## Domain Analyse

### Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 19.03.2012 | 1.0 | Erste Version des Dokuments | DT |
| 20.03.2012 | 1.1 | Review | CH |
| 26.03.2012 | 1.2 | Durchführung und Resultat hinzugefügt | LE |
| 27.03.2012 | 1.3 | Review Test-Durchführung | CH |
| 27.03.2012 | 1.4 | Review, Test-Resultat | DT |
| 02.04.2012 | 1.5 | Review | CH |
| 04.05.2012 | 1.6 | Ideensammlung Demomodus | DT |
| 07.05.2012 | 1.7 | Review Demomodus | CH |
| 09.05.2012 | 1.8 | Umsetzung Demomodus Kraftfeld | DT |
| 15.05.2012 | 1.9 | Externes Design | CH |
| 15.05.2012 | 1.10 | Review Externes Design | DT |
| 15.05.2012 | 1.11 | Umsetzung Demomodus Teaser | DT |
| 22.05.2012 | 1.12 | Review Umsetzung Demomodus Teaser | CH |
| 22.05.2012 | 1.13 | Domain Models | CH |
| 23.05.2012 | 1.14 | Verfügbarkeit Daten | CH |
| 24.05.2012 | 1.15 | Review Korrektur Markus Stolze | DT |
| 25.05.2012 | 1.16 | Administration der Videowall hinzugefügt | LE |
| 27.05.2012 | 1.17 | Review Domain Models, Verfügbarkeit Daten | DT |
| 27.05.212 | 1.18 | Review Administration der Videowall | CH |
| 27.05.2012 | 1.19 | Review Administration der Videowall | DT |
| 29.05.2012 | 1.20 | Screen Map | CH |
| 04.06.2012 | 1.21 | Review Screen Map | DT |
| 04.06.2012 | 1.22 | Guidelines, Design Entscheide | CH |
| 04.06.2012 | 1.23 | Ergänzung Kinect Bedienung | DT |
| 04.06.2012 | 1.24 | Demomodus Zustandsdiagramm angepasst und beschrieben | LE |
| 05.06.2012 | 1.25 | Review Guidelines | DT |
| 06.06.2012 | 1.26 | Prozessdiagramm | CH |
| 08.06.2012 | 1.27 | Review Design Entscheide, CI/CD Absprache mit HSR | DT |
| 09.06.2012 | 1.28 | Review und Korrekturen | LE |
| 10.06.2012 | 1.29 | Review Systemübersicht | CH |
| 10.06.2012 | 1.30 | Review Poster-Prozess | DT |
| 11.06.2012 | 1.31 | Korrekturen Markus Stolze übernommen | DT |

### Systemübersicht

Das System ist in mehrere Tiers unterteilt. Dabei handelt es sich um die Folgenden:

* HSR Videowall mit Kinect
* Service Server mit Datenbank
* Webserver
* Mobiltelefon
* Sekretariats-Computer

Dieser Aufbau entspricht dem gewünschten Endsystem. Der in dieser Bachelorarbeit effektiv umgesetzte Teil findet sich im (TODO ref: Entwurf unter Systemübersicht).



Abbildung 1 - Systemübersicht, gewünschtes Endsystem

#### Videowall mit Kinect

Über die Videowall können sich Nutzer über verschiedene Themen (beispielsweise Poster von Bachelorarbeiten, das Mittagsmenu der Mensa oder das Wetter) informieren. In Zukunft sind auch Optionen wie z.B. Minispiele denkbar. Die Wall wird mittels Kinect gesteuert. Die dafür benötigten Daten werden durch WCF vom Service Server geladen.

#### Service Server mit Datenbank

Auf dem Service Server werden die verschiedenen Daten, welche die Videowall benötigt, abgelegt. Diese können mittels WCF über den Webserver verwaltet oder auf der Videowall angezeigt werden.

Dieser Server könnte auch in die Cloud (Windows Azure[[1]](#footnote-1)) ausgelagert werden, falls für die Daten kein eigener Server betrieben werden soll.

#### Webserver

Der Webserver bietet einerseits eine Administrationsoberfläche für das Sekretariat an, um die Daten verwalten zu können. Dies aber nur, sofern die Daten nicht direkt in der Typo3 Datenbank des HSR Webauftritts[[2]](#footnote-2) verwaltet werden (siehe auch TODO ref Typo3 Vorschlag).   
Andererseits können auf dem Webserver per Mobiltelefon spezifische Informationen zu den auf der Wall dargestellten Daten abgerufen werden.   
Beide Anforderungen sind einfach über einen Webserver realisierbar, es muss so keine zusätzliche Applikation auf den Zielgeräten installiert werden. Beide Verbindungen basieren auf dem Protokoll HTTPS. Die durch das Sekretariat getätigten Änderungen werden vom Webserver aus mittels WCF an den Service Server weitergeleitet.

#### Mobiltelefon

Über den Browser des Mobiltelefons können spezifische Informationen zu den visualisierten Daten der Videowall abgerufen werden.

#### Sekretariats-Computer

Die Administrationsoberfläche kann über den Browser eines Sekretariats-Computers aufgerufen werden. Über diese können die Daten der Videowall verwaltet werden.

### Daten

#### Prozessdiagramm

##### Ist-Prozess

Die Poster durchlaufen, von ihrer Entstehung bis zu ihrer Ausstellung, den nachfolgenden Prozess:

Abbildung 2 - Poster Ist-Prozess

##### Soll-Prozess

Auf der in Zukunft an der Hochschule installierten Videowall sollen die Bachelorposter aller Studiengänge verfügbar sein. Daher soll ein Poster in Zukunft den folgenden Prozess zu durchlaufen haben:

Abbildung 3 - Poster Soll-Prozess

#### Domain Models

Auf der Videowall sollen verschiedene Inhalte präsentiert werden. Die Präsentation der Poster oder das Mittagsmenu sind Beispiele für solche Inhalte. Eine Anforderung an das Framework ist es daher, dass diese Inhalte hinzugefügt werden können.

##### Framework VideoWall



Abbildung 4 - Domain Model VideoWall

Wie in Abbildung 4 - Domain Model VideoWall ersichtlich ist, verwaltet die *VideoWall* mehrere *VideoWallApplications*. Zudem wird immer eine *VideoWallApplication* von der *VideoWall* angezeigt.   
Subklassen der *VideoWallApplication* sind die *PosterApplication*, die *LunchMenuApplication* oder jeglicher denkbarer Inhalt, welcher auf der *VideoWall* präsentiert werden soll. Diese Inhalte werden durch die Subklasse *AnyApplication* veranschaulicht. Eine Subklasse der *VideoWallApplication* verfügt über folgende Attribute:

|  |  |  |
| --- | --- | --- |
| Attribut | Beschreibung | Beispiel |
| DemomodeText | Der Teaser-Text, welcher im Demomodus angezeigt wird (siehe hierzu I.1.3.2.2.2 Umsetzung des Demomodus „Teaser“). | Hunger? |
| MainView | Die View, welche als Einstiegspunkt in die Applikation dient. | - |
| Name | Der Name der Applikation. | Mittagsmenu |

Tabelle 1 - Attribute VideoWallApplication

Wie die Attribute auf GUI-Ebene eingesetzt werden, kann im Kapitel TODO User Environment Diagram nachgelesen werden.

##### PosterApplication



Abbildung 5 - Domain Model PosterApplication

Die *PosterApplication* verwaltet ihrerseits *Posters*. Ein *Poster* verfügt über folgende Attribute:

|  |  |  |
| --- | --- | --- |
| Attribut | Beschreibung | Beispiel |
| Authors | Die Namen der Autoren, welche das Poster erstellt haben. | Lukas Elmer, Christina Heidt, Delia Treichler |
| DatePublished | Das Datum der Publikation des Posters. | 24.05.2012 |
| DegreeCourse | Der Studiengang, für welchen das Poster erstellt wurde. | Informatik |
| Document | Das Poster selbst. | - |
| Name | Der Name der Arbeit. | HSR Videowall |

Tabelle 2 - Attribute PosterApplication

##### LunchMenuApplication



Abbildung 6 - Domain Model LunchMenuApplication

Die *LunchMenuApplication* verwaltet das *Lunchmenu*. Dieses verfügt über folgende Attribute:

|  |  |  |
| --- | --- | --- |
| Attribut | Beschreibung | Beispiel |
| Date | Das Datum des Lunchmenus. Dabei handelt es sich immer um das Datum des aktuellen Tags. | 22.05.2012 |

Tabelle 3 - Attribute LunchMenu

Das *LunchMenu* selber bietet verschiedene *Dishes* an. Ein *Dish* hat folgende Attribute:

|  |  |  |
| --- | --- | --- |
| Attribut | Beschreibung | Beispiel |
| Name | Der Name des Dishes. | Poulet im Kokosnussmantel auf Karottenstiften mit Salbei  Thai-Currysauce  Basmatireis  Fleisch aus der Schweiz |
| Price | Der Preis des Dishes. | INT 8.00 EXT 10.60 |
| Type | Der Typ des Dishes. | Tagesteller |

Tabelle 4 - Attribute Dish

#### Verfügbarkeit der Daten

##### Poster

Die Poster sind bei den jeweiligen Sekretariaten in digitaler Form vorhanden. Sie sind zu einem grossen Teil als PDF abgespeichert, einige wenige Poster sind PowerPoint-Präsentationen. Dies kommt daher, dass die Vorlage für das Poster eine PowerPoint-Präsentation ist.   
Die Poster, welche nicht im PDF-Format vorliegen, müssen manuell als PDF abgespeichert werden. (TODO: stimmt das oder implementieren wir das jetzt noch?)

##### Mittagsmenu

Die Daten des Mittagsmenus werden auf der Internetseite der Mensa HSR Hochschule Rapperswil der SVGroup[[3]](#footnote-3) abgerufen. Die benötigten Informationen zum Mittagsmenu werden aus dem HTML-Dokument herausgelesen und in der Applikation dargestellt.

### Graphical User Interface (GUI)

#### Empirischer formativer Test zur Eruierung der Navigationsart

Die Videowall wird mittels Gesten gesteuert. Um herauszufinden, welche Gesten Nutzer intuitiv benutzen würden, wurde ein Test durchgeführt. Dieser wurde als ein Wizard of Oz - Experiment durchgeführt (TODO: link auf Testdokumentation in Realisierung und Test). Der Test sollte auch zeigen, ob das erarbeitete GUI für den Benutzer einfach verständlich ist.

Es wurde folgendermassen vorgegangen:

* Am 14.03.2012 wurden erste Ideen zum GUI gesammelt.
* Nach dem Meeting vom 16.03.2012 wurden die am Meeting erhaltenen Inputs in die Test-Erarbeitung miteinbezogen.
* Am 27. März 2012 wurde der Test durchgeführt.

##### Ideensammlung

Am 14.03.2012 wurden Ideen zum GUI der Videowall gesammelt und Skizzen erstellt. Parallel dazu wurde überlegt, wie der Test ablaufen könnte.   
Bei der Sammlung von Ideen für die Applikation selbst wurden auch erste Vorschläge für einen Demomodus festgehalten (siehe Abbildung 7 - Anforderungen an den Test). Dieser wurde aber erst später ausgearbeitet (siehe Kapitel I.1.3.2 Demomodus), da er für den Wizard of Oz - Test zur Eruierung der Navigationsart noch nicht benötigt wurde.

Folgende Überlegungen wurden gemacht:

Der Test wird als PowerPoint-Präsentation vorbereitet und mit einem Beamer projiziert. Je nach dem, wohin in der Applikation die Testperson navigiert, wird eine andere Folie der Präsentation eingeblendet. Dazu bestehen keine fliessenden Übergänge, damit der Aufwand zur Erstellung des Tests klein gehalten werden kann. Der Testperson soll zusätzlich ein Laserpointer zur Verfügung stehen, mit welchem sie Schaltflächen anwählen kann, da dies nicht über Gesten allein möglich ist. In der Applikation würde dies später ähnlich gelöst werden, indem der Nutzer seine Hand als Pointer verwenden kann, um Schaltflächen zu aktivieren.



Abbildung 7 - Anforderungen an den Test

Die Abbildung 7 - Anforderungen an den Test zeigt, welche Anforderungen mit dem Test abgedeckt werden sollen. Der Test prüft das Browsen der Poster und die Navigation zwischen den verschiedenen Ansichten. In diesen werden beispielsweise die Poster, das Mittagsmenu der Mensa, das Wetter oder Informationen zu Veranstaltungen an der HSR dargestellt. Zwischen diesen soll einfach gewechselt werden können. Bei den Postern soll es zudem möglich sein, die Auswahl auf eine bestimmte Abteilung einzuschränken. Diese Anforderungen werden getestet, indem die Testpersonen ein oder mehrere für den Test erarbeitete Szenarien durchlaufen wird.

Die Nutzer sollen zu Beginn von der Videowall angelockt werden. Dies soll über einen Demomodus geschehen. Eine Idee dazu ist, dass dem Nutzer ein Titel eines Posters als Schriftzug folgt, sobald dieser den Bereich betritt, in dem er von Kinect erkannt wird. Es können maximal sechs verschiedene Schriftzüge zur gleichen Zeit sechs Personen folgen, da dies die maximale Anzahl an Personen ist, die Kinect gleichzeitig erkennen kann. Ein anderer Vorschlag ist, das Poster in Stücke zerschnitten darzustellen, wobei die einzelnen Stücke ungeordnet auf der Wall angezeigt werden und sich bewegen. Sobald nun jemand erkannt wird und sich diese Person zur Videowall hindreht, so vereinigen sich die Teile zu einem Poster. Im Test wird der Demomodus weggelassen, da es nicht möglich ist, ihn im Wizard of Oz - Experiment umzusetzen. Es ist bekannt, dass je nach Teaser die Interaktion mit der Videowall variieren kann.



Abbildung 8 - Posteransicht

Die Abbildung 5 - Posteransicht stellt die Ansicht der Poster dar. Links und rechts des aktuell angezeigten Posters sind Teile des vorangehenden und des nachfolgenden Posters sichtbar. Dies veranschaulicht dem Benutzer, dass noch mehr Poster existieren und es möglich ist, zwischen ihnen zu navigieren.   
Am oberen Rand befinden sich Schaltflächen, über welche zwischen den Ansichten gewechselt werden kann.   
In der Mitte des unteren Randes werden die Skelette der Personen, die von Kinect erkannt worden sind, angezeigt. Die Person, welche die Applikation steuern kann, wird gekennzeichnet.   
Die Interaktive Hilfe in der rechten unteren Ecke wird in der Testpräsentation nicht vorkommen. Mit dem Test kann auch validiert werden, ob die Steuerung genug intuitiv ist, dass eine Hilfe überflüssig ist.



Abbildung 9 - Unterteilung in Tabs

Die obenstehende Abbildung 6 - Unterteilung in Tabs zeigt auf, dass die einzelnen Ansichten (obere Tab-Reihe) weiter unterteilt werden können (untere Tab-Reihe). Befindet man sich nun bei in der Ansicht der Poster, dient die zweite Reihe Tabs dazu, dass die Auswahl an Postern auf eine bestimmte Abteilung (z.B. Informatik) eingeschränkt werden kann.



Abbildung 10 - Zonenmarkierung

Des Weiteren ist es für den Nutzer wichtig zu wissen, in welchem Abstand zur Videowall er stehen muss, um erkannt zu werden. Es sind zwei Zonen denkbar, eine Browsing- und eine Lese-/Interaktionszone. Wie in Abbildung 7 - Zonenmarkierung ersichtlich ist, könnten diese direkt am Boden vor der Videowall gekennzeichnet werden. Diese Markierungen würden zusätzlich auf die Videowall aufmerksam machen und der Nutzer wüsste immer, ob er im richtigen Bereich steht. Dieser Teil würde im Test mit Klebstreifen am Boden umgesetzt werden.

##### Ausarbeitung

Im Meeting vom 16.03.2012 wurde die Alternative, die Applikation nur mit der Hand als Zeiger (also ohne Gesten) zu bedienen, vorgeschlagen. Es gibt zwei Auffassungen der Steuerung von Kinect, die getestet werden können:

1. Konzept: Meine Hand ist die Maus

2. Konzept: Mein Körper ist die Maus (Steuerung mit Gesten)

Beim ersten Konzept könnte, wie bereits im Unterkapitel I.1.3.1.1 Ideensammlung beschrieben, ein Laserpointer genutzt werden, um den Pointer des Nutzers zu repräsentieren.

Bei der Diskussion über das zweite Konzept entstand die Idee, neben der Identifikation der Gesten zusätzlich die rechte Hand des Benutzers zu identifizieren, da gewisse Elemente nicht mit Gesten angesteuert werden können. Durch die Projizierung mit dem Beamer entsteht durch die Testperson ein Schatten an der Leinwand. Dieser könnte genutzt werden, um das Skelett zu simulieren. An der Stelle, wo nun der Schatten der Hand ist, könnte zusätzlich ein „Bällchen“ angezeigt werden. Dadurch wissen die Personen, wo ihre Hand ist und können diese an die von ihnen gewünschte Stelle bewegen. Problematisch ist hierbei, dass der Schatten wohl über die gesamte Applikation reichen müsste, damit der Nutzer alle aufgezeigten Elemente erreichen kann.

Die Markierungen am Boden, welche die Zonen umschreiben (siehe Abbildung 7 - Zonenmarkierung), könnten genauso gut in der Applikation selbst ersichtlich sein. Diese könnten unten in der Mitte zusammen mit dem Skelett angezeigt werden. So sieht der Nutzer immer, ob er sich in der richtigen Zone befindet.

Um die erarbeiteten Ideen sinnvoll zu testen, soll die Testapplikation interaktiv sein und daher sogleich als WPF-Applikation umzusetzen. Diese soll in etwa wie folgt aussehen:



Abbildung 11 - Skizze Testapplikation

Durch die oberen Tabs kann zwischen den verschiedenen Ansichten (Poster, Mensamenu) gewechselt werden. Befindet man sich in der Browsing-Zone, wird das Poster etwas kleiner dargestellt, dafür hat das Menu oben mehr Platz. Wechselt man in die Lese-Zone, so vergrössert sich das Poster und das Menu wird dafür kleiner. Die Zonen werden unten in der Mitte angezeigt. Dort befindet sich auch das Skelett, welches die Bewegungen der Person imitiert und angibt, in welcher Zone sie steht.

##### Umsetzung

Nicht alle in den Kapiteln I.1.4.1.1 Ideensammlung und I.1.4.1.2 Ausarbeitung beschriebenen Ideen konnten für die Testapplikation umgesetzt werden. Daher nachfolgend eine Zusammenfassung, was und was nicht für den Test umgesetzt wurde:

* Der Test wird als WPF-Applikation, nicht als PowerPoint-Präsentation, realisiert.
* Es wird kein Laserpointer, dafür werden die Handbewegungen des Benutzers mit der Maus simuliert.
* Es wird nur das Skelett des Benutzers angezeigt, auf Markierungen am Boden wird verzichtet.
* Für die Anzeige des Skeletts wird Kinect verwendet, jedoch nicht für die Steuerung.
* Der Demomodus wird nicht in diesen Test miteinbezogen und soll zu einem späteren Zeitpunkt umgesetzt werden.

Nachfolgend ist die Applikation, welche beim Test eingesetzt werden wird, genauer beschrieben und es wird gezeigt, wie sie gesteuert wird.



Abbildung 12 - Testapplikation

Die Testapplikation besteht aus zwei Ansichten. In der einen können Poster gelesen werden, in der anderen Ansicht wird das Mittagsmenu der Mensa angezeigt.

Die blauen Punkte in der Abbildung 12 - Testapplikation dienen der Beschriftung der einzelnen Komponenten in der Poster-Ansicht:

1. Das Menu. Hier kann zwischen den Ansichten (hier Poster und Mittagsmenu) gewechselt werden.
2. Der Navigationspfeil nach links. Er wird dazu benutzt, um nach links zum vorhergehenden Poster zu navigieren.
3. Der Navigationspfeil nach rechts. Er wird dazu benutzt, um nach rechts zum nachfolgenden Poster zu navigieren.
4. Die Hand. Sie symbolisiert die Hand der Testperson und befindet sich dort, wo die Testperson hinzeigt. Die Mauszeiger-Hand wird am Computer von den Testüberwachern bewegt (manuelle Steuerung), und zwar synchron zu den Bewegungen der Hand der Testperson.
5. Das Skelett der Testperson. Es dient dazu, der Testperson zu zeigen, dass sie erkannt wird und merkt, dass sie durch Körperbewegungen die Applikation steuern kann. Das Skelett wird mithilfe von Kinect angezeigt.

Damit eine Schaltfläche effektiv gedrückt wird, muss die Testperson ihre Hand eine Weile darüber halten. Dabei wir über der Mauszeiger-Hand ein Uhr-Symbol angezeigt. Dies dient der Testperson als Feedback, damit diese weiss, dass die Applikation die Geste erkannt hat.

##### Durchführung & Fazit

Am 27. März 2012 wurde der Test als Wizard of Oz - Experiment durchgeführt. Für weitere Informationen siehe (TODO link Testdokument Wizard of Oz Test).

#### Demomodus

Damit Personen, welche das Verwaltungsgebäude der HSR passieren, mit der Videowall interagieren, müssen sie zuerst einmal auf diese aufmerksam und auch von ihr angezogen werden. Zu diesem Zweck wird ein Demomodus, der die Aufmerksamkeit und das Interesse der Passanten auf sich lenkt, erstellt.

Bei der Erarbeitung des Wizard of Oz - Experiments (Unterkapitel I.1.4.1 Empirischer formativer Test zur Eruierung der Navigationsart) wurden bereits erste Ideen für einen Demomodus festgehalten. Trotzdem wurden nochmals Ideen gesammelt um eine möglichst breite Auswahl an Ideen für einen Demomodus zu erhalten.

##### Ideensammlung und Besprechung

Im Sprint 9, im Zeitraum vom 23. bis am 30. April 2012, überlegte jedes Teammitglied für sich allein, wie der Demomodus umgesetzt werden könnte und hielt die Ideen fest. Am 01.05.12 wurden die unterschiedlichen Ideen im Team diskutiert. Die Ideen werden nachfolgend erläutert.



Abbildung 13 - Demomodus, Ideen 1-3

Der Demomodus der Idee 1 zeigt eine Hand. Der Schriftzug fordert die Passanten auf, die Hand zu bewegen. Eine Schwierigkeit ist, dass der Benutzer nicht weiss, wie er die Hand bewegen kann. Zudem ist dieser Demomodus zu wenig attraktiv, da er statisch ist.

Bei der Idee 2 wird der Benutzer direkt angesprochen. Um herauszufinden, was die Videowall für Informationen zur Frage oder Aussage auf der Wall bietet, stellt sich der Nutzer auf die am Boden markierte Stelle. Somit kann er von Kinect erkannt werden und gelangt dann in die Applikation mit der Ansicht, welche im Zusammenhang mit dem zu Beginn gezeigten Begriff oder Spruch steht. Im Falle der Skizze (siehe Idee 3, Abbildung 9 - Demomodus, Ideen 1-3) fragt die Wall: „Hunger?“. Stellt sich nun eine Person an die bezeichnete Stelle, so wird das Mittagsmenu angezeigt.

Idee 3 zeigt ein Skelett, welches Passanten mit Winken oder anderen Gesten dazu auffordert, näher zu kommen. Das auf der Wall angezeigte Skelett befindet sich immer auf gleicher Höhe mit dem Benutzer. Bewegt sich der Nutzer also beispielsweise nach links, so bewegt sich das Skelett ebenfalls nach links. Sobald der Benutzer in einen bestimmten Bereich des Sensors eingetreten ist (in Abbildung 9 - Demomodus, Ideen 1-3 durch die zwei senkrechten Striche in der Mitte markiert), so übernimmt er das Skelett und die Applikation startet. Bei dieser Lösung besteht die Problematik, dass der Demomodus wenig Zusammenhang mit der Applikation selbst hat.



Abbildung 14 - Demomodus, Ideen 4-8

Bei der Idee Nummer 4 werden Objekte, verteilt über alle Monitore, dargestellt. Diese Idee ist an das Video über die interaktive Projektion einer Chanel Boutique in Paris[[4]](#footnote-4), welches bereits in der Konkurrenzanalyse (TODO: Ref Vorstudie Konkurrenzanalyse) referenziert wurde, angelehnt. Als Objekte sind hier beispielsweise Dreiecke, Puzzleteile oder kleine Stücke eines Posters denkbar. Wird nun ein Passant, welcher an der Wall vorbeiläuft, mit der Tiefenkamera erkannt, verschieben sie durch seine Bewegungen die Objekte auf der Wall. Werden mehrere Personen erkannt, welche beispielweise auch noch aus zwei verschiedenen Richtungen kommen, werden die Objekte von beiden Seiten verdrängt und bewegen sich in alle Richtungen. Bleibt der Benutzer über eine gewisse Zeitspanne vor der Wall stehen, so setzen sich die Teilchen zu einem Ganzen zusammen. Danach wird vom Demomodus in den Interaktionsmodus gewechselt.

Der Demomodus Idee 5 ist ein Vorhang, durch dessen schmale Öffnung ein Teil eines Posters sichtbar ist (diese Idee stammt vom Betreuer Markus Stolze). Die Öffnung des Vorhangs bewegt sich synchron mit der Position des Nutzers vor der Wall. Die Breite der Vorhangöffnung ist bestimmt durch den waagrechten Abstand der Hände des Benutzers. Die Vorzüge dieser Variante sind die einfach Implementation und die deutliche Verbindung zur Applikation selbst. Allerdings ist der Demomodus nicht sehr spannend und wird das Interesse der Passanten nur für kurze Zeit wecken können.

Idee Nummer 6 stellt das Skelett des Passanten, welcher bereits erkannt wurde, ganz gross auf der Wall dar. Kann kein Benutzer erkannt werden, wird nichts auf der Wall dargestellt, was der Nachteil dieser Lösung ist. Das grosse Skelett imitiert alle Bewegungen des Nutzers. Es wird so unmissverständlich klar, wie die Steuerung der Applikation vor sich geht. Nach Ablauf eines Timers wechselt die Anzeige zur eigentlichen Applikation.

Bei der Idee 7 wird die Applikation im Hintergrund schwach angezeigt, davor befindet sich Nebel. Durch Wischbewegungen des Benutzers kann der Nebel entfernt werden und die eigentliche Applikation kommt dahinter zum Vorschein.

Die Idee 8 für den Demomodus zeigt einen Lock-Screen mit einem Vorhängeschloss, analog zu dem eines Smartphones (TODO ref <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=8,046,721.PN.&OS=PN/8,046,721&RS=PN/8,046,721>). Durch das Vorbeilaufen an der Wall, durch näher kommen oder weiter weg gehen kann die Applikation entsperrt werden. Dabei öffnet sich dann der Screen, auf wessen Symbol das Schloss beim Entsperren geschoben wurde. Läuft ein Passant nun von rechts nach links an der Videowall vorbei, so schiebt er das Vorhängeschloss auf das Symbol mit dem Teller und so zeigt die Wall das Mittagsmenu der Mensa an.  


Abbildung 15 - Demomodus, Idee 12, Erweiterung zu Idee 8

Die Idee 12 aus obiger Abbildung zeigt eine Weiterentwicklung der Idee 8. Am Boden wird eine Markierung angebracht, welche einem besser verständlich macht, dass man die Wall steuern kann, indem man seine Position ändert. Dieses Kreuz am Boden bietet nun eine zusätzliche Navigationsmöglichkeit zum simplen Steuern durch die Hand (siehe dazu Kapitel I.1.4.1 Empirischer formativer Test zur Eruierung der Navigationsart). Durch farbliche Kennzeichnung der Schaltflächen der Applikation können diese auch durch Positionsänderungen analog der Markierung am Boden betätigt werden, und nicht mehr nur alleine durch das Nutzen der Hand als Pointer.   
Die Erweiterung der Steuerung durch die Nutzung der Markierungen am Boden kann auch für andere Ansichten der Applikation verwendet werden.



Abbildung 16 - Demomodus, Ideen 9 und 10

Bei der Idee 9 wachsen oder erscheinen im Demomodus Pflanzen, Roboter und andere Gegenstände, die thematisch mit den Studiengängen an der HSR zu tun haben. Wird ein Passant erkannt, so reagiert einer der Gegenstände auf ihn. Bleibt ein Passant nach der Erkennung stehen, so verschwinden alle Gegenstände und die Applikation kommt zum Vorschein. Als Alternative könnten auch Seifenblasen auf der Wall dargestellt werden. Wird eine Person erkannt, so kann diese durch Bewegungen diese Seifenblasen zerplatzen lassen. Der Aufwand für die Implementierung einer dieser Ideen ist jedoch gross, und zudem würden sich die Ideen auch eher als Minigame anstatt als Demomodus eignen.

Der Demomodus der Idee 10 zeigt verschiedene Schriftzüge, beispielsweise die Titel von zufällig ausgewählten Postern. Sobald ein Passant erkannt wurde, sammeln sich diese Schriftzüge um die Hand des Benutzers.



Abbildung 17 - Demomodus, Idee 11

Die Idee 11 dient vor allem der korrekten Positionierung des Benutzers vor der Wall. Passanten, welche weit entfernt von der idealen Position stehen, werden auf der Wall als kleine, durchsichtige Skelette (hier sind auch alternative Anzeigen denkbar) repräsentiert. Je näher man zur Mitte steht, desto grösser und deutlicher zeigt die Wall das Skelett an. Eine Markierung am Boden soll dem Benutzer helfen, sich ideal zu positionieren. Ein Nachteil dieser Idee ist der fehlende Zusammenhang zwischen dem Demomodus und der Applikation.

##### Auswahl der besten Idee für den Demomodus

Nach dem Zusammentragen und Besprechen aller Ideen am 01.05.12, wählte das Team die folgenden drei Ideen zur weiteren Vertiefung aus: Nummer 2, 4 und 8 resp. 12.

Am 02.05.12 wurden diese drei Ideen auch noch mit Markus Stolze besprochen. Dabei kam zur Sprache, dass sich die Idee 8 resp. 12 nicht eignet, da durch den positiv ausgefallenen Usability Test (I.1.4.1 Empirischer formativer Test zur Eruierung der Navigationsart) die Steuerung auf „Die Hand als Maus“ festgelegt wurde.   
Der Demomodus aus den Ideen 8 und 12 bringt mehrere Schwierigkeiten mit sich. Das Angebot von zwei Navigationsmöglichkeiten (Hand und Markierung am Boden) kann verwirrend sein. Die Sperrung der Wall mit einem symbolischen Vorhängeschloss wirkt sich eher negativ auf das mögliche Interesse der Benutzer aus, da das Schloss als Interaktions-Verbot aufgefasst werden könnte. Bei einem Smartphone macht solch eine Sperrung durchaus Sinn, damit nicht unabsichtlich irgendwelche Schaltflächen betätigt werden. Bei der Wall ist dies hingegen nicht nötig. Zudem ist die Anzahl der Favoriten-Programme in dieser Ansicht mit dem Lock-Kreuz auf vier Stück beschränkt.

Als Favorit zur Umsetzung wurde die Idee 4 gewählt:  
Die Idee 4 mit den Objekten, die durch Bewegungen von Passanten durcheinander gewirbelt werden, bringt viel Bewegung und hat daher eine grosse Anziehungskraft. Des Weiteren ist kein Skeletal Tracking nötig, zur Umsetzung wird der Tiefensensor genutzt. Die Problematik der verzögerten Erkennung des Skeletts eines Passanten besteht hier also nicht. Erschwerend ist hier nur die eher aufwändige Implementation der Idee. Trotzdem soll diese Idee als Demomodus umgesetzt werden.

Die Idee 2 ist die Alternative, falls die Umsetzung der Idee 4 aus Zeitgründen nicht klappen sollte:  
Die Idee 2 mit dem Anzeigen eines Begriffes oder Spruches hat den Vorteil, dass sie ohne grossen Aufwand implementiert werden kann. Zudem hat der Demomodus konkret etwas mit der Applikation selbst zu tun. Nachteilig erweist sich, dass dieser Modus wenig Dynamik hat. Diese Idee wird daher als Alternative zur Idee 4 beibehalten.

Die anderen Design-Vorschläge zu den Demomodi (siehe I.1.4.2.1 Ideensammlung und Besprechung) wurden hier so ausführlich dokumentiert, damit allfällige weitere Arbeiten hierauf aufsetzen können.

Die Umsetzung des Demomodus ist im Kapitel (TODO: Ref Umsetzung im Entwurf) festgehalten.

#### Screen Map



Abbildung 18 - Screen Map

Die Screen Map bezieht sich auf die im Kapitel I.1.2.1 Domain Models vorgestellten Klassen und Attribute.  
Solange kein Skelett erkannt wird, befindet sich die Applikation im Demomodus (mehr Informationen hierzu im Kapitel I.1.3.2 Demomodus) und der Demomodus Screen wird angezeigt. Auf diesem wird der *DemomodeText* einer zufällig ausgewählten *VideoWallApplication* dargestellt, diese Applikation und somit der Text wechseln periodisch. In Fall der Abbildung 17 - Screen Map handelt es sich bei der *VideoWallApplication* um die *LunchMenuApp*.  
Sobald ein Skelett erkannt wird, wechselt die Anzeige auf der Videowall vom Demomodus in den Interaktionsmodus. Im Interaktionsmodus Screen können über das Menu die verschiedenen *VideoWallApplications* angezeigt werden. Für jede *VideoWallApplication* wird ein „Knopf“ erstellt, mit dessen Betätigung die jeweilige *VideoWallApplication* ausgewählt werden kann. Der Inhalt der momentan ausgewählten *VideoWallApplication* wird in der Mitte des Screens dargestellt, im Beispiel der Abbildung 17 - Screen Map ist das der Inhalt der *LunchMenuApp*. Bei einem Ansichtswechsel auf die *PosterApp* würde in der Screen-Mitte der Inhalt der *PosterApp* visualisiert werden.

TODO: Wie ändert sich diese Beschreibung nach der Diskussion mit der Kommunikationsstelle?

#### Design Entscheide

Die Design Entscheide für die verschiedenen Applikationen werden nachfolgend aufgelistet.

##### Steuerung mit der Hand

Das externe Design der Videowall-Applikation und der Poster-Applikation (Plug-in) wurde mehrheitlich aus dem ersten Usability Test (TODO Verlinkung Wizard of Oz-Test Realisierung und Tests) übernommen, da die Nutzer das dort verwendete Design gut verstanden. Um eine Schaltfläche auswählen zu können, soll der Nutzer eine Zeitlang mit dem Handcursor darauf verweilen. Damit das vom Nutzer verstanden wird, wird zusätzlich eine Animation gestartet, sobald der Nutzer sich auf der Schaltfläche befindet.

Abbildung 19 - Handcursor Animation

Aus dem Test hat sich ergeben, dass die Applikation nicht nur mit der rechten, sondern auch mit der linken Hand bedienbar sein soll. Deshalb wurde ein zweiter Handcursor für die linke Hand erstellt, welcher angezeigt wird, sobald die Applikation mit der linken Hand bedient wird.

##### Demomodus

In der Ideensammlung für den Demomodus (I.1.3.2.1 Ideensammlung) wurde das Design des Demomodus grob definiert. Der Hintergrund sollte eine auffällige Farbe haben. Auf dem Hintergrund soll ein Teasertext angezeigt werden und darunter die Aufforderung „Komm näher“. Die Farbe und der Text dienen dem Anlocken der Passanten, damit diese darauf mit der Videowall interagieren. Sobald der Nutzer von Kinect erkannt wurde, beginnt ein Timer herunterzuzählen. Nach dessen Ablauf wechselt die Applikation in den Interaktionsmodus und der Benutzer kann die Applikation steuern.

Der Demomodus wurde nach dessen Implementation mit einem Usability Test geprüft (siehe TODO Usability-Test Demomodus). Aus dem Test konnte folgender Schluss gezogen werden: Wenn ein Passant im Demomodus von Kinect erkannt wird, soll beim Anzeigen des Timers gleichzeitig das Skelett des Nutzers angezeigt werden, damit er neugierig wird und stehen bleibt. Er wird sofort erkennen, dass er das Skelett ist und somit nach dem Wechseln in den Interaktionsmodus die Applikation problemlos bedienen kann.  
Dieser Redesign Entscheid wurde sogleich nach der Durchführung des Tests implementiert.

##### Menu mit Tabs

Das Menu der Applikation besteht aus Tabs. Diese Tabs stellen Elemente dar, welche die meisten Nutzer von Browserapplikationen her kennen und daher verstehen sollten. Dieser Sachverhalt wurde beim Test (TODO Test 4 Grafisches Design) validiert. Nach dem Test wurden noch farbliche Anpassungen vorgenommen, welche dem Nutzer verdeutlichen, welches Tab gerade aktiv ist.

Zwei weitere Punkte, welche aber im Rahmen der Bachelorarbeit nicht umgesetzt werden können, ergaben sich aus dem Usability Test. Der Handcursor dreht sich über jedem anwählbaren Element, was für den Nutzer verwirrend sein kann. Im Menu dreht sich der Handcursor also beispielsweise auch, wenn er sich über dem bereits aktiven Tab befindet. Auch soll der Handcursor, wenn er sich über einem nicht anwählbaren Element befindet, abgeschwächt oder durchgestrichen dargestellt werden. Diese zwei Anforderungen wurden in Form von User Stories in den Backlog (TODO Backlog) aufgenommen.

##### Corporate Design HSR

Das Corporate Design der HSR [hsr2011] legt verschiedene Farben, Schriftarten sowie die Verwendung des HSR Logos fest. Diese Vorgaben wurden für die Videowall so gut als möglich eingehalten.

Am 05.06.12 wurde das Design mit Michael Lieberherr und Oliver Kirchhofer von der Kommunikationsstelle der HSR besprochen. Aus dem Gespräch ergaben sich folgende zwei Änderungen, die sogleich umgesetzt wurden:

* Verwendung der Schriftart Arial ist in Ordnung
* Pfeile in der Poster-Applikation kleiner machen (Iceberg Buttons)

Das daraus resultierende Externe Design wird im Kapitel I.1.3.5 Externes Design aufgezeigt.

Es wurden noch weitere Design Vorschläge gemacht, welche bei einer Weiterentwicklung der Applikation umgesetzt werden könnten:

* Das HSR Logo kommt oben links hin. Der Hintergrund des Logos muss weiss sein, dies bedarf Anpassungen im Menu.
* Der Abstand zwischen den Tabs im Menu soll grösser sein.
* Es sollen Gesten verwendet werden können, damit für das Browsen der Poster keine Pfeile mehr angezeigt werden müssen.
* Das Menu soll einem Home-Screen, auf welchem das HSR Logo platziert wird, weichen. Von da aus können die Inhalte der Videowall durch das Anwählen der Buttons angeschaut werden. Über einen Home-Button im unteren Bereich neben dem Skelett kommt man zurück zum Home-Screen.

#### Externes Design

Nach den in Kapitel I.1.3.4 Design Entscheide aufgelisteten Design Entscheiden, wurde das Externe Design erarbeitet.

##### Videowall-Applikation

Oben befinden sich die Tabs. Aktiviert man eines davon, so wird in die entsprechende Applikation gewechselt. Ist das Tapp aktiviert, so ist dieses weiss. Die nicht aktiven Apps sind blau hinterlegt. In der Mitte wird die aktive Applikation angezeigt. Unten links wird zu jeder Zeit das HSR Logo angezeigt. In der Mitte des unteren Bereichs wird das Skelett des Nutzers in blauer Farbe angezeigt. Der Cursor ist als Hand dargestellt. Wird mit der rechten Hand bedient, so wird ein Icon mit einer rechten Hand angezeigt. Wird mit der linken Hand bedient, so ist der Cursor eine linke Hand.



Abbildung 20 - Externes Design, Videowall-Applikation

##### Poster-Applikation (Plug-in)



Abbildung 21 - Externes Design, Poster-Applikation (Plug-in)

Die Poster-Applikation zeigt in der Mitte jeweils ein Poster an. Mit den Pfeilen links und rechts davon kann zum vorherigen oder zum nächsten Poster navigiert werden. Die Pfeile sind in Blau gehalten.

##### Mittagsmenu-Applikation (Plug-in)

Die Abbildung 19 - Externes Design, Videowall Applikation zeigt auch die Mittagsmenu-Applikation. Diese wurde sehr schlicht gehalten. Besonders relevante Texte, wie das Datum und die Kategorie, wurden im Schriftbild hervorgehoben.

##### Demomodus

Für den Demomodus „Teaser“ (siehe I.1.3.2.2.2 Umsetzung des Demomodus „Teaser“) wurde ein externes Design erarbeitet. Sobald der Demomodus aktiv ist, wird auf der Videowall eine zufällig ausgewählte Farbe gezeigt.

In der Mitte wird jeweils der Teaser-Text der im Hintergrund aktiven Applikation angezeigt. Dies könnte beispielsweise die Mittagsmenu-App sein, was dann dazu führt, dass der entsprechend passende Text: „Hunger?“ angezeigt wird. Unterhalb dieses Textes befindet sich noch ein Zusatztext, welcher die Passanten dazu animieren soll, sich der Wall zu nähern.



Abbildung 22 - Externes Design, Teaser-Text

Sobald sich ein Passant der Wall genähert hat und dessen Skelett erkannt wurde, beginnt ein Countdown und das erkannte Skelett wird angezeigt.



Abbildung 23 - Externes Design, Countdown

Der Countdown ist dazu da, dass der Nutzer einerseits eine Rückmeldung auf sein Näherkommen erhält und anderseits darüber informiert wird, wie lange er noch warten muss, bis es weiter geht. Zusätzlich wurde nach einem Usability Test (TODO Dok Tests) das Skelett visualisiert, damit dem Nutzer klar ist, dass er von der Applikation erkannt wurde. Ist der Countdown bei 0 angekommen, wird vom Demomodus in den Interaktionsmodus gewechselt. Auch wird dem Nutzer schon zu diesem Zeitpunkt der Handcursor angezeigt, damit die Verwendung von ihm in der Wartezeit erlernt werden kann.

Entfernt sich ein Nutzer vor Ablauf des Countdowns von der Wall, so wird wieder der Teaser-Text angezeigt.

#### Guidelines

Die „Microsoft for Kinect Human Interface Guidelines“[[5]](#footnote-5) definieren eine Reihe von Prinzipien. Dieses Dokument war jedoch erst ab dem 21.05.2012 online verfügbar. Die Bachelorarbeit startete aber bereits am 20.02.2012, daher war bis zu diesem Zeitpunkt schon ein Grossteil der Applikation umgesetzt. Trotz dem Fehlen der Guidelines zu Beginn der Bachelorarbeit konnten die Anforderungen und Empfehlungen aus den Guidelines erfüllt werden, was auf die zahlreich durchgeführten Usability Tests (siehe TODO: Link) zurückzuführen ist.

Die Guidelines sind in den nachfolgenden Kapiteln kurz zusammengefasst und es ist beschrieben, wie diese im Projekt umgesetzt wurden.

##### Human Interface Guidelines

###### Best Practices for Designing Interactions

Das Kapitel beschreibt, wie Stimme und Gesten für die Steuerung von Kinect verwendet werden. Unter dem Wort „Gesten“ versteht das Team bestimmte Abläufe von Bewegungen. Die Guidelines bezeichnen aber auch eine simple Handbewegung (nicht bestimmt in welche Richtung) als Geste: „*Basic Gesture Types Gestures can take many forms, from using your hand to target something on the screen, to specific, learned patterns of movement, to long stretches of continuous movement.“*1.   
Im Falle der Videowall werden lediglich *„Innate Gestures“*1 verwendet, welche wie folgt beschrieben werden: *„Innate gestures are ones that the user intuitively knows or that make sense based on the users’ understanding of the world.“*1. Dies trifft auf das Konzept „Meine Hand ist die Maus“ (siehe TODO Usability Test) zu.   
Weiter ist festgehalten, dass ein visuelles Feedback vorhanden sein soll. Der Nutzer soll unter anderem wissen, ob er von Kinect erkannt wurde, ob er die Kontrolle hat und ob er am richtigen Ort steht. Diese Anforderungen werden alle durch die Darstellung des Skeletts und der Hand gewährleistet.   
Auch soll die Verwendung von Kinect aus einer gewissen Distanz geschehen. Die Videowall hat eine imposante Grösse. Damit diese überschaubar ist, muss der Nutzer automatisch ein paar Meter Abstand nehmen.   
Gesten sollen immer mit beiden Händen (ob rechts oder links) ausführbar sein. Dies ist mit dem Handcursor möglich.  
*„Whether using gesture, or voice, or both, providing good feedback is critical to making users feel in control and helping them understand what’s happening.”*1 – Sobald sich der Handcursor auf einem auswählbaren Element befindet, wird ein Fortschrittskreis um die Hand angezeigt. Auswählbare Elemente sind zudem immer in der Farbe Blau (TODO siehe Externes Design) gehalten.

###### Basic Interactions

In diesem Kapitel wird beschrieben, wie Interaktionen gestaltet werden sollen.   
Es besagt, dass Interaktionen abbrechbar sein sollen. Bewegt man den Handcursor von einer auswählbaren Schaltfläche weg, bevor der Fortschrittskreis vollständig ausgefüllt ist, verschwindet die Fortschrittsanzeige wieder. Auch das Wechseln vom Demomodus in den Interaktionsmodus (siehe I.1.3.2.2.2 Umsetzung des Demomodus „Teaser“) kann abgebrochen werden, indem sich der Nutzer aus dem von Kinect erkennbaren Bereich entfernt.   
Im Kapitel steht weiter, dass die Inputmethode möglichst praktisch sein soll für die gestellten Aufgaben. Die Schaltflächen sind gross genug, damit sie problemlos mit dem Handcursor angewählt werden können.  
Weiter wird die Zone für das Handtracking angesprochen: *„The most common way of targeting, and the way targeting is done for Kinect on Xbox360 is with a cursor visual that is controlled by hand movement. The simplest way of implementing a cursor is to define a Physical Interaction Zone and just do a direct mapping of the horizontal and vertical position of the hand in the Zone to the screen. This is how the Xbox360 cursor is implemented.”*1 – dies beschreibt die genau gleiche Idee, wie sie auch für dieses Projekt für das Handtracking (TODO Dokument Entwurf) umgesetzt wurde.  
Wie in den Guidelines empfohlen, wurde der Jitter beim Handcursor, also das Ruckeln der Hand auf dem Monitor auch ohne Bewegung des Nutzers, verringert.   
Auch wurde für das Konzept „Meine Hand ist die Maus“ ein Cursor in Form einer Hand genommen. Verwendet der Nutzer die linke Hand, so wird der Handcursor als linke Hand dargestellt und umgekehrt. Auch dies wird in den Guidelines vorgeschlagen.

###### Distance-Dependent Interactions

Das Kapitel beschreibt verschiedene Interaktionszonen und was für Interaktionen sich in den jeweiligen Zonen am besten eigenen. Für die Videowall wurden nur der *„Far(2.0-4.0 Meters)“*1- und der *„Out of Range (>4 Meters)“*1-Bereich verwendet. Im Bereich mit mehr als 4 Metern Abstand vom Sensor soll der Nutzer animiert werden, näher zu kommen: *„Your UI should focus on informing users that there is an interesting interaction available and enticing them to move closer. Visuals must be very large and simple”*1 – Der Demomodus verfügt über eine Hintergrundfarbe und einen grossen Teasertext (siehe I.1.3.5 Externes Design) und erfüllt daher dieses Kriterium.

###### Multimodality

Dieses Kapitel beschäftigt sich mit dem Einsatz von mehreren Inputmethoden – einerseits die Stimme, andererseits Gesten. Da für die Steuerung der Videowall-Applikation lediglich Gesten verwendet werden, wird auf dieses Kapitel nicht weiter eingegangen.

###### Multiple Users

Das Kapitel „Multiple Users“ beschäftigt sich mit der Möglichkeit mehrere Nutzer zu tracken. Für diese Arbeit wurde das Single „Driver Model“ übernommen: *„This model assigns one of the users as the “driver” at any given time and only registers actions taken by that user. The driver role can be selected or transferred in a number of ways, including choosing the first user to engage, or the user that is closer to the sensor. This is one way to avoid conflicting inputs. This model is usually indicated by having visuals that show which person is being tracked and only having one cursor on the screen at any time.“*1. Durch das dargestellte Skelett ist klar, welche Person die Applikation gerade bedient. Zudem gibt es nur einen Handcursor.

### Installation, Betrieb und Administration der Videowall

Im folgenden Kapitel wird beschrieben, wie die Videowall gewartet werden soll. Dazu gehören einerseits der allgemeine Betrieb, sowie andererseits die Administration der Inhalte der Videowall.

Dieses Kapitel beschreibt den aktuellen Stand und muss bei einer Weiterführung des Projektes nochmals überarbeitet und verfeinert werden.

#### Installation

##### Hardware

Die Installation der Monitore wird von einer externen Firma organisiert. Dazu müssen noch genauere Abklärungen gemacht werden.

Beim Testsetup wurden zwei Matrox Grafikkarten in einen Computer eingebaut und installiert. Mehr dazu kann im Kapitel Hardware Evaluation und Test nachgelesen werden(TODO: ref).

##### Software

Die Software ist die gleiche wie diejenige des Testhardware Setups, eine Beschreibung dazu kann im Kapitel Software (TODO: ref) gefunden werden.

#### Betrieb

Da die Videowall in der Nacht nicht genutzt wird, ist es möglich, die Wall in dieser Zeit auszuschalten. Dadurch kann der Stromverbrauch gesenkt und die Abnutzung der Hardware, speziell der Monitore, verringert werden.

Folgendes Vorgehen wird vorgeschlagen:

Da um etwa 7.30 Uhr die ersten Personen an der HSR eintreffen, wird der Videowall PC um etwa 7 Uhr über eine Zeitschaltuhr hochgefahren. Damit verbleiben noch 30 Minuten für allfällige Updates und den Systemstart.

Falls eine neue Version der Software oder der Plug-ins existiert, wird ein automatisches Deployment durchgeführt. Die Informationen und Dateien für dieses Deployment werden automatisch von einem Deployment Server (beispielsweise Team Foundation Server[[6]](#footnote-6)) heruntergeladen. Sollte beim Deployment etwas schief gehen, wird der Videowall PC wieder heruntergefahren.

Nach dem automatischen Deployment wird die Videowall-Applikation gestartet und die Plug-ins werden geladen. Die Bildschirme der Videowall um etwa 7.30 Uhr eingeschaltet. Die Videowall läuft dann den ganzen Tag lang ohne Unterbruch. Sollte ein Fehler auftreten, wird zuerst versucht, die Applikation neu zu starten. Schlägt dies fehl, so wird die Videowall heruntergefahren.

Sobald die Videowall nicht mehr gebraucht wird (ca. um 20 Uhr) wird der Videowall PC heruntergefahren und die Bildschirme ausgeschaltet.

Sollte zu irgendeinem Zeitpunkt ein Fehler auftreten, so wird automatisch ein Mail mit dem Log und einem Stack Trace generiert und an die Verantwortlichen geschickt, um herauszufinden, um für einen Fehler es sich handelt. Zusätzlich wird die Videowall von einem externen Tool (Bsp. Nagios[[7]](#footnote-7)) überwacht, welches bei einem Problem ebenfalls die Verantwortlichen benachrichtigt.

Abbildung 24 - Betrieb der Videowall

#### Zu verwaltende Daten

Bis zum Ende des Projektes sind folgende zu verwaltende Inhalte vorgesehen:

##### Plug-ins

Plug-ins sind Applikationen, die auf der Videowall dargestellt werden können und müssen verwaltet werden.

Durch das Plug-in System können Studenten Innovation in die Applikation einfliessen zu lassen. Da die Videowall aber an einem prominenten Ort steht und die HSR direkt repräsentiert, ist es wichtig, dass nicht beliebige Inhalte auf der Videowall publiziert werden (z.B. gewaltverherrlichende oder erotische Inhalte). Ein weitere Problematik besteht darin, dass durch die Plug-ins die Stabilität der Videowall negativ beeinträchtigt werden könnte.

Um also qualitativ hochwertige und politisch korrekte Plug-ins sicherzustellen, ist es notwendig, die interessierten Studenten auf gewisse Restriktionen und Regeln aufmerksam zu machen. Es wird vorgeschlagen, dass die Studierenden nach der Entwicklung eines Plug-ins zu einem Code Review eingeladen werden, bei dem der Quellcode des Plug-ins analysiert wird. Zusätzlich müssen die Studenten mit ihrem Namen dafür bürgen, dass durch ihre Erweiterung keine politisch unkorrekten Inhalte auf der Videowall erscheinen. Sollte dies doch passieren, sind im Vorhinein Massnahmen zu definieren, welche bei einer Verletzung der Vorschriften eingeleitet werden.

Folgender Ablauf ist für das Erstellen und Deployen eines Plug-ins denkbar:

Abbildung 25 - Initialer Deployment Prozess

##### Poster-Applikation

Jedes Semester entstehen neue Bachelorposter, welche für die Videowall aufbereitet werden müssen. Es ist noch nicht bestimmt, wer diese Aufgabe übernehmen wird.   
Folgender Ablauf ist denkbar:

1. Die Bachelorposter werden von den Studierenden erstellt.
2. Die Studiengangleiter sind verantwortlich, diese Poster in elektronischer Form entgegenzunehmen und inhaltlich zu kontrollieren.
3. Die Poster werden von den Studiengangleitern dem Sekretariat der HSR übergeben.
4. Das Sekretariat pflegt die Inhalte über ein CMS Interface in die Videowall-Applikation ein.
5. Berichtigungen können dem Sekretariat gemeldet werden, welches die Korrekturen ins CMS Interface einpflegt.

##### SV Group Mensa Mittagsmenu

Das Mittagsmenu ist an den Wochentagen auf der Internetseite der SV Group verfügbar (Typo3 CMS der SV Group) (siehe Unterkapitel I.1.2.2.2 Mittagsmenu). Leider bietet die SV Group keine Schnittstelle ausser der HTTP/HTML-Version für das Menu an. Um das Mittagsmenu trotzdem aktuell zu halten, wird der Menuplan beim Start der Applikation in HTML heruntergeladen, die nötigen Informationen herausgelesen und in einer Form zwischengespeichert, die sich für WPF eignet. Die HTTP/HTML-Schnittstelle ist technisch gesehen labil. Eine Design-Änderung an der Website der SV Group könnte dazu führen, dass das Menu nicht mehr richtig eingelesen werden kann. Daher wurden spezifisch für das Mensamenu Unit Tests geschrieben. Sie bieten dem Entwickler die Möglichkeit, die Schnittstelle einfach und schnell zu testen. Dazu wird eine aktuelle Version des HTML heruntergeladen und im Unit Test eingebunden.

Um das Menu zu aktualisieren, gibt es zwei Möglichkeiten:

* Die Applikation wird neu gestartet.
* Es wird ein Cronjob oder ein (Dispatcher)Timer mit hohem Aktualisierungsintervall programmiert, der beispielsweise alle zwei Stunden das Mittagsmenu aktualisiert.

#### Datenverwaltung der Plug-ins

Jedes Plug-in kann über eigene Daten verfügen. Die Poster-Applikation benötigt beispielsweise die Bilder der anzuzeigenden Poster. Das Framework könnte ein Interface zur Verfügung stellen, über welches die Daten der Plug-ins verwaltet werden können. Die Entwickler eines Plug-ins definieren die Objekte, welche verwaltet werden sollen. Das Framework generiert dann automatisch eine Benutzeroberfläche für deren Bearbeitung.

Alternativ könnte jedes Plug-ins eine eigene Administrationsoberfläche anbieten. Da bei dieser Variante Funktionen (beispielsweise das Speichern der Daten in einer Datenbank) redundant programmiert werden müssten und die Bedienung nicht einheitlich wäre, ist sie jedoch weniger geeignet.

#### Administrationsoberfläche

Für die Administrationsoberfläche der Inhalte gibt es verschiedene Möglichkeiten. Die drei wichtigsten sind in den nachfolgenden Unterkapiteln kurz beschrieben. Vor der Umsetzung ist die ausgewählte Lösungsmöglichkeit mit der Kommunikationsstelle zu validieren.

Das Bachelorteam hätte bei genügend Zeit die Lösungsvariante, welche im Unterkapitel I.1.4.5.1.2 Web Interface und Typo3 Extension mit Iframe beschrieben ist, für die Videowall umgesetzt.

##### Administration über Typo3 CMS [[8]](#footnote-8)

Das Sekretariat der HSR arbeitet bereits mit einem Typo3 CMS[[9]](#footnote-9). Eine Integration der Administration der Videowall in dieses System wäre eine Option. Dabei könnte auf zwei Arten vorgegangen werden:

###### Typo3 Extension mit Typo3 Datenbank

Es wird eine Typo3 Extension[[10]](#footnote-10) für den Administrationsbereich entwickelt. Diese Applikation wird sehr einfach gehalten und zeichnet sich vor allem durch XML Konfigurationen aus, die das Datenbankschema beschreiben. Durch die Installation der Extension im Typo3 wird die Datenbank automatisch erzeugt und die Inhalte können sofort über den Administrationsbereich von Typo3 bearbeitet werden. Die grafische Oberfläche des Administrationsbereichs wird automatisch vom Framework generiert und sieht so wie eine gewöhnliche Typo3 Extension aus. Gewisse Standardfunktionalitäten wie die Archivierungsoption oder das Ausblenden von einzelnen Datensätzen werden ebenfalls durch das Framework angeboten.

Der Nachteil an dieser Lösungsvariante ist, dass die Daten von der Typo3 Datenbank zur Videowall migriert werden müssen. Eine Synchronisation des unidirektionalen Informationsflusses von der Typo3 Datenbank zur Videowall kann aber auf einfache Art per Cronjob (z.B. alle 15 Minuten) eingerichtet werden.

Diese Variante eignet sich gut, wenn alle Personen, die an der Videowall etwas ändern müssen, Zugriff auf Typo3 haben. Speziell für das Sekretariat ist diese Art von Interface einfach zu bedienen, es wird bereits täglich benutzt.

###### Web Interface und Typo3 Extension mit Iframe

Wie in der ersten Lösungsvariante (siehe Unterkapitel I.1.4.5.1.1 Typo3 Extension mit Typo3 Datenbank) ist auch hier eine Typo3 Extension vorgesehen. Dieses Mal wird die Extension aber so erstellt, dass nur ein Iframe programmiert wird, das auf einen anderen Web Server verweist. Somit kann die Administrationsoberfläche Typo3-technologieunabhängig entwickelt werden, zum Beispiel mit ASP.NET MVC3.

Die Hauptvorteile dieses Ansatzes sind, dass die Administrationsoberfläche nicht mit Typo3 programmiert werden muss und trotzdem ins Typo3 integriert ist. Die Administrationsoberfläche kann auch ohne Typo3 bearbeitet werden, gegebenenfalls mit einem SSO. Die Implementierung dieser Lösungsvariante ist allerdings etwas zeitaufwändiger.

##### Administration über Web Server

Ähnlich wie in der zweiten Typo3-Lösungsvariante I.1.4.5.1.2 Web Interface und Typo3 Extension mit Iframe beschrieben ist, wird bei dieser Lösung auf einem Web Server (z.B. mit ASP.NET MVC3) eine Administrationsoberfläche entwickelt, die gegebenenfalls mit dem SSO der HSR gekoppelt wird. Als Transportprotokoll dient HTTPS/HTML.

Die Vorteile liegen darin, dass das System klar von anderen Applikationen abgegrenzt ist. Auch ist es einfach möglich, eine mobile Applikation mit HTML5 zu entwickeln. Als Nachteil ist jedoch aufzuführen, dass ohne Typo3 Extension die Benutzer auf eine separate URL zugreifen müssen und ihnen das System nicht sofort bekannt vorkommt. Für diese Variante könnte statt ASP.NET auch Silverlight eingesetzt werden.

##### Administration über WPF-Applikation

In dieser Variante geht es darum, einen WPF Client zu schreiben, mit dem die Inhalte bearbeitet werden können. Als Transportprotokoll würde WCF eingesetzt werden.

Bei dieser Variante können grosse Teile aus den Daten- und Serviceschichten (TODO: Ref Dokument Entwurf, Kapitel Logische Sicht) der bestehenden Software wiederverwendet werden, was ein Vorteil ist. Nachteilig ist, dass durch WPF die Plattform eingeschränkt wird und die Entwicklung einer mobilen Applikation so nicht möglich ist.

Diese Variante wird nicht empfohlen.

1. <https://www.windowsazure.com/> [↑](#footnote-ref-1)
2. <http://www.hsr.ch/> [↑](#footnote-ref-2)
3. <http://hochschule-rapperswil.sv-group.ch/de.html> [↑](#footnote-ref-3)
4. <http://www.youtube.com/watch?v=CLD1wVbcD8w&feature=related> [↑](#footnote-ref-4)
5. [microsoft12.2] Microsoft Corporation, „Kinect for Windows Human Interface Guidelines”, <http://www.microsoft.com/en-us/kinectforwindows/develop/learn.aspx>  
   letzter Zugriff 04.06.2012 [↑](#footnote-ref-5)
6. <http://msdn.microsoft.com/en-us/vstudio/ff637362.aspx> [↑](#footnote-ref-6)
7. <http://www.nagios.org/> [↑](#footnote-ref-7)
8. <http://typo3.org> [↑](#footnote-ref-8)
9. <http://www.hsr.ch/typo3/index.php> [↑](#footnote-ref-9)
10. <http://typo3.org/extensions/repository/> [↑](#footnote-ref-10)