster verantwortlich. Erster Ansprechpartner für die Verwaltung und möglichen Korrekturen der Dokumente war der Dokumentenwart. Der Kümmerer sorgte für die Organisation von gemeinsamen Kneipen- und Grillabenden. Um die Organisation des Urlaubs kümmerte sich der Urlaubswart und der Kassenwart, um die Korrektheit des PG-Kassenstandes.

Dann kam der große Tag an dem wir unsere ersten praktischen Erfahrungen mit der MHP-Plattform sammelten. Die anfängliche Euphorie wurde durch Ernüchterung verdrängt, da nur wenige funktionierende Beispiele oder gute Literatur im Netz zu finden waren. Nach einiger Zeit begannen wir aber erste Fortschritte zu erzielen und die ersten GUI-Elemente wurden auf der Referenzimplementierung ausprobiert. In dieser Hinsicht ist der größte Unterschied zu den herkömmlichen Vorlesungen zu sehen, bei denen man sich mit der Theorie auseinandersetzt und vielleicht noch ein abstraktes Beispiel dazu bespricht. Als vorteilhaft erwies sich der Projektpartner CeWe Color, der für unsere Mediengalerie zur Seite stand, da uns so auch Einflüsse aus der Wirtschaft zuteil wurden.

4.9.2 Freizeitaktivitäten und PG-Fahrt

Um den Gruppenzusammenhalt zu fördern, fanden in regelmäßigen Abständen Kneipentreffen an wechselnden Orten innerhalb Oldenburgs statt. Es gab auch ein gemeinsames Nudelessen, Grillabende und ein Fußballabend (EM-Spiel Deutschland gegen Holland). Dies diente zur Taktikanalyse unseres Trainings für das alljährlich stattfindende Fußballturnier der Fakultät Informatik, für das Projektgruppen automatisch qualifiziert sind. Unsere Trainingseinheiten fanden schon recht früh im Jahr statt. Viele Freundschaftsspiele gegen andere Projektgruppen förderten unsere Spielpraxis. In der Mitte unserer Projektgruppenzeit fand unsere PG-Fahrt nach Berlin statt, wo wir in den drei Tagen zwei Firmen bzw. Projekte besuchten. Zum einen das Projekt Broadcast_Mobile_Convergence (bmco) und das Fraunhofer Institut FOKUS. Einige Erfahrungen, die wir dort gesammelt haben, konnten wir in unserer Anwendung umsetzen.

4.9.3 Danksagung

Ein recht herzlicher Dank für die erfolgreiche und lehrreiche Unterstützung während der Projektgruppenzeit geht an:

- die Betreuer(innen) Frau Susanne Boll, Frau Sabine Thieme und Herrn Ansgar Scherp

- den Kooperationspartner CeWe Color (auch für die schönen T-Shirts für das Fußballturnier)
- die involvierten Mitarbeiter des OFFIS

A Gesamtglossar

API: Eine *Application Programming Interface* (Dt.: Schnittstelle für die Anwendungsprogrammierung) ist eine definierte und dokumentierte Software-Schnittstelle zwischen zwei verschiedenen Programmen, meist zwischen einem Betriebssystem und einem Anwendungsprogramm.

Automatisierte Tests: Automatisierte Tests sind Testfälle, die mit Hilfe eines Testwerkzeuges wie JUnit gespeichert werden und beliebig oft auf „Knopfdruck“ automatisch wiederholt werden können.

Benutzungsoberfläche: Die Benutzungsoberfläche ist eine visuelle Programmoberfläche, die die Interaktion des Benutzers mit dem Softwaresystem ermöglicht.

Benutzerprofil: Datensammlung über einen Nutzer, mit deren Hilfe zum Beispiel dargebotene Inhalte auf den entsprechenden Nutzer angepasst werden können.

Broadcast: Beim Broadcasting werden Datenpakete ausgesendet an alle, die mit einem digitalen Rundfunkempfänger ausgestattet sind und nicht nur an einen bestimmten Empfänger.

Client: Rechner oder Programm zum Abruf von Daten von einem Server über ein Computer-Netz.

Cookies: Kleine Informationsdateien, die von einer Homepage an den eigenen Web-Browser geschickt und von diesem auf der Festplatte des Anwenders gespeichert werden. Diese Informationen werden vom Browser bei einem erneuten Aufruf dieser Homepage zurückgeschickt. Dadurch können beispielsweise auf einigen Internet-Seiten eigene, individuelle Elemente erzeugt werden, die sofort erscheinen, sobald man diese Seite anwählt. Cookies gibt es auch auf der ↑MHP.

Computer Aided Software Engineering-Tool (CASE-Tool): Ein Computer Aided Software Engineering-Tool ist ein Werkzeug, das zur Unterstützung der Softwareentwicklung eingesetzt wird.

Client-Server-Funktion: Client und Server sind Softwareprogramme, wobei der Client über ein Rechnernetz Daten vom entfernten Server abruft und der Server diese Daten zur Verfügung stellt.

DAB: *DAB* ist die Abkürzung für *Digital Audio Broadcasting*. Es handelt sich dabei um ein standardisiertes Verfahren zur digitalen übertragung

von Radiosignalen. Neben einer verbesserten Klangqualität bietet die digitale übertragungstechnik auch die Möglichkeit, zusätzliche Informationen zu übermitteln.

Digitales Fernsehen: Bezeichnung von Fernsehen, bei dem die Bild- und Tondaten digital über Satellit, Kabel oder Antenne übertragen werden.

DRC: Sonys Algorithmus zur Verarbeitung digitaler Signale. Dabei werden horizontale und vertikale Bildinformationen in Echtzeit verdoppelt, um klare und präzisere Bilder zu erzielen. Mit einer bis zu vierfach höheren Auflösung sind die empfangenen Bilder detaillierter als die der ursprünglichen Bildquelle.

Drehbuch: Ein Drehbuch ist eine inhaltliche, grafische Beschreibung und eine Beschreibung des Handlungsablaufes eines Multimediasystems.

DVB: *DVB* ist die Abkürzung für *Digital Video Broadcasting*. Hier handelt es sich allgemein um das Verfahren zur Verbreitung von Digitalem Fernsehen, welches vom DVB-Projekt entwickelt wurde. Verbesserte übertragungstechniken sowie Fehlererkennungs- und -korrekturverfahren sorgen für eine bessere Bild- und Tonqualität (DVD-Qualität und Mehrkanalton). Aufgrund des Einsatzes von Datenreduktionsverfahren passen in einen analogen übertragungskanal etwa vier digitale Programme.

DVB-C: *DVB-C* ist ein auf das Kabelfernsehen abgestimmter DVB-Standard, d. h. die Daten werden über das normale Kabelnetz verbreitet. Für den Empfang benötigt der Zuschauer einen Kabelanschluss und eine DVB-C-taugliche Set-Top-Box.

DVB-S: *DVB-S* ist ein auf das Satellitenfernsehen abgestimmter DVB-Standard. Die Daten werden über die bestehenden Satelliten verbreitet, für den Empfang benötigt der Zuschauer einen Satelliten-Schüssel und eine DVB-S-taugliche Set-Top-Box.

DVB-T: *DVB-T* ist ein DVB-Standard, welcher die Verbreitung von Digitalem Fernsehen über terrestrische Sender ermöglicht. Für den DVB-T Empfang benötigt der Zuschauer eine Antenne (je nach Signalstärke eine kleine Stabantenne, Zimmerantenne oder auch Dachantenne) und eine DVB-T-taugliche Set-Top-Box.

DSL: steht für *Digital Subscriber Line* und stellt ähnlich wie ein Modem eine Verbindung zum Internet oder anderen Diensten über das Telefonnetz her. Da beim DSL mehr Frequenzbereiche benutzt werden, steht eine wesentlich höhere Bandbreite zur Verfügung.

DSM-CC: Abkürzung für *Digital Storage Media Command and Control*. Definiert ein Protokoll für Sitzungen zwischen unterschiedlichen Netzwerken und remote control eines Servers mit MPEG-2-Inhalten.

EPG: Der Begriff *EPG* steht für *electronic program guide* bzw. *elektronischer Programmführer*. Beim Digitalen Fernsehen werden neben den Video- und Audiodaten einige Informationen über das laufende Programm zum Zuschauer übertragen. Set-Top-Boxen verfügen in der Regel werksseitig über einen EPG, welcher die vorliegenden Informationen auf dem Bildschirm darstellt. Dies sind meistens kurze Informationen zum laufenden Programm oder zur jeweils nachfolgenden Sendung eines Senders. Der *EPG*, der von der Projektgruppe implementiert werden soll, geht über die beschriebene und übliche Funktionalität hinaus.

Evolutionäre Entwicklung: Zuerst wird eine Nullversion des Softwaresystems erstellt, die den wichtigsten Anforderungen entspricht. Diese Nullversion wird dann schrittweise unter Einbeziehung und Rücksprache mit dem Kunden bzw. den Anwendern zu dem gewünschten Softwaresystem weiterentwickelt.

Funktionale Anforderungen: Funktionale Anforderungen beschreiben die dem Benutzer angebotene Funktionalität des Softwaresystems.

GUI: Englische Abkürzung für den Begriff *Graphical User Interface*. GUI bedeutet im Deutschen *Graphische Benutzungsoberfläche* und bezeichnet die visuelle Programm-Oberfläche, die aus Bildelementen besteht. Das GUI ist die *Benutzungsschnittstelle zwischen Anwender und Applikation*.

***Hello-World* -Prototyp:** Ein kleines Programm, welches den kurzen Satz *Hello World!* ausgibt. Es wird traditionell zu Testzwecken geschrieben, um sich mit der grundsätzlichen Struktur und den elementaren Möglichkeiten einer Programmiersprache vertraut zu machen.

IP-Streaming: Das versenden von Audio und Videodateien zu PCs oder Fernsehern über IP.

IRT: Abkürzung für *Institut für Rundfunktechnik*. Das IRT hat seinen Sitz in Braunschweig und ist der Hersteller der Referenzimplementierung für MHP Applikationen.

Java: Objektorientierte Programmiersprache (OOP) der Firma Sun Microsystems, die besonders geeignet ist zur Entwicklung von interaktiven

Programmen (mit Grafiken, Animationen etc.) innerhalb von Webseiten (World Wide Web). Das Besondere an Java-Programmen ist, dass sie unabhängig vom jeweiligen Betriebssystem laufen, also z. B. gleichermaßen auf Apple-Computern wie auf Windows-PCs.

Java-AWT: Das *Java Abstract Window Toolkit* ist eine umfangreiche Java-Klassenbibliothek zur Gestaltung von graphischen Benutzungsoberflächen.

Linux: Frei verfügbares Betriebssystem für PCs.

LVDS: Steht für *Low Voltage Differential Technology* und ist ein differentialer Datenübertragungsstandard über Kupferkabel mit einer übertragungsrate von bis zu 1.2 Gbps.

MHEG: steht für Multimedia and Hypermedia coding Expert Group und ist ein internationaler Standard (ISO/IEC 13522-5/6/7) zum Kodieren und Dekodieren von Multimedia -Präsentationen. Sie ist eine Art Beschreibungssprache für multimediale Präsentationen.

Meilenstein: Ein Meilenstein ist ein feststehender Termin; oft repräsentiert durch ein fertig gestelltes Dokument oder Software, die in der vorhergehenden Phase oder Iteration erstellt worden ist.

MPEG-2: Die Abkürzung *MPEG* steht für *motion picture experts group.* Dabei handelt es sich im Zusammenhang mit DVB um ein Datenkompressionsverfahren, welches bei der Verbreitung von Digitalem Fernsehen benutzt wird. Eingesetzt wird die zweite Norm dieses Verfahrens (MPEG-2), dabei wird die anfallende Datenmenge vor der übertragung zum Zuschauer um durchschnittlich 96% reduziert. Das MPEG-2 - Verfahren ist daher eine wichtige Voraussetzung für das Digitale Fernsehen, denn ohne eine Datenkomprimierung würden die vorhandenen übertragungskapazitäten nicht ausreichen.

Multimedia: Multimedia alle computergestützen interaktive Online-,Offline - Medien- und Kommunikationsprodukte die mindestens drei Darstellungsformen (z.B. Text,Bild,Bewegtbild und Ton) beinhalten. Zusammengesetzter Begriff aus den englischen Begriffen "multi"[mehrfach] und "media"[Medien].Kombination bzw Verwendung mehrerer Medien. Medien lassen sich in Bezug auf ihre zeitliche Komponente in zwei Klassen unterteilen: Ein Medium wird als diskret oder zeitunabhängig bezeichnet, wenn seine Verarbeitung zeitunkritisch ist (Text, Graphik, Bilder). Ein Medium wird als kontinuierlich oder zeitabhängig bezeichnet, wenn seine darzustellenden Informationen nicht nur durch ihre

Werte vermittelt werden, sondern auch durch den zeitlichen Verlauf ihres Auftretens (Video, Animation, Audio).

Multimedia Home Platform (MHP): Die Multimedia Home Platform ist eine einheitliche, offene Schnittstelle zur Entwicklung von interaktiven, multimedialen Mehrwertdiensten für digitale TV-Programme und Dienste.

NFS: Abkürzung für *Network File System*. Erlaubt das simulieren von Festplattenpartitionen für ferngesteuerte (*remote*) Maschinen (z.B. die TU-MSF100) als wenn sie eine eigene lokale Festplatte hätten.

Nichtfunktionale Anforderungen: Nichtfunktionale Anforderungen beschreiben Anforderungen eines Benutzers, die nicht zu der eigentlichen Funktionalität des Softwaresystems zählen, wie z.B. Zuverlässigkeit, Effizienz, Portabilität der Software.

OpenTV: ist das Betriebssystem der Set - Top - Box. Das Betriebssystem ermöglicht dem Receiver, interaktive Anwendungen darzustellen. Es basiert auf den bekannten Programmiersprachen HTML und Java. Zudem ist OpenTV eine kommerzielle Plattform.

Objektkarussell: ist Bestandteil des Play-Out-Centers, Anwendungen und Daten werden in das *Objektkarussell* eingestellt und zyklisch in den *MPEG-2*-Datenstrom eingefügt. Bei Bedarf können diese von der Set-Top-Box jederzeit ohne lange Wartezeiten geladen und gespeichert werden.

Playout-Center: Es ist eine Funktionseinheit, in der Geräte zur Codierung, Multiplex-Bildung, Modulation und Satelliten-Ausstrahlung von digitalen Signalen enthalten sind. Die Sendeabwicklung und -überwachung der einzelnen Programme wird von Playout-Center vorgenommen.

Programmieren im Großen: Als Programmieren im Großen wird die Integration kleinerer Softwareeinheiten, sogenannte Softwaremodule oder -komponenten, zu einem Softwaresystem bezeichnet.

Programmieren im Kleinen: Als Programmieren im Kleinen wird die Implementation kleinerer Softwareeinheiten, sogenannte Softwaremodule oder -komponenten, bezeichnet.

Projektgruppe (PG): Eine Projektgruppe besteht an der Universität Oldenburg in der Regel aus sechs bis zwölf Teilnehmerinnen und Teilnehmern, die gemeinsam ein komplettes Projekt im Umfang von vier

Modulen (jeweils zwei pro Semester) durchführen. Ziel ist es, anhand eines gegebenen Problems die vollständige Entwicklung von der Problemanalyse bis hin zur Realisierung des Systems durchzuführen.

Proprietär: Herstellerspezifische, nicht-öffentliche Techniken und Verfahren.

Prototyp: Ein Prototyp ist ein ablauffähiges Modell des zukünftigen Produkts, um die Umsetzung von Anforderungen und Entwürfen in Software zu demonstrieren und mit ihnen zu experimentieren.

PVR: *PVR* steht für *personal video recorder*. Dies ist ein Ersatz für den VHS-Recorder und nimmt TV-Sendungen auf eine Festplatte auf. Zu den üblichen Features gehört die Wahl der Aufnahmequalität, die Verwaltung der Aufnahmen und eine schnelle Rück- und Vorspulfunktion bei der Wiedergabe. Damit der Begriff *personal* an Bedeutung gewinnt, könnten *PVRs* in Zukunft Sendungen automatisch aufzeichnen, die den Zuschauer interessieren.

Redesign: Eine Phase im Entwicklungsprozess von Software, in der Entwurfsentscheidungen, evt. bis hin zu sehr grundsätzlichen, erneut überdacht, einer Prüfung unterzogen und evt. geändert werden.

Referenzimplementierung: Implementiert den MHP Standard exakt und läuft auf Pentium-kompatiblen Plattformen unter Windows oder Linux. Kann auf andere Plattformen in Verbindung mit geeigneten Java Virtual Machines und plattformspezifischen Anpassungen portiert werden und ist in Java programmiert. Die MHP-Referenzimplementierung ist zudem parametrisierbar, d.h. die sie kann in Produkte, wie z.B. in Set-Top-Boxen, integriert werden und dient dort als das MHP-Betriebssystem des jeweiligen Gerätes.

Receiver: Gerät zur Umwandlung von, z.B. Satellitensignalen in analoge Signale für den Fernseher.

RS232: Bidirektionale Schnittstelle in Form einer 9 oder 25 poligen Buchse. Signale können gleichzeitig von jeder Station zu der jeweils anderen geschickt werden: So kann z.B. ein Gerät gleichzeitig Messdaten übertragen, während es verschiedene Steuerungsparameter vom lesenden Computer erhält.

Rolle: Beschreibt die notwendigen Erfahrungen, Kenntnisse und Fähigkeiten, über die ein Mitarbeiter verfügen muss, um eine bestimmte Aktivität durchzuführen.

Rollenkonzept: Das Rollenkonzept ist ein dynamisches Modell der Aufgabenverteilung, in dem die verschiedenen Aktivitäten ↑Rollen und keinen Mitarbeitern zugewiesen werden.

Rückkanal: Es ist ein bidirektionaler Kanal, d. h. man kann sowohl Daten empfangen als auch senden. Innerhalb der ↑MHP dient es dazu, Informationen zum Diensteanbieter zurückzusenden und ermöglicht damit echte Interaktivität. Wird bei einer stationären ↑Set-Top-Box zumeist durch ein analoges Modem oder DSL, bei ↑DVB-T dagegen, damit die Mobilität nicht verloren geht, durch Umts oder GPRS, realisiert.

Workstation: Englische Bezeichnung für einen Rechner der meist zum Arbeiten eingesetzt wird.

SDK: Abkürzung für *Software Development Kit*. Umfassender Begriff für Entwicklungsumgebung, enthält meist alle zum Erstellen von Anwendungen notwendigen Programme und Bibliotheken.

Server: Rechner oder Programm, das in einem Netz zentrale Dienste zur Verfügung stellt.

Set-Top-Box: Zusatzgerät einschließlich Decoder zum Empfang digitaler Fernsehprogramme. Eine *Set-Top-Box* kann allen vorhandenen Fernsehgeräten vorgeschaltet werden, so dass sie auch für das digitale Fernsehen genutzt werden können.

Schnittstelle: Mit *Schnittstelle* bezeichnet man sowohl den Verbindungspunkt zwischen zwei Hardware-Systemen (Geräten, Gerätekomponenten) als auch die übergabestelle zwischen zwei Software-Systemen (zum Beispiel Anwendungsprogrammen, Datenübertragungsprotokollen, Datenbanken).

Softwarearchitektur: Eine Softwarearchitektur beschreibt die Struktur eines Softwaresystems durch Systemkomponenten und Beziehungen zwischen den Systemkomponenten.

Transport Stream: Einer von zwei Datenströmen, die von MPEG-2-Systemen vorgegeben werden. Ein Transport Stream (TS) besteht aus 188-Byte-Paketen.

Trailer: Ein Kurzfilm, der den Versuch unternimmt, dem Zuschauer einen Eindruck vom eigentlichen Film zu vermitteln. Dieser wird häufig als Werbung für den Film verwendet.

Unified Modeling Language (UML): Die Unified Modeling Language ist eine einheitliche Modellierungssprache für die objektorientierte Softwareentwicklung.

Use-Case (Anwendungsfall): Use-Cases sind Modellierungselemente der ↑UML und beschreiben die ↑funktionalen Anforderungen eines Softwaresystems.

B Seminararbeiten

Die Seminararbeiten befinden sich zudem auf der PG-CD. Im folgenden werden die Themengebiete der Seminararbeiten aufgelistet und kurz erklärt:

- DVB und die MHP (Carsten Zimmermann): Zusammenhang zwischen den DVB Empfangsstandards und der MHP
- Die Multimedia Home Platform (Martin Willkomm): Genaue Definition und Erläuterung der Eigenschaften bzw. Funktionalitäten der MHP
- Multimedia on TV (Jan Schoof): Vergleich von Standards wie MHEG und OpenTV mit der MHP
- Prozessmodelle zur Multimedia/Software-Entwicklung (Sascha Heider): Vorstellung verschiedener Modelle zur Softwareentwicklung
- Marktforschung Hardware (Andre Dromowicz): Analyse der sich auf dem Markt befindlichen MHP fähigen Hardware
- Marktforschung Software (Sebastian Stockfleth): Analyse der sich auf dem Markt befindlichen Software zur Entwicklung von auf der MHP basierenden Applikationen
- Personalisierung und Multimedia auf dem Fernseher (Christian Wegener): Übersichtsarbeit zu den Themenkomplexen Personalisierung und Multimedia mit dem Fokus auf Anwendungen und Dienste für den Fernseher.
- Geschäftsmodelle für Multimedia on TV / eCommerce (Michael Hasseler): Vorstellung verschiedener Geschäftsmodelle
- Sicherheit mit der MHP (Daniel Thobe): Erläuterung von sicherheitsrelevanten Aspekten die MHP betreffend