

Thema

„Public Screen als Informationsplattform“

Vorlesung

Programmierung mobiler Geräte in Java

Gruppenmitglieder

Esther Burger: Projektleitung und Programmierung

Michael Kaiser: Programmierung

Nadine Fendt: Design

Sandra Richter: Design

Inhaltsverzeichnis

I. Einführung in das Thema

1. Situation
2. Projektbeschreibung
3. Rahmenbedingungen

II. Bereich „Information“ – Allgemeines

1. Anforderungsanalyse und –bewertung
2. Inhaltliche Entscheidungen
3. Technologieeinsatz

III. Bereich „Information“ – Design

1. Screen-Format und -Aufteilung
2. Corporate Design

IV. Bereich „Information“ – Programmierung

1. Übersicht
2. Implementierung
 - Veranstaltungen
 - Gebäudeplan
 - Voting
 - Benutzungshinweise
 - Informationsmenü
3. Integration
4. Test
5. Probleme und Mängel

V. Resümee

1. Kritische Reflektion
2. Ausblick

VI. Anhang

I. Einführung in das Thema

1. Situation

Autoren: Burger, Kaiser

Die mobilen Geräte der jüngsten Zeit bieten über ihre eigentliche Funktionalität hinaus eine große Anzahl an technischen Möglichkeiten. Auf Grund dessen ist eine der größten Herausforderungen für Softwareentwickler, dieses Potential durch die Entwicklung neuer Anwendungen zu erschließen.

Die Veranstaltung „Programmierung mobiler Geräte in Java“ bietet den Studenten der Studiengänge Informatik, Wirtschaftsinformatik, Multimedia und Kommunikationsdesign im Rahmen einer Projektarbeit die Möglichkeit, einen Einblick in die Anwendungs-Entwicklung für mobilen Endgeräten zu bekommen, und zusätzlich auch noch Erfahrungen in den Bereichen Projektmanagement und Software-Implementierung zu sammeln.

2. Projektbeschreibung

Autoren: Burger, Kaiser

Bei dem Projekt im Sommersemester 2006 sollten die verschiedenen Einsatzbereiche und Techniken untersucht werden, wie die in Mobiltelefonen verfügbaren standardisierten Schnittstellen zum Datenaustausch genutzt werden können, um mit einem (Informations-)Terminal zu kommunizieren.

Als konkreter Anwendungsfall wurde deshalb für das gerade neu entstehende Gebäude der Fachbereiche Informatik und Gestaltung die Installation eines sogenannten „Public Screen“ geplant. Dieser soll zukünftig im Vorraum der neuen Mensa platziert werden und den Nutzern die Möglichkeit geben, über verschiedene Anwendungen auf ihren Mobiltelefonen mit der Anzeige zu interagieren.

3. Rahmenbedingungen

Autoren: Burger, Kaiser

Um den unter Punkt 2 („2. *Projektbeschreibung*“) beschriebenen Anwendungsfall bis zum Ende des Semesters umsetzen zu können, mussten zunächst einige wichtige Punkte geklärt werden. Bei diesen handelte es sich um die Festlegung des idealen Installationsorts, die Bestimmung der Zielgruppe, die Einigung auf gewisse technische Standards und die Einteilung der Studenten in einzelne Projektteams.

So hat sich beispielsweise der Vorraum der neuen Mensa als Installationsort für den Public Screens bestätigt, da dieser zukünftig eine zentrale Anlaufstelle für die Studenten der FH Augsburg darstellen wird und deshalb optimal dafür geeignet ist, eine große Nutzergruppe anzuziehen.

Diese Nutzer- oder auch Zielgruppe wird nach Meinung der am Projekt beteiligten Personen sowohl aus Studenten, als auch aus Besuchern der Fachhochschule bestehen.

Als technische Rahmenbedingungen wurde auf Serverseite, also auf Seite des Public Screen J2SE, auf Clientseite (Mobiltelefon) J2ME und als Schnittstelle zum Datenaustausch Bluetooth festgelegt.

Das Gesamtprojekt wurde in dieser ersten Projektphase in vier Themenbereiche untergliedert, für die jeweils ein Team aus drei bis fünf Studenten zuständig waren:

1. Bereich „Information“:

Konzeptfindung und softwaretechnische Realisierung in Bezug auf die Nutzung des Systems als Informationsquelle.

2. Bereich „Spiele“:

Konzeptfindung und softwaretechnische Realisierung in Bezug auf die Nutzung des Systems als Spieleplattform.

3. Bereich „Multimedia“:

Konzeptfindung und softwaretechnische Realisierung in Bezug auf die multimedialen Nutzungsmöglichkeiten des „Public Screen“.

4. Bereich „Technik“:

Schaffung eines Frameworks zur Bereitstellung von Basisfunktionalitäten und zur einfachen Integration der verschiedenen Team-Applikationen.

Die folgende Abhandlung beschäftigt sich mit dem Bereich „Information“.

II. Bereich „Information“ – Allgemeines

1. Anforderungsanalyse und -bewertung

Autoren: Burger, Kaiser

Um einen Überblick über die Anforderungen an des System in der Rolle einer Informationsquelle für Studenten und Besucher zu erhalten, wurden zunächst Ideen gesammelt. Diese ließen sich im wesentlichen in die Kategorien Informationen zum Download aufs Handy und Information zur Darstellung auf dem Public Screen unterteilen. Im Folgenden ist das Ergebnis der ersten Anforderungssammlung ersichtlich:

1. Ein Benutzer soll in der Lage sein im räumlichen Umkreis von einigen Metern per Handy eine Verbindung zum Rechner des Public Screen herzustellen, die den Download von
 - aktuellen Stundenplänen
 - fachbereichsbezogenen Veranstaltungsterminen
 - hochschulweiten Veranstaltungsterminen
 - kulturellen Veranstaltungsterminen (z.B. Party, Theater, Filmclub)
 - Informationen über Vorlesungsausfälle oder Raumänderungen ermöglicht.
2. Ein Benutzer soll in der Lage sein nach der Herstellung einer Bluetooth

Verbindung, die angezeigten Inhalte auf dem Public Screen zu bestimmen.

Dieses Feature soll für die Ausgabe von

- Anzeigen aller Art
 - aller oben genannten Inhalte
 - einer interaktiven Gebäudeführung
- verwendet werden.

Nach der Sammlung der Ideen und Anforderungen erfolgte deren Bewertung. Für die Bewertung spielen folgende Kriterien eine Rolle:

- Korrektheit
- Machbarkeit
- Notwendigkeit (gefordert)
- Priorität
- Nutzbarkeit, Nützlichkeit
- Benutzerfreundlichkeit

Bei der Überprüfung der Anforderungen anhand der genannten Kriterien (siehe Abb.1) stellte sich heraus, dass der Public Screen als reines Download-Center gewisse Nachteile aufweist.

	korrekt	machbar	notwendig	priorisiert	nutzbar / nützlich	benutzer- freundlich
Download von Informationen	ja	ja	ja	nein	nein	nein
Anzeige von Informationen	ja	ja	ja	nein	nein	ja

Abb.1: Bewertung der Anforderungen

Dies lässt sich dadurch begründen, dass an der FH Augsburg die Versorgung mit aktuellen Informationen ausreichend über bestehende Systeme wie der FH-Webseite, häufig benutzten Mailverteilern, und herkömmlichen Aushängen gegeben ist. Der Aktualitätsgrad der meisten Informationen erfordert es zudem nicht, dass Studenten während des Aufenthalts an der FH laufend informiert werden müssten.

Durch die geringen Ausmaße von Handydisplays und dem relativ langsamen Verbindungsaufbau mit Bluetooth gestaltet sich außerdem die Nutzung bestehender Informationssysteme (Webseite, E-Mail am Rechner) wesentlich komfortabler.

Und auch die Idee der Nutzung des Public Screens als reinen Informationsdarstellungs-Bildschirm schien bei kritischer Betrachtung zunächst wenig überzeugend, da die Anzeige für einen zu langen Zeitraum mit Darstellungen belegt wäre, die für Dritte völlig uninteressant sind. Dies widerspräche somit der Grundidee des Systems, mehrere Menschen gleichzeitig anzusprechen.

Die einzige Ausnahme stellte deshalb zunächst die Idee der Implementierung eines interaktiven 3D-Gebäudeplans dar. Durch eine schnelle und komfortable Menüführung sollte die Benutzungsdauer durch einzelnen Anwender minimiert werden und die Applikation sollte zusätzlich graphisch so ansprechend gestaltet sein, dass auch unbeteiligte Zuschauer unterhalten werden.

Im Rahmen dessen sollte den Studenten und Besuchern der FH Augsburg die gewünschten Informationen zum einen auf eine spielerische Art und Weise nahe gebracht werden, und zum anderen sollten sie auch gleichzeitig eine räumliche Vorstellung davon bekommen, wo sich die Informationsquelle, Institution etc. konkret befinden.

Dadurch sollte der Bereich „Information“ eine Entwicklung weg von der Umsetzung als reine Informationsplattform, und hin zum ansprechenden „interaktiven Spielzeug“, das für ein breites Publikum ansprechend ist, durchlaufen.

Die Idee einer interaktiven Gebäudeführung im 3D-Format wurde allerdings in der Diskussion mit den Professoren und den anderen Projektteams als zu unübersichtlich eingestuft. Des weiteren wurde in der Diskussion erwähnt, dass die Downloadmöglichkeit doch noch einmal überdacht werden sollte und auch die Realisierung eines Votings in Betracht gezogen werden sollte.

Deshalb wurden die Projektthemen von Seiten des Teams „Information“ noch mal überarbeitet und einige Entscheidungen in diesem Bereich getroffen (siehe „2. Inhaltliche Entscheidungen“).

2. Inhaltliche Entscheidungen

Autor: Burger

Auf Basis der aus den Diskussionen mit den Professoren und Kommilitonen gewonnenen Anregungen wurden teamintern weitere Analysen und Überlegungen durchgeführt, die letztendlich zu folgenden Ergebnissen führten:

1. Realisierung eines 2D-Gebäudeplans

- Kurzbeschreibung:

Bietet die Möglichkeit nach bestimmten Institutionen und FH-spezifischen Suchbegriffen zu suchen, sich eine ausführliche Wegbeschreibung auf dem Public Screen anzusehen und diese Beschreibung sowie weitere Informationen anschließend auf ein Mobiltelefon herunterzuladen.

- Beweggründe:

Hilfestellung v.a. für Studienanfänger, um sich an der FH Augsburg und deren Institutionen einfacher und schneller zurecht zu finden.

2. Realisierung einer Downloadmöglichkeit von Veranstaltungen

- Kurzbeschreibung:

Bietet die Möglichkeit sich eine Liste von aktuellen Veranstaltungen herunterzuladen oder eigene Veranstaltungen einzustellen.

- Beweggründe:

Näherbringen des Veranstaltungsangebot auf einen Blick, sowie Aufmerksammachen auf Veranstaltungen, die anderenfalls evtl. keine Beachtung finden würden

3. Realisierung eines Votings

- Kurzbeschreibung:

Möglichkeit in bestimmten Intervallen für ein Voting-Thema abzustimmen sowie eigene Voting-Vorschläge einzubringen

- Beweggründe:

Bietet den Studenten die Möglichkeit zu den verschiedensten Themen ihre Meinung zu äußern

4. Realisierung von Benutzungshinweise

- Kurzbeschreibung:

Erläuterung der Nutzungsmöglichkeiten des Public Screen durch eine Animation auf dem Screen

- Beweggründe:

Erleichterung des ersten Kontakts mit dem System

Eine detailliertere Beschreibung der einzelnen Unterbereiche erfolgt im Abschnitt „IV. Bereich „Information“ – Programmierung“.

3. Technologieeinsatz

Autor: Burger

Wie bereits in Punkt 3 („3. Rahmenbedingungen“) in Abschnitt 1 erläutert, wurden zu Beginn des Projekts einige technische Rahmenbedingungen festgelegt. Im weiteren Projektverlauf sind dann noch einige teamübergreifende sowie teaminterne Technologien hinzugekommen, auf die im Folgenden ebenfalls ausführlicher eingegangen wird.

Zu Beginn des Projekts „Public Screen“ wurde u.a. entschieden, dass auf Clientseite J2ME eingesetzt wird. J2ME steht für Java Micro Edition und ist eine „abgespeckte“ Version der Programmiersprache Java, die zur Entwicklung von Applikationen für sogenannte „embedded devices“, wie Mobiltelefone, PDAs u.ä. eingesetzt wird. Durch seine unterschiedlichen Konfigurationsmöglichkeiten und Profile kann es für ein breites Spektrum an mobilen Endgeräten verwendet werden (weitere Informationen siehe <http://java.sun.com/javame/index.jsp>).

Zur Erleichterung der Entwicklung von J2ME-Applikationen stellt Sun Microsystems ein sogenanntes Wireless Toolkit zur Verfügung. Dabei handelt es sich um ein leichtgewichtiges Werkzeug, das beispielsweise verschiedene Emulatoren sowie Performance-Optimizer and Tuning-Features bereitstellt. Durch diese Möglichkeit der Simulation des Verhaltens von Mobiltelefonen auf dem PC wird, wie bereits erwähnt, die Applikations-Entwicklung erheblich effizienter und leichter. Im Rahmen des Projekts „Public Screen“ wurde das Wireless Toolkit in der Version 2.2 verwendet (weitere Informationen siehe <http://java.sun.com/products/sjwtoolkit/overview.html>).

Des Weiteren wird das Entwickeln für J2ME von vielen IDE-Herstellern durch eigene Module unterstützt. So gibt es für die Entwicklungsumgebung Eclipse ein Plugin, welches das Einbinden des J2ME Wireless Toolkit in die Umgebung ermöglicht (siehe <http://eclipseme.org/>). Dadurch konnten sich die einzelnen Projektteams verstärkt auf die eigentliche Entwicklungsarbeit konzentrieren, ohne sich um die spezifischen Anforderungen von J2ME im Detail kümmern zu müssen.

Innerhalb des Teams „Information“ stand des Weiteren zur Diskussion, ob zur Gestaltung der Benutzeroberflächen auf dem Mobiltelefon die High-Level-API, die Low-Level-API oder J2ME Polish eingesetzt werden soll.

Mit J2ME Polish kann die Benutzeroberfläche eines jeden Mobiltelefon-Modells mit Hilfe einer einfachen CSS-Datei an dessen spezifisches Oberflächen-Design angepasst werden und ermöglicht damit die eigene Applikation stimmig in das Erscheinungsbild zu integrieren. J2ME Polish kann wie das Wireless Toolkit direkt in Eclipse eingebunden werden (siehe <http://www.j2mepolish.org/>). Die Entscheidung fiel dann allerdings gegen Polish aus, da zum einen die gestalterischen Anforderungen für die „Informations“-Applikation nicht so anspruchsvoll waren, als dass sie die Einrichtung von verschiedenen CSS-Dateien für die vielen verschiedenen Mobiltelefon-Modelle gerechtfertigt hätten. Zum anderen hätte das Projektteam, das sich mit der Schaffung des technischen Frameworks beschäftigt hat, ihre komplette Struktur im Nachhinein auf Polish ausrichten müssen, was ebenfalls einen zu hohen Aufwand erfordert hätte. Deshalb wurde dann teamintern beschlossen die High-Level-API zu verwenden, da diese im Gegensatz zu Polish oder auch zur Low-Level-API auf jedem beliebigen Mobiltelefon-Modell lauffähig ist.

Für die Server-Seite wurde bei Projektstart festgelegt, dass für deren Applikationsentwicklung die Java Standard Edition (J2SE) verwendet wird. Für die grafischen Darstellungen der einzelnen Teams auf dem Public Screen war zunächst ein mit Java Swing realisierte Lösung angedacht. Diese wurde aber im Laufe des Projekts verworfen und stattdessen wurde eine Lösung mit Flash bevorzugt, da die von den Bereichen „Spiele“ und „Multimedia“ benötigten Gestaltungsanforderungen mit Flash viel einfacher zu realisieren waren.

Wie aus den Ausführungen in Abschnitt 4 („IV. Bereich „Information“ – Programmierung“) ersichtlich werden wird, wurde des Weiteren auf Server-Seite für

die persistente Datenhaltung in den Unterbereichen Gebäudeplan, Veranstaltungen und Voting eine Datenbank benötigt. Da das Team „Technik“ ihre Datenbank („Hypersonic“) aus zeittechnischen Gründen den anderen Teams leider nicht rechtzeitig zur Verfügung stellen konnte, wurde von unserer Seite zu Testzwecken eine MySQL-Datenbank eingerichtet. Die Entscheidung viel deshalb auf eine MySQL-Datenbank, weil diese einfach zu installieren und mit Hilfe von PhpMyAdmin auch sehr einfach einzurichten und zu administrieren ist.

III. Bereich „Information“ – Design

1. Screen-Format und -Aufteilung

Autor: Fendt

Für das Grundformat des Screens mussten wir uns zunächst über die verschiedenen Möglichkeiten der Darstellung informieren.

Als Screen-Systeme sind verschiedene möglich: LCD-, Plasma- und Rückprojektionsgeräte, alle ähnlich TV-Geräten. Diese sind allerdings durch max. Diagonale von 1,5m relativ klein für größere Benutzergruppen.

Geeigneter sind dagegen Beamer, da sie einfach größere Projektionen ermöglichen. Dabei ist hohe Lichtstärke zu beachten, da sonst durch Lichteinfall Störungen auftreten; ein wesentliches Kriterium für die Nutzbarkeit von Videoprojektoren bei größeren Projektionsabständen; was gleichbedeutend mit großer Projektionsflächen ist, ist die Helligkeit, die durch den Einsatz von Metalldampflampen erreicht wird. Handelsübliche Videoprojektoren verfügen über Leuchtstärken zwischen 1.000 und 3.000 ANSI-Lumen, große Modelle für Konferenzräume durchaus auch 4000 bis 10000 ANSI-Lumen.

Für eine gute Ausleuchtung nicht komplett abgedunkelter, größerer Konferenzräume sollte man sich deutlich am oberen Ende des Machbaren bewegen.

Unterschiede Beamertechnik

DLP (Spiegel+Farbrad)

- + kein Nachleuchten/Nachziehen des Bildes, kein Einbrennen, höherer Kontrast (durch das tiefere Schwarz), weniger stark ausgeprägte Pixelstruktur als LCD-Projektoren
- feste Auflösung des Eingangssignals erforderlich, "Regenbogeneffekte" bei Geräten mit Farbrad, wenn das Farbrad keine hohe Umdrehungsgeschwindigkeit hat (herstellerabhängig). bei Darstellung einzelner Grau-/ Farbwerte kann es zu sichtbaren Flimmern kommen, teilw. fehlende Farbtreue: Problem, sattes Grün, rot und orange darzustellen, Lüftergeräusch

LCD

- + relativ preiswert, klein und leicht,
- Einbrennen möglich, starke Pixelstruktur, Lüftergeräusch, teure Ersatzteile, feste Auflösung des Eingangssignals

Kosten

Die Preise variieren zwischen Herstellern und Händlern zum Teil deutlich. Deshalb ist ein Preisvergleich im Internet durchaus lohnenswert. Außerdem sind die Kosten für eine Ersatzleuchte erheblich, derzeit liegen sie zwischen 200 und 650 EUR. Eine Lampe hat eine Lebensdauer zwischen 1.000 und 4.000 Betriebsstunden. Selbst durch kleinere Erschütterungen kann diese erheblich verkürzt werden.

Bildschirmattribute

Produkt	Homepage	Technologie	Helligkeit Ansi-Lumen	Kontrast (max)	Bildschirmdiagonale		Seitenverhältnis		Bildgröße (cm)		Auflösung (pixel)		Pixelgröße (mm)	
					Zoll	cm	horizontal	vertikal	Breite	Höhe	horizontal	vertikal	Breite	Höhe
Jungmann CommandView CVC 84	http://jungmann.de	Rückprojektion	800	500:1	84,0	213,4	5	4	163,0	130,4	1280	1024	1,27	1,27
Sony VPL-VW100 *	www.sony.de	Aufprojektion (LCD)	800*	15000:1	100,0	254,0	16	9	221,4	124,5	1920	1080	1,15	1,15
Samsung PS-63P5H	www.samsung.de	Plasma	1000 cd/m2	3000:1	63,0	160,0	15	8	128,5	78,3	1366	768	1,02	1,02
30.000€ alternativ														
Sanyo PLC-XL15 (ohne Objektiv)	www.beamershop24.de	Aufprojektion (LCD)	7700	1000:1	109,0	254,0	4	3	203,2	152,4	1600	1200	1,27	1,27
Sanyo PLC-XF80A (ohne Objektiv)	www.beamershop24.de	Aufprojektion (LCD)	6500	1300:1	109,0	254,0	4	3	203,2	152,4	1400	1050	1,45	1,45
1. Wahl														
EIKI LC-XT3 (ohne Objektiv)	www.beamer.de	Aufprojektion (LCD)	10000	1000:1	109,0	254,0	4	3	203,2	152,4	1024	768	1,98	1,98
24h Betrieb														
Sharp XG-PH50XNL (ohne Objektiv)	www.beamer.de	Aufprojektion (DLP)	4000	1000:1	109,0	254,0	4	3	203,2	152,4	1024	768	1,98	1,98
Panasonic PT-D5500	www.beamer.de	Aufprojektion (DLP)	6000	1600:1	109,0	254,0	4	3	203,2	152,4	1024	768	1,98	1,98

* Bildschirmdiag. = 100" bei 3,1-5,3m Entf.

Abb.2: Übersicht über die technischen Daten unterschiedlicher Beamer

Grundformat und Auflösung

Für unsere Anwendungen gehen wir von dem Grundformatformat 1024 x 768 aus. Sollte der Beamer eine höhere Auflösung bieten, kann die Aufteilung proportional vergrößert angepasst werden.

Lösungsansätze Screen-Aufteilung

Varianten der räumlichen Darstellung.

Es stellte sich die Frage, wie der Screen von den drei Anwendungen (Aquarium, Spiel und Infobereich) genutzt werden kann.

1. Jede Anwendung erhält den gleichen Raum, egal, ob sie gerade aktiv oder inaktiv ist.

Vorteil: Alle Anwendungen gleichzeitig sichtbar

Nachteil: Aufteilung wirkt unübersichtlich, verwirrend; schlecht erkennbar

2. Aktive Anwendungen treten in den Vordergrund, inaktive sind nur verkleinert zu sehen. Dabei sind Anwendungen mit gleichen Status gleich groß.

Vorteil: Eindeutige Erkennbarkeit des Anwendungsstatus

Nachteil: Raum nicht optimal ausgenutzt, verschenkter Platz zu Lasten der aktiven Anwendungen, Skalierung problematisch

3. Aktive Anwendungen im Vordergrund, Inaktive „ausgegraut“; jede Anwendung mit

eigener Größe je nach Anforderungen.

Vorteil: Deutliche Statusdarstellung, jede Anwendung für Nutzer sichtbar, bessere Raumausnutzung

Nachteil: Immer noch „verschenkter“ Platz

4. Aktive Anwendungen im Vordergrund, inaktive verkleinert; dynamische Raumausnutzung.

Vorteil: Alle Anwendungen gleichzeitig sichtbar, gute Raumausnutzung

Nachteil: Eventuell Formatprobleme für die einzelnen Anwendungen, Problem vieler aktiver Fenster

5. Aktive Anwendungen im Vordergrund, Inaktive unsichtbar; Aufteilung des Screens in entsprechende Fenster.

Vorteil: Für Inaktive wird kein Platz verschwendet

Nachteil: Skalierung der Fenster = Formatänderungen! Bei vielen offenen Anwendungen kaum noch lesbar

6. Timeslots: Jede Anwendung erhält ihr eigenes „Zeitfenster“; beispielsweise wechselt alle 15 Minuten die aktive Anwendung, die anderen laufen im Hintergrund.

Vorteil: Jede Anwendung kann auf voller Fläche dargestellt werden; bessere Erkennbarkeit

Nachteil: Keine Abfrage nach Bedarf, Wartezeiten

7. Prioritäten: Die Anwendung, der die höchste Priorität eingeräumt wurde, kann andere laufende Anwendungen niederer Prioritäten unterbrechen, bzw. in den Hintergrund verdrängen.

Vorteil: Jede Anwendung kann auf voller Fläche dargestellt werden; bessere Erkennbarkeit

Nachteil: Unterbrechung anderer Nutzer

8. Mischformen:

Timeslot + feste Bereiche

Zunächst unterteilt man die Anwendung in 2 Bereiche:

Spiel und Information, die einen festen Bereich des Screens innehaben. Damit verhindert man zu kleine Darstellung der Anwendungen. Die beiden spielerischen Anwendungen, wechseln im 15min.-Takt, sofern für beide gleichzeitig Bedarf besteht. Der Informationsbereich deckt einen kleineren Raum ab, in dem die Bereiche je nach Anfrage dynamisch wechseln.

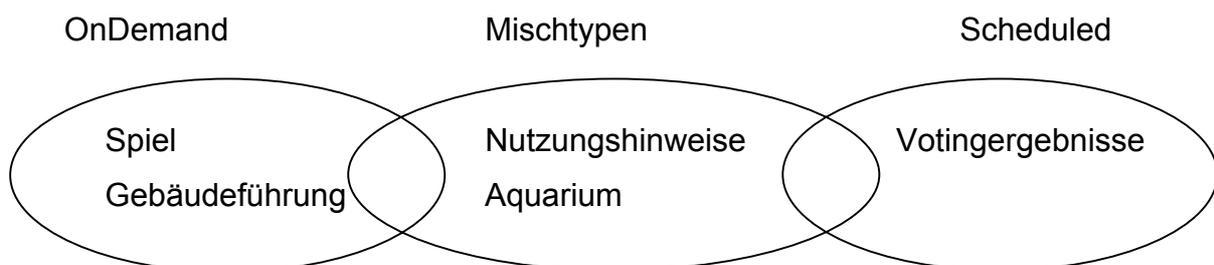
Vorteil: Keine Skalierung oder Verzerrung der Formate

Nachteil: Infobereich zu klein für Gebäudeführung und Benutzeranleitung; Anwendungen nicht fullscreen möglich

Kombination aus prioritätsbasiertem Scheduling und Timesharing

(Konzept M. Kaiser)

Einteilung in OnDemand- und Scheduled-Anwendungen



OnDemand-Anwendungen werden durch Benutzer gestartet, Scheduled-Anwendungen werden vom System gestartet, wenn keine Anforderung für eine OnDemand-Anwendung vorliegt. Jede Scheduled-Anwendung bekommt einen bestimmten Zeitslot. OnDemand-Anwendungen erhalten eine Priorität: Bei Anforderung einer höherpriorären Anwendung wird eine laufende Anwendung nach Ablauf eines Countdowns unterbrochen - jeder Anwendung wird jedoch eine Mindestlaufzeit auf dem Screen zugestanden; Rückmeldung: „nächste Anwendung in ... Sekunden“.

Vorteil: Jede Anwendung kann den ganzen Bildschirm ausnutzen

Nachteil: Zeitliche Einschränkungen

Eine Möglichkeit ist, jeder Anwendung zu gestatten, den ganzen Bildschirm auszunutzen, d.h. immer nur eine zu zeigen. Dabei müsste ein System erstellt werden, das den Zugriff regelt; so könnten sogenannte „Time-Slots“ periodische Zyklen für jede Anwendung bereitstellen. Eine Alternative wären unterschiedliche Prioritäten für die einzelnen Bereiche; so könnte beispielsweise der Zugriff auf Informationen höhere Priorität als Spiele genießen. Auch bei einer Mischform stellt sich die Frage nach der Regelung; welche Anwendung ist wichtiger? Benutzer der Spiele würden sich sicher gestört fühlen, wenn man „ihr“ Programm unterbricht; aber andererseits ist es genauso unsinnig, auf eine einfache Information 10 Minuten zu warten. Auf diese Weise würde man sicher viele potenzielle User vergraulen.

Die Überlegung, den Bereich gerade nicht genutzter Anwendungen dynamisch auf die anderen beiden zu verteilen, wurde verworfen, um den Benutzern jederzeit einen Überblick über das gesamte Angebot zu geben.

Daher fiel die Entscheidung schließlich auf die dritte Möglichkeit:

Jeder Anwendung wird ein bestimmter Bereich auf dem Screen zugewiesen. Alles läuft gleichzeitig ab. Wichtig ist dabei zu beachten, bei laufenden Animationen nicht einen zu überladenen Eindruck zu erwecken.

Aufteilung

Die Wahl fiel auf eine Aufteilung nach dem Prinzip des goldenen Schnittes.

Prinzip: Zwei Strecken stehen im Verhältnis des Goldenen Schnittes, wenn sich die größere zur kleineren verhält wie die Summe aus beiden zur größeren. Dieses Verhältnis wird meist mit dem griechischen Buchstaben Φ (Phi) bezeichnet. Bezeichnet man die längere Strecke mit a und die kürzere mit b, dann gilt damit

$$\Phi = a/b = (1 + \sqrt{5}) / 2 \approx 1,618033988\dots$$

Nach diesem Prinzip erfolgte nun die Screenaufteilung:

Die Gesamthöhe verhält zu der Höhe der größten Anwendung wie die Größte zur Nächstkleineren und diese wiederum zur Kleinsten.

Um die Anwendungen nicht direkt an die Nächste angrenzen zu lassen und dadurch entstehende harte Übergänge zu vermeiden, entschied ich mich für einen Zwischenraum.

Die Maße:

Ausgangsformat: 1024 x 768

Format der Anwendungen:

Breite: 994

Höhe: Aquarium 355, Info 218, Spiel 135

Zwischenräume 15

Die Verteilung stellt sich also auf dem Screen folgendermaßen dar:

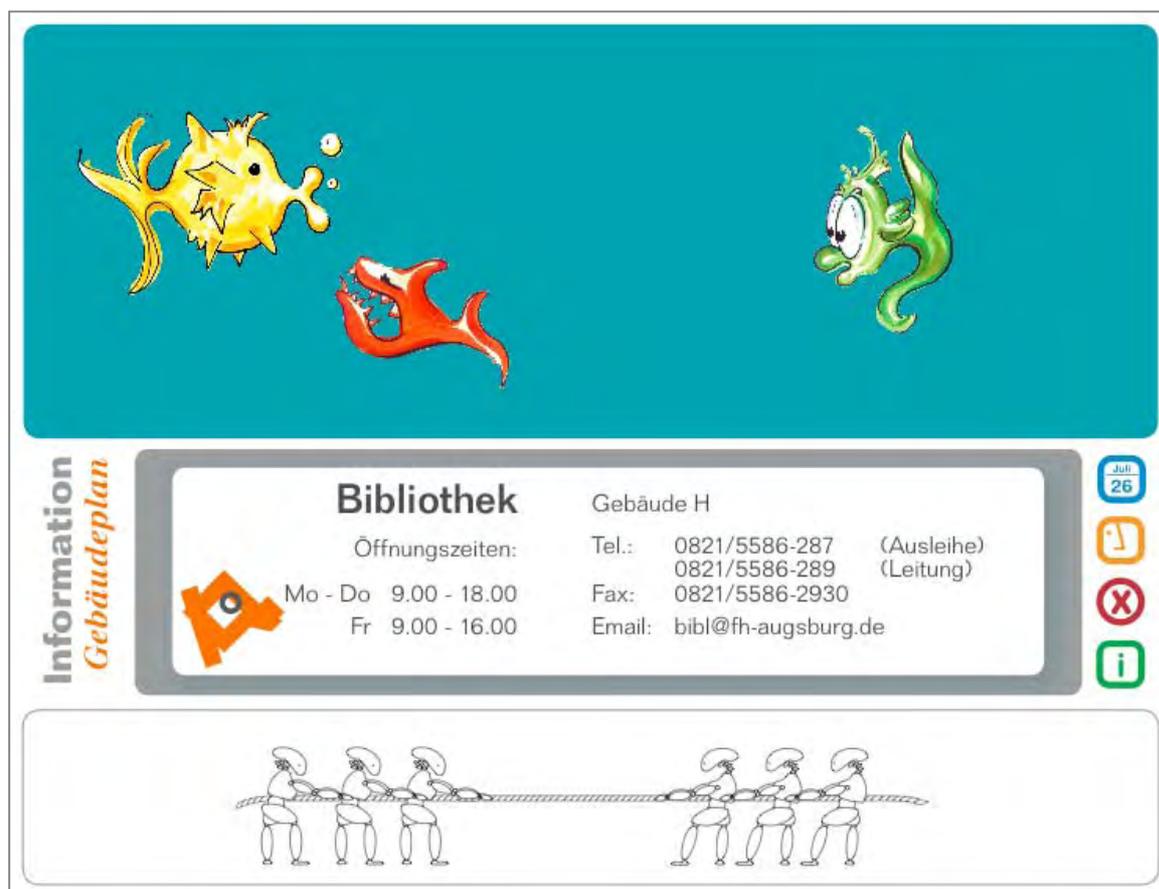


Abb.3: Beispiel für die Screen-Aufteilung

Raster

Um die verfügbare Fläche optimal auszunutzen, dabei aber nicht einen chaotischen Eindruck zu erwecken, ist ein Grundraster sehr hilfreich.

Zunächst waren die Versuche noch relativ grob und einfach, schließlich wurde es sehr fein gewählt. Basis ist nun ein Raster aus 30 Spalten und 25 Zeilen. Dies ermöglicht einen flexiblen Umgang mit den unterschiedlichen Inhalten.

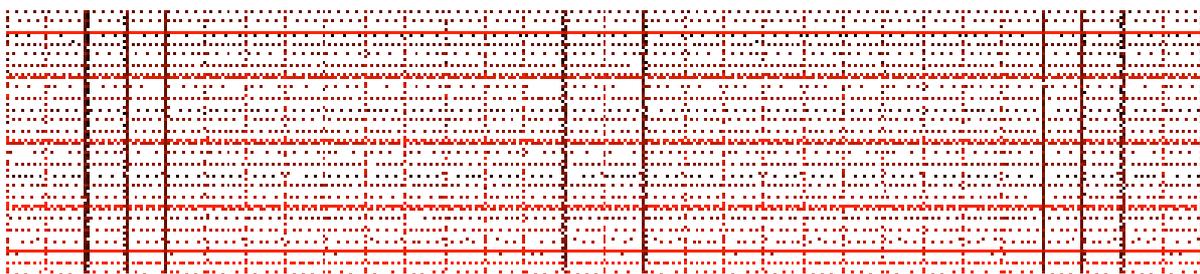


Abb.4: Raster



Abb.5: Umsetzung des Rasters „Gebäudeplan“

Als Konstante wird zur Orientierung auf der linken Seite der Bereichsname angezeigt; der Unterbereich hervorgehoben durch die zugehörige Farbe und Schrift abweichend von der Grundschrift (Berthold Walbaum book medium italic statt Folio). In der Mitte befindet sich der eigentliche Informationsbereich, eingerahmt durch einen grauen Fond, um ihn von den Anwendungen darüber bzw. darunter

abzugrenzen. Hier taucht das Icon des jeweiligen Bereichs in der linken unteren Ecke zur zusätzlichen Orientierung auf. Die Informationen werden jeweils einer erläuternden Grafik zugeordnet, so z.B. die Studentenkazlei und die Wegbeschreibung, bzw. der Gebäudegrundriss.

Auf der rechten Seite zeigt eine Leiste die Icons übereinander angeordnet, als Überblick über die weiteren Auswahlmöglichkeiten des Informationsbereiches.

Analog erfolgt die Darstellung für die anderen Bereiche:



Abb.6: Umsetzung des Rasters „Start“

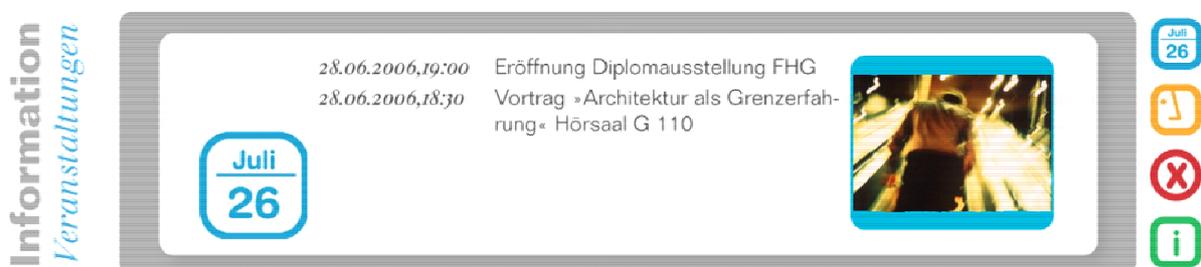


Abb.7: Umsetzung des Rasters „Veranstaltungen“

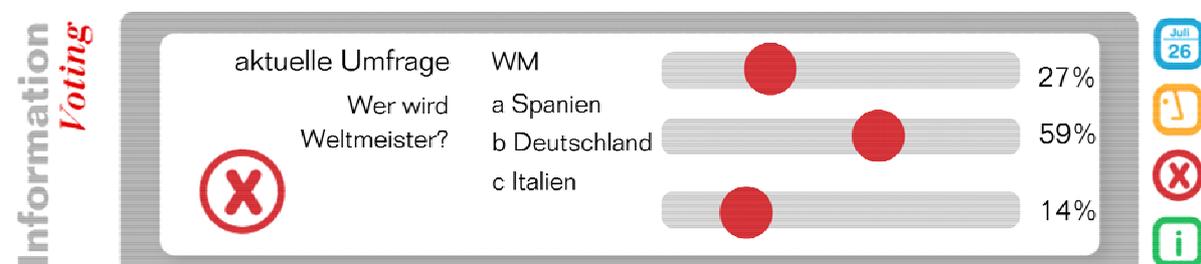


Abb.8: Umsetzung des Rasters „Voting“

Die Animationen sind möglichst einfach gehalten, um die Informationen ruhig und sachlich zu vermitteln und um nicht mit den anderen, spielerischen Anwendungen in Konkurrenz zu treten. Sie beschränken sich auf eine Pfadanimation der Wegbeschreibung, einen einfachen Bildwechsel der Veranstaltungen und die Anzeige des Votingergebnisses. Die Benutzungshinweise fallen dabei ein wenig aus dem Rahmen, da sie als Film die Verwendung des Screens erklären sollen.

Die Formgebung ist schlicht und zurückhaltend, leicht abgerundete Flächen in Verbindung mit Schrift und einfachen Grafiken. Die Farbigkeit ist ebenfalls minimalistisch; die Flächen grau und weiß, wichtige Elemente betont mit einer Akzentfarbe, die dem jeweiligen Bereich zugeordnet wurde. Sie orientiert sich damit an dem Corporate Design der FH, da der Screen mit der Mensa an einem ihrer zentralen Anlaufpunkte positioniert wird und sich so in das Gesamtbild einfügt.

2. Corporate Design

Autor: Richter

Recherche FH Augsburg Corporate Design

Definition CD

Corporate Design bezeichnet einen Teilbereich der Corporate Identity (CI) und beinhaltet das gesamte visuelle Erscheinungsbild.

Dazu gehören die Gestaltung der Kommunikationsmittel (z.B. Logo, Definition von Farben, Fonts und Schriften).

Die Gestaltung aller Elemente des Corporate Design geschieht unter einheitlichen Gesichtspunkten, um bei jedem Kontakt einen Wiedererkennungseffekt zu erreichen.

Um eine möglichst effiziente Implementierung des Corporate Designs gewährleisten zu können, hat die FH Augsburg die einzelnen visuellen Basiselemente sowie exemplarische Anwendungen in Form eines Corporate-Design-Handbuchs dokumentiert und uns zur Verfügung gestellt.

FH Corporate Design

Schrift für Fließtexte Berthold Akzidenz-Grotesk Light

Überschriften Berthold Akzidenz-Grotesk Medium

Fließtext steht im linksbündigen Flattersatz-Zeilenabstand

120% des Schriftgrades

Überschriften in drei Graden, ausgehend vom Schriftgrad des Fließtextes

Absatz-Typ	SG	Abstand	Farbe
Überschrift 1.Grades	125%	3x	orange
Überschrift 2.Grades	125%	2x	schwarz
Überschrift 3.Grades	100%	1x	schwarz

Fließtext 100% 1x schwarz

Sonderfall Websites

Wenn, wie auf Webseiten, nur Systemschriftarten verwendet werden können, wird Arial oder Helvetica verwendet, da diese der Azidenz-Grotesk näher kommen als die häufigverwendete Verdana. In Flash und anderen Autorensystemen, die die Möglichkeit bieten Schriften einzubinden, wird wie gewohnt die Akzidenz-Grotesk verwendet. Textlinks sind schwarz und unterstrichen. Bei Mouseover wechselt die Farbe zu orange.

Farben

Für alle digitalen Dokumente werden websichere RGB Werte verwendet, für Druckdokumente die CMYK-Werte nach Euroskala oder die Pantone Werte.

Farbe	Pantone	CMYK	RGB
Orange	151	0/48/95/0	255/102/0
Dunkelgrau	Cool Gray 10	0/0/0/60	102/102/102

Mittelgrau	Cool Gray 6	0/0/0/30	153/153/153
Hellgrau	Warm Gray 2	0/2/5/10	204

Bildsprache

Generell soll ein sauberer, weißer Look vorherrschen, also weiße Flächen dominieren. Bilder und farbig ausgezeichnete Elemente setzen Akzente. Es ist darauf zu achten, dass bei den Bildern immer mit dem definiertem Orange „verwandte“ Farben das Klima bestimmen. Bilder werden nicht eingefärbt. Bilder werden nicht durch Entsättigung stilisiert. Normale Bildbearbeitung ist selbstverständlich erlaubt und erwünscht. Bilder sollten kein dominantes, konkurrierendes Farbklima besitzen.

Logo Varianten



Abb.9: Logo Varianten

Adaption des FH CDs auf den Infobereich

Vorüberlegungen

Wichtig ist eine einheitliche Gestaltung für Public Screen und Handyanwendungen, um dem User die Wiedererkennung und Bedienung zu erleichtern. Dabei sollte das Design die Benutzerführung, die Anwendungen und Informationen unauffällig unterstützen und nicht von den eigentlichen Inhalten ablenken. Die Gestaltung am

Public Screen kann dabei aufwendiger sein als auf dem Handy, da hier die Darstellungsmöglichkeiten durch das kleine Handy Displays stark eingeschränkt sind.

Bildformat Handy

Größe_12 x 12 Pixel

Auflösung_72 dpi

Bildformat Public Screen

Größe_128 x 128 Pixel

Auflösung_72 dpi

Überlegung zur Schriftwahl

Da der Abstand zwischen Public Screen und Benutzer sehr groß ist, und die Darstellung auf dem Handydisplay wiederum sehr klein, sollte die verwendete Schrift sehr gut lesbar sein. Serifenschriften und Schriften mit ausgeprägtem Fett/Fein-Kontrast sind daher ungeeignet; besser sind serifenlose mit klaren Formen. Außerdem weichen wir so auch nicht vom CD der FH Augsburg ab.

Schrift Entscheidung Public Screen

Wir entscheiden uns für Folio.

Mit der Folio konkurrieren wir nicht mit dem Logofont und bleiben in den Rahmenbedingungen des CD.

Folio Light

Folio Medium

Folio Bold

Schrift Entscheidung Handy-Display

Keine spezifische Schrift, da der Content auf jedem Handy dargestellt werden soll.

Arial oder Helvetica sind Standard Fonts.

Farbwelt

Wichtig ist, die Menüführung klar zu betonen und nicht von den Inhalten abzulenken, da diese eher aufwendig und auffällig sind. Als Basis eignen sich daher z.B. verschiedene Grautöne, wie sie auch im Erscheinungsbild der FH verwendet werden.

Buttons und ihre Farben

Die Farben der vier Elemente im Infobereich sollten einprägsam und ihrer Bedeutung nach den vier Bereichen logisch zu zuordnen sein und auch im Sinne der Farblehre verwendet werden.

Die Buttons wurden noch mal überarbeitet.

Überdenken der Symbole



Abb.10: Symbol-Entwürfe

Die Symbole sollten leicht zuzuordnen sein. Die neuen eindeutigeren Symbole z.B. das Kalenderblatt für Veranstaltungen aber auch das i Symbol, das für Information steht, wurde erkannt. Die Idee ein Ausschnitt des Logos für das Symbol der Gebäudeführung

zu wählen, ist den Studenten bekannt und jedem ist klar, das dieses Icon "irgendwas mit dem Gebäude "zu tun hat.

Also hatten wir unser Ziel erreicht, die Bedeutung der stilisierten Piktogramme so zu gestalten, das sie von unserer Zielgruppe intuitiv richtig erkannt werden.

Unterbereiche

icons fürs handy

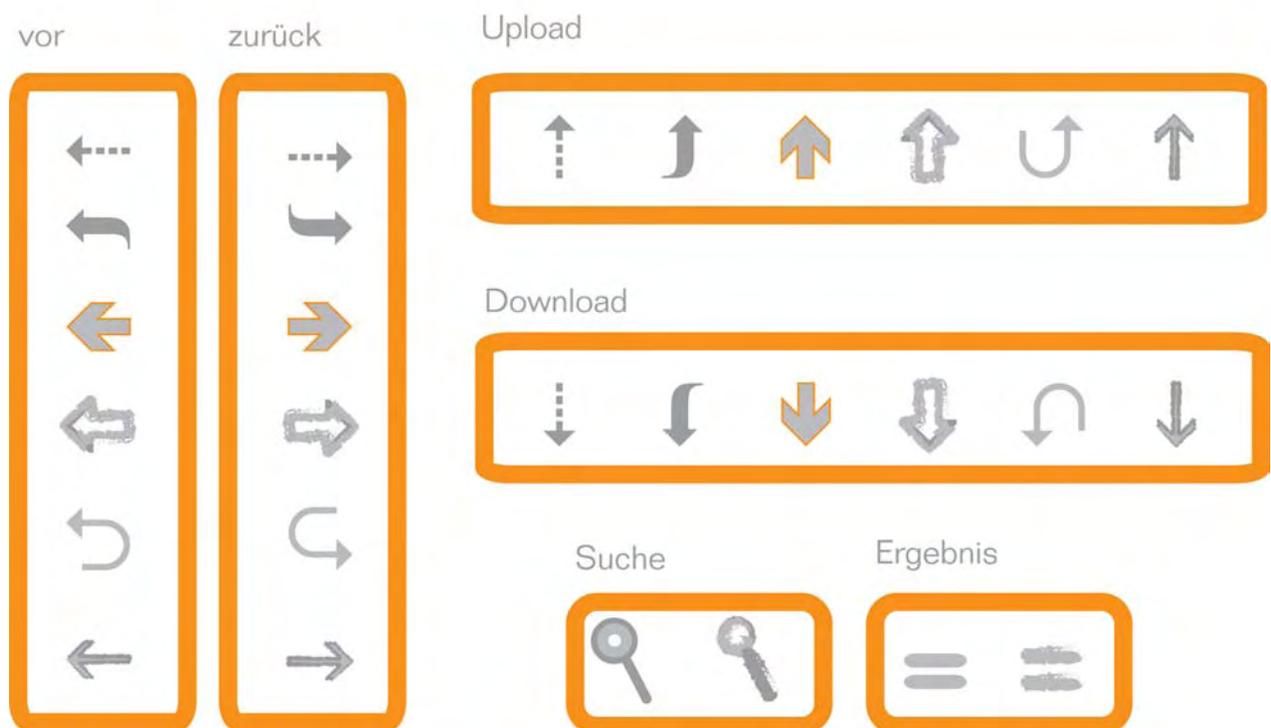


Abb.11: Icon-Entwürfe für das Handy

Finale Design Buttons auf dem Handydisplay



Abb.12: Handy-Buttons

Finale Design Buttons auf dem Screen



Abb.13: Screen-Icons

Entwicklung der Benutzeranleitung

Zusätzliche Komponenten

Jeder der vier Oberbereiche sollte eine Opener Seite anzeigen, bevor die Benutzeroberfläche des gewählten Oberbereichs dargestellt werden.

Erste Layout Entwürfe der Opener Seiten

Die Vorschläge halten sich im Groben am CD. Allerdings sind die Fotos und Darstellungen nicht gut gewählt und fungieren hier mehr als Platzhalter für die entsprechenden Bilder. Nach der ersten Gruppenpräsentation ist klar, dass noch mehr das Format (924 x 218 Pixel) des Public Screen einbezogen werden muss, um schöne Ergebnisse zu erzielen.



Abb.14: Finales Design Opener Veranstaltung und Gebäudeführung



Abb.15: Finales Design Opener Voting und Benutzeranleitung

Storyboard

Das Storyboard wurde einmal aufgeteilt in die grundsätzliche Bewegung und ihrer Wirkung.

Der Infobereich sollte mehr im Hintergrund bleiben und nur bei Bedarf in den Vordergrund treten. Also kam die Idee mit weichen Blenden und eher im Bereich 15 f/ps die einzelnen Elemente zu animieren.

Animationsablauf

1. Der Benutzer soll erfahren was alles möglich ist
2. Idee davon bekommen wie man den Screen bedient

Deutlich werden soll außerdem dass es drei Grundbereiche gibt:

- Spiele
- Information
- Aquarium

1. Einblenden der "schematisierten" Screen Übersicht
2. Einblenden eines "schematisierten" Handy Navigationskreuzes

Auf dem Display wird nacheinander, vom Hauptmenü aus, die Menü Navigation gezeigt bis der zu behandelnde Menüpunkt erscheint (z.B. Veranstaltung) dann wird eine Auswahl getroffen (z.B. Veranstaltung) und auf dem Handycross angezeigt.

Der ausgewählte Button wird größer und als nächstes Bild erscheint die jeweilige Eingangsseite, damit der User genau weiß welchen Bereich er gewählt hat. Das Action Script läuft ab.

Nach Ablauf des Action Script wird der Button aus dem Bereich, in dem der User sich befindet "herausgelöst" und es erscheinen gegebenenfalls noch Hinweise (z.B. Download/Upload). Die schematisierte Screenübersicht "fährt" ins Bild und die Auswahl und der Ablauf und die Bewegung der Animation wiederholen sich bis man am Ende der Benutzeranleitung die Übersicht der Oberbereiche sieht und die Animation beginnt von vorne.

Das FH-Logo wechselt die Position.

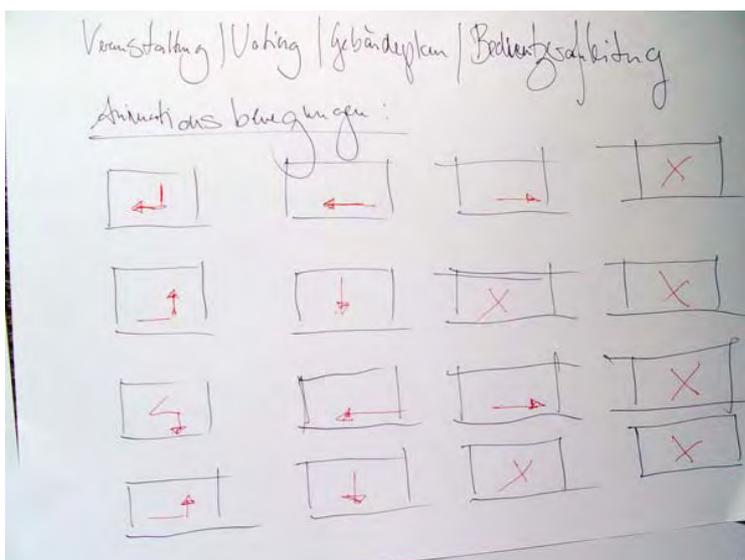


Abb.16: Storyboard - Benutzeranleitung - Analyse - Ablauf der Animationsbewegungen

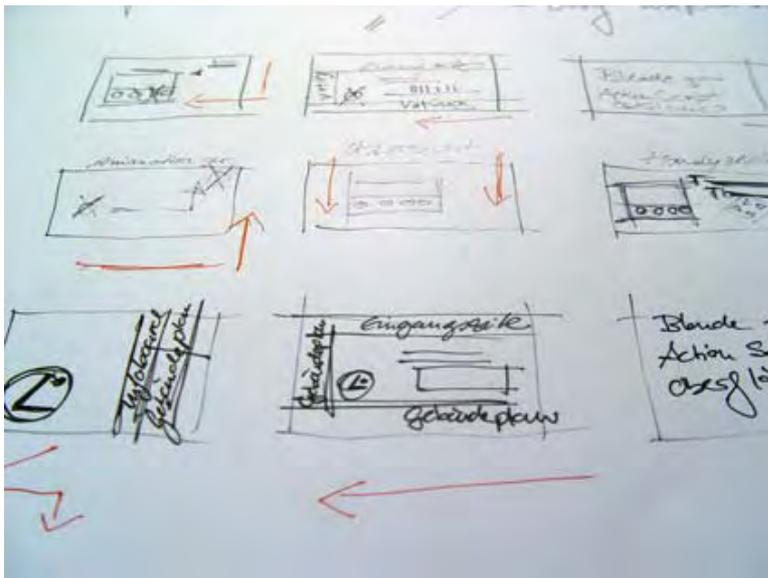


Abb.17: Storyboard - Benutzeranleitung - Ablauf der Animation

IV. Bereich „Information“ - Programmierung

1. Überblick

Autor: Burger

Um im Bereich „Information“ das Projekt „Public Screen“ zu einem erfolgreichen Abschluss zu bringen, mussten verschiedene Phasen durchlaufen werden (siehe Abb. 18).



Abb.18: Projekt-Phasen im Bereich „Information“

So wurde, wie bereits im Abschnitt 2 („II. Bereich „Information“ – Allgemeines“) beschrieben, zunächst eine Anforderungsanalyse und – bewertung durchgeführt (siehe Punkt 1, Abschnitt 2). Anschließend wurden auf Basis dieser Analyse und auf Grund von Anregungen von Seiten der projektbetreuenden Professoren und den anderen Projektteam-Mitgliedern die entgeltigen Projekt-Inhalte im Bereich „Information“ festgelegt (siehe Punkt 2, Abschnitt 2). Nachdem nun die theoretischen

Grundlagen feststanden, erfolgte die Einarbeitung in die unter Punkt 3 („3. *Technologieeinsatz*“) beschriebenen Technologien. Des weiteren mussten die eigenen Anwendungen implementiert werden (siehe Punkt 2, Abschnitt 4), diese dann in die Umgebung des Team „Technik“ integriert (siehe Punkt 3, Abschnitt 4) und anschließend die komplette Anwendung auf einem Mobiltelefon getestet werden (siehe Punkt 4, Abschnitt 4).

Diese gerade genannten letzten drei Projekt-Phasen werden nun im folgenden ausführlicher beschrieben.

2. Implementierung

Autoren: Burger, Kaiser

Veranstaltungen

Autor: Burger

Die Veranstaltungen bieten den Nutzern die Möglichkeit sich eine Liste der aktuellsten Veranstaltungen herunterzuladen, diese Liste auf dem Mobiltelefon abzurufen oder eigene Veranstaltungen einzustellen.

Dabei wird wie folgt vorgegangen. Je nach Intention wählt man aus dem Veranstaltungs-Menü einen der drei Punkte „Veranstaltungsliste“, „Download“ oder „Upload“ aus (siehe Abb. 19). Über den Joystick des Mobiltelefons gelangt man anschließend in den ausgewählten Bereich.

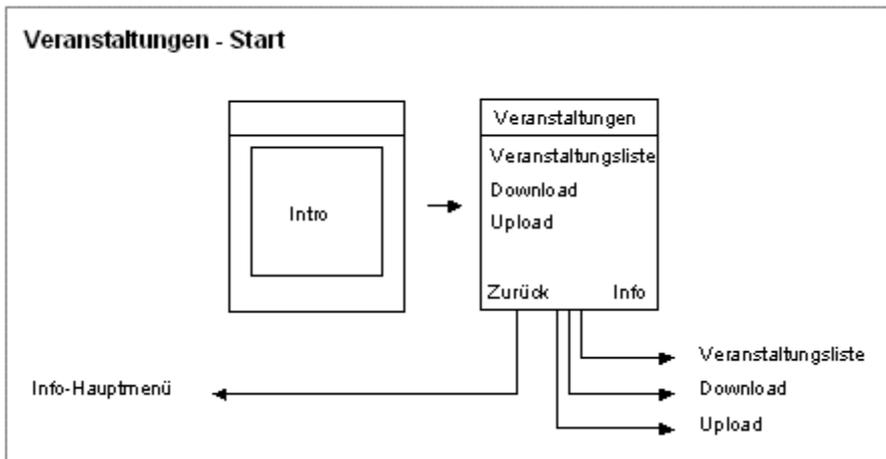


Abb.19: Startbildschirm von „Veranstaltungen“

Hat der Nutzer sich beispielsweise für den Punkt „Veranstaltungsliste“ entschieden, erscheint entweder – wenn bisher noch keine Download erfolgt ist – eine Anzeige auf dem Mobiltelefon-Display, dass bisher noch keine Veranstaltungen einsehbar sind, oder es erscheint die Liste der heruntergeladenen Veranstaltungen (Details siehe Abb. 20)

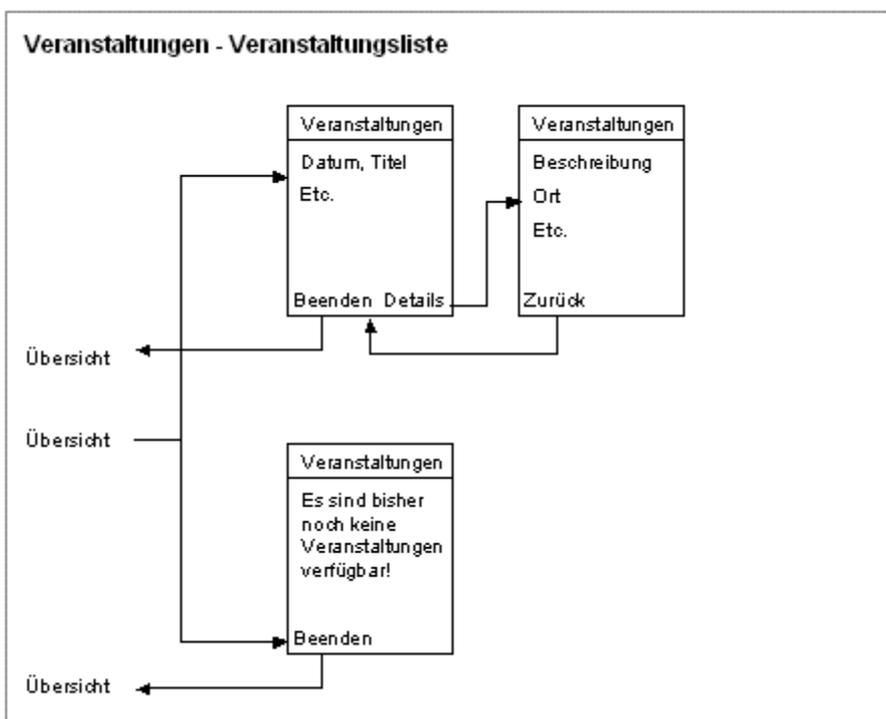


Abb.20: Ablaufdiagramm „Veranstaltungsliste“

Hat der Nutzer hingegen die Option „Download“ gewählt, kann er durch das unter Abbildung 21 aufgezeigte Vorgehen, die aktuellste Veranstaltungsliste vom Server herunterladen.

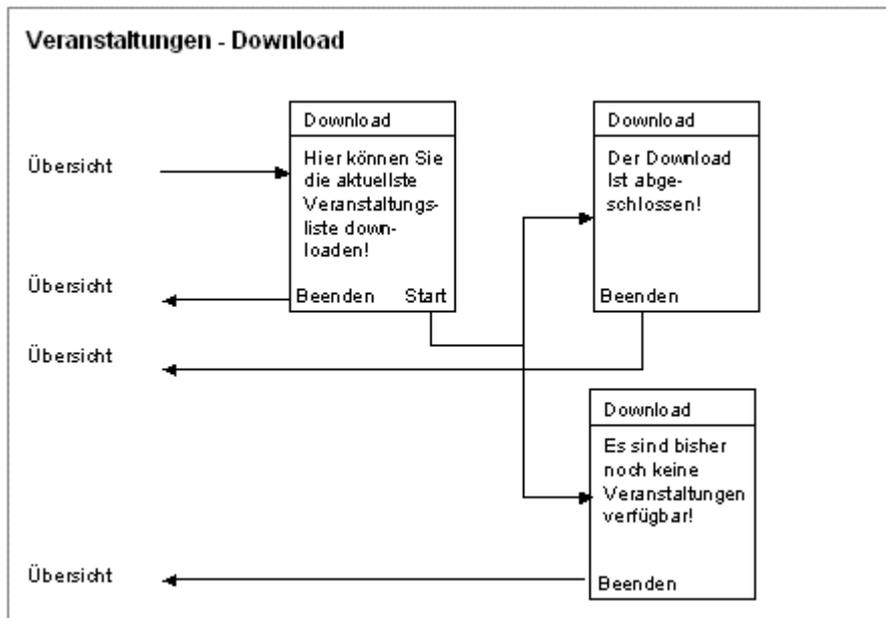


Abb.21: Ablaufdiagramm „Download“

Will der Nutzer aber eigene Veranstaltungen einstellen, dann wählt er den Punkt „Upload“ und füllt zunächst einmal das Upload-Formular aus.

Dieses beinhaltet sowohl Pflichtfelder (Titel, Ort, Startdatum, Startzeit) als auch optionale Eingabefelder (Beschreibung, Enddatum, Endzeit).

Durch verschiedenste Sicherheitsvorkehrungen ist sichergestellt, dass zum einen durch einen versehentlichen Rückkehrversuch in das Veranstaltungs-Menü keine Eingaben verloren gehen und zum anderen werden Fehl- und Falscheingaben verhindert. Des weiteren ist vorgesehen, dass nur registrierte Benutzer Veranstaltungen hochladen können.

Sind alle Eingaben korrekt und vollständig und der Nutzer registriert, wird man zunächst aufgefordert die gemachten Eingaben zu überprüfen, bevor anschließend die neue Veranstaltung über Bluetooth zum Server übertragen wird (siehe Abb. 22).

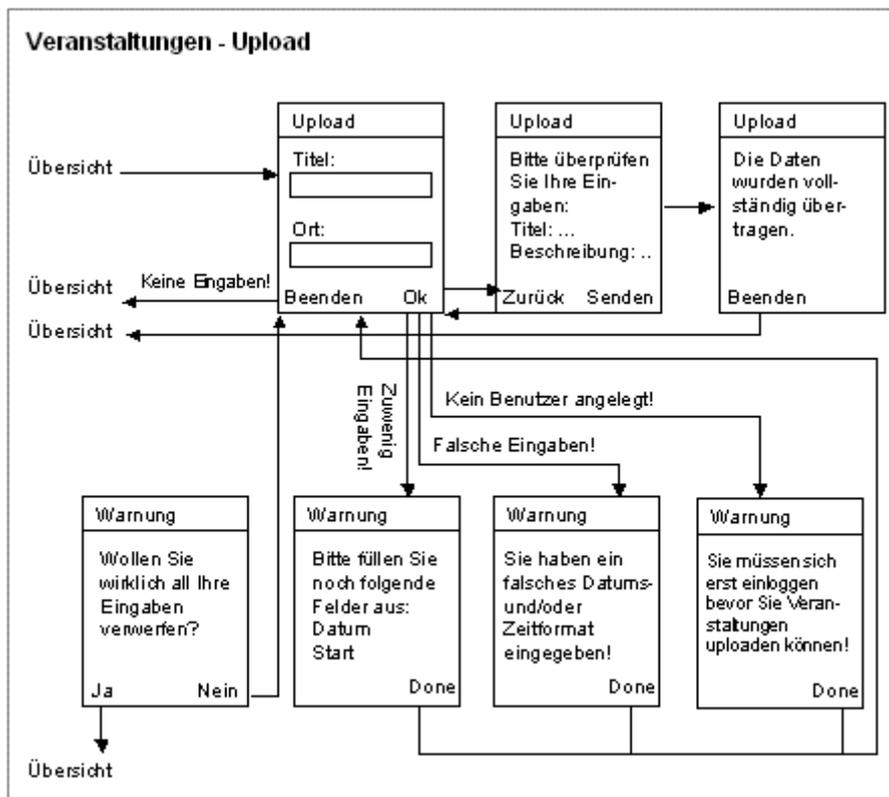


Abb.22: Ablaufdiagramm „Upload“

Programmietechnisch umgesetzt sind die gerade beschriebenen Abläufe auf Client-Seite in den drei Klassen ‚VList‘, ‚VDownload‘ und ‚VUpload‘. Auf Server-Seite existiert die Klasse ‚VeranstaltungenPlugIn‘.

Die Kommunikation zwischen Client und Server funktioniert mit verschiedenen Typen von Messages, die über Bluetooth versandt werden.

Die Speicherung und der Abruf der Veranstaltungsinformationen erfolgt auf Client-Seite im bzw. aus dem Telefoninternen RecordStore und auf Server-Seite in bzw. aus der Datenbank-Tabelle ‚Veranstaltungen‘ der MySQL-Datenbank über den Aufruf von Methoden in der Klasse ‚Database‘.

In Abbildung 23 kann man beispielsweise mit Hilfe eines Sequenzdiagramms sehen, wie die Kommunikation und die Speicherung der heruntergeladenen Veranstaltungen in der Klasse ‚VDownload‘ abläuft.

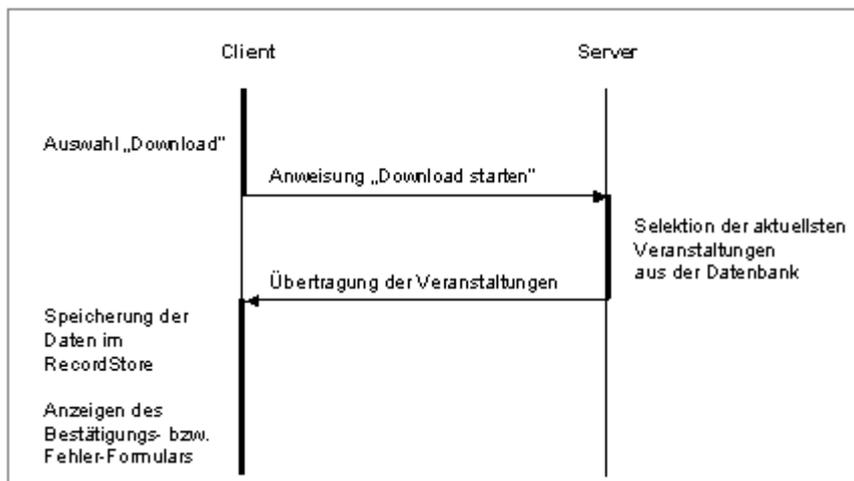


Abb.23: Sequenzdiagramm „Download“

In der Klasse ‚VList‘ hingegen erfolgt keine Kommunikation mit dem Server, sondern lediglich das Auslesen der beim Download gespeicherten Veranstaltungen aus dem RecordStore. Falls keine Veranstaltungen im RecordStore vorhanden sind, wird eine entsprechende Meldung auf dem Display ausgegeben.

Aus dem Sequenzdiagramm in Abbildung 24 wird ersichtlich, dass in der Klasse ‚VUpload‘ wieder eine Kommunikation zwischen dem Client und der Klasse ‚VeranstaltungenPlugIn‘ auf dem Server stattfindet und auch ein Eintrag der neuen Veranstaltungsdaten in die Veranstaltungs-Tabelle der MySQL-Datenbank erfolgt.

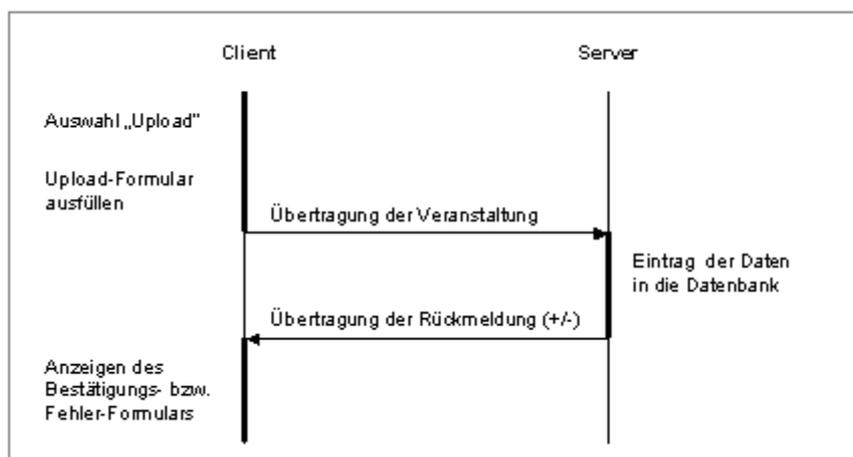


Abb.24: Sequenzdiagramm „Upload“

Gebäudeplan

Autor: Burger

Der Informations-Bereich „Gebäudeplan“ bietet den Nutzern die Möglichkeit nach bestimmten Institutionen und FH-spezifischen Suchbegriffen zu suchen, sich eine ausführliche Wegbeschreibung auf dem Public Screen anzusehen und diese Beschreibung sowie weitere Informationen anschließend auf ein Mobiltelefon herunterzuladen. Ziel dabei ist es, dass der Nutzer genau die Informationen erhält, die er benötigt, und zugleich einen räumlichen Eindruck davon bekommt, wo sich die jeweilige Institution/ausführende Stelle auf dem weitläufigen Gelände der FH Augsburg befindet.

Um den Gebäudeplan zu nutzen, muss der Nutzer zunächst einmal im Info-Menü den entsprechenden Menüpunkt auswählen, um anschließend in das Start-Menü des Gebäudeplans zu gelangen (siehe Abb. 25).

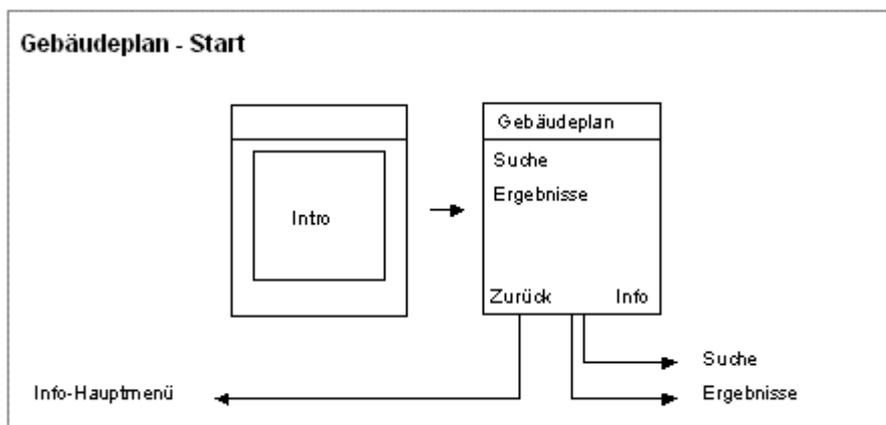


Abb.25: Startbildschirm von „Gebäudeplan“

Danach kann der Nutzer durch das unter „*Abb.26: Ablaufdiagramm ‚Suche‘*“ beschriebene Vorgehen seine(n) gesuchte(n) Institution bzw. Suchbegriff auswählen, die anschließend auf dem Public Screen gestartete Animation der Wegbeschreibung betrachten und bei Notwendigkeit auch das Suchergebnis auf sein Mobiltelefon herunterladen.

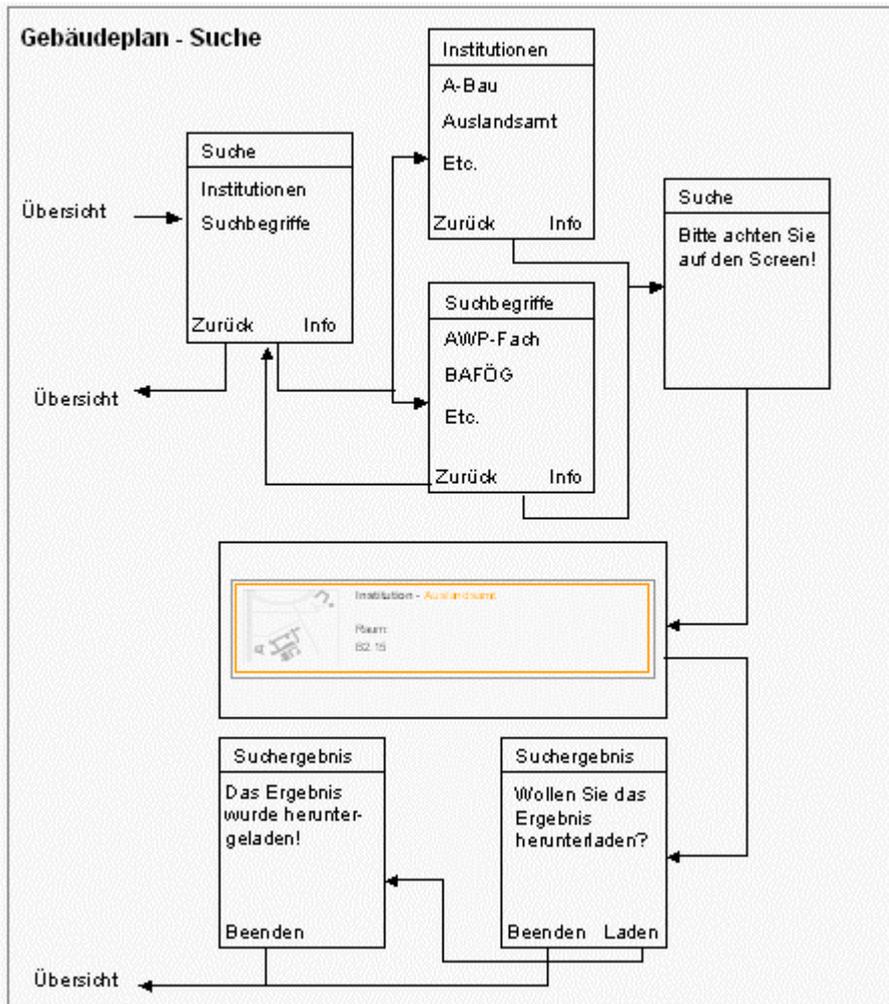


Abb.26: Ablaufdiagramm „Suche“

Falls der Nutzer die Download-Funktion in Anspruch nimmt, wird das Ergebnis über Bluetooth vom Server auf das Mobiltelefon übertragen. Der Nutzer kann dieses unter dem Menü-Punkt „Ergebnisse“ einsehen und auch wieder löschen (siehe Abb. 27).

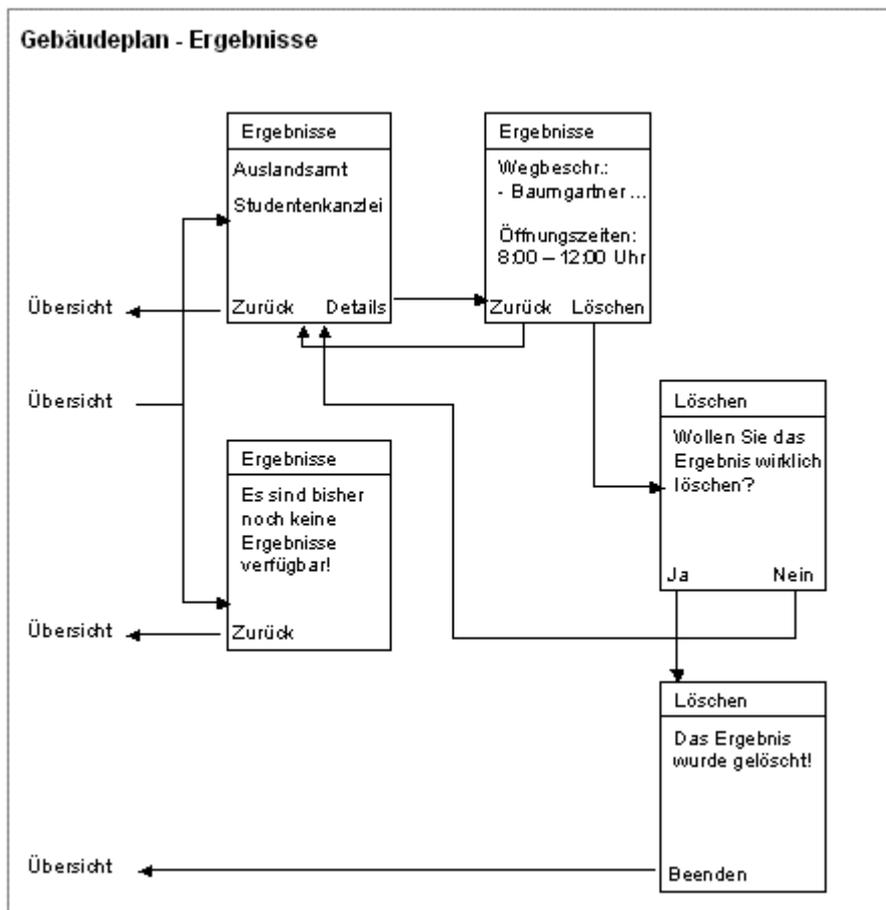


Abb. 27: Ablaufdiagramm „Ergebnisse“

Programmiertechnisch umgesetzt sind die gerade beschriebenen Abläufe auf Client-Seite in den zwei Klassen ‚GebäudeSuche‘ und ‚GebäudeErgebnisse‘ und auf Server-Seite in der Klasse ‚GebäudePlugIn‘.

Die Kommunikation sowie die Speicherung bzw. der Abruf der Daten erfolgt auch hier - wie bereits im Absatz „Veranstaltungen“ beschrieben - über verschiedene Message-Typen bzw. durch das Schreiben/Lesen in/aus dem RecordStore bzw. der Tabelle ‚Gebäude‘ in der MySQL-Datenbank.

Der grobe Ablauf einer solchen Kommunikation sowie Speicherung von Daten in der Klasse ‚GebäudeSuche‘ ist aus Abbildung 28 ersichtlich.

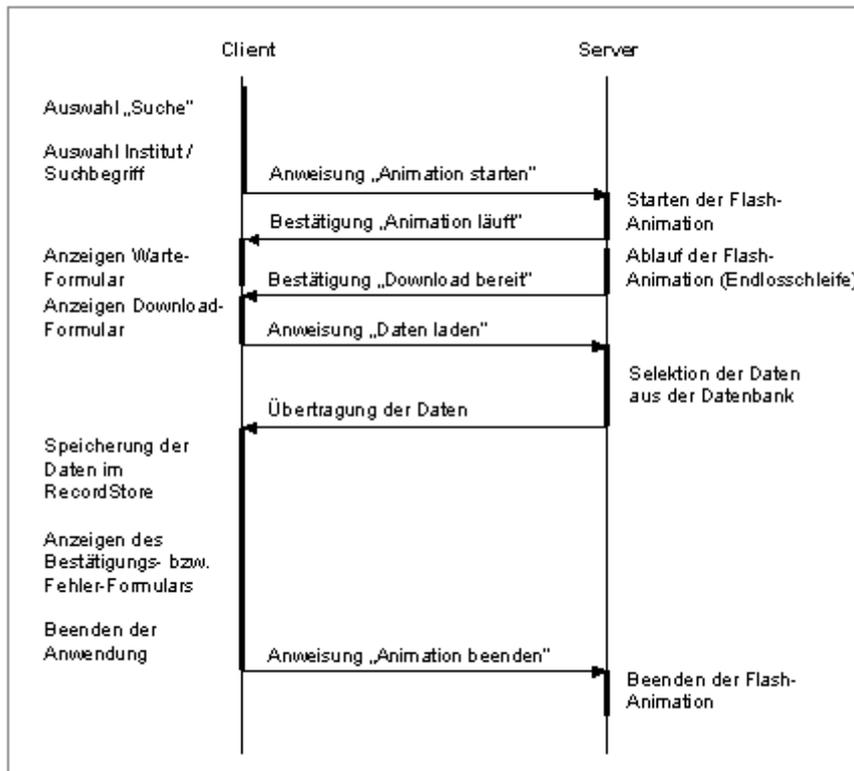


Abb.28: Sequenzdiagramm „Suche“

In der Klasse ‚GebäudeErgebnisse‘ findet wiederum keine Kommunikation mit dem Server statt. In dieser werden allerdings Datensätze aus dem RecordStore ausgelesen oder aus diesem gelöscht.

Voting

Autor: Kaiser

Die Voting-Funktion des Informationsbereichs ermöglicht Benutzern zum einen ihre Stimme zu einem aktuellen Voting abzugeben, und zum anderen selbst Themenvorschläge für Votings zu unterbreiten.

Um eine dieser Funktionen zu nutzen wählt der Benutzer im Menü „Informationen“ den Unterpunkt „Voting“. Daraufhin gelangt der Anwender, nachdem das Intro angezeigt wurde, ins Voting-Menü. Dort sind die beiden Menüpunkte „Teilnehmen“ und „Upload“ zu finden.

Teilnahme am aktuellen Voting:

Wählt der Benutzer "Teilnehmen", dann sendet die Handyapplikation (im folgenden Client) über Bluetooth eine Actionmessage des Typs "GET_VOTING". Dieser Message wird der Benutzername als Parameter angehängt. Auf Serverseite kann dann mittels der Tabelle "participants" festgestellt werden ob der Benutzer bereits eine Stimme für das laufende Voting abgegeben hat. Abhängig davon wird eine Variable vom Typ Boolean belegt.

Beim Erhalt einer "GET_VOTING"-Message überprüft die Serverapplikation darüber hinaus, ob derzeit ein aktuelles Voting zur Verfügung steht. Dazu wird auf eine Variable der Klasse currentVoting zugegriffen.

Läuft zurzeit kein Voting, sendet der Server eine Actionmessage vom Typ "NO_VOTING" an den Client zurück. Der Benutzer erhält daraufhin die Meldung "Derzeit ist leider kein Voting verfügbar" auf seinem Handydisplay.

Wenn ein aktuelles Voting verfügbar ist, erstellt die Serverapplikation eine Voting-Message mit allen Daten. Ein Voting besteht immer aus einem Titel, einer Frage und einer variablen Anzahl von Auswahlalternativen (minimal 2). Die entsprechenden Informationen werden ebenfalls aus der Klasse currentVoting gewonnen. Nachdem der Voting-Message die bereits erwähnte boolesche Variable mit dem Erlaubnis-Flag hinzugefügt wurde, wird sie an den Client gesendet.

Tritt bei der Behandlung einer Voting-Anforderung Serverseitig ein Fehler auf, so erhält der Client eine Actionmessage vom Typ "ERR_VOTINGDOWNLOAD". Daraufhin wird auf dem Handydisplay die Meldung "Beim Laden des aktuellen Votings ist ein Fehler aufgetreten. Bitte versuchen Sie es später erneut." ausgegeben.

Bei einem fehlerfreien Verlauf überprüft die Client-Applikation beim Erhalt der Voting-Message zunächst das Erlaubnis-Flag. Ist der Wert 0, wird dem Benutzer nur Titel und Frage des aktuellen Votings mit der Bemerkung "Sie haben ihre Stimme für dieses Voting bereits abgegeben" angezeigt. Der Benutzer hat dann nur noch die Möglichkeit mit der "Zurück"-Taste wieder ins Voting-Menü zu gelangen – ein zweites Mal abzustimmen ist damit unmöglich. Die "NO_VOTING"-Actionmessage wird auch versendet wenn auf Serverseite ein Fehler auftritt.

Ist das Erlaubnis-Flag gesetzt werden ebenfalls Titel und Frage des Votings auf dem Handydisplay angezeigt. Wird die "Weiter"-Taste gedrückt, folgt die Anzeige der

Auswahlalternativen als "Exclusive"-Liste. Nach dem Markieren einer Alternative kann die Auswahl mit dem Betätigen der "Senden"-Taste zum Server geschickt werden.

Dazu wird eine Textmessage, die den Text der ausgewählten Alternative und nochmals den Benutzernamen enthält, verwendet.

Beim Empfang dieser Message vermerkt die Serverapplikation bei der entsprechenden Auswahlalternative in der Tabelle "alternatives" den Treffer und berechnet umgehend den Stand des Votings neu. Der aktuelle Stand wird in der bereits mehrfach erwähnten Klasse currentVoting gespeichert. Um eine weitere Stimmabgabe durch den Benutzer zu verhindern wird außerdem der Benutzername in der Tabelle "participants" zusammen mit dem Primärschlüssel des Votings gespeichert. Der bereits beschriebene Mechanismus stellt dies sicher.

Um dem Benutzer die erfolgreiche Registrierung seiner Stimme zu signalisieren, sendet die Serverapplikation eine "VFeedback"-Message an den Client. Diese Message enthält die bisherige Anzahl der Teilnehmer am aktuellen Voting, die Auswahlalternativen sowie ihre prozentualen Stimmanteile.

Diese Informationen werden dem Benutzer als Dankeschön für die Teilnahme am Voting auf dem Handydisplay angezeigt.

Sollte während der Behandlung der Stimmabgabe auf Serverseite ein Fehler auftreten erhält der Client eine Actionmessage vom Typ "ERR_VOTE". Daraufhin wird die Meldung "Ihre Stimme konnte wegen eines technischen Problems nicht berücksichtigt werden. Bitte versuchen Sie es später erneut." ausgegeben.

Eine differenziertere Fehlerbehandlung mit entsprechenden Rollbacks bei der Datenbankmanipulation ist auf jeden Fall notwendig, konnte jedoch nicht mehr implementiert werden.

Upload eines Votings:

Um einen Vorschlag für ein Voting zu unterbreiten wählt der Benutzer im Voting-Menü den Punkt "Upload". Daraufhin erfolgt die Anzeige einer Eingabemaske auf dem Handydisplay. Hier sind Titel, Frage und die Auswahlalternativen anzugeben. Werden dabei die Formatierungsregeln nicht eingehalten, erfolgt nach der Betätigung der "Weiter"-Taste eine entsprechende Meldung. Sind hingegen alle Eingaben korrekt, erhält der Benutzer nochmals eine Übersicht. Er hat die Möglichkeit mit der

“Zurück”-Taste wieder zur Eingabemaske zu gelangen und Korrekturen vorzunehmen. Mit dem Betätigen der “Senden”-Taste veranlasst der Benutzer die Übermittlung der Daten zum Server.

Dafür wird eine “Voting”-Message verwendet. Dieser Messagetyp ist bereits aus dem Abschnitt “Teilnahme am aktuellen Voting” bekannt. Zu Verwechslungen bei der Verwendung dieses Message-Typs kann es jedoch nicht kommen, da durch den Absender implizit festgelegt ist, ob es sich um den Upload eines Voting-Vorschlags oder um die Übermittlung der Votingdaten bei der Teilnahme handelt. Ergänzend zu sagen ist, dass eine “Voting”-Message neben Titel, Frage und Auswahlalternativen auch eine Variable für einen Benutzernamen beinhaltet. Diese Variable wird beim Upload verwendet um den Autor zu übermitteln. Der Benutzername des Autors wird automatisch aus dem Recordstore geladen und angefügt.

Auf Serverseite werden die Informationen aus der Message direkt an die Datenbank angefügt. Dabei bilden die Attribute Titel, Frage und Autor den Primärschlüssel der Tabelle “votings”. Die Auswahlalternativen werden mit dieser Attributkombination als Fremdschlüssel in der Tabelle “alternatives” gespeichert. Das aktuelle Datum wird als Einstellungsdatum in “votings” hinzugefügt. Die beiden Felder Start- und Enddatum werden mit “null” belegt. Das Startdatum wird vom Voting-Scheduler gesetzt wenn der Abstimmungszeitraum des Votings beginnt, das Enddatum wird beim Ablauf des Abstimmungszeitraums eingetragen. Der Abstimmungszeitraum kann in der Klasse “Database” festgelegt werden, und ist für jedes Voting gleich.

Tritt beim Upload eines Votings Serverseitig kein Fehler auf, so sendet der Server eine Actionmessage vom Typ “OK_UPLOADVOTING” an den Client. Dem Benutzer wird daraufhin die Meldung “Vielen Dank für ihre Angaben! Ihr Vorschlag wurde gespeichert.” angezeigt.

Tritt Serverseitig ein Fehler auf, erhält der Client eine Actionmessage des Typs “ERR_VOTINGUPLOAD”. Der Benutzer wird darüber mit der Meldung “Ihre Angaben konnten wegen eines technischen Problems nicht gespeichert werden. Bitte versuchen Sie es später erneut.” informiert.

Voting-Scheduler:

Der Voting-Scheduler ist ein eigenständiges Server-PlugIn. Er ist dafür zuständig, dass ein Voting nach Ende seines Abstimmungslaufzeit beendet, und ein neues

Voting zur Abstimmung bereitgestellt wird. Dazu greift der Voting-Scheduler regelmäßig (alle 5 Minuten) auf die Tabelle "votings" zu. Diese Tabelle enthält alle Votingtitel mit Frage, Einstellungsdatum, sowie Start- und Endedatum.

Folgendes Aktivitätsdiagramm verdeutlicht die Arbeitsweise des Schedulers:

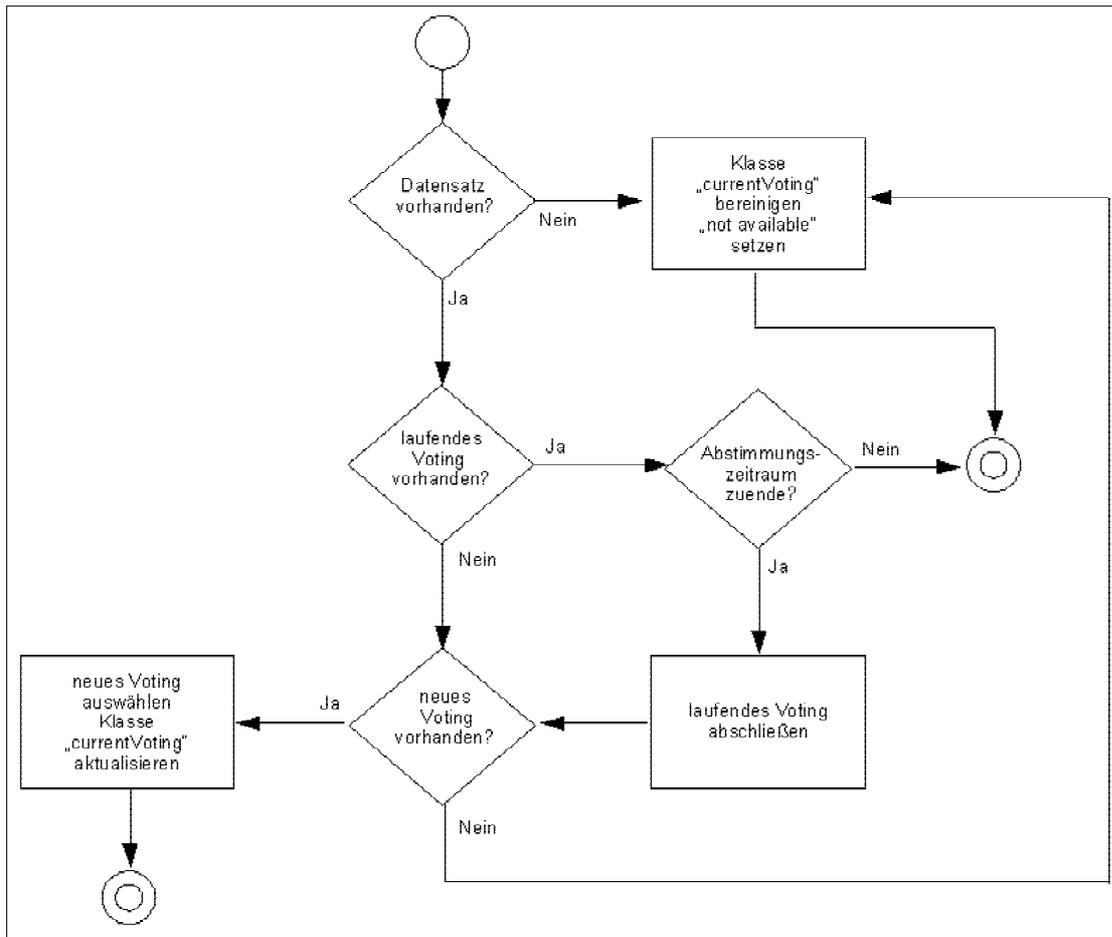


Abb.29: Arbeitsweise des Voting-Schedulers

Dieser Aufbau ermöglicht den vollautomatischen Betrieb der Voting-Software. Selbst nach Systemausfällen ist ein manueller Eingriff nicht notwendig.

Das entgültige Ergebnis eines Votings wird erst nach dem Ende des Abstimmungszeitraums öffentlich auf dem Screen bekannt gegeben. Die Anzeige vom Ergebnissen wird vom "InfoScreenScheduler" angestoßen. Mit der Implementierung dieses Server-PlugIns konnte jedoch aus Zeitgründen nicht mehr

begonnen werden. Die Darstellung wurde von den Gestaltern unseres Teams realisiert und erfolgt in Macromedia Flash.

Benutzungshinweise

Autor: Kaiser

Um Benutzern einen Überblick darüber zu verschaffen welche Möglichkeiten das System bietet und wie es bedient wird, wurde der Bereich „Benutzungshinweise“ geschaffen. Vermittelt werden sollen diese Informationen mit Hilfe einer ansprechenden Simulation.

Das Abspielen dieser Simulation auf dem Screen kann vom Benutzer per Handy angestoßen werden. Dazu muss der Punkt „Benutzungshinweise“ im Hauptmenü des Informationsbereichs ausgewählt werden. Auf dem Handydisplay erscheint daraufhin die Anweisung „Drücken Sie Start...“ Beim Betätigen der Start-Taste sendet die Clientapplikation eine Actionmessage des Typs „START_USERGUIDE“ an den Server.

Beim Empfang der Message wird über einen Funktionsaufruf des technischen Frameworks die Animation auf dem Screen gestartet. Eine Rückmeldung des Servers an den Client erfolgt in diesem Fall nicht.

Möchte der Benutzer die Anzeige der Animation vorzeitig beenden, genügt die Betätigung der Taste „Beenden“. Die Rückkehr ins Hauptmenü des Informationsbereichs auf dem Handy erfolgt daraufhin automatisch. Die Animation wurde von den Gestaltern unseres Teams in Flash realisiert.

Informationsmenü

Autor: Kaiser

Innerhalb des Informationsbereichs war uns eine saubere einheitliche Menüstruktur besonders wichtig. Deshalb haben wir auf Clientseite eine Klasse allein für die Menüführung angelegt. Die Klasse „InfoMenu“. Beim Start des Informations-Plugins auf Clientseite erfolgt zunächst ein Funktionsaufruf des Singleton-Objekts dieser

Klasse. Die Darstellung der Menüs und die Navigationslogik ist größtenteils innerhalb dieser Klasse implementiert. Erst bei der Auswahl bestimmter Menüpunkte erfolgt der Einsprung in Funktionen der Klassen für Voting, Veranstaltungen, Benutzungshinweise oder Gebäudeführung. Dort sind in der Regel weitere Untermenüs implementiert, die auf die jeweiligen Anforderungen zugeschnitten sind. Innerhalb der Hauptmenüs stehen dem Benutzer durchgehend drei Funktionstasten zur Verfügung: „Auswahl“, „Zurück“, „Info“. Durch die Verwendung der High-Level-UI ist die Belegung der Tasten mit den Funktionen von Handy zu Handy unterschiedlich, jedoch immer einheitlich!

Die Funktionen „Auswahl“ und „Zurück“ sind für die Navigation in den Menüs gedacht. Beim Betätigen der „Info“-Taste erscheint eine Meldung die dem Benutzer beschreibt welche Möglichkeiten das aktuelle Menü eröffnet.

3. Integration

Autor: Burger

Wie bereits in Abschnitt 1, Punkt 3 („3. Rahmenbedingungen“) erwähnt, hatte das Team „Technik“ die Aufgabe ein Framework zu schaffen, in das die Applikationen der anderen Teams eingebettet werden können und in dem Daten und Informationen über Bluetooth versandt werden können. Um dies zu ermöglichen wurde von Seiten des Team „Technik“ eine Art PlugIn-Konzept entwickelt (siehe Abb. 30).

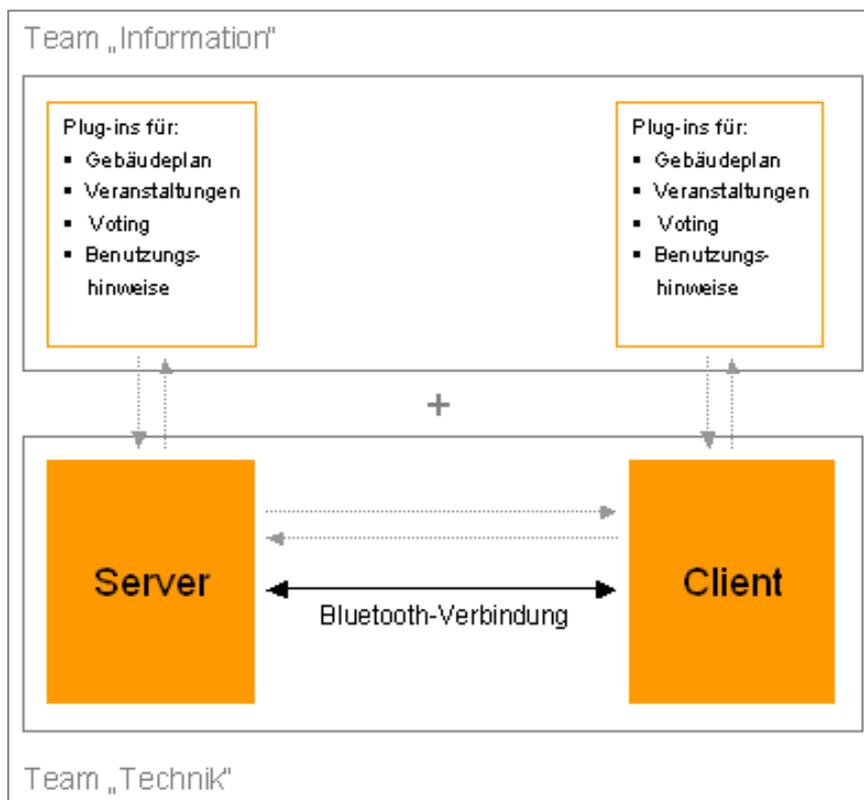


Abb.30: Integration durch PlugIns

Bei diesem Konzept wurden für jedes Projektteam eigene Projekte angelegt, deren Namen im Bereich „Information“ ‚PublicScreenClientInfo‘, ‚PublicScreenServerInfo‘ und ‚PublicScreenShared‘ lauten.

Das Projekt ‚PublicScreenClientInfo‘ beinhaltet dabei alle Klassen für die Client-Seite und ‚PublicScreenServerInfo‘ enthält alle Klassen für die Server-Seite. Im Projekt ‚PublicScreenShared‘ hingegen sind diejenigen Klassen vorhanden, die sowohl Client als auch Server betreffen.

Innerhalb dieser Projekte wurden von Seiten des Team „Technik“ bereits einige Klassen angelegt, die zur Integration der Gruppen-Anwendungen in das Framework zwingend erforderlich sind (siehe Anhang).

Des Weiteren musste von Seiten des Teams „Information“ weitere Klassen, die bereits teilweise unter *„2. Implementierung“* erwähnt wurden, angelegt und an den diversen Managern des Team „Technik“ (z.B. PluginManager, MessageManager etc.) registriert werden.

Für die Kommunikation zwischen Client und Server wurden sowohl die vom Team „Technik“ bereitgestellte Message- und Action-Typen verwendet als auch gruppeneigene Typen angelegt und eingesetzt.

Für die Speicherung der Informationen wurde auf Server-Seite, wie bereits erwähnt, nicht die Datenbank des Team „Technik“ verwendet, sondern eine eigene MySQL-Datenbank eingerichtet. Auf Client-Seite hingegen wurde zur Speicherung der bereitgestellte RecordStoreManager sowie die in diesem Zusammenhang vorhandenen Methoden zum Schreiben, Lesen und Löschen verwendet. Um diese Funktionalität einwandfrei nutzen zu können, mussten allerdings noch eigene Deserializer geschrieben werden.

Zur Darstellung der Inhalte auf dem Screen fiel die Entscheidung wie unter Abschnitt 2 „3. *Technologieeinsatz*“ beschrieben, auf Flash. Um eine Kommunikation zwischen Flash und der serverseitigen Java-Gesamtanwendung zu ermöglichen, stellte das Team „Technik“ wieder einige Basis-Klassen sowie Flash- bzw. ActionScript-Dateien (z.B. JavaFlashClient.as) bereit. Von Seiten des Team „Information“ mussten dann noch eigene Flash- bzw. ActionScript-Dateien (info.swf, Info.as) kreiert, die zur Kommunikation notwendigen Code-Zeilen in die jeweiligen Klassen eingefügt und diese Klassen dann am FlashManager registriert werden.

4. Test

Autor: Burger

Nachdem also die Anwendungen implementiert und in die Umgebung des Team “Technik” integriert wurden, mussten sie noch auf ihre Praxistauglichkeit hin untersucht werden.

Um einen Praxistest durchführen zu können, benötigt man:

- einen Bluetooth-Dongle
- ein Mobiltelefon, das sowohl Java als auch Bluetooth unterstützt
- einen Computer, der als Server fungiert

Hat mal all diese Utensilien bereitgestellt, müssen folgende Schritte befolgt werden:

1. Bilden der entsprechenden Jar- und Jad-Datei
2. Übertragung der Jar-Datei auf ein Mobiltelefon mittels Bluetooth, Infrarot oder Kabel
3. Installation der Jar-Datei auf dem Mobiltelefon
4. Starten des Servers
 - 4.1 Starten der MySQL-Datenbank
 - 4.2 Starten der Klasse ‚FlashMain‘ als SWT-Applikation im Projekt ‚PublicScreenServerInfo‘
 - 4.3 Öffnen der Datei ‚container.swf‘ im Projekt ‚FlashFiles‘ (direkt oder im Internet Explorer)
 - 4.4 Starten der Klasse ‚Main‘ im Projekt ‚BlueCoveCOMM‘
5. Starten des MIDlets auf dem Mobiltelefon
6. Durchführung des Test (Registrieren > Verbinden > Auswahl Menü-Punkt Informationsbereich > ...)

5. Probleme und Mängel

Autor: Burger

Die größten Probleme und Mängel sind v.a. in den Projektphasen Implementierung, Integration und Test aufgetreten.

Implementierung

Autor: Burger

Mängel, die auf Grund der Zeitknappheit gegen Ende des Projekts nicht mehr beseitigt werden konnten, sind beim Punkt „Veranstaltungen“ das Datumsformat (YYYY-MM-DD) und die Möglichkeit Veranstaltungen hochzuladen, auch wenn der Nutzer nur als Gast registriert ist.

Beim Unterbereich „Gebäude“ sind für die unzähligen Institutionen und Suchbegriffe bisher nur vier Punkte beispielhaft umgesetzt worden. Bei diesen handelt es sich in

der Rubrik Institutionen um die Punkte ‚Bibliothek (Zentral)‘ und ‚Studentenkanzlei‘ und in der Rubrik Suchbegriffe um ‚Auslandsstudium‘ und ‚Bibliotheksausweis‘.

Des weiteren fehlt noch ein sogenannter „InfoScreenScheduler“, also eine Klasse in Java, die den zeitlichen Ablauf der einzelnen Screen-Applikationen im Bereich „Information“ steuert. Bisher wurde der Aufruf der Screen-Applikationen durch eine Aktion auf dem Mobiltelefon ausgelöst (z.B. Start des Veranstaltungen-Downloads), was aber nur eine Übergangslösung zu Test- und Demonstrationszwecken darstellen sollte, die aber leider dann aus zeitlichen Gründen nicht mehr auf die Ideallösung, also den Screen-Scheduler, umgestellt werden konnte.

Da das Team „Information“ geringe Kenntnisse im Bereich der Flash-Programmierung aufwies, wurde die Flash-Anwendung für die verschiedenen Unterbereiche auf eine sehr einfache Art und Weise kreiert. Und zwar wurden die Flash-Animation für die vier Informationsbereiche in der Zeitleiste einer einzigen Flash-Datei (info fla) hintereinander gebaut und die jeweiligen Startpunkte durch ein entsprechendes Label gekennzeichnet. Durch den Aufruf von Funktionen in der zugehörigen as-Datei (Info.as), wird dann einfach nur an das jeweilige Label gesprungen und der dort beginnende Film bis zum nächsten Stop-Punkt abgespielt. Eine „schönere“ Lösung war leider auf Grund der begrenzten Flash-Kenntnisse sowie der knappen Zeit leider nicht möglich.

All diese Mängel müssten durch die folgende Projektarbeit beseitigt werden.

Integration

Autor: Burger

Die größten Probleme und damit den größten Zeitaufwand verursachte allerdings die Integration der eigenen Anwendungen in die Umgebung des Team „Technik“.

So funktionierte beispielsweise zunächst das Versenden von Messages nicht und auch das Zusammenspiel zwischen Flash und Java bereitete einige Schwierigkeiten.

Und es gab noch weitere kleine „Stolpersteine“, deren Erwähnung an dieser Stelle aber zu weit gehen würde.

Doch mit tatkräftigen Unterstützung des Team „Technik“ konnten die meisten der gerade beschriebenen Probleme gelöst werden.

Ein Problem, das bis zuletzt leider nicht gelöst werden konnte, ist, dass durch einen vermeintlichen Fehler im MessageManager keine Messages zum Mobiltelefon verschickt werden können, die eine Größe von mehr als 100 bis 150 Byte aufweisen. Dadurch musste für das Versenden der Veranstaltungsliste sowie der Wegbeschreibung des Gebäudeplans eine Umgehungslösung gefunden werden. Bei dieser werden die Daten gesplittet, mit Hilfe mehrerer Messages an das Mobiltelefon versandt und dort wieder zusammengesetzt.

Test

Autor: Burger

Bei der Durchführung von Praxistests traten v.a. Probleme in Bezug auf die verschiedenen Mobiltelefon-Modelle auf.

So war beispielweise die Jar-Datei auf einigen Mobiltelefonen grundsätzlich nicht lauffähig.

Es konnte aber auch passieren, dass die Jar-Datei der Gesamtanwendung im Vergleich zur verfügbaren Speicherkapazität auf dem Mobiltelefon einfach zu groß war und deshalb nicht auf dem Mobiltelefon installiert werden konnte.

Des Weiteren gab es Probleme mit der grafischen Darstellung von z.B. den Intros auf den verschiedenen Mobiltelefonen, da diese je nach Modell unterschiedlich stark abweichen konnte.

Beispiele bei dem die Jar-Datei einfach nicht lauffähig war, sind die Mobiltelefon-Modelle Nokia 6600 und Motorola A780. Beim Nokia 6600 bestand der Grund dafür in einer zu alten CLDC-Version (1.0 statt der benötigten 1.1), wohingegen beim Motorola keine plausible Erklärung für die Nichtlauffähigkeit gegeben werden konnte.

Des Weiteren wurde untersucht, ob mit den Mobiltelefonen vom Typ Siemens S65, Sony Ericsson K700i und Nokia 6230i ein Praxistest durchzuführen ist.

Dabei kamen folgende Vor- bzw. Nachteile zum Vorschein:

- Siemens S65
 - (+) Platz für die Gesamtanwendung
 - (-) kein Zustandekommen einer Bluetooth-Verbindung mit dem Server
- Sony Ericsson K700i
 - (+) Platz für die Gesamtanwendung
 - (-) kein Empfang von Messages vom Server
 - (-) keine zentrierte Ausrichtung der Intro-Bilder
- Nokia 6230i
 - (-) nur Platz für Einzelanwendungen
 - (-) fehlerhafte Darstellung der Intro-Bilder
 - (+) Senden und Empfangen von Messages

Da das Nokia 6230i das einzige Mobiltelefon war, auf dem das Senden und Empfangen, also die Kommunikation zwischen Client und Server, in beide Richtungen einwandfrei möglich war, wurden dieses primär als Test-Modell verwendet.

V. Resümee

1. Kritische Reflektion

Autor: Burger

Betrachtet man die ausgearbeiteten Ideen und das umgesetzte Resultat so kann man feststellen, dass eine weitestgehende Übereinstimmung zwischen beiden herrscht. Das Projekt verlief zwar zum einen oder anderen Zeitpunkt auf Grund der unter Abschnitt 4 Punkt 5 beschriebenen Probleme nicht immer ganz reibungslos, allerdings sind letztendlich doch nur sehr wenige schwerwiegende Mängel bestehen

geblieben. Das Ziel einen lauffähigen Prototypen zu entwickeln wurde also durchaus erreicht.

2. Ausblick

Autor: Burger

Da das Projekt „Public Screen“ im folgenden Wintersemester von einer Gruppe von Multimedia-Studenten weitergeführt wird, sehe wir, das Team „Information“, durchaus gute Chancen, dass das Gesamtprojekt zu einer Reife gelangt, um tatsächlich im Frühjahr 2007 im Vorraum der neuen Mensa platziert und genutzt zu werden.

VI. Anhang

Projekt	Package	Klasse	Team
publicScreenClientInfo	publicScreen.client.info	AnInfoExamplePlugIn	Technik
		Benutzungshinweise	Information
		InfoClientConfiguration	Technik
		InfoMenu	Information
		InfoMidlet	Technik
		StartMidlet	Technik
		Voting	Information
	publicScreen.client.info.gebaeude	GebaeudeErgebnisse	Information
		GebaeudeSuche	Information
	publicScreen.client.info.ressources	[Images]	Information
	publicScreen.client.info.veranstaltungen	VDownload	Information
		VList	Information
		VUpload	Information
publicScreenServerInfo	publicScreen.server.info	AnInfoExamplePlugIn	Technik
		BenutzungshinweisePlugIn	Information
		Database	Information
		FlashMain	Technik
		GebaeudePlugIn	Information
		InfoServerConfiguration	Technik
		Main	Technik
		VeranstaltungenPlugIn	Information
		VotingPlugIn	Information
		VotingScheduler	Information
publicScreenSharedInfo	publicScreen.shared.info	ClientInfoComponentIDList	Technik
		ServerInfoCpmonentIDList	Technik
	publicScreen.shared.info.data	GebaeudeKey	Information
		StoreValue	Information
		StoreValueV	Information
		VeranstaltungenKey	Information
	publicScreen.shared.info.deserializer	GebaeudeRSDeserializer	Information
		GKeyRSDeserializer	Information
		VDownloadMessageDeserializer	Information
		VeranstaltungenRSDeserializer	Information
		VFeedbackMessageDeserializer	Information
		VKeyRSDeserializer	Information
		VotingMessageDeserializer	Information
		VUploadMessageDeserializer	Information
	publicScreen.shared.info.messages	ActionTypes	Information
		VDownloadMessage	Information
		VFeedbackMessage	Information
VotingMessage		Information	
VUploadMessage		Information	