## 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 |

### Daten

#### 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 1 - Domain Model VideoWall

Wie in Abbildung 1 - 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 erdenkbarer Inhalt, welcher auf der *VideoWall* präsentiert werden soll. Diese Inhalte werden durch mit der 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 2 - 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 3 - 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-Präsentation. 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.

##### Mittagsmenu

Die Daten des Mittagsmenus werden auf der Internetseite der Mensa HSR Hochschule Rapperswil der SVGroup abgerufen. (TODO <http://hochschule-rapperswil.sv-group.ch/de.html>). 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 Benutzer intuitiv benutzen würden, wurde ein Test durchgeführt. Dieser wurde als ein Wizard of Oz Experiment durchgeführt (TODO: link auf Testdokumentation). Der Test sollte auch zeigen, ob das erarbeitete GUI für den Benutzer einfach verständlich ist.

##### Ideensammlung

Am 14.03.2012 wurden erste 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 4 - 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.

Der Test wird als Powerpoint Präsentation vorbereitet und mit einem Beamer projiziert. Je nachdem, 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 4 - Anforderungen an den Test

Die Abbildung 1 - 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 Szenarien der in der Vorstudie erarbeiteten Personas (TODO link Vorstudie) durchlaufen.

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 Wand angezeigt werden und sich bewegen. Sobald nun jemand erkannt wird und sich diese Person zur Wand 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 5 - Posteransicht

Die Abbildung 2 - 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 6 - Unterteilung in Tabs

Die obenstehende Abbildung 3 - 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 7 - Zonenmarkierung

Des Weiteren ist es für den Nutzer wichtig zu wissen, in welchem Abstand zur Wand er stehen muss, um erkannt zu werden. Es sind zwei Zonen denkbar, eine Browsing- und eine Lese-/Interaktionszone. Wie in Abbildung 4 - 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, Kinect 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. Hypothese: Meine Hand ist die Maus

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

Bei der ersten Hypothese könnte, wie bereits im Unterkapitel I.2.1.1.1 Ideensammlung beschrieben, ein Laserpointer genutzt werden um den Pointer des Nutzers zu repräsentieren.

Bei der Diskussion über die zweite Hypothese 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 4 - 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. Deshalb wurde dem Team geraten, diesen Test nicht in PowerPoint sondern gleich als WPF Applikation umzusetzen. Diese soll in etwa wie folgt aussehen:



Abbildung 8 - 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.

##### Durchführung & Fazit

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

Das Fazit des Tests ist, dass die Hypothese „Meine Hand ist die Maus“ bestätigt werden konnte. Die Steuerung wird also so festgelegt, dass die Hand des Benutzers die Maus auf der Videowall steuert.

#### Demomodus

##### Ideensammlung

Damit Personen, welche das Gebäude 4 der HSR passieren, mit der Videowall interagieren, müssen sie erstmals 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.

###### Sammlung und Besprechung der Ideen

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 9 - 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 7 - Demomodus, Ideen 1-3) fragt die Wand: „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 7 - 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 10 - Demomodus, Ideen 4-8

Bei der Idee Nummer 4 werden Objekte, verteilt über alle Monitore, dargestellt. 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. 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 kurz 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. Durch das Vorbeilaufen an der Wall oder 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 Wand vorbei, so schiebt er das Vorhängeschloss auf das Symbol mit dem Teller und so zeigt die Wall das Mittagsmenu der Mensa an.  


Abbildung 11 - 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.2.1 Empirischer formativer Test). 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 12 - 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 ein. 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 13 - Demomodus, Idee 11

Die Idee 11 dient vor allem der korrekten Positionierung des Benutzers vor der Wall. Passanten, welche weit entfernt von der ideale Position stehen, werden auf der Wand 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 (siehe I.2.1 Empirischer formativer Test) 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.  
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 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.

##### Umsetzung

###### Besprechung Demomodus „Kraftfeld“

Am 07.05.2012 besprach das Team, wie bei der Umsetzung des ausgewählten Demomodus „Kraftfeld“ vorgegangen werden soll. Es handelt sich hierbei um den Demomodus, bei dem durch Vorbeilaufen die über alle Monitore verteilten Objekte (z.B. kleine Stücke eines Posters) bewegt werden können (siehe auch I.2.2.1 Sammlung und Besprechung der Ideen, Beschreibung zu Idee Nummer 4).

Die Abbildung 12 - Teilaufgaben des Demomodus "Kraftfeld" zeigt, dass die Applikation aus sechs Teilaufgaben bestehen müsste. Der erste Punkt ist das Generieren von Screenshots (1), welche dann in Teilchen zerschnitten werden. Weiter müssen diese Teilchen über den ganzen Bildschirm verteilt angezeigt werden(2). Damit bereits hier Dynamik im Spiel ist, benötigt jedes einzelne Teilchen eine Grundanimation (z.B. eine leichte Hin- und Herbewegung). Der dritte Punkt ist das Zusammenfügen der Teilchen (3) zu einem Ganzen, dem Ursprungsbild. Als Nächstes müssen die Bewegungen der Teilchen (4), die durch das Passieren der Videowall ausgelöst wird, festgelegt und implementiert werden. Dazu mehr im nachfolgenden Abschnitt, der die Abbildung 13 - Bewegungsart der Teilchen beschreibt. Abschliessend folgt das Wechseln vom Demomodus in den Interaktionsmodus (5) und umgekehrt (6).  
Ein Usability Test und das Umsetzen der allfällig dadurch entstandenen Verbesserungsansätze runden die Implementation ab.



Abbildung 14 - Teilaufgaben des Demomodus "Kraftfeld"

Die Umsetzung des im obigen Abschnitt aufgelisteten Punktes Nummer 4 benötigt mathematische Vorarbeit. Die Abbildung 13 - Ideen zur Bewegungsart der Teilchen zeigt Ansätze, wie die Bewegungen der Teilchen berechnet werden könnte.



Abbildung 15 - Ideen zur Bewegungsart der Teilchen

Im oberen Teil der Abbildung soll mit den Abstandsangaben, die vom Tiefensensor der Kinect erfasst werden, gearbeitet werden. Der Tiefensensor misst für jedes Pixel, wie weit der darauf zu sehende Mensch oder Gegenstand vom Sensor entfernt ist. Mit Hilfe der daraus gewonnenen Zahlwerte könnten nun Geradensteigungen und Vektorrichtungen für die Bewegung der Teilchen, welche auf den Wall verteilt dargestellt werden, ausgerechnet werden.

Im unteren Teil der Abbildung wird im Hintergrund ein feines Raster über die Monitore gelegt. Ein Quadrat dieses Rasters beinhaltet mehrere Pixel. Wird nun das Skelett des Benutzers erkannt, so werden die Rasterquadrate, von denen ein oder mehrere Pixel im Bereich des Skeletts sind, als besetzt markiert (grün schraffierte Fläche). Teilchen, die sich auf diesen besetzen Rasterquadraten befinden, suchen sich nun den kürzesten Weg auf ein freies Quadrat. Teilchen, welche bereits auf einem freien Quadrat dargestellt werden, bewegen sich nicht.

Fazit

Beim Notieren der Teilaufgaben, welche alle erledigt werden müssen, um den Demomodus umsetzen zu können, wurde dem Team bewusst, wie viel Arbeitsstunden die Implementation benötigen würde.   
Die Zeit für die Durchführung der Bachelorarbeit ist jedoch beschränkt. Zum Zeitpunkt der Besprechung des Demomodus befand sich das Team bereits in der 11. Arbeitswoche von total 16 Wochen. Da, durch die Hochschule vorgegeben, bis zum Arbeitsende zahlreiche Dokumente erstellt werden müssen, sind die letzten vier Wochen (13-16) der übrig gebliebenen Zeit bereits komplett verplant. Für den aktuellen Sprint (Woche 11) gilt es, zuerst die geplanten, höher priorisierten User Stories abzuarbeiten.  
Das Team kam daher zum Schluss, den Demomodus „Kraftfeld“ aus zeitlichen Gründen nicht umzusetzen. Es wird daher die Alternative umgesetzt. Erklärungen dazu sind im Unterkapitel I.2.4 Umsetzung des Demomodus „Lockspruch“ zu finden.

###### Umsetzung des Demomodus „Teaser“

Auch dieser Demomodus wurde in einzelne Teilaufgaben unterteilt. Wie bei der Idee I.2.3.1 Besprechung Demomodus „Kraftfeld“ muss es möglich sein, zwischen dem Interaktions- und Demomodus zu wechseln. Sobald der Demomodus angezeigt wird, soll der Hintergrund auf eine zufällig ausgewählte Farbe gesetzt werden. Zudem soll auch ein Teaser-Text zur jeweilig im Hintergrund aktiven App angezeigt werden. Dabei könnte es sich, wie in der Abbildung Abbildung 7 - Demomodus, Ideen 1-3 ersichtlich, um einen Text wie „Hunger? – Dann stell dich hier hin“ handeln.

Die nachfolgende Abbildung 13 - Zustandsdiagramm Interaktions- und Demomodus zeigt das Zustandsdiagramm, welches den Wechsel vom Interaktionsmodus in den Demomodus und zurück aufzeigt.   
Zu Beginn befindet sich die Applikation im Interaktionsmodus. Der Timer ZuDemomodus startet, sobald die Applikation bereit ist. Solange im Interaktionsmodus ein Skelett erkannt wird, kann die Applikation bedient werden, der Timer ZuDemomodus wird immer wieder zurückgesetzt. Wird kein Skelett mehr erkannt, läuft der Timer Schritt für Schritt ab. Wird vor Ablauf der definierten Zeit (z.B. 10 Sekunden) wieder ein Skelett erkannt, wird der Timer ZuDemomodus zurückgesetzt, die Applikation kann weiter bedient werden. Läuft die vordefinierte Zeit jedoch ab, findet dann der Wechsel in den Demomodus statt.  
Im Demomodus läuft zu Beginn kein Timer. Erst wenn ein Skelett erkannt wird, startet der Timer ZuInteraktionsmodus. Wird vor Ablauf der definierten Zeit (z.B. 5 Sekunden) plötzlich kein Skelett mehr erkannt, wird der Timer gestoppt und zurückgesetzt. Wird während der ganzen vordefinierten Zeit ein Skelett erkannt, so läuft der Timer vollständig ab und die Applikation wechselt in den Interaktionsmodus.



Abbildung 16 - Zustandsdiagramm Interaktions- und Demomodus

##### Externes Design

Für den Demomodus „Teaser“ wurde ein externes Design erarbeitet. Sobald der Demomodus aktiv ist, wird auf der Videowall eine zufällig ausgewählte Farbe gezeigt. Bei den dafür definierten Farben handelt es sich um das HSR-Blau, ein Pink, ein Orange und ein Grün. Die Farbauswahl kann bei Bedarf später noch erweitert werden.

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 17 - 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 18 - 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.

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

#### Screen Map



Abbildung 19 - Screen Map

Das Screen Map bezieht sich auf die im Kapitel I.1.2.1 Domain Models vorgestellten Klassen und Attribute. Solange kein Skelett erkannt wurde, befindet sich die Applikation im Demomodus (mehr Informationen hierzu im Kapitel I.1.3.2 Demomodus) und es wird daher der Demomodus Screen angezeigt. In diesem wird immer der *DemomodeText* einer zufällig ausgewählten *VideoWallApplication* dargestellt. In diesem Fall handelt es sich bei dieser um die *LunchMenuApp*. Sobald ein Skelett erkannt wird, wechselt die Anzeige vom Demomodus in den Interaktionsmodus. Im Interaktionsmodus Screen können die verschiedenen *VideoWallApplications* im Menu ausgewählt 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. Da momentan die *LunchMenuApp* aktiv ist, wird auch deren Inhalt angezeigt. Würde nun auf die *PosterApp* gewechselt werden, würde in der Mitte der Inhalt der *PosterApp* visualisiert werden.

#### Design Entscheide

#### Guidelines

Die Human Interface definieren eine Reihe von Prinzipien. Dieses Dokument war jedoch erst am 21.05.2012 online verfügbar. Zu diesem Zeitpunkt war jedoch schon ein Grossteil der Applikation umgesetzt. Unabhängig davon konnten die Anforderungen aus den Guidelines erfüllt werden. Dies kann grösstenteils auf die Usability Tests (siehe TODO) zurückgeführt werden. Die Guidelines sind in den nachfolgenden Kapiteln kurz zusammengefasst und es wird erwähnt wie diese im Projekt umgesetzt wurden.

##### Human Interface Guidelines

###### Best Practices for Designing Interactions

Dieses Kapitel Best Practices for Designing Interactions beschreibt wie Stimme und Gesten für die Steuerung von Kinect verwendet werden. Für das Team werden Gesten als ein bestimmter Ablauf von Bewegungen angenommen. Die Guidlines verstehen 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.“. In Falle der Videowall werden lediglich „Innate Gestures“ 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.“. Dies trifft für die Methode „Meine Hand ist die Maus“ (siehe TODO Usability Test) zu. Zudem soll ein visuelles Feedback vorhanden sein. Der Nutzer soll unter anderem wissen ob Kinect einem erkannt hat, ob man die Kontrolle hat und ob man am richtigen Ort steht. Diese Anforderungen werden alle mit der Darstellung des Skeletts und der Hand gewährleistet. Auch soll die Verwendung von Kinect mit 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 ausführbar sein, dies ist mit der Hand als Maus 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.” – Sobald sich der Handcursor auf einem auswählbaren Element befindet, wird eine Ladeleiste 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. Interaktionen sollen auch abbrechbar sein. Bewegt man den Handcursor von einer auswählbaren Schaltfläche weg, verschwindet die Ladeleiste wieder. Auch das Wechseln vom Demomodus in den Interaktionsmodus (siehe I.1.3.2.2.2 Umsetzung des Demomodus „Teaser“) kann abgebrochen werden, indem der Nutzer sich vom erkennbaren Bereich entfernt. Die Inputmethode soll möglichst praktisch sein für die gestellten Aufgaben. Die Grössen der Schaltflächen sind gross genug, dass sie mit dem Handcursor ausgewählt werden können.

„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.” – dies beschreibt die genau gleiche Idee wie sie auch für dieses Projekt für das Handtracking (TODO Dokument Entwurf) umgesetzt wurde.

Der Jitter bei dem Handcursor wurde, wie in den Guidelines empfohlen, verringert. Auch wurde für die Methode „Meine Hand ist die Maus“ ein Cursor in Form einer Hand genommen. Verwendet der Nutzer die linke Hand, so die der Cursor als linke Hand dargestellt und umgekehrt. Auch dies wird von den Guidelines vorgeschlagen.

###### Distance-Dependent Interactions

Das Kapitel beschreibt verschiedene Interaktionszonen und was für Interaktionen sich dort am besten eigenen. Für Die Videowall wurde nur der „Far(2.0-4.0 Meters)“- und der „Out of Range (>4 Meters)“-Bereich verwendet. Im zweiteren Bereich soll der Nutzer dazu 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”

#### Multimodality

###### Multiple Users

### Installation, Betrieb und Administration der Videowall

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

Da am Ende des Projekts (15.06.2012) nicht bekannt sein wird, welche Hardware tatsächlich angeschafft wird, muss dieses Kapitel 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

Die aktuelle Applikation ist nicht darauf ausgelegt, 24 Stunden und 365 Tage im Jahr ohne Neustart zu laufen.  
Die Gebäude werden um 20.00 geschlossen, Studenten und Mitarbeiter haben mit dem Badge aber weiterhin Zutritt zu den Gebäuden. Da sich ab 20.00 aber nur noch vereinzelte Personen an der HSR aufhalten werden, macht es keinen Sinn, dass die Videowall 24 Stunden in Betrieb ist. Es wird darum vorgeschlagen, die Applikation ein Mal pro Tag neu zu starten, zum Beispiel:

* Herunterfahren um 20.00 Uhr, da um diese Zeit die HSR-Gebäude geschlossen werden.
* Starten um 7.00 Uhr, da um diese Zeit die HSR-Gebäude geöffnet werden.

Dieses Vorgehen würde auch den Stromverbrauch wesentlich reduzieren.

Beim Start der Applikation wird sogleich eine Synchronisation mit anderen Datenquellen durchgeführt. So wird zum Beispiel das Mittagsmenu neu geladen und neue Videowall Extensions werden heruntergeladen (TODO: von wo?) und installiert.

In der Informatik ist es heutzutage praktisch unmöglich, allen Fehlern vorzubeugen. Deshalb ist es nötig, für den produktiven Betrieb eine Notifikation für aufgetretene Fehler an die Betreiber zu senden und die Fehler zu protokollieren. Zusätzlich sollte die Applikation bei einem Fehler neu gestartet werden. Diese Features sind bei Ende der Bachelorarbeit noch nicht realisiert und sollten vor der Inbetriebnahme der Videowall noch umgesetzt werden.

#### Zu verwaltende Daten

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

##### Extensions

Extensions sind Applikationen, die auf der Videowall dargestellt werden können und müssen verwaltet werden. Folgender Ablauf ist dafür vorgesehen:

1. Die für die Technik verantwortliche Stelle (IFS/INS => TODO: Refs) gibt bekannt, dass für die Videowall Extensions entwickelt werden können. Dies könnte auf verschiedene Arten geschehen, beispielsweise im Modul Software Engineering 2 Projekt oder in einer Studien- oder Bachelorarbeit.
2. Die Entwickler programmieren eine oder mehrere Extensions für die Videowall.
3. Ist eine Applikation fertig entwickelt, so melden sich die Programmierer bei der für die Technik verantwortlichen Stelle (IFS/INS, Markus Stolze TODO: Refs).
4. Für das Deployment der Extension müssen zwei Bedingungen erfüllt werden:
   1. Die Entwickler müssen schriftlich garantieren, dass keine unangebrachten Inhalte (gewaltverherrlichend, erotisch, usw.) in ihrer Extension dargestellt werden. Die Konsequenzen des Nichteinhaltens werden im Vorhinein durch die verantwortliche Stelle definiert und den Entwicklern offengelegt.
   2. Die verantwortliche Stelle führt ein Code Review der Extension durch, um die Qualität und die Sicherheit des Gesamtsystems zu gewährleisten.
5. Das Deploymnet wird durch die für die Technik verantwortliche Stelle (INS, TODO ref) durchgeführt. Alternativ dazu könnte ein Web-Interface zur Verfügung gestellt werden. Die Entwickler der Extension erhalten einen Zugang und können die Extension eigenständig hochladen.

##### 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 Mittagsmenü aktualisiert.

#### Datenverwaltung der Extensions

Jede Extension 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 Extensions verwaltet werden können. Die Entwickler einer Extension definieren die Objekte, welche verwaltet werden sollen. Das Framework generiert dann automatisch eine Benutzeroberfläche für deren Bearbeitung.

Alternativ könnte jede Extension 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.

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 [[1]](#footnote-1)

Das Sekretariat der HSR arbeitet bereits mit einem Typo3 CMS (TODO: auf was muss hier verwiesen werden? <http://www.hsr.ch>) 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 (TODO: ref <http://typo3.org/extensions/repository/>) (nicht zu verwechseln mit einer Extension für die Videowall) 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 da durch WPF die Plattform eingeschränkt wird und die Entwicklung einer mobilen Applikation so nicht möglich ist.

Diese Variante wird nicht empfohlen.

1. <http://typo3.org> [↑](#footnote-ref-1)