

SmartMirror

Fakultät Informatik Mathematik

OTH Regensburg

Human Computer Interaction

Projektbericht

Wintersemester 2020/2021

12. Februar 2021

Patrick Gruber

patrick.gruber@st.oth-regensburg.de

Michael Lazik

michaell.lazik@st.oth-regensburg.de

Tobias Gubo

tobiasl.gubo@st.oth-regensburg.de

Marcus Müller

marcus.mueller@st.oth-regensburg.de

Inhaltsverzeichnis

1	Planen des menschenzentrierten Gestaltungsprozesses	1
1.1	Festlegen der Ressourcen	1
2	Verstehen und Festlegen des Nutzungskontexts	2
2.1	Beschreibung des Nutzungskontexts	2
3	Erarbeitung von Gestaltungslösungen zur Erfüllung des Nutzungskontexts	3
3.1	Erstellen der Interaktionsspezifikation	3
3.2	Erstellen der Informationsarchitektur	4
3.3	Spezifikation des UI	5
3.4	Prototyping	6
3.5	UI Gestaltungsrichtlinien	7
4	Aufbau Software System	9
4.1	Nutzungsanforderungen spezifizieren	9
4.2	Nutzungsanforderungen priorisieren	9
5	Evaluierung der Gestaltungslösung anhand der Nutzungsanforderung	10
5.1	Entwicklungsbegleitende Usability-Tests	10
5.2	QuantitativeEvaluierungen	10

1 Planen des menschenzentrierten Gestaltungsprozesses

Nur wer sein Ziel kennt, findet den Weg. - Laozi

Wie jedes gute Projekt beginnt man mit der grundlegenden Planung und Strukturierung

1.1 Festlegen der Ressourcen

Der erste und elementarste Schritt war es sich zu überlegen, wie ein Produkt wie der SmartMirror überhaupt realisiert werden kann. Dafür haben wir uns im Internet auf die Suche gemacht und sind dort an mehreren Stellen auf DIY-Projekte gestoßen, die eine ähnliche Idee umgesetzt haben. Die fundamentalen Bestandteile waren jedoch oft sehr vergleichbar. Man benötigt einen Einwegspiegel, der das Licht von einer Seite durchlässt und von der anderen Seite verspiegelt ist. Mit einem LED-Display hinter der Spiegelscheibe lässt sich so eine Anzeigefläche erschaffen, die für den Betrachter nur sichtbar ist, wenn sie beleuchtet ist. Die Art und Weise, wie das Display angesprochen wird ist wieder eine freiere Entscheidung. Wir haben uns dafür entschieden einen RaspberryPi 3b zu verwenden. Dies hat mehrere Gründe, die im Laufe des Berichts noch genauer beleuchtet werden. Montiert wird das Spiegeldisplay in einem Rahmen aus Holz, um ein einheitliches Erscheinungsbild zu kreieren. Als letztes wichtiges Element ist noch der LeapMotion-Controller zu nennen, der eine einfache Möglichkeit bildet um die Gestik des Nutzers zu erkennen, um somit die Interaktion zwischen Mensch und Maschine zu ermöglichen.

Grobe Kostenaufstellung

<u>Name</u>	<u>Zulieferer</u>	<u>Preis</u>
Spiegelglas 70x100cm	GlasStar	ca. 125€
Display 17" mit Controller	Amazon	ca. 160€
LeapMotion Controller	AdaFruit	ca. 100€
Holzrahmen 70x100cm	Amazon	ca. 50€
Raspberry Pi	Amazon	ca. 40€
Kabel und Verbindungen	Amazon	ca. 50€
Gesamt		ca. 525€

Aufgabenaufteilung

Patrick Gruber	Erstellen der Kostenaufstellung
Tobias Gubo	
Michael Lazik	
Marcus Müller	

2 Verstehen und Festlegen des Nutzungskontexts

Nachdem die Ressourcen festgelegt worden sind, musste der Nutzungskontext definiert, verstanden und festgelegt werden. Hierbei wurde überlegt, wer die Benutzer sind und wie die Benutzer mit dem Smart Mirror interagieren werden.

2.1 Beschreibung des Nutzungskontexts

Unsere Primären (Direkten) Benutzer sind die Probanden, die unser Smart Mirror testen und Fazite daraus ziehen.

Die Sekundären (Indirekten) Benutzer sind die Mitglieder des Projektteams, die aus den Fazite den Smart Mirror weiter entwickeln.

Mit dem Smart Mirror sollen den Benutzern innerhalb kürzester Zeit die wichtigsten Informationen am Spiegel gezeigt werden.

Dabei sollen dem Benutzer gezeigt werden:

- Uhrzeit
- Wetter (Uhrzeitbedingt)
- RSS-Feed

Zudem kann der Benutzer seine individuellen Informationen anzeigen lassen. In unserem Projekt sind das:

- Termine
- ToDo-Liste
- Fahrzeiten ÖPNV
- Verkehrslage
- Speißeplan

Für den Smart Mirror benötigt der Nutzer keine Ausrüstung. Sie müssen nur wissen, wie man mit dem Smart Mirror interagiert und wie man durch den Pageflow die Seiten wechselt. Der Smart Mirror kann an jeden von den Benutzern beliebig ausgewählten Ort stehen. Das Projektteam hat sich jedoch auf das Schlafzimmer als Örtlichkeit für den Smart Mirror festgelegt.

Damit der Nutzungskontext aus Sicht der primären Benutzer verstanden ist, wurden Interviews mit den Probanden geführt. Dabei wurden vor allem zu Beginn möglichst offene, neutrale und allgemeine Fragen gestellt, also das "Warum?", die dann immer mehr in die Materie des Smart Mirrors gingen, also das "Wie?". Hier konnten die Probanden aus ihre Vorstellung eines Smart Mirrors offenbaren und das Projektteam konnte anhand dieser Informationen die Nutzungsanforderungen festlegen.

Aufgabenaufteilung

Patrick Gruber	
Tobias Gubo	
Michael Lazik	
Marcus Müller	

3 Erarbeitung von Gestaltungslösungen zur Erfüllung des Nutzungskontexts

3.1 Erstellen der Interaktionsspezifikation

Anhand der zuvor, durch Umfragen und Formulare erlangten Informationen über das potentielle Nutzungsverhaltens der User, wurden nun die möglichen Varianten der Interaktion mit dem Spiegel validiert.

Dabei standen Folgende Optionen zur Verfügung:

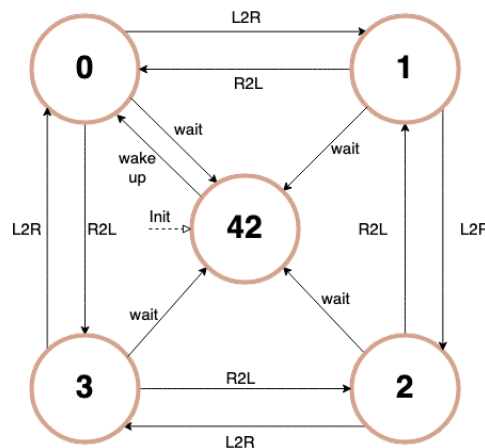
- Toucheingabe
- Sprachsteuerung
- Gestensteuerung

Hierbei stellte sich heraus, dass die Sprachsteuerung oftmals von den Nutzern als negativ empfunden wurde, da hier das Gefühl entsteht, dass der Spiegel alles mithört was in der Nähe davon gesprochen wird. Die Toucheingabe wurde ausgeschlossen, da hier einfach das Problem besteht, dass der Spiegel durch die Fingerabdrücke verschmutzt wird, so dass häufiges reinigen notwendig wäre.

Letztendlich fiel die Wahl nun auf die Gestensteuerung, da hier die Umfragen aufzeigen, dass Gesten für viele Nutzer eine intuitive Möglichkeit darstellt, einen Spiegel zu bedienen.

Daraufhin wurde noch validiert welche konkreten Gesten für den Nutzer am einfachsten zu benutzen waren. Diese Entscheidung geht Hand in Hand mit Punkt "4.2 Erstellen der Informationsarchitektur" da das Layout der Informationen eine maßgebliche Rolle für die Art wie darauf Navigiert werden muss, spielt.

Dabei wurde das folgende Konzept herausgearbeitet:



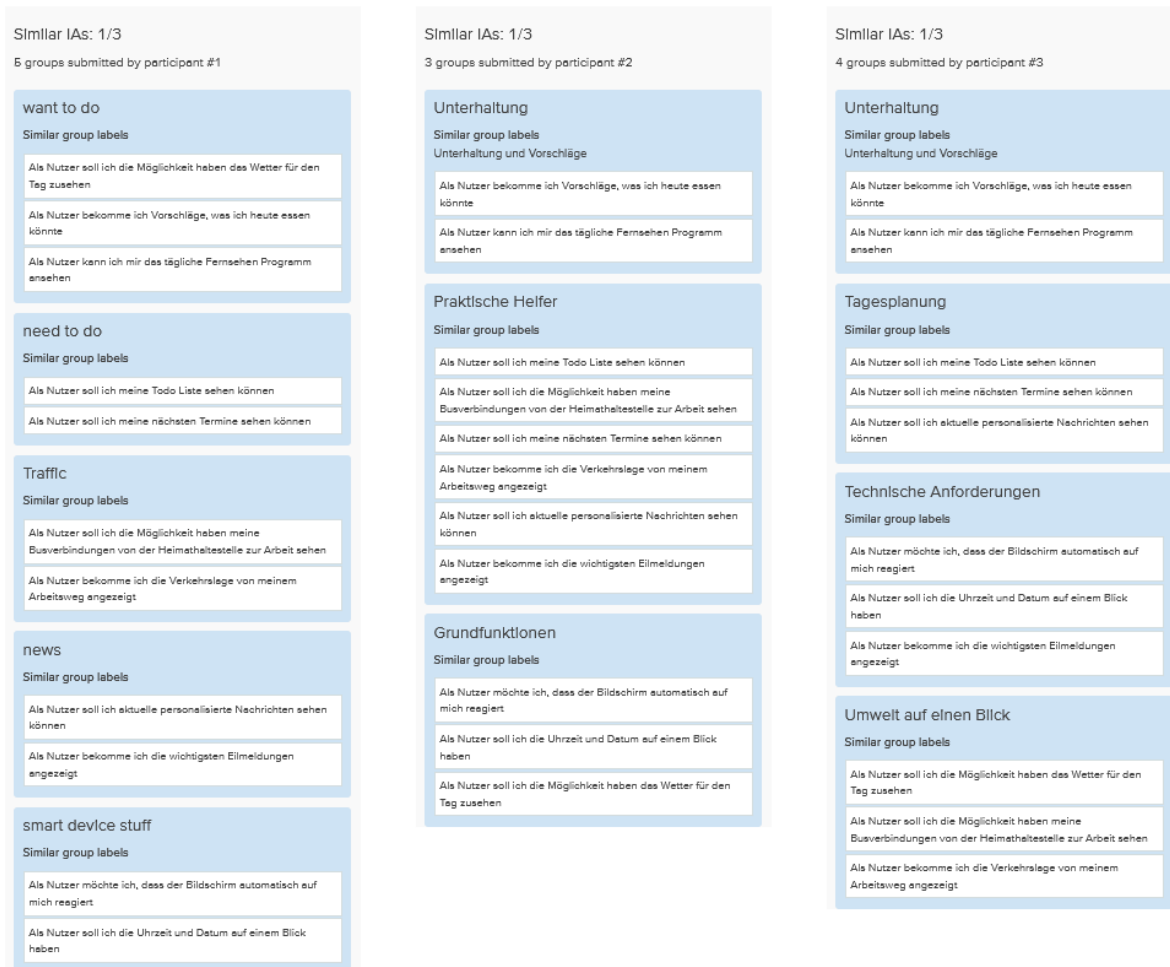
Zustandsdiagramm des Systems

Auf diesem Bild ist zu sehen wie die 5 unterschiedlichen Seiten des Spiegels mittels den Gesten swipe Links nach Rechts oder swipe Rechts nach Links erreicht werden können.

Diese beiden Gesten stellten für die grundsätzliche Bedienung die schönste Möglichkeit dar, den Spiegel zu bedienen. Zusätzlich können die Bewegungen swipe Hoch und swipe Runter für erweiterte Funktionalitäten wie Bildschirmhelligkeit oder Ähnliches genutzt werden.

3.2 Erstellen der Informationsarchitektur

Begonnen wurde die Gestaltung der Informationsarchitektur mittels eines **Trello-Boards**. Dabei wurden erst auf Basis von Nutzer Feedback eine Sammlung an interessanten Features erstellt. Daraufhin wurde mittels **Optimal Workshop** und Card Sorting analysiert, wie Nutzer Informationen und Features gruppieren würden. Das gab ein klares Bild über die Struktur der Informationen am Spiegel.



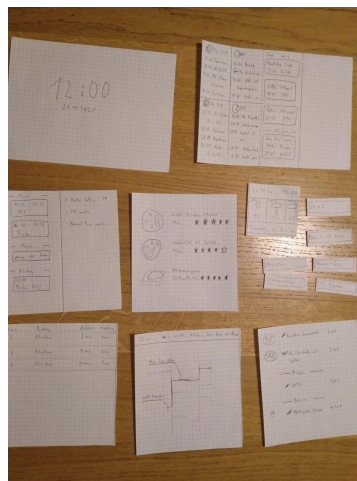
Informationarchitecture

Nachdem die Gruppen festgelegt wurden, zeichnete sich ein klares Bild ab, welche Features wie am Spiegel angezeigt werden solle:

- HomeScreen
 - Uhrzeit
 - Datum
- Termine & ToDo
 - Termine für heute
 - ToDo-Liste für heute
- ÖPNV
 - Die Abfahrtszeiten vom aktuellen Standort zu einem festlegbaren Ort
- Verkehrslage
 - Der aktuelle Weg zur Arbeit samt Stauinformation
- Speißeplan
 - Die Gerichte des heutigen Tages

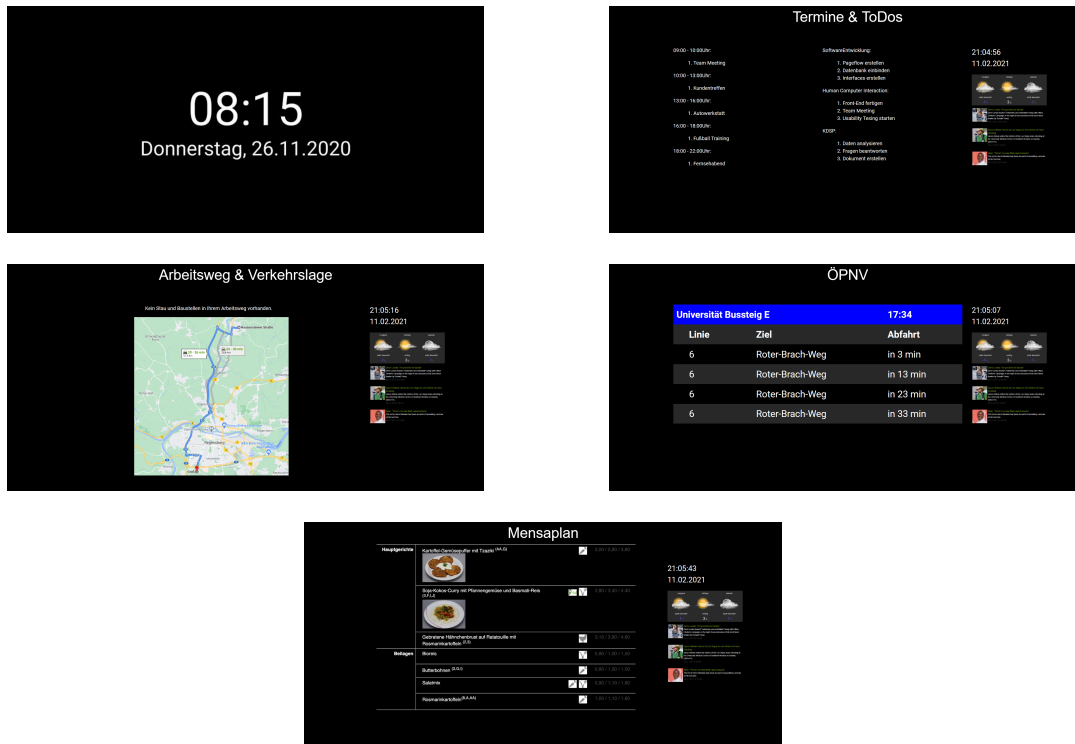
3.3 Spezifikation des UI

Die Entwicklung der UI begann mit Paper Prototyping. Hierbei wurden die ersten ersten Entwürfe auf basis der im vorherigen Schritt erstellten Informationsarchitektur erstellt.



Paper sketching

Danach ging es weiter mit dem Design der unterschiedlichen Seiten mittels Powerpoint. Hier wurden konkrete Designs mit den verschiedensten UI Elementen ausgearbeitet.



3.4 Prototyping

Der Prototyp wurde mittels dem Frontend Framework vue.js und node.js als Backend entwickelt. Dabei wurden die einzelnen Seiten auf basis von den Powerpoint Designs in html Nachprogrammiert. Zur Steuerung des Spiegels wurde ein Leap Motion Sensor mittels des bereitgestellten JS SDKs in den node Server integriert. Darin wurde Gestenerkennung implementiert. Die erkannten gesten werden nun mittels Web Sockets an das vue.js Frontend gesendet. Hier wurden die Gesten nun zur Navigation des UIs verwendet.

Aus Usability zwecken wurden zwei Varianten des Gestenerkennungs-Algorithmuses erstellt. Eine Version die die Gesten nach einer bestimmten Distanz auslöst (also mehrfach swipe mit einer Handbewegung möglich) und eine die bei einer Bewegung nur eine Geste erkennt. Nach User Testing stellte sich heraus, dass die zweite Variante deutlich intuitiver für den User ist, da diese immer versuchen, wie am Smartphone, durchgängige Wischgesten zu tätigen.

Am Schluss wurde ein Konzept zur Durchführung von Usability Tests erstellt. Das erstellte Dokument enthält eine kurze Vorstellung des Projektes, Hinweise zum Usability Testing und Aufgaben, die der Teilnehmer am Spiegel durchführen soll. Die Aufgaben behandeln das Navigieren zwischen den Seiten und das Herauslesen von Informationen. Nach Durchführung des Tests sollen in einem offenen Gespräch zusätzliche Informationen gewonnen werden.

3.5 UI Gestaltungsrichtlinien

Um für ein vereinheitlichtes Design zu sorgen, wurde immer die gleiche Schriftart und die gleiche Hintergrundfarbe verwendet. Um Inkonsistenzen zu vermeiden, wurden Komponenten, die auf mehreren Seiten auf die gleiche Art und Weise dargestellt werden, als Fragment in die Seiten eingebunden. Das wären unter anderem diese Komponenten:

- Uhrzeit
- Wetter
- RSS-Feed

Aufgabenaufteilung

Patrick Gruber	<ul style="list-style-type: none"> • Validierung der Interaktionsmöglichkeiten • Entwurf der Navigations zwischen den Seiten • Erstellen des Trello-Boards • Durchführen des Card-Sortings • Paper Prototyping • Implementierung der Gestenerkennung • Erstellung eines Konzepts zur Durchführung von Usability Tests am Prototypen
Tobias Gubo	<ul style="list-style-type: none"> • Validierung der Interaktionsmöglichkeiten • Entwurf der Navigations zwischen den Seiten • Erstellen des Trello-Boards • Durchführen des Card-Sortings • Paper Prototyping • Prototyping Setup von vue.JS und node.JS einrichten • Implementierung der Socket Kommunikation • Implementierung der Gestenerkennung • Erstellung eines Konzepts zur Durchführung von Usability Tests am Prototypen
Michael Lazik	<ul style="list-style-type: none"> • Validierung der Interaktionsmöglichkeiten • Entwurf der Navigations zwischen den Seiten • Erstellen des Trello-Boards • Durchführen des Card-Sortings • Paper Prototyping • Implementierung der Spiegel UI mittels vue.JS • Erstellung eines Konzepts zur Durchführung von Usability Tests am Prototypen
Marcus Müller	<ul style="list-style-type: none"> • Validierung der Interaktionsmöglichkeiten • Entwurf der Navigations zwischen den Seiten • Erstellen des Trello-Boards • Durchführen des Card-Sortings • Paper Prototyping • Erstellung eines Konzepts zur Durchführung von Usability Tests am Prototypen

4 Aufbau Software System

4.1 System

4.2 Frontend

4.3 Backend

Aufgabenaufteilung

Patrick Gruber	
Tobias Gubo	
Michael Lazik	
Marcus Müller	

5 Evaluierung der Gestaltungslösung anhand der Nutzungsanforderung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

5.1 Entwicklungsbegleitende Usability-Tests

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

5.2 QuantitativeEvaluierungen

Aufgabenaufteilung

Patrick Gruber	
Tobias Gubo	
Michael Lazik	
Marcus Müller	