

Bauhaus-Universität Weimar
Fakultät Medien
Studiengang Medieninformatik

Bauhausboards - Interactive Door Signs for the Office

Bachelorarbeit

André Karge
geb. am: 01.08.1990 in Jena

Matrikelnummer 110033

1. Gutachter: Junior-Prof. Dr. Florian Echtler
2. Gutachter: Prof. Dr. Eva Hornecker

Datum der Abgabe: 24. Dezember 2015

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Weimar, 24. Dezember 2015

.....
André Karge

Zusammenfassung

Diese Bachelorarbeit befasst sich mit dem Entwurf, der Umsetzung und dem Test von personalisierten digitalen Türschildern zur Präsentation und Kommunikation im Bürobereich der Bauhaus Universität Weimar (BUW), genannt Bauhausboards.

Ziel ist es, eine Anwendung zu erstellen, mit der Nutzer in Büros individuelle Daten präsentieren können.

Vor den meisten Büros sind dazu Pinnwände oder Whiteboards angebracht. Diese bieten jedoch nicht die Möglichkeit, den Inhalt anzupassen oder hinterlassene Nachrichten zu lesen, wenn man nicht persönlich vor Ort ist.

Zu diesem Zweck wurde eine Web-Applikation entworfen, welche auf Tablet-PCs angezeigt wird und mit denen Nutzer im Büro sowie vorbeigehende Besucher kommunizieren können.

Diese muss leicht bedienbar und zusätzlich von Desktop-PCs oder Smartphones benutzbar sein.

Nutzer in den Büros können vom Arbeitsplatz aus oder von unterwegs individuelle Daten präsentieren sowie ihren aktuellen Status aktualisieren.

Besucher können dadurch vor Ort über aktuelle Arbeiten, kurzfristige Abwesenheit oder andere interessante Informationen in Kenntnis gesetzt werden. Zudem haben sie die Möglichkeit bestimmten Nutzern des Raumes Nachrichten zu schreiben.

Die Applikation wurde, um Interaktion und Benutzbarkeit zu erproben, einem Testlauf unterzogen und das Ergebnis anschließend ausgewertet.

Inhaltsverzeichnis

1 Einleitung	1
2 Verwandte Arbeiten	3
2.1 Hermes	3
2.2 NetBoards	6
2.3 Andere	8
3 Vorstudie	10
3.1 NetBoards Experiment	10
3.2 Auswertung	12
4 Bauhausboards	14
4.1 Entwurf	14
4.1.1 Entwurf des Systems	14
4.1.2 Entwurf des Interfaces	17
4.1.3 Entwurf der Datenbank	20
4.2 Umsetzung	22
4.2.1 Funktionen	22
4.2.2 Editor	25

INHALTSVERZEICHNIS

4.2.3	Probleme	27
4.3	Vergleich mit NetBoards und Hermes	28
5	Studie	30
5.1	Entwurf	30
5.2	Durchführung	32
5.3	Auswertung	33
5.3.1	Interview	33
5.3.2	UEQ Fragebogen	38
6	Fazit	40
Literaturverzeichnis		42
Glossar		44
Anhang		46

Kapitel 1

Einleitung

In Forschungseinrichtungen und in Unternehmen arbeiten viele Menschen mit unterschiedlichen Aufgaben und Tätigkeiten. Manche wissen jedoch nichts darüber was andere Mitarbeiter machen, wenn sie nicht direkt mit ihnen zusammen arbeiten. Einige Mitarbeiter wollen ihre Kollegen oder Gäste über ihre aktuelle Arbeit, wichtige Termine oder kommende Veranstaltungen informieren. Andere möchten gern den Inhalt ihrer letzten wissenschaftlichen Ausarbeitung, die Ergebnisse der letzten Konferenz, Noten einer bestimmten Prüfung oder einfach nur private Daten präsentieren. Um diese Informationen zur Verfügung zu stellen werden in den meisten Büros Pinnwände oder Whiteboards neben den Türen aufgehängt (Abb. 1.1). Um kurze Nachrichten zu hinterlassen, kleben einige Personen Post-It's an ihre Tür oder hängen "Do-Not-Disturb"-Schilder an den Türknauf, wenn sie nicht gestört werden wollen.

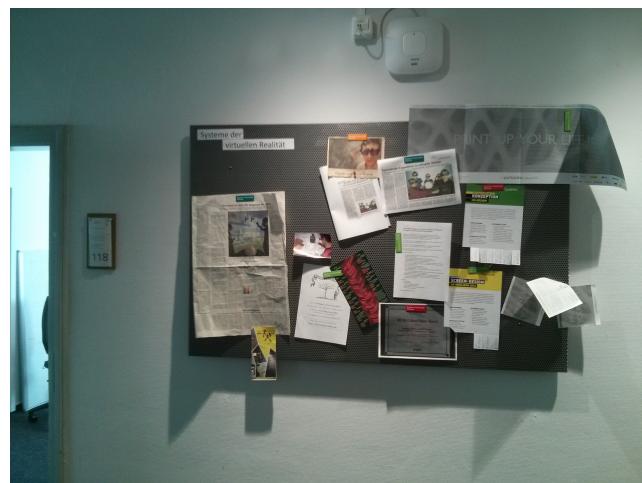


Abbildung 1.1: Typische Pinnwand der Medienfakultät der BUW

Aktuell müssen die Personen des Raumes solche Informationen direkt an die Wand schreiben oder ausdrucken und danach dort aufhängen. Wenn man aber nicht persönlich vor Ort ist, kann man die Daten an der eigenen Pinnwand nicht selbst anpassen. Es kann zu Problemen kommen, wenn man zum Beispiel einen Termin vereinbart hat und sich dann verspätet, wenn man krank geworden ist und deswegen den ganzen Tag nicht anzutreffen ist oder auf einer Reise ist und andere am Arbeitsplatz direkt daran teilhaben lassen möchte. Pinnwände und Whiteboards bieten zudem auch nicht die Möglichkeit digitale Informationen, wie Videos, animierte Gifs¹ oder die neuesten Twittermeldungen anzuzeigen.

Eine Lösung für diese Probleme ist die Anbringung eines digitalen Türschildes, mit welchem Nutzer direkt oder aus der Ferne interagieren können.

Zu Beginn galt es zu klären, welchen Weg ich mit dieser Arbeit einschlagen sollte. Ich hatte die Möglichkeiten, ein bereits vorhandenes Projekt zu verwenden, anzupassen und zu verbessern oder ein eigenes System von Grund auf selbst zu entwerfen und umzusetzen. Deshalb führte ich ein Experiment mit dem NetBoards System [1, 2] durch, um herauszufinden, ob dies ein geeignetes System für die Verwendung an der Medienfakultät der BUW wäre. Zudem ging es darum die Bedürfnisse und das Verhalten der Benutzer im Bezug zu interaktiven Türschildern zu ermitteln.

Nach der Durchführung des Experiments stellte sich heraus, dass das NetBoards System nicht der richtige Ansatz für die Umgebung war, was in Kapitel 3.2 erklärt wird. Deswegen entschied ich mich dazu, ein eigenes System zu entwerfen.

Die Entwicklung meines Systems teilte sich in drei Abschnitte: Planungsphase, Implementationsphase und Studienphase. Die Planungsphase diente dazu, die Systemumgebung und den Aufbau der Applikation zu definieren. Darunter fielen die Wahl des Anwendungstyps und der Programmiersprache sowie der Entwurf einer Datenbank und des Benutzerinterfaces.

In der Implementationsphase befasste ich mich damit, die geplanten Ideen in ein lauffähiges Programm umzusetzen.

Das Programm musste nach seiner Fertigstellung getestet werden. Deswegen habe ich in der letzten Phase eine Studie geplant und durchgeführt. Dafür habe ich in der Medienfakultät der BUW vor bestimmten Büros Tablets, auf denen das Programm lief, angebracht, die Interaktionen damit archiviert und im Anschluss die Testnutzer befragt. Die Ergebnisse der Befragung wurden ausgewertet und fließen in die nächste Version des Programms ein.

¹Ein Gif (Graphic Interchange Format) ist ein Grafikformat mit dem mehrere Bilder in einer Datei gespeichert und von Webbrowsern als Animation interpretiert werden können.

Kapitel 2

Verwandte Arbeiten

Die Idee für interaktive Türschilder wurde schon in verschiedenen wissenschaftlichen Ausarbeitungen sowie in der Praxis erarbeitet. Die Gruppe um Keith Cheverest et al. von der Universität Lancaster hat mehrere Paper für ihr Türschild-System “Hermes” [3] bereits ab 2003 veröffentlicht. Das Projekt “NetBoards” von Errol Wood [1] von der Universität Cambridge behandelt eine aktuellere Version interaktiver Türschilder. Zudem gab es noch einige andere Arbeiten, die sich mit ähnlichen Ansätzen befassten und teilweise als Grundlage für diese beiden Projekte dienten.

2.1 Hermes

Das Hermes System [3–6] ist eines der ersten Versionen interaktiver digitaler Türschilder im Bürobereich. Es diente dazu herauszufinden, ob die „traditionelle Methode Nachrichten in einem halbwegs öffentlichem Raum, wie beispielsweise vor Büros in einem Forschungsinstitut, mittels Post-It-Zetteln zu hinterlassen durch eine digitale Methode verbessert werden könnte“ [3, 4]. Zu diesem Zweck haben die Forscher ein digitales asynchrones Nachrichtensystem entwickelt, welches direkte Interaktion mittels eines vor den Büros angebrachten Interfaces, sowie Fernzugriff durch ein Web-Portal und SMS ermöglicht. Das System bestand aus einem Web-Server und mehreren PDA’s, welche an die Wände neben den Büroeingängen angebracht wurden (Abb. 2.1). Der Server wurde mit Java-Servlets realisiert und generierte HTML Webseiten zur Anzeige auf den Displays. Er bot eine zentrale Datenbank für alle Daten des Systems und diente als Kommunikationseinheit mit einem SMS-Gateway. Eines der Hauptblickpunkte von Hermes war, dass das System leicht einsetzbar sein sollte. Deswegen fiel die Entscheidung zur Wahl der Geräte auf PDA’s. Zu der Zeit, als das Paper entstand, waren PDA’s die beste Wahl für kleine, handliche Geräte, welche per WLAN mit einem Server kommunizieren konn-



Abbildung 2.1: Das erste Hermes Display [3]

ten. Zudem mussten die Geräte mit der Umweltpolitik der Universität übereinstimmen und sollten daher nicht viel Energie verbrauchen. Um die Geräte vor Diebstahl zu schützen, wurden Aluminiumhüllen für die PDA's entworfen, welche neben den Büroeingängen an die Wand geschraubt werden konnten. Um unerwünschte Interaktion zu verhindern, wurden die Hüllen so konzipiert, dass die Tasten nicht direkt zugänglich waren.

Die Funktionen von Hermes wurden in zwei Perspektiven aufgeteilt: Die Besitzerperspektive und die Besucherperspektive.

Besitzerperspektive

„Diese Perspektive ermöglicht dem Besitzer des Displays entweder Nachrichten oder Bilder zu erstellen, die dann direkt auf dem PDA angezeigt werden oder Nachrichten zu lesen, die von Gästen für ihn hinterlassen wurden“ [3]. Zudem bot es die Funktion, animierte Gifs hochzuladen. Es war den Nutzern möglich die Perspektive von einem PC oder direkt am Display aufzurufen, wodurch sie schnell beim Verlassen des Büros eine Nachricht schreiben konnten. Um sich zu authentisieren mussten sie entweder ihren Nutzernamen und ihr Passwort eingeben oder sich mit einem iButton¹ anmelden.

¹Ein iButton ist ein Mikrochip in einer Metallhülle und dient beispielsweise zum Authentisieren des Besitzers [7].

Besucherperspektive

„Besucher müssen sich vor dem PDA befinden, um mit dem System interagieren zu können“ [3]. Sie konnten dem Besitzer des Raumes Nachrichten schreiben. Anderen Besuchern war es jedoch nur möglich die Nachricht, die der Besitzer des Raumes eingestellt hat, zu sehen. Die Nachrichten von anderen Besuchern konnte nur der Besitzer einsehen, wodurch ein Vorteil im Bereich der Privatsphäre gegenüber Post-It-Zetteln erzielt wurde.

Um das System zu testen haben die Entwickler eine Langzeitstudie über 15 Monate durchgeführt. Dabei haben sie im Vorfeld Bedenken geäußert, dass „die Nutzer Zeit brauchen bis sie neue Technologien in ihre tägliche Routine eingliedern“ [3]. Eines der enttäuschensten Ergebnisse war, dass es Nutzer gab, die trotz aufgehängtem Hermes Display weiterhin Post-It's an die Türen geklebt haben [3]. Ein weiteres Ergebnis der Studie war, dass die Nutzer erst Vertrauen aufbauen mussten. „Nutzer gebrauchen ein neues System erst, wenn sie wissen, dass es funktioniert“ [3]. Durch Verbindungsprobleme über das WLAN kam es dazu, dass das System ab und an nicht reagierte, wodurch es für die Nutzer schien, das Gerät wäre abgestürzt. Um dies zu beheben, wurde das Metallgehäuse an der Stelle des WLAN Adapters geöffnet, damit das Gehäuse das Signal nicht weiter abschwächen konnte. Es kam auch vor, dass Besucher vergaßen ihre Nachricht abzuschicken, wodurch deren Nachricht nicht mehr privat war und andere Nutzer sie sehen konnten. Das ermöglichte anderen Nutzern die Nachricht zu verändern oder sie zu missbrauchen um unangebrachte Nachrichten zu schreiben. Wenn der eigentliche Benutzer sich zudem auch authentisiert hat wurde die Nachricht dann in seinem Namen geschickt [4]. Manche Nutzer fanden es frustrierend, nachdem sie eine temporäre Mitteilung eingestellt hatten, eine alte Nachricht wiederherzustellen, da sie wieder durch den Prozess einer neuen Mitteilung gehen mussten. Als Lösung dafür wurde eine Möglichkeit eingebaut, mit der die Nutzer eine Standardnachricht einstellen konnten, die nach Entfernen der temporären Nachricht wieder angezeigt wurde [4].

Um das Hermes System zu erweitern wurde später ein Foto-Display (Abb. 2.2) entwickelt [6]. Da die Hermes-Displays sehr klein waren wurde dieses Display ausschließlich für Bilder verwendet. Mit dieser Erweiterung war es den Nutzern möglich Bilder per MMS oder Bluetooth an das Gerät zu schicken oder mit der vorhandenen Displaykamera ein Bild zu schießen.

Da dieses System schon 2003 entwickelt wurde, dienten dessen Ausarbeitungen und Forschungsergebnisse als Inspiration oder Grundlage für andere Projekte, wie beispielsweise für das NetBoards System.



Abbildung 2.2: Das erste Hermes Foto-Display [8]

2.2 NetBoards

NetBoards [1, 2] ist ein weiteres interessantes Projekt im Bereich interaktiver digitaler Türschilder. Es wurde 2014 an der Universität von Cambridge entwickelt und zielte darauf ab „die bereits vorhandenen Whiteboards und Pinnwände neben den Universitätsbüros mit großen, touch-fähigen Bildschirmen zu erweitern oder zu ersetzen“ [1].

Im Vergleich zum Hermes System hatte sich einiges bezüglich Hardware geändert. „Displays sind billiger, größer, hochauflösender und energieeffizienter geworden“ [1]. Dadurch konnten für NetBoards größere Bildschirme mit höherer Auflösung (Abb. 2.3) verwendet werden als beim Hermes System.

Die ersten Prototypen des NetBoard Systems bestanden jeweils aus einem 22 Zoll großem Touchscreen-Monitor, der von einem Raspberry Pi² betrieben wurde. Im ersten Experiment der Entwickler wurden fünf Prototypen verwendet, um herauszufinden, ob die Boards akzeptiert werden und um typische Nutzerverhalten zu bestimmen. „Diese wurden vor den Büros von Doktoranden, Post-Doktoranden und eines Professors aufgehängt, um herauszufinden, wie diese unterschiedlichen Mitarbeiter mit dem System umgehen würden“ [1]. „Die Probanden konnten bei diesem Experiment Skizzen und Nachrichten zeichnen sowie Bilder per Web-Interface hochladen“ [1]. Zudem war es möglich, Websei-

²Ein Raspberry Pi ist ein kompakter Einplatinencomputer [9].



Abbildung 2.3: Das NetBoards Display [1]

ten zu laden, die dann in einer Hintergrundschicht angezeigt werden konnten. Während des Experiments fanden die Entwickler heraus, dass die Boards schnell Aufmerksamkeit erzeugten. „Vorbeigehende Leute begannen spontan mit den Bildschirmen zu interagieren oder sogar die Besitzer direkt auf die Boards anzusprechen“ [1]. Als Ergebnis des Experiments zeigte sich, dass zwei wichtige Faktoren die Akzeptanz der Nutzer gegenüber den Boards beeinflussten:

- **Zuverlässigkeit:** „Nutzer werden ein System nur benutzen wenn sie denken es funktioniert“ [1]
- **Benutzerfreundlichkeit:** „Angewöhnung der Nutzer an das System ist unwahrscheinlich, wenn das System schwer zu benutzen ist oder es zu viel Übung erfordert“ [1]

Da für die Prototypen Touchscreens mit SAW-Technologie³ eingesetzt wurden, war die Interaktion mit den Boards sehr ungenau, wodurch die Benutzerfreundlichkeit stark beeinträchtigt wurde. Zudem waren die Raspberry Pi's zu schwach, um ein funktionsreiches Interface zu betreiben.

Nach dem Experiment passten die Entwickler ihr System an. Sie ersetzten die

³Surface-Acoustic-Wave (dt.: Akustische Oberflächenwelle) ist eine Technik um die Position von Fingern auf Displays zu ermitteln.

Raspberry's mit leistungsfähigeren Maschinen, die jeweils in der Lage waren zwei Monitore simultan anzusprechen. Um es von überall erreichen zu können, wurde das NetBoards-System mit Web-Technologien implementiert. „Es musste möglich sein, mit mobilen Telefonen oder Tablet-PC's das System zu erreichen“ [1]. Der Server bestand deswegen aus einem einfachen Webserver, der beim Aufruf eine HTML Webseite auslieferte. Frontend Funktionen⁴ wurden per Javascript realisiert, wobei die Ansicht der Displays in drei Schichten aufgeteilt wurde [1]:

- **UI (Benutzerinterface)**
für Interface Komponenten und Details über die Bewohner des Büros
- **Board Content (Board-Inhalt)**
für gezeichnete Nachrichten, Skizzen und Bilder
- **Web-Page Background (Hintergrund Webseite)**
zur Anzeige einer beliebigen Webseite

Mit diesen neuen Spezifikationen wurde ein Testlauf über mehrere Monate durchgeführt. „Die Boards erzeugten in diesem neuen Test erhöhte Aufmerksamkeit und die Besitzer nutzten sie mehr, um ihre Kollegen von ihren Abwesenheiten zu informieren, selbst wenn sie nicht persönlich vor Ort waren“ [1]. „Im ersten Test änderten die Nutzer nur den Inhalt ihres eigenen Boards, was sich im zweiten Test änderte. Sie begannen auch den Inhalt der Boards ihrer Kollegen zu ändern.“ [1]

Da NetBoards eines der neuesten Systeme für interaktive Türschilder ist, flossen in die Ausarbeitung Teile von früheren Forschungen ein, unter anderem aus dem Hermes System, aber auch von diversen anderen Projekten.

2.3 Andere

Andere wissenschaftliche Arbeiten befassen sich teilweise mit dem Konzept von interaktiven Türschildern oder wurden nur theoretisch erarbeitet.

Im Paper “Dynamic Door Displays” von David Nguyen et al. ging es darum, eine Möglichkeit zu entwickeln, um die üblichen Anzeigefähigkeiten von Bürotüren, wie beispielsweise Post-It's oder Türschilder, zu erweitern. Dies umfasste automatische Updates und maßgeschneiderte Displays zur Präsentation

⁴Frontend Funktionen sind Funktionen auf die jeder Nutzer ohne Einschränkung zugreifen kann.

von privaten Informationen [10]. Die Ausarbeitung ähnelte in gewisser Hinsicht dem Hermes-System, nur das anstelle eines PDA's ein Hewlett-Packard 620LX Handheld-Computer zum Einsatz kam. Leider wurde nur ein Prototyp entwickelt, das System danach aber nicht weiter geführt. Jedoch bot dieses Projekt einige Grundlagen für Hermes sowie NetBoards.

In der Arbeit "Semi Public Displays for Small, Co-located Groups" von Elaine M. Huang et al. ging es um Displays in halb öffentlichen Räumen. „Das System beinhaltete die Bereitstellung von Informationen an abgeschiedenen Orten oder für Leute, die keine Kenntnis über die Arbeiten ihrer Kollegen haben“ [11]. Es bot mehrere voneinander unabhängige Funktionen für die Darstellung und Interaktion auf peripheren Displays an. Im NetBoards Projekt wurde jedoch entschieden, Funktionen zusammenzuführen, wodurch beispielsweise die Anzeige einer statischen Webseite mit der Funktion eines Editors für Skizzen verbunden wurde.

Die wissenschaftliche Ausarbeitung "UniCast, OutCast & GroupCast: Three Steps Toward Ubiquitous, Peripheral Displays" von Joseph F. McCarthy et al. handelt von drei verschiedenen Display-Varianten: OutCast - Display außerhalb eines Büros, UniCast - Display in einem Büro und GroupCast - Display in einem Gemeinschaftsbereich [12]. Die OutCast Displays sind eine weitere Variante von interaktiven Türschildern im Bürobereich. Sie wurden dazu genutzt um Biographien, Kalender-, Positions-, Projektinformationen, Demonstrationen oder Textnachrichten anzeigen zu lassen. Dieses Projekt war eines der ersten Systeme für die Anbringung von Bildschirmen außerhalb von Büros zur Informationsdarbietung. Deshalb hatte es auch gewissen Einfluss auf die späteren Entwicklungen.

Das Paper "Sharing Multimedia Content with Interactive Public Displays: A Case Study" von Elizabeth F. Churchill et al. [13, 14] umfasste einen ähnlichen Ansatz wie NetBoards. Es beschrieb die Entwicklung von interaktiven Poster-Boards, genannt "Plasma Poster", um Informationen innerhalb von Arbeitsgruppen zu teilen. Diese Plasma Poster waren dazu gedacht, damit Arbeitskollegen in öffentlichen Bereichen, wie zum Beispiel in einer Küche oder einem Foyer, untereinander Informationen austauschen konnten. Das System wurde zudem in einer 14 Monate andauernden Studie getestet.

Kapitel 3

Vorstudie

Zu Beginn war es wichtig, eine Anforderungsanalyse durchzuführen. Deshalb musste analysiert werden, in welchem Rahmen sich meine Arbeit bewegen sollte. Es bestanden die Optionen, ein fertiggestelltes Projekt anzupassen und zu verbessern oder ein vollkommen eigenes System zu entwickeln.

Um zu erproben, wie ein bereits fertiges System funktioniert und wie es von Studenten und Mitarbeitern der Medienfakultät der BUW aufgenommen werden würde, habe ich ein bereits vorhandenes Projekt verwendet und in der Praxis getestet. Da das Hermes System schon relativ alt war und hauptsächlich für PDA's entworfen wurde, kam es für diesen Test nicht in Frage. Die Entscheidung fiel auf die zweite Version des NetBoards Projekts [2], da dieses Projekt das aktuellste war.

3.1 NetBoards Experiment

Als Grundlage für mein Experiment diente ein virtueller Debian-Linux Server mit 1,5GHz AMD CPU und 2GB RAM auf dem Apache2 als Webserver und Python2 für serverseitige Scripts liefen. Das öffentlich zugängliche NetBoards2 Projekt wurde installiert und konfiguriert.

Nachdem der Server eingerichtet war, stellte sich die Frage, was als Anzeigegerät dienen sollte. Das originale Projekt von E. Wood wurde für 22 Zoll Monitore mit Touch-Oberfläche entworfen. Diese wurden vertikal neben die Büroeingänge gehängt.

Da für mein Experiment nicht die notwendigen Ressourcen vorhanden waren, um den genauen Versuchsaufbau nachzuempfinden, fiel die Wahl des Anzeigegerätes auf einen kostengünstigen Tablet-PC.

Dies war ein 10 Zoll Tablet mit 1GHz Cortex A8 Dual Core CPU, 1GB RAM und Android Version 4.4.

Damit die Benutzer nicht mit den auf dem Gerät installierten Applikatio-

nen interagieren konnten, wurde eine Kiosk-Applikation¹ installiert. Diese App schränkt die Interaktionsmöglichkeiten der Nutzer nur auf eine ausgewählte Webseite ein. Nur der Administrator des Gerätes hatte die Möglichkeit, diese zu ändern oder die Applikation ganz zu beenden.

Ein weiteres Problem war die Befestigung des Displays. Ich entschied mich dafür, das Tablet an den bereits vorhanden Türschildern anzubringen (Abb. 3.1).



Abbildung 3.1: Versuchsaufbau des Experiments

Da sich die Rückwand des Tablets abschrauben ließ, konnte ich zwei Löcher hinein bohren. Diese dienten zur Anbringung eines Drahtes, welcher über das Türschild gehängt werden konnte. Für das Experiment war diese Aufhängung vollkommen ausreichend.

Für die Stromversorgung wurde das Ladekabel des Tablets verlängert, damit es an eine Steckdose im Büro angeschlossen werden konnte.

Als Testnutzer für das Experiment konnte ich zwei Mitarbeiter der Professur für Webtechnologien und Informationssysteme gewinnen. Diese Nutzer teilten sich ein Büro, wodurch nur ein Anzeigegerät benötigt wurde. Nach einer Einweisung zur Benutzung des Systems stellte sich heraus, dass beide Tester nicht mit der Interaktivität der Boards einverstanden waren. Jeder Gast konnte den dargestellten Inhalt beliebig ändern. Durch die Anonymität der Gäste hätte es zu missbräuchlicher Nutzung und Vandalismus kommen können.

Aus diesem Grund habe ich für die Nutzer zwei verschiedene Ansichten erstellt:

¹<http://www.android-kiosk.com> (Stand 10.12.2015)

- eine Backend-Sicht, auf der sie ihre Änderungen machen konnten
- eine Frontend-Sicht, die auf dem Tablet angezeigt wurde, nach 5 Minuten den aktuellen Stand abspeicherte und die Sicht auf den aktuellsten Stand der Backend-Sicht setzte

3.2 Auswertung

Das Experiment lief 12 Tage. Die Nutzer erstellten sich zu Beginn jeweils einen Nutzeraccount und richteten ihr Board ein. Jeder der beiden lud ein Profilbild hoch und gab seinen Namen sowie seinen akademischen Grad an. Einer der Tester benutzte das Display um das Banner der Professur zu präsentieren (Abb. 3.2(a)) und um Gäste beispielsweise darüber zu informieren, dass er sich zur Zeit im Büro befand (Abb. 3.2(b)).

Positiv war, dass das Board viel Aufmerksamkeit zu erzeugen schien. Viele vorbeigehende Nutzer blieben stehen, um sich das Display genauer anzusehen und interagierten sogar damit. Somit entstanden viele Zeichnungen, wovon viele leider keinen tieferen Sinn hatten wie beispielsweise die Zeichnung in Abb. 3.2(c). Jedoch gab es auch ab und an Nachrichten, die direkt an die Besitzer des Boards gerichtet waren, wie das Bild einer Kaffeetasse mit einem Fragezeichen aus Abb. 3.2(d), was die Frage nach einer Kaffeepause darstellen sollte.

Nach Beendigung des Experiments führte ich ein Gespräch mit den Testnutzern sowie mit einigen Gästen.

Es stellte sich dabei heraus, dass durch die geringe Auflösung des Tablets den meisten Besuchern gar nicht bewusst wurde, dass zwei Nutzer auf dem Board angemeldet waren. Man musste manuell in der Ansicht scrollen, um den Inhalt des zweiten Nutzers einsehen zu können. Jedoch war auch das ein Problem, da Scrolling nur bei ausgeblendeter Sidebar möglich war. Die meisten Nutzer interagierten demnach nur mit der oberen Hälfte des eigentlichen Inhalts.

Ein weiteres Problem: das Tablet hatte eine sehr geringe Leistung und das Touch-Interface war nicht genau. Deshalb waren Interaktionen mit dem Board langsam und ruckelig.

Positive Äußerungen gab es bezüglich angezeigter Statusmeldungen, wie beispielsweise "I am inside". Die meisten Besucher fanden diese Meldung hilfreich, weil sie deshalb wussten, ob der Mitarbeiter zur Zeit im Raum war. Weil zum Empfangen einer Benachrichtigung eine Browser-App erforderlich war, hatten sie keine Möglichkeiten darüber benachrichtigt zu werden, wenn jemand etwas auf ihrem Board änderte. Das war den Testern leider nicht bewusst. Besser wäre es gewesen, wenn die Besitzer eine Email erhalten hätten, sobald jemand etwas mitteilen wollte.



Abbildung 3.2: Einige Beispielbilder

Als Schlussfolgerung des Experiments entschied ich mich dazu, ein eigenes System zu entwerfen. Es war sinnvoller, dieses System nach den benötigten Anforderungen zu entwickeln, anstatt NetBoards so anzupassen, dass es die Anforderungen erfüllte. Das neue Projekt sollte gut auf Tablets laufen und für eine kleinere Bildschirmgröße angepasst sein. Außerdem sollte jeder Nutzer eine eigene Sicht bekommen, da sonst auf dem Tablet zu wenig Platz wäre. Nur die Besitzer durften die Möglichkeit bekommen, den angezeigten Inhalt zu ändern. Den Besuchern musste es dennoch möglich sein, mit den Besitzern in Kontakt zu treten, welche darauf eine Benachrichtigungs-Email erhalten sollten.

Kapitel 4

Bauhausboards

Das neue Projekt bekam den Namen “Bauhausboards”. In einer Entwurfsphase wurde entschieden, wie das System aussehen und welche Funktionen es bieten sollte. Dabei flossen die Ergebnisse der Vorstudie und anderer Projekte mit ein.

Die Voraussetzung war, dass als Anzeigegeräte für die Türschilder weiterhin Tablets genutzt werden sollten. Daraufhin wurde der Entwurf in ein Programm umgesetzt, wobei manche geplanten Funktionen geändert, entfernt oder neue hinzugefügt wurden.

4.1 Entwurf

4.1.1 Entwurf des Systems

Die Planung begann damit, die Plattform des Systems auszuwählen. Es gab die Möglichkeit eine Android-Applikation zu erstellen, die auf jedem Tablet hätte installiert werden müssen. Dabei gab es jedoch das Problem, dass die Nutzer auch von unterwegs mit der Anwendung interagieren sollten, wofür ein eigenständiger Server für die Datenverwaltung notwendig gewesen wäre. Außerdem müsste durch die unterschiedlichen Tablet- und Smartphone-Betriebssysteme wie Android, Apple iOS oder Windows Mobile, die App für jede Architektur portiert werden.

Deswegen fiel, wie bei NetBoards, die Wahl auf eine Web-Applikation, da jedes Tablet, jedes Smartphone und jeder PC Webseiten darstellen kann und es nicht nötig ist, extra eine App auf den Geräten zu installieren. Ein zentraler Webserver generiert eine Webseite mit allen nötigen Frontend Funktionen, die von allen Geräten mit Web-Browser (Clients) angezeigt werden kann. Dadurch schränkte sich der Programmieraufwand auf Server- und Client-Funktionen ein.

Die Applikation sollte nicht bei jeder kleinsten Interaktion mit der Webseite diese neu laden müssen. Das hätte einen viel zu großen Kommunikationsaufwand erzeugt, da bei jeder Kommunikation statische Daten übertragen werden wären. Deswegen musste der Server nur beim ersten Aufruf das Grundgerüst der Webseite mit allen Client-Funktionen ausliefern. Ein Teil der Funktionen würde dann Daten nachladen oder an den Server schicken, wodurch ein dynamischer Inhalt erzeugt wird. Dies sollte mit dem “CRUD” (Create, Read, Update and Delete) Prinzip realisiert werden. Der Server wartet, nachdem er eine Seite ausgeliefert hat, auf spezifische Anfragen des Clients und liefert dementsprechend Daten aus (Read) oder ändert den Zustand der Daten (Create, Update und Delete). Zudem gewährleistet dieses Prinzip, dass bei Verbindungsabbrüchen die Seite auf dem Client weiter läuft und nur so lange keine Daten ändern oder nachladen kann, bis die Verbindung wieder aufgebaut wurde.

Die anfallenden Daten des Systems mussten in irgend einer Form auf dem Server gespeichert werden. Die beste Methode dafür war die Nutzung einer Datenbank. Anders als bei NetBoards, wo alle Daten als JSON¹-Dateien auf dem Server gespeichert wurden, sah ich diese Methode als geeigneter an. Mit einer Datenbank war es möglich Relationen besser darzustellen, Konsistenz zu bewahren und Redundanz zu vermeiden. Zudem sind damit Daten zentralisiert zusammengefasst, wodurch eine bessere Suche möglich ist.

Die Wahl des Datenbanksystems fiel auf SQLite², da es das am meisten verbreitete relationale Datenbanksystem der Welt ist. Es war einfach zu benutzen und hatte eine gute Dokumentation.

Das Gerüst der Webseite sollte auf HTML basieren, wobei die hauptsächlichen Client-Funktionen üblicherweise in Javascript umgesetzt werden. Da die Clients viele Funktionen bieten und der Server nur zum Ausliefern der Webseite und für den Datenaustausch dienen sollte, machte es Sinn für den Server Node.js³ zu verwenden.

In der Abb. 4.1 ist der Aufbau des Systems zusammengefasst. Auf dem Server befinden sich der Webserver und die Datenbank. Der Webserver stellt Anfragen an die Datenbank und erhält dafür entsprechende Ergebnisse zurückgeliefert. Wenn ein Client die Adresse des Servers aufruft generiert der Webserver die Webseite und sendet sie mit allen benötigten Funktionen an den Client. Solange die Seite nicht explizit neu geladen werden soll, kommuniziert der Client

¹“JSON ist ein, für Menschen leicht zu lesendes und Maschinen leicht zu parsendes, Datenaustauschformat.” [15]

²<https://www.sqlite.org> (Stand 01.12.2015)

³“Node.js ist serverseitiges Javascript basierend auf Google Chrome’s V8 Javascript Engine.” [16]

danach nur durch CRUD-Interaktionen mit dem Server.

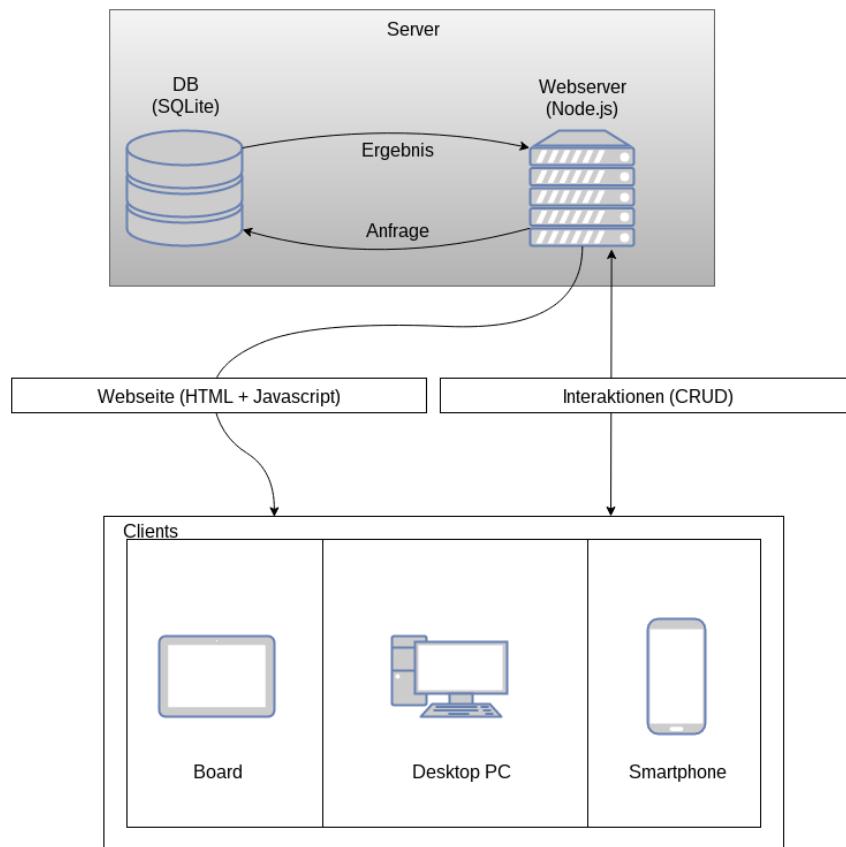


Abbildung 4.1: Bauhausboards - Systemaufbau

Da die Nutzer die Möglichkeit bekommen sollten, individuellen Inhalt präsentieren zu können, musste ein geeigneter Editor programmiert werden. Damit sollte es möglich sein, einfache Zeichnungen anzufertigen, Texte zu schreiben oder Bilder hochladen zu können. Es war wichtig, dass der Editor die Funktionen eines Whiteboards in manchen Zügen übernehmen oder möglicherweise sogar verbessern konnte.

Folgende Whiteboard-Funktionen musste der Editor daher abbilden:

- Zeichnen mit verschiedenfarbigen Stiften
- Entfernen von Zeichnungen mit einem Schwamm
- Anheften von Bildern oder Zettel per Magnet
- Neuanordnung von angehefteten Elementen

Als Grundlage für den Editor entschied ich mich für die Javascript Bibliothek Paper.js⁴. Es ist ein Open Source Framework für die Darstellung und Manipulation von Vektorgrafiken, welches auch die Grundlage des Editors aus dem NetBoards Projekt bildete [1].

Eine weitere wichtige Entscheidung war, dass nur die an den Boards registrierten Nutzer den dargestellten Inhalt ändern können sollten, da sich in der Vorstudie herausstellte, dass es durch die Anonymität der Besucher häufig zu unerwünschten Änderungen kam. Dadurch sollte ein Sicherheitsaspekt geschaffen werden, weil keine andere Person die Daten ändern durfte außer derjenige, der sie erstellt hat.

Den Besuchern sollte jedoch weiterhin die Möglichkeit geboten werden, mit den registrierten Nutzern in Kontakt zu treten. Dies konnte mit einer separaten Nachrichtenfunktion realisiert werden.

Zur Planung erstellte ich zu Beginn einen Paper-Prototype⁵ für das Interface und die Funktionen, die es bieten sollte.

Dieser Entwurf befindet sich im [Anhang].

4.1.2 Entwurf des Interfaces

Hauptansicht

Da den Gästen in der Vorstudie nicht bewusst war, dass mehrere Personen auf dem Board registriert waren, was der Größe des Tablets zu verschulden war, musste der vorhandene Platz anders aufgeteilt werden. Das NetBoards System teilte den Bereich verschiedener Benutzer räumlich, indem es den verfügbaren Platz durch die Anzahl der registrierten Benutzer teilte. Auf einem 10,1 Zoll großem Tablet macht dieser Ansatz jedoch wenig Sinn, da bei mehr als einem Benutzer der verfügbare Platz zu klein wäre (Abb. 4.2).

Deswegen sollte jeder registrierte Nutzer die ganze Fläche zu Verfügung haben, wobei nach einer festgelegten Zeit die Sicht wechselt und die Daten des nächsten Nutzers angezeigt werden.

Die Datenansicht sollte in zwei Schichten aufgeteilt werden. Zum einen eine Editor-Schicht für selbst gezeichnete Skizzen, Texte, Bilder und animierte Gifs sowie eine Hintergrundschicht, in der die Nutzer eine Webseite anzeigen lassen können. Da NetBoards diese Funktion bereits bot und die Testnutzer in der Vorstudie diese ebenfalls nutzten, entschied ich mich, sie auch umzusetzen.

Um im System navigieren zu können musste es dafür ein Menü geben, das auf den Tablets nicht viel Platz benötigte. Die beste Variante dafür war eine Side-

⁴<http://paperjs.org> (Stand 01.12.2015)

⁵Ein Paper-Prototype ist ein auf Papier gezeichneter Entwurf.

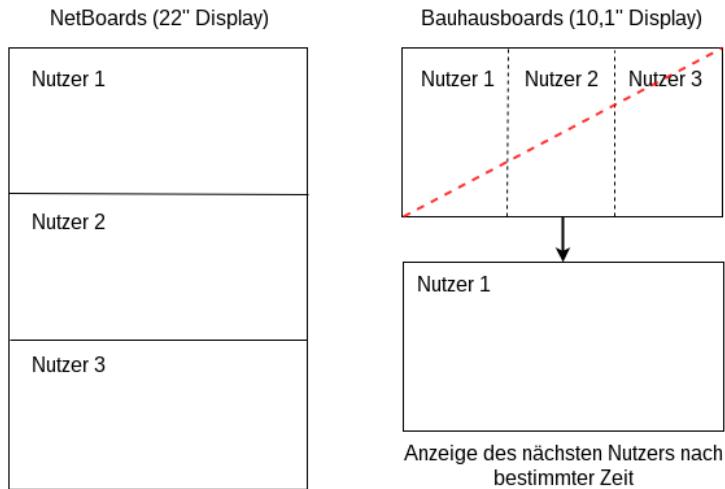


Abbildung 4.2: Vergleich der Platzverteilung NetBoards - Bauhausboards

bar, die standardmäßig eingeklappt ist und sich per Klick oder Swipe-Geste⁶ öffnen ließ. In diesem Menü sollten alle Navigationselemente untergebracht sein.

Um Informationen über die Nutzer anzeigen zu können, wurde eine zusätzliche Fläche benötigt. Dafür sollte im oberen Bereich des Displays ein Teil nur für solche Informationen reserviert sein. In diesem Header konnte man das Profilbild, den Namen und eine Beschreibung des aktuell ausgewählten Nutzers darstellen. In der Vorstudie wurde es sehr gut aufgenommen, dass einer der Nutzer seinen aktuellen Status angegeben hat. Deswegen sollte diese Funktion neben den Nutzerinformationen im Header zu finden sein.

Damit die Gäste mit den Besitzern in Kontakt treten können, musste es eine Nachrichtenfunktion geben. Dabei sollten die Benutzer, die eine Nachricht erstellen wollen, einen oder mehrere Besitzer auswählen dürfen. Danach soll der Editor aktiviert werden, wo die Besucher eine Zeichnung oder einen Text erstellen können. Die Besitzer müssen, nachdem die Nachricht abgeschickt wurde, darüber informiert werden, dass ihnen eine Nachricht hinterlassen wurde.

In Abb. 4.3 ist der konzeptionelle Aufbau der Hauptansicht mit Sidebar, Header und Nutzerinhalt skizziert. Zudem sind auf der linken Seite von Abb. 4.4 die Funktionen der Hauptansicht (Main View) im System eingeordnet.

⁶Eine Swipe-Geste ist eine Fingergeste, bei der eine Aktion durch seitliches Wischen des Fingers auf dem Interface ausgeführt wird.



Abbildung 4.3: Aufbau der Hauptansicht

Benutzer-Backend

Die Besitzer der Boards sollten ihren öffentlichen Inhalt ändern, einen Status setzen und empfangene Nachrichten lesen können. Dafür müssen sie sich vorher im Benutzer-Backend authentisieren.

Das Backend soll den Nutzern die Möglichkeit bieten, ihre Daten, wie das Profilbild, die Benutzerbeschreibung oder das Passwort ändern zu können.

Um die angezeigten Daten der Hauptseite anzupassen, wird der selbe Editor verwendet, der auch bei der Erstellung von Besuchernachrichten zum Einsatz kommt.

Für den Status sollen die Nutzer einen kurzen Text und den Endzeitpunkt festlegen, der dann auf der Hauptseite angezeigt wird.

Außerdem mussten sie die Möglichkeit bekommen, erhaltenen Nachrichten einzusehen zu können, wobei ungelesene Nachrichten dabei besonders gekennzeichnet werden.

Die Änderung der Benutzerdaten sollten in einem gesonderten Einstellungsreich möglich sein.

Da das Benutzer-Backend auch vom aufgehängten Display aus erreichbar sein sollte, damit die Nutzer beim Verlassen des Raumes schnell einen Status setzen konnten, mussten sie sich schnell authentisieren können. Deswegen entschied ich mich für zwei Authentisierungsmethoden. Ein vierstelliger Pin sollte zur schnellen Anmeldung in das Backend dienen. Die Benutzereinstellungen konnten dann nach einer klassischen Passwortauthentisierung vorgenommen werden.

In Abb. 4.4 wird der Aufbau der Seitenstruktur mit dem Weg von der Hauptansicht (Main View) zum Benutzer-Backend (User Backend) dargestellt. Zudem

wird gezeigt, welche Funktionen wo zu finden sind und in welchen davon der Editor zum Einsatz kommt.

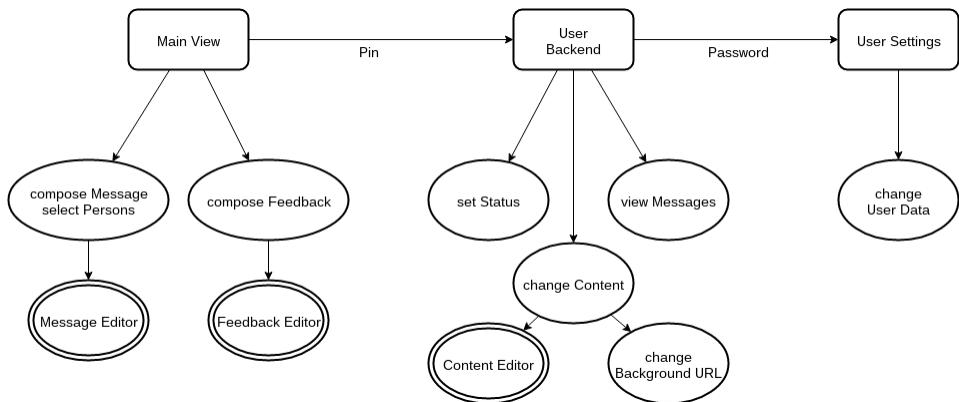


Abbildung 4.4: Bauhausboards - Interface Funktionen mit Editorkomponenten

Administrator-Backend

Der Administrator der Boards muss die Funktionen bekommen, um das System verwalten zu können. Das beinhaltet das Erstellen, Ändern und Entfernen von Nutzern, sowie das Registrieren von neuen Tablets.

Zudem soll er für Studienzwecke die Inhalte und Nachrichten der Benutzer einsehen können.

4.1.3 Entwurf der Datenbank

Durch die geplanten Funktionen mussten anfallende Daten in einer Datenbank gespeichert werden. Es bestand die Möglichkeit, eine dokumentenbasierte oder eine relationale Datenbank dafür zu nutzen. Eine dokumentenbasierte Datenbank hätte den Vorteil gehabt, dass alle Anfragen eine JSON-Zeichenkette zurückgegeben hätten. JSON wird häufig in der Verbindung mit Javascript und dessen Serverkommunikation verwendet. Relationale Datenbanken hingegen verwenden eine Tabellenstruktur für die Speicherung von Daten. Mit ihnen lassen sich Beziehungen zwischen den Daten einfacher darstellen.

Für die Datenbank von Bauhausboards entschied ich mich für eine relationale Datenbank.

Der Datenbankentwurf (Abb. 4.5) beinhaltete 6 Objektklassen. Diese Objektklassen werden in der Datenbank jeweils als Tabelle gespeichert. Um alle Objekte eindeutig beschreiben zu können, wurde in jeder Tabelle eine ID als Primärschlüssel⁷ hinzugefügt.

⁷Ein Primärschlüssel ist ein Attribut zur eindeutigen Identifikation eines Datensatzes.

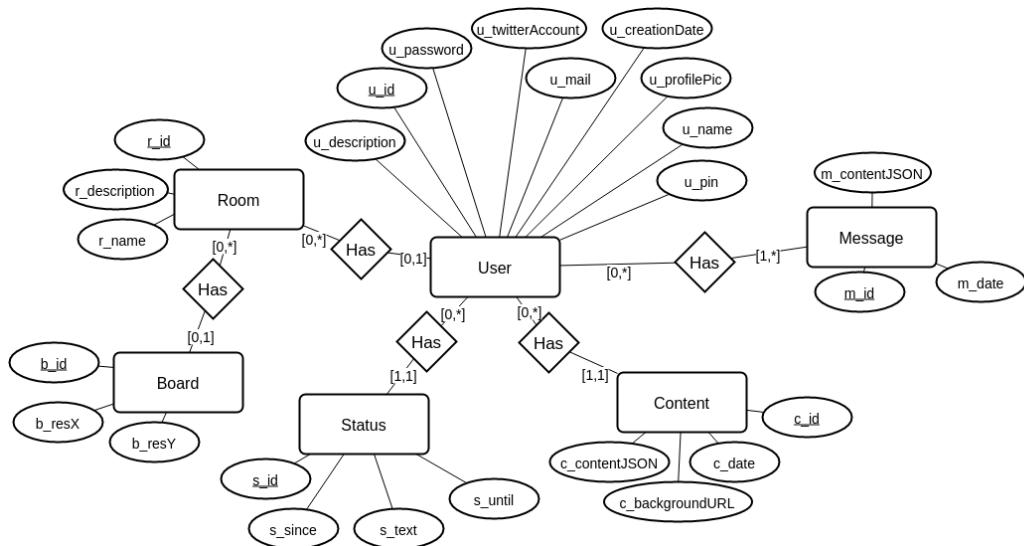


Abbildung 4.5: Bauhausboards - Datenbankentwurf

User Tabelle

Die User-Tabelle ist die Haupttabelle und dient zur Speicherung von Nutzeraccounts. Ein Nutzer muss auf jeden Fall einen Namen, eine Emailadresse, ein Passwort und einen Pin haben. Zusätzlich können die Nutzer auch ein Profilbild, eine eigene Beschreibung und einen Twitter-Account angeben. Wenn ein Benutzeraccount erstellt wird, soll das aktuelle Datum mitgespeichert werden. Ein Benutzer kann entweder in einem oder keinem Raum arbeiten. Er kann beliebig viele Status und Inhalte erzeugen und beliebig viele Nachrichten erhalten.

Board Tabelle

Diese Tabelle ist die digitale Repräsentation von tatsächlich aufgehängten Displays. Da verschiedene Tablets eine unterschiedliche Auflösung haben können, musste diese in der Datenbank für jedes Board gespeichert werden. Ein Board kann entweder noch keinem oder genau einem Raum zugeordnet sein.

Room Tabelle

In dieser Tabelle werden die möglichen Räume, vor denen Boards angebracht werden können, registriert. Jeder Raum hat einen Namen und möglicherweise eine Beschreibung. Ein Raum kann entweder keinem oder mehreren Boards zugeordnet sein. Zudem können beliebig viele Personen in einem Raum arbeiten.

Content Tabelle

Mit dieser Tabelle werden die Inhalte der Nutzer gespeichert. Ein Inhalt besteht aus einem JSON-Objekt und einem Erstelldatum. Wenn für den Inhalt eine Hintergrundwebseite angegeben wurde, wird deren Adresse ebenfalls an dieser Stelle eingetragen. Jeder Inhalt hat genau einen Nutzer, der ihn erstellt hat.

Status Tabelle

Die Status Tabelle dient zur Speicherung von Nutzerstatus. Jeder Nutzerstatus hat einen Statustext, ein Erstelldatum, einen Endzeitpunkt und genau einen Ersteller.

Message Tabelle

In der Message Tabelle werden Nachrichten abgelegt, die von Besuchern erstellt wurden. Eine Nachricht hat einen Nachrichteninhalt und ein Erstelldatum. Da die Nachricht an mehrere Besitzer adressiert sein kann, hat jede mindestens einen Empfänger.

Die Relationen zwischen User und Message, sowie zwischen User und Room müssen jeweils über eine dritte Tabelle dargestellt werden. In diesen Tabellen sind dann nur die Primärschlüssel der beiden beteiligten Objekte gespeichert.

4.2 Umsetzung

Die im Entwurf geplanten Konzepte wurden dann in ein lauffähiges System umgesetzt (Abb. 4.6). Dabei änderten sich einige Ansätze und andere kamen neu hinzu.

4.2.1 Funktionen

Nachdem der Aufbau des Systems geplant war, mussten diese Pläne in ein lauffähiges Programm umgesetzt werden. Um die Entwicklung mit Node.js zu vereinfachen, wurde das Framework Express⁸ verwendet. Damit ließ sich eine einfache Webapplikation erstellen.

Da die hauptsächlichen Funktionen auf den Clients ausgeführt werden sollten, mussten diese in Javascript geschrieben werden. Am wichtigsten war, dass sich die Seite nicht neu lud, wenn man durch sie navigierte. Dies wurde mit

⁸<http://expressjs.com> (Stand 02.12.2015)

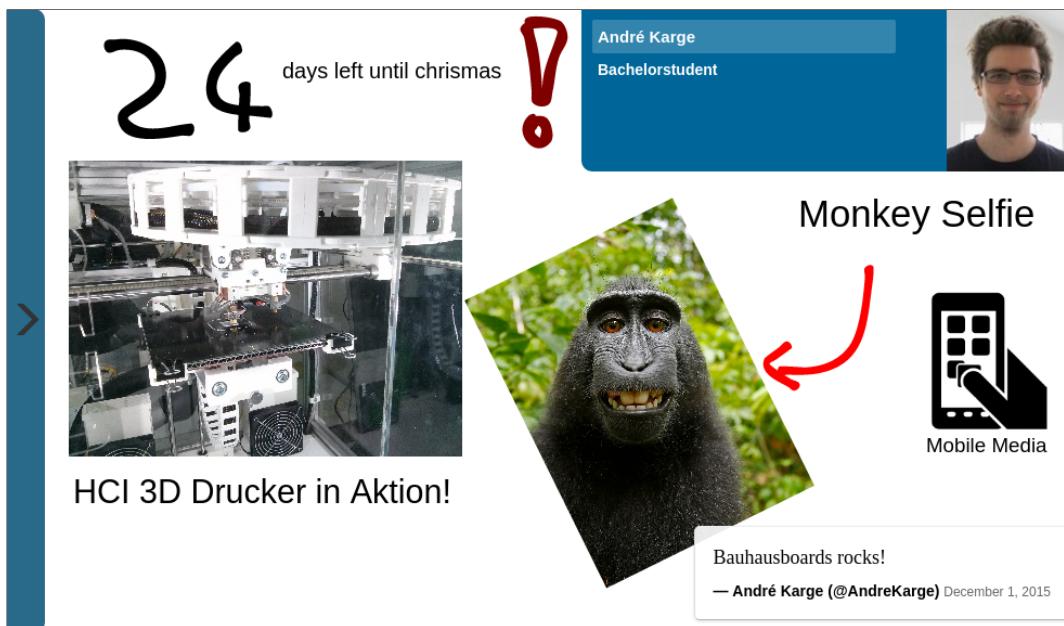


Abbildung 4.6: Bauhausboards - Hauptseite

jQuery⁹ realisiert. jQuery ist eine Bibliothek zur einfachen Manipulation der HTML-Struktur einer Webseite. Mit ihr war es einfach, die dargestellte Ansicht anzupassen.

Falls an einer Stelle der Seite Daten vom Server benötigt wurden oder Daten auf dem Server geändert werden mussten, wurde dies mit Ajax-Anfragen durchgeführt. Ajax (Asynchronous Javascript and XML) ist eine Funktion für asynchronen Datenaustausch zwischen einem Client und einem Server. Während ein Client auf die Ajax-Antwort des Servers wartet, können weiterhin andere Funktionen ausgeführt werden.

Die Sidebar sollte keinen Platz wegnehmen und musste daher ein- und ausklappbar sein. Zudem sollten die Benutzer direkt erkennen, dass es sich um eine Sidebar handelt, wodurch ein Pfeil als Kennzeichnung daran angebracht wurde (Abb. 4.7). Damit sie auf Tablets leichter zu öffnen war, nutzte ich TouchSwipe¹⁰. Mit dieser Bibliothek konnten Swipe-Gesten zum Aufruf von Funktionen genutzt werden.

Die Passwörter und Pins durften nicht im Klartext gespeichert werden. Dafür wurde die crypto-js¹¹ Bibliothek verwendet, um die sicherheitsrelevanten Daten mit SHA256¹² zu hashen und sie danach in der Datenbank zu speichern

⁹<https://jquery.com> (Stand 10.12.2015)

¹⁰<http://labs.rampinteractive.co.uk/touchSwipe> (Stand 05.12.2015)

¹¹<https://code.google.com/p/crypto-js> (Stand 05.12.2015)

¹²SHA256 (Secure Hash Algorithm 256) ist eine Krypto-Hashfunktion.

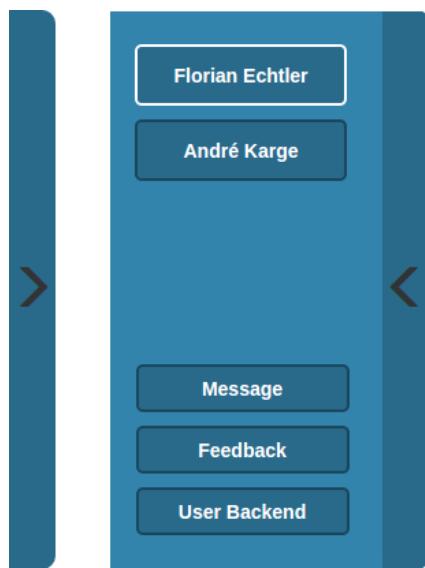


Abbildung 4.7: Bauhausboards - geschlossene und offene Sidebar

oder sie mit den Werten der Datenbank abzugleichen.

Damit die Besitzer der Boards benachrichtigt werden, wenn Gäste ihnen Nachrichten hinterlassen, musste der Server Emails verschicken können. Die Bibliothek `emailjs`¹³ brachte diese Funktion mit. Zusätzlich benötigte der Server zum Verschicken von Emails einen eigenen Mailserver oder wenigstens einen Account an einem anderen Mailserver. In den Benachrichtigungsmails sollte ein Link mit einem Token¹⁴ zu der jeweiligen erhaltenen Nachricht im System sein. Der Token wurde mit `crypto-js` beim Abspeichern der Nachricht auf dem Server erzeugt. Wenn der Empfänger auf den Link klickt, wird der Token geprüft und die entsprechende Nachricht angezeigt. Dafür muss er sich nicht extra authentisieren.

Die Farbgestaltung der Seite wurde schlicht gehalten und orientierte sich an den Hausfarben der Medienfakultät der BUW.

Während der Umsetzung änderte sich die Entscheidung, alle Funktionen in einer einzigen Seite anzubieten, vom geplanten Konzept.

Es war ursprünglich geplant, dass der Administrator normal über die Seite die Administratorfunktionen ausführen kann. Da diese Funktionen aber rein funktional waren und nicht auf den Boards gebraucht wurden, ergab es mehr Sinn diese gesondert in eine zusätzlichen Webseite auszulagern.

¹³<https://travis-ci.org/eleith/emailjs> (Stand 05.12.2015)

¹⁴Ein Token ist ein Attribut zur Freigabe von Daten zwischen Parteien. Wer den Token hat, kann die Datei lesen.



(a) Nutzer verfügbar (b) Nutzer nicht verfügbar

Abbildung 4.8: Bauhausboards - Nutzerstatus

Bauhausboards rocks!
— André Karge (@AndreKarge) December 1, 2015

Abbildung 4.9: Bauhausboards - Twitterfunktion

Zur geplanten Statusfunktion kam während der Umsetzung noch die Idee auf, dass sich Nutzer als “Abwesend” markieren konnten. Dazu sollte im Benutzer-Backend eine zusätzliche Option eingebaut werden. Um diesen Status zu kennzeichnen, wurden dabei die Profilbilder der Nutzer ausgegraut und ein Text darüber angezeigt, dass der Nutzer zur Zeit nicht verfügbar ist (Abb. 4.8). Außerdem kam eine Twitter-Funktion hinzu. Die Besitzer der Boards konnten einen Twitteraccount in ihren Einstellungen eintragen, wodurch auf ihrer Seite ihr aktuellster Tweet geladen und angezeigt wurde (Abb. 4.9).

4.2.2 Editor

Der Kern der Client-Funktionen war der Editor zum Erzeugen von Nutzer-Inhalten und Besuchernachrichten. Da Paper.js nur ein Framework war, musste das Interface dafür von Grund auf implementiert werden. Voraussetzungen für Paper.js war, dass die Webseite mit dem neuesten HTML5 Standard entwickelt wurde, weil diese Version ein neues Canvas-Element bot, mit dem Grafiken gezeichnet werden konnten.

Die Funktionen des Editors erfüllten die Anforderungen des Entwurfs sehr gut. Mit einer Stiftfunktion ist es möglich, Zeichnungen per Hand anzufertigen. Dabei kann der Benutzer die Linien in vier verschiedenen Linienstärken zeichnen.

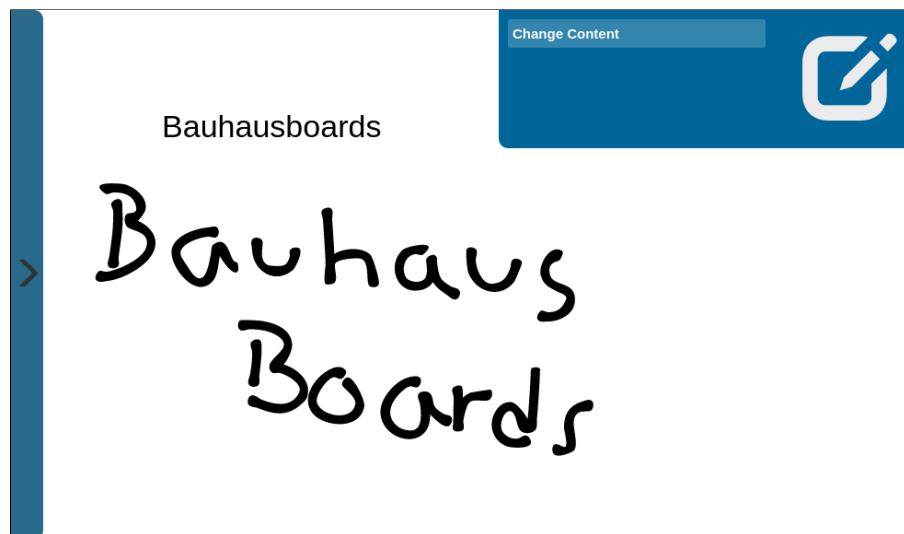


Abbildung 4.10: Bauhausboards - Text Sketch

Außerdem wurde ein Textwerkzeug implementiert, mit dem ein Text per Tastatur eingegeben werden konnte (Abb. 4.10).

Damit das Gezeichnete beliebig auf der Zeichenfläche angeordnet werden kann, musste es ein Auswahlwerkzeug geben (Abb. 4.11). Die Nutzer konnten damit einen Auswahlrahmen um Objekte ziehen, welche dadurch markiert wurden. Die Markierung zeichnete sich durch ein Auswahlrechteck mit Kreisen an den Ecken, in der oberen Mitte sowie einem kleinen Menü aus. Durch Ziehen des Auswahlrechteckes wurde die Auswahl an eine gewünschte Stelle verschoben. Zudem war es möglich, die Auswahl durch Klick auf die Kreise in den Ecken des Auswahlrechteckes zu skalieren oder durch Klick auf den Kreis in der oberen Mitte zu rotieren. Die Funktionen des kleinen Menüs beinhalteten das Löschen oder Kopieren der Auswahl, sowie das Platzieren vor oder hinter anderen Objekten auf der Zeichenfläche.

Da es für Whiteboards und Tafeln Stifte oder Kreide in unterschiedlichen Farben gibt, musste es für den Editor eine Farbpalette geben, mit der man die Farbe der Zeichnungen und Texte ändern konnte.

Die Funktion eines Tafelschwamms wurde mit einem einfachen Radiergummi-Werkzeug implementiert.

Wie in jedem Editor, musste es eine Möglichkeit geben, die letzte Interaktion rückgängig zu machen oder wiederherzustellen. Dafür wurden Undo- und Redo-Funktionen eingebaut.

Um Bilder darzustellen, konnten diese von einem anderen Browserfenster oder einem Dateibrowser per Drag-and-Drop in den Editor geschoben werden, wodurch diese auf den Server geladen und im Editor angezeigt wurden. Damit



Abbildung 4.11: Bauhausboards - Objekt Selektion

war es auch möglich animierte Gifs hochzuladen.

Da man mit dem Editor alle Objekte verschieben, skalieren und rotieren konnte, hatte er einen Vorteil gegenüber klassischen Whiteboards und Tafeln. Dort war es bisher nur möglich, per Magnet angebrachte Papiere zu verschieben oder zu rotieren. Der Editor hatte durch die Skalierung von Objekten deswegen einen Freiheitsgrad mehr. Zeichnungen konnten nur entfernt und an anderer Stelle wieder neu gezeichnet werden, welche in der digitalen Variante nur verschoben werden brauchten. Außerdem konnte man am Editor nachträglich die Farbe von Zeichnungen ändern, was am Whiteboard nicht möglich ist. Den größten Vorteil hat der Editor durch die Möglichkeit, animierte Gifs anzeigen zu lassen, da Whiteboards und Tafeln nur statischen Inhalt präsentieren können.

4.2.3 Probleme

Während der Arbeit an Bauhausboards traten diverse Probleme auf.

Da das HTML5 Canvas Element nur statischen Inhalt anzeigen konnte, musste die Anzeige von animierten Gifs auf andere Art gelöst werden. Dazu wurde eine Gif-Schicht über den Editor gelegt. Wenn sich ein Gif-Objekt in der Editor-Schicht befand, wurde dieses Objekt in die Gif-Schicht geladen und die Transformationen aus dem Editor auf die Kopie angewandt.

Ein weiteres Problem war, dass durch die horizontale Ausrichtung der Tablets der Platz des Headers zu groß war, da er über die gesamte Breite des Tablets

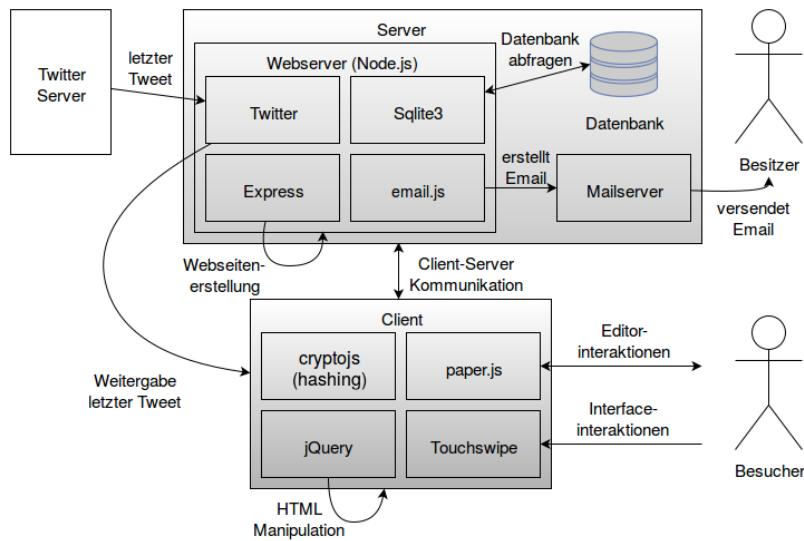


Abbildung 4.12: Bauhausboards - Aufteilung der Systembibliotheken

lief. Weil er aber nur das Profilbild, den Namen, die Beschreibung und den Status eines angezeigten Nutzers beinhalten sollte, wurde er verkleinert. Dadurch entstand mehr Platz für den Inhalt der Nutzer.

Hinzu kam das Problem, dass Desktop PC's eine größere Auflösung hatten als Tablets. Deswegen konnten Zeichnungen, die außerhalb der Board-Auflösung erstellt wurden, nicht auf den Tablets angezeigt werden. Die Lösung dafür war, die Größe des Editors auf die Auflösung des Tablets anzupassen und bei größeren Auflösungen diese Größe mit einem Rahmen zu kennzeichnen.

4.3 Vergleich mit NetBoards und Hermes

Im Vergleich zu NetBoards wurde bei Bauhausboards der Platz nicht durch die Anzahl der angemeldeten Benutzer geteilt. Das hatte den Vorteil, dass jeder Benutzer die ganze Fläche zur Verfügung bekam. Dadurch konnte jedoch immer nur ein Nutzer zur selben Zeit angezeigt werden.

Bauhausboards bot, wie Hermes, die Funktion, den Besitzern der Boards Nachrichten zu hinterlassen, wohingegen NetBoards diese Funktion nicht hatte.

Bei NetBoards kann jeder Gast den angezeigten Inhalt nach Belieben ändern. Da sich in der Vorstudie von Bauhausboards gezeigt hatte, dass diese Funktion so nicht angebracht war, durften bei Bauhausboards nur die Besitzer diesen ändern und die Besucher in einer extra Funktion Nachrichten hinterlassen. Wenn dann ein Besucher einem Besitzer eine Nachricht hinterlässt, wird dem

Empfänger, wie beim Hermes System, eine Email geschickt.

Der Editor von Bauhausboards basiert auf der gleichen Bibliothek, wie bei NetBoards. Zudem wurde die Idee zur Anzeige von Webseiten als Hintergrundschicht ebenfalls vom NetBoards System übernommen.

Kapitel 5

Studie

Um das System in der Praxis zu testen, musste im Anschluss eine Nutzerstudie durchgeführt werden. Dies beinhaltete die Planung und Durchführung der Studie, sowie die Auswertung der Ergebnisse. Es mussten die Rahmenbedingungen der Studie geplant und Testpersonen angeworben werden, die darauf in einem Interview Fragen beantworten und einen standardisierten Fragebogen zur allgemeinen Benutzung ausfüllen mussten.

5.1 Entwurf

Nachdem die Umsetzung des Bauhausboards Systems abgeschlossen war, galt es einen Test mit Mitarbeitern der Medienfakultät der BUW durchzuführen. Dafür wurde am internen Rechenzentrum der Uni ein Linux Server mit 2,53GHz Intel Xeon Dualcore, 2GB RAM und Ubuntu Version 14.04 aufgesetzt. Auf dem Server liefen Node.js, damit Bauhausboards ausgeführt werden konnte und Postfix¹ als Mailserver. Dieser Mailserver konnte nur Emails an Adressen innerhalb des Universitätsnetzes verschicken, was für die Studie jedoch kein Problem darstellte.

Als Displays wurden neue Tablets angeschafft, die mehr Leistung besaßen, als das Tablet aus der Vorstudie. Es handelte sich dabei um vier Blaupunkt Discovery 1000c mit 10,1 Zoll Display, 1,33GHz AllWinner A33 Quadcore, 1GB RAM und Android 5. Mit vier Tablets konnten für die Studie vier Räume mit Displays ausgestattet werden.

Damit die Benutzer, wie in der Vorstudie, nicht mit den vorinstallierten Applikationen der Tablets interagieren konnten, musste auch auf den neuen Tablets eine Kiosk-Applikation² installiert werden.

¹<http://www.postfix.org> (Stand 10.12.2015)

²<http://www.android-kiosk.com> (Stand 10.12.2015)



(a) 3D gedrucktes Eckstück



(b) fertige Boardaufhängung

Abbildung 5.1: Ergebnis der Vorarbeit

Zusätzlich zum Kiosk wurde eine Tasker-App³ installiert. Damit konnte man Aufgaben nach bestimmten Systemereignissen ausführen lassen. Es wurde Task erstellt, der beim Start des Android-Systems automatisch die Kiosk-App startete, falls das System neu bootete.

Als Diebstahlsicherung konnte man auf den Tablets Google Locate aktivieren. Es diente zur Ortung der Geräte, falls diese gestohlen worden wären.

Um die Displays aufhängen zu können, musste eine geeignete Befestigungsmöglichkeit gefunden werden. Jedoch sollten keine neuen Löcher zur Aufhängung an den Wänden neben den Büros entstehen. Deswegen mussten die Displays erneut mit Hilfe der bereits aufgehängten Türschilder befestigt werden. Da bei dieser Studie die Tablets nicht, wie in der Vorstudie, durch Löcher in der Tablet-Rückwand angebracht werden konnten, musste die Befestigung anders konstruiert werden. Zu diesem Zweck wurden 3D gedruckte Eckstücke (Abb. 5.1(a)), mit denen die Tablets befestigt werden konnten, auf eine Plexiglasplatte geschraubt (Abb. 5.1(b)). Diese Aufhängung nutzte die Bohrlöcher für die bereits vorhandenen Türschilder, wodurch keinerlei Eingriff in die Bausubstanz entstand.

Damit sich die Testnutzer mit der grundlegenden Funktion von Bauhausboards vertraut machen konnten, habe ich ein Benutzerhandbuch erstellt, welches sich im [Anhang] befindet.

³<http://tasker.dinglisch.net> (Stand 10.12.2015)

5.2 Durchführung

Da nur vier Räume mit Tablets ausgestattet werden konnten, war es nötig, mit den vorhandenen Mitteln möglichst viele Testnutzer organisieren zu können. Mir war es möglich, einige Mitarbeiter der Professur für Systeme der virtuellen Realität (VR) und der Professur für grafische Datenverarbeitung (CG) zur Teilnahme an meiner Studie zu gewinnen.

Im Fachbereich der virtuellen Realität gab es drei Büroräume. Zwei davon mit jeweils drei und einer mit zwei Mitarbeitern. An der Professur für grafische Datenverarbeitung arbeiteten zwei Mitarbeiter, die sich einen Raum teilten. Diese Räume befanden sich alle im selben Gang der Fakultät.

Da ein Mitarbeiter der VR Professur durch seine Forschungsarbeit nicht in seinem Büro war, nahm er nicht an der Studie teil.

Somit konnten insgesamt 9 Doktoranden und Post-Doktoranden als Testnutzer gewonnen werden.



Abbildung 5.2: Aufgehängtes Board

Bevor die Studie begann, wurden die Boards angebracht (Abb. 5.2). Außerdem musste es eine gute WLAN Verbindung für sie geben, was durch zwei WLAN-Access-Points im Gang gewährleistet wurde. Nachdem die Tablets angebracht waren, wurde für jeden Benutzer ein Account erstellt, die Boards eingerichtet und die Nutzer kurz eingewiesen.

Die Dauer der Studie beschränkte sich leider nur auf zwei Wochen. Es sollte mindestens ein Wochenende eingeplant sein, weil sich das Benutzerverhalten nach diesen zwei freien Tagen ändern konnte.

Während der Studie kam es gelegentlich dazu, dass einige Testnutzer Fragen zu bestimmten Funktionen hatten, welche ich direkt geklärt habe. Das System wurde gut genutzt und einige Gäste hinterließen den Testern Nachrichten.

5.3 Auswertung

Nach Abschluss der Studie wurden alle Teilnehmer interviewt. Um diese Interviews zu strukturieren, erstellte ich im Vorfeld einen Fragenkatalog, welcher ebenfalls im [Anhang] zu finden ist. Zudem sollten die Testnutzer einen standardisierten Fragebogen zur allgemeinen Auswertung ausfüllen.

5.3.1 Interview

Der Fragenkatalog der Interviews zielte darauf ab, die Erfahrungen und Anregungen sowie das Verhalten der Nutzer im Bezug auf bestimmte Funktionen des Bauhausboards Systems herauszufinden.

Zur besseren Auswertung wurden die Gespräche aufgezeichnet, wodurch ungefähr 5 Stunden und 30 Minuten an Gesprächsmitschnitten entstanden.

Die Fragen wurden in unterschiedliche Kategorien eingeteilt. Es ging darum, herauszufinden, wie es für die Testnutzer war, Nachrichten zu empfangen und ob sie selber Nachrichten geschrieben haben. Außerdem wurde die Benutzung des Status- und Inhalts-Systems evaluiert und es musste untersucht werden, wie die Testnutzer die Benutzung des Editors empfanden. Zum Schluss wurden sie danach gefragt, was sie von den allgemeinen Boardfunktionen hielten und was sie für zusätzliche Anmerkungen haben.

Nachrichten empfangen

Es galt herauszufinden, wofür die Besucher die Nachrichtenfunktion genutzt haben. Als erstes ging ich mit den Testnutzern deren empfangenen Nachrichten durch. Ich fragte sie, um was es in diesen Nachrichten ging.

Dies waren größtenteils Spaßnachrichten (Abb. 5.3) oder hatten keinen speziellen Bezug. Ein Testnutzer bezeichnete seine erhaltenen Nachrichten als “temporäres Graffiti”. Teilweise gab es auch Nachrichten mit direktem Bezug. Einige Nutzer wurden durch die Nachrichten gefragt, ob sie mit dem Autor essen gehen wollten oder sind auf ihren angezeigten Inhalt angesprochen worden. Außerdem wollte ein Nutzer wissen, ob der Besitzer zur Zeit in der Uni anzutreffen war.

Da es in den Nachrichten keine Kennzeichnung für Autoren gab war ein weiterer Bestandteil der Befragung, wie die Autoren der Nachrichten identifiziert werden konnten. Das wurde von den meisten Testnutzern als größte Schwäche

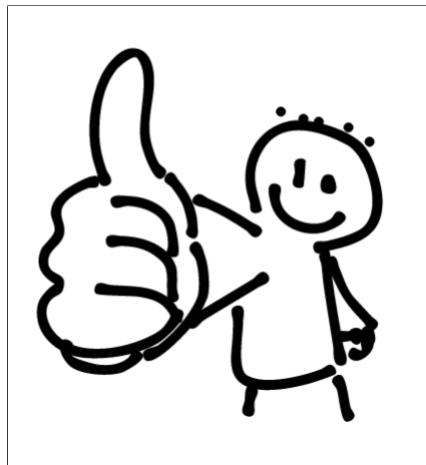


Abbildung 5.3: Ein Beispiel einer Spaß-Nachricht

bemängelt. Sie konnten bei bestimmten Nachrichten nur raten, wer sie erstellt hatte.

Interessant war es zu wissen, wie die Testnutzer, ohne Nutzung des Bauhausboards, für Gäste erreichbar waren. Der Großteil meinte darauf, dass sie üblicherweise per Email erreichbar waren. Andere sagten, Besucher könnten sie per Post-It-Zetteln oder Klopfen an ihrer Tür erreichen.

Für die Kommunikation untereinander gaben einige an, Telefon, SMS oder Instant-Messenger zu nutzen.

Auf die Frage, wie schnell die Testnutzer ihre Nachrichten gelesen haben, meinten fast alle, dass sie direkt die Mail erhalten und auf den eingebetteten Link geklickt haben. Die Funktion des direkten Nachrichtenaufrufs per Link in der Email wurde von allen Testnutzern als sehr gut empfunden, da sie sich somit nicht extra im System anmelden mussten.

Ich fragte die Nutzer, ob es für sie in irgend einer Form anders war, über das Board eine Nachricht zu erhalten. Da mit dem System Zeichnungen verschickt werden konnten meinte ein Testnutzer, dass es eine persönliche Note verschafft: "Die Zeichnungen machen es anders". Die anderen sahen es als eine normale Email-Nachricht, was demnach gut war. Es wurde besonders gut aufgenommen, dass die Tester von den Besuchern über das System angesprochen werden konnten, wenn sie sich derzeit nicht im Büro aufhielten und die Besucher dafür nicht extra ein eigenes Gerät benötigten.

Zur Verbesserung des Nachrichtensystems meinten einige, dass es vereinfacht werden müsste, da sie es als zu kompliziert erachteten. Es sollten weniger Klicks notwendig sein, bis eine Nachricht abgeschickt werden könnte und weniger Untermenüs enthalten. Außerdem meinten sie, dass die Funktion, eine Nachricht an mehrere Personen gleichzeitig erstellen zu können, nicht gebraucht wird.

Nachrichten schreiben

Ich befragte die Testnutzer, ob sie in der Laufzeit der Studie einem anderen Nutzer eine Nachricht hinterlassen haben. Die meisten meinten jedoch, dass sie nur über die Besitzerrolle das System nutzten, weil sie entweder keine Zeit hatten, jemanden eine Nachricht zu schreiben oder persönlich mit diesen Personen redeten. Da sich die Testnutzer untereinander gut kannten, war es für sie einfacher, miteinander zu reden anstatt über das Board eine Nachricht zu schreiben. Einem der Nutzer war der Prozess, eine Nachricht zu erstellen, zu aufwändig und er hat deswegen keine Nachrichten geschrieben.

Nutzerinhalt

Am wichtigsten war es, in Erfahrung zu bringen, was die Testnutzer während der Studie für Inhalte erzeugten und wie für die Benutzung dieser Funktion war. Hauptsächlich nutzte man es dazu, um Spaß-Informationen zu präsentieren. Dazu wurden meist Bilder und Gifs hochgeladen. Einige Tester nutzten diese Funktion auch, um die aktuelle Arbeit zu beschreiben (Abb. 5.4). Einer meinte: er nutze es gern “um von außen einen schnellen Eindruck zu gewähren, was ich mache”.

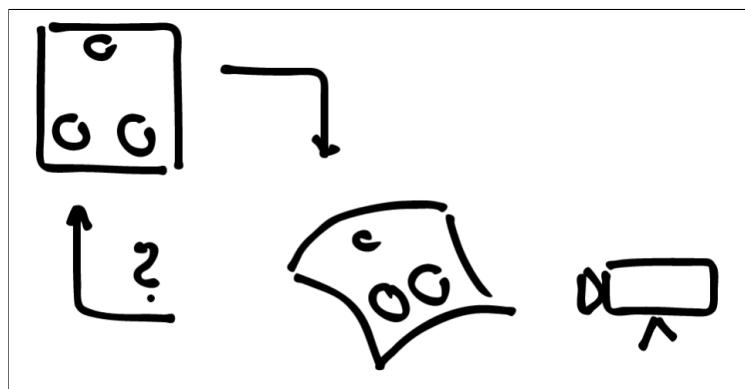


Abbildung 5.4: Ein Beispiel eines Nutzerinhalts zur aktuellen Arbeit

Einige Tester nutzten diese Funktion zum Anzeigen ihres Status, obwohl es dafür ein eigenes Feature gab.

Ansonsten wurde die Hintergrundschicht von einigen zur Anzeige von Webseiten, wie der Seiten der jeweiligen Professuren, genutzt. Ein Nutzer meinte, dass der Editor zum Erzeugen von Inhalten vollkommen ausreichen würde und die Hintergrundschicht maximal als Poweruser⁴-Funktion angeboten werden soll-

⁴Ein Poweruser ist ein Nutzer, der deutlich mehr Kenntnisse über das System hat, als normale Nutzer.

te.

Leider nutzte keiner der Tester die Twitterfunktion. Das lag daran, dass alle Tester keinen Twitteraccount besaßen, bis auf einen, der jedoch nicht seine privaten Twittermeldungen in der Universität präsentieren wollte.

Einige Tester fanden den Ansatz der Informationsdarbietung interessant, da es eine neue Technologie war und man damit sogar bewegten Inhalt erstellen konnte.

Bedauerlicherweise gab es auch Nutzer, die diese Funktion gar nicht nutzten. Sie meinten, dass sie keine Zeit hatten, einen Inhalt zu erstellen oder ihnen einfach nichts eingefallen ist.

Allerdings war es sehr erfreulich, dass es Besucher gab, die auf den Inhalt einiger Tester eingegangen sind und diese explizit darauf angesprochen haben. Dadurch wurde sogar über das System eine Gesprächsgrundlage geschaffen.

Die Inhaltsfunktion als solche wurde von den meisten positiv aufgenommen, da sie einen persönlichen Aspekt bot: "Sehr gut war die Möglichkeit, selber kreativ zu werden". Außerdem wurde es begrüßt, dass über das Board Informationen, Bilder und animierte Gifs präsentiert werden konnten.

Einige waren jedoch der Meinung, dass das System zu kompliziert sei und es zu viele Funktionen bot. Einem Nutzer waren zu viele Schritte nötig, bis er seinen Inhalt ändern konnte.

Status

Die meisten Tester haben diese Funktion gelegentlich genutzt. Einige haben ihren Status jedoch lieber mit der Inhaltsfunktion angegeben. Nur ein Nutzer gab seinen Status beim Verlassen des Raumes am Board an, die anderen taten dies an ihrem PC.

Als Statustexte wurde zum Beispiel angegeben, ob man derzeit beschäftigt war: "busy" oder wo man zu finden ist: "im DBL zu finden".

Einige Nutzer waren der Meinung, dass es zu kompliziert ist, einen Status zu setzen. Es waren ihnen zu viele Klicks notwendig und das Interface zu schwer zu bedienen.

Von einigen Testnutzern wurde es gut aufgenommen, dass es diese Funktion gab und dass sie damit angeben konnten, wo sie sich befanden oder wenn sie nicht gestört werden wollten. Zudem wussten Gäste dadurch, wann ein bestimmter Nutzer wieder anzutreffen war und mussten sich nicht bei dessen Kollegen nach ihm erkundigen.

Auf die Frage, ob die Nutzer das Feature häufiger verwenden würden, wenn es vordefinierte Status geben würde, antworteten die meisten mit ja. Die anderen

meinten, dass sie es nicht für nötig hielten, ihren aktuellen Status anzugeben, da sie eh die meiste Zeit im Büro waren. Sie sahen es als impliziten Status an, wenn sie entweder nicht da waren oder ihre Tür geschlossen war: "Wenn die Tür zu ist, bin ich nicht da".

Die Gespräche ergaben, dass die Funktion zur Abwesend-Markierung fast nicht benutzt wurde und deswegen nicht weiter in dieser Form gebraucht wird.

Einige Tester meinten, dass der Status im Header zu unoffensichtlich ist und daher besser gekennzeichnet werden sollte.

Editor

Fast alle Tester haben den Editor genutzt, um sich Inhalte zu erzeugen. Es wurden eigene Arbeiten skizziert, Texte geschrieben oder einfach nur Spaßzeichnungen erstellt. Die meisten Benutzer meinten, dass der Editor sich schön bedienen lasse, es jedoch einige Unstimmigkeiten gäbe.

Ein Problem war die Identifizierbarkeit der Editor-Icons. Da diese nicht wie in üblichen Editoren waren, konnten nicht alle eindeutig identifiziert werden, was von den meisten Testern bemängelt wurde. Die Nutzer hätten es besser gefunden, wenn sich die Gestaltung der Icons nach den üblichen Standards richten würde.

Da manchen Benutzern anfangs nicht klar war, dass sie Bilder und Gifs per Drag-and-Drop hochladen konnten, nutzten sie dafür nur die Hintergrundschicht. Die Mehrzahl der Nutzer meinte deswegen, dass eine eigene Schaltfläche, um Bilder hochladen zu können, angebrachter wäre.

Einigen Testern war es beim Zeichnen nicht klar, welches Werkzeug sie zur Zeit ausgewählt hatten und wünschten sich eine bessere Kennzeichnung.

Es wurde sich positiv über die Benutzung des Editors geäußert. Einige Tester meinten, dass der Editor allein schon vollkommen als einzige Bauhausboardfunktion ausreichen würde.

Boardfunktionen

Für einige Testnutzer war es nicht ersichtlich, wo sie sich im Backend befanden und ob sie noch angemeldet waren. Deswegen äußerten mehrere Benutzer der Wunsch, das Frontend auch vom Benutzer-Backend zu trennen und auf den Displays nur das Frontend anzubieten.

Weiterhin sollte der durchschnittliche Klickaufwand reduziert werden, damit sich Funktionen schneller durchführen lassen.

Als sie nach den Authentisierungsmethoden gefragt wurden, meinten die meisten, dass zwei Authentisierungen überflüssig sind. Wenn das Frontend vom Backend getrennt wäre, bräuchte man nur ein Passwort im Backend. Einer der Nutzer war sogar der Meinung, die Authentisierung komplett wegzulassen:

“auch wenn es Vandalismus Tür und Tor öffnet”.

Außerdem waren einige Tester nicht mit dem zeitlichen Wechsel der Nutzer auf der Hauptseite einverstanden. Sie hätten es lieber gehabt, wenn alle Nutzer parallel angezeigt werden würden, wodurch alle Benutzer und deren Status auf einen Blick sichtbar wären.

Ich befragte die Testnutzer, welche Vor- und Nachteile sie in Bauhausboards gegenüber Post-It's, Whiteboards und Pinnwänden sahen. Fast alle Benutzer meinten, dass es sehr gut war, dynamischen Inhalt erstellen zu können und dass sie über Besuchernachrichten informiert werden. Die Boards produzieren nicht wie Post-It's Müll und können nicht einfach entfernt werden. Zudem wissen nur der Ersteller und der Empfänger einer Nachricht deren Inhalt.

Als Nachteil wurde die Tatsache angesehen, dass die Boards kontinuierlich Energie verbrauchen und teurer als Post-It's sind. Einer der Testnutzer war der Meinung, dass die Boards im Vergleich zu Post-It-Statusmeldungen nur periphere Aufmerksamkeit erzeugen.

Allgemeine Anmerkungen

Leider war die Laufzeit der Studie nur zwei Wochen, wodurch die Testnutzer nur sehr wenig Zeit hatten, sich an die Boards zu gewöhnen und deren Funktionalität in ihren Arbeitsprozess mit einzubinden.

Einige der Nutzer empfanden es als lästig, dass die Sidebar sich nach einer kurzen Zeit selbstständig schloss: “Du sitzt am Schreibtisch, arbeitest und hast dir deine Werkzeuge hingelegt. Dann kommt ständig deine Freundin und räumt sie wieder weg.” Deswegen sollte sie bei der Benutzung des Editors entweder dauerhaft angezeigt oder die Editorwerkzeuge ausgelagert werden.

Ein Nutzer meinte, dass die Boards sehr gute Notleuchten für den Gang sind, da sie im Dunkeln ausreichend Beleuchtung boten.

5.3.2 UEQ Fragebogen

Um das System zusätzlich auszuwerten, sollten die Testnutzer einen Fragebogen ausfüllen. Das User Experience Questionnaire⁵ (UEQ) ist eine einfache und schnell durchführbare Möglichkeit, interaktive Systeme zu evaluieren. Es beinhaltet einen Fragebogen mit 26 sorgfältig ausgewählten Fragen zu 6 Klassen: Attraktivität, Durchschaubarkeit, Effizienz, Steuerbarkeit, Stimulation und Originalität. Zu jeder Frage gab es 7 Ankreuzmöglichkeiten. Sie gingen von -3 (unattraktiv) bis 3 (attraktiv). Diese Fragen durften nicht abgeändert oder entfernt sowie keine neuen hinzugefügt werden.

⁵<http://www.ueq-online.org> (Stand 10.12.2015)

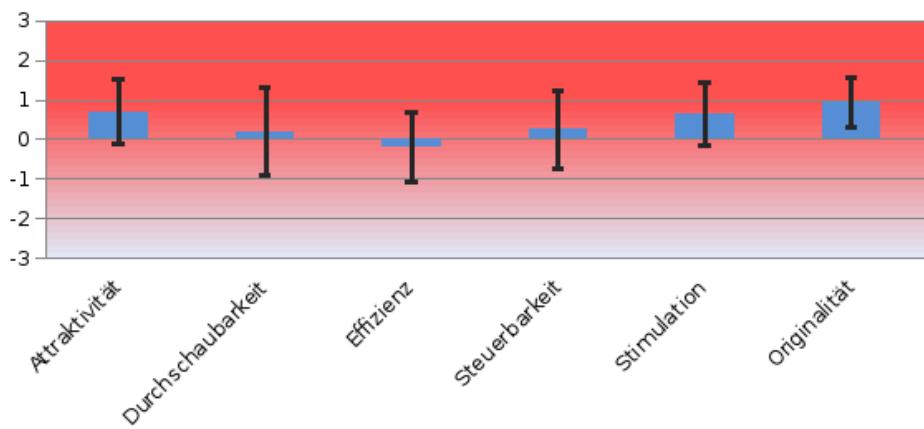


Abbildung 5.5: UEQ - Klassen

In Abb. 5.5 kann man erkennen, dass die Testnutzer das System vorwiegend als attraktiv, stimulierend und originell empfanden.

Da Bauhausbaords ein neues System für die Testnutzer war, ist die Attraktivität annähernd bei 1 und daher positiv.

Viele Nutzer meinten, dass die Boards im Gang eine höhere Aufmerksamkeit erzeugten. Zum einen, da die Tablets von selbst leuchten und zum anderen, da sie animierte Gifs anzeigen konnten. Deswegen ist das Ergebnis in der Klasse Stimulation ungefähr 1 ebenfalls positiv.

Die Idee, digitale Türschilder zu benutzen schienen die Tester als originell zu empfinden, da es im Testergebnis den höchsten Durchschnittswert erreichte. Die Werte für die Klassen Durchschaubarkeit, Effizienz und Steuerbarkeit sind annähernd neutral. Dies liegt möglicherweise daran, dass für einige Benutzer manche Funktionen zu kompliziert waren und zu viele Klicks erforderten. Einige Funktionen waren zudem nicht auf den ersten Blick ersichtlich und die Genauigkeit des Touch-Interfaces für manche Nutzer zu ungenau.

Kapitel 6

Fazit

Die Idee, interaktive digitale Türschilder zu nutzen, wurde sehr gut aufgenommen.

Die größten Vorteile waren, laut Aussage der Testnutzer, die Möglichkeiten animierten Inhalt und digitale persönliche Zeichnungen darzustellen sowie diese Darstellung aus der Ferne anzupassen.

Leider stellte sich heraus, dass die Dauer der Studie zu kurz war.

Die Nutzer hatten nicht genügend Zeit, um sich vollständig an das System zu gewöhnen. Für manche Testnutzer bot es zudem zu viele Funktionen und war an manchen Stellen zu kompliziert.

Als Schlussfolgerung müssen einige Änderungen am System durchgeführt werden. Am wichtigsten ist, dass das Frontend vom Benutzer-Backend getrennt wird. Dadurch brauchen die Benutzer nur noch ein Passwort, da die Backend-Schnittstelle auf den Boards wegfällt.

Außerdem sollen auf der Frontend-Sicht alle im Raum angemeldeten Benutzer in einer Übersicht auf einmal sichtbar sein.

Besucher können den Besitzern auch weiterhin Nachrichten hinterlassen, nur muss der Prozess vereinfacht werden und die Autoren sollten sich in gewisser Form kennzeichnen können.

Die Angabe eines Status war zu kompliziert und wird vereinfacht. Außerdem soll es vordefinierte Statustexte geben. Da die Abwesend-Markierung fast gar nicht benutzt wurde, muss sie komplett überarbeitet oder entfernt werden.

Die Icons des Editors müssen an die gängigen Standards von Zeicheneditoren angepasst werden.

Des weiteren soll es eine Poweruser Einstellung geben, mit der fortgeschrittene Funktionen angeboten werden können. Für die normalen Benutzer werden nur die Basisfunktionen angeboten.

Ferner könnten für Poweruser noch zusätzliche Funktionen hinzugefügt werden.

Es könnten Benutzer in Gruppen, wie beispielsweise alle Mitarbeiter einer Professur, eingeteilt werden, womit Gruppenbenachrichtigungen oder Gruppenvorlagen möglich wären.

Besuchernachrichten sollten als Bild direkt in der Email mit verschickt werden. Damit müssen die Nutzer nicht extra auf einen Link klicken und sehen den Inhalt direkt.

In den Interviews mit den Testnutzern gab es einige Erweiterungsvorschläge. Es sollte andere Nachrichtenoptionen geben, wie zum Beispiel Audio-, Video- oder Fotonachrichten.

Zudem gab es den Vorschlag, im Editor geometrische Formen oder Smileys anzubieten.

Für die Statusmeldungen hätten einige Nutzer gern eine Kalenderschnittstelle, mit der sie ihre Google-Kalender auslesen lassen können.

Nachdem die Änderungen am System vorgenommen worden sind, müssen diese in einer weiteren Studie getestet werden. Der Test sollte mehr Testnutzer und eine längere Laufzeit haben, damit sich die Testnutzer besser an das System gewöhnen und es in ihren alltäglichen Arbeitsprozess aufnehmen können.

Mit Bauhausboards wurde ein neues Projekt von interaktiven Türschildern speziell für die Anwendung an der Bauhaus Universität erstellt, welches einen neuen Kanal zur Kommunikation zwischen Besuchern und Besitzern sowie die Möglichkeit zur Präsentation von digitalen Informationen schuf.

Literaturverzeichnis

- [1] Erroll Wood and Peter Robinson. Netboards: Investigating a collection of personal noticeboard displays in the workplace. *ACM ITS 2014*, pages 177–183, November 2014. 1, 2, 2.2, 2.2, 2.3, 4.1.1
- [2] NetBoards. <https://github.com/errollw/smartboards2>. Stand: 04.11.2015. 1, 2.2, 3
- [3] Keith Cheverest and Dan Fitton. Experiences managing and maintaining a collection of interactive office door displays. *Ambient Intelligence*, pages 394–409, januar 2003. 2, 2.1, 2.1, 2.1, 2.1
- [4] Keith Cheverest et al. Exploring the evolution of office door displays. *Public and Situated Displays - Social and Interactional Aspects of Shared Display Technologies*, pages 141–169, 2003. 2.1, 2.1
- [5] Keith Cheverest et al. Exploring the utility of remote messaging and situated office door displays. In *Human-Computer Interaction with Mobile Devices and Services*, Udine, Italy, 2003. Springer.
- [6] Keith Cheverest et al. Exploring bluetooth based mobile phone interaction with the hermes photo display. In *MobileHCI '05 Proceedings of the 7th international conference on Human computer interaction with mobile devices and services*, New York USA, 2005. ACM. 2.1, 2.1
- [7] iButton. <https://www.maximintegrated.com/en/products/comms/ibutton.html>. Stand: 04.11.2015. 1
- [8] Keith Cheverest et al. The design, deployment and evaluation of situated display-based systems to support coordination and community. In *Ubiquitous Display Environments*. SpringerLink, 2012. 2.2
- [9] Raspberry Pi. <https://www.raspberrypi.org>. Stand: 13.11.2015. 2
- [10] David H. Nguyen, Joe Tullio, Tom Drewes, and Elizabeth D. Mynatt. Dynamic door displays. 2.3

- [11] Elaine M. Huan and Elizabeth D. Mynatt. Semi-public displays for small, co-located groups. *CHI '03 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 49–56, 2003. 2.3
- [12] Joseph F. McCarthy, Tony J. Costa, and Edy S. Liangosari. Unicast, outcast & groupcast: Three steps toward ubiquitous, peripheral displays. In *Ubicomp 2001: Ubiquitous Computing*. SpringerLink, 2001. 2.3
- [13] Elizabeth F. Churchill et al. The plasma poster network: Posting multimedia content in public places. *INTERACT 2003, Ninth IFIP TC13 International Conference on Human-Computer*, 2003. 2.3
- [14] Elizabeth F. Churchill et al. Sharing multimedia content with interactive public displays: A case study. In *DIS '04 Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 7–16, Camebridge, MA, USA, 2004. 2.3
- [15] JSON. <http://www.json.org>. Stand: 29.11.2015. 1
- [16] Node.js. <https://nodejs.org>. Stand: 24.11.2015. 3

Glossar

Android

Android ist ein Betriebssystem von Google für mobile Geräte

Apache-Webserver

Ein Apache-Webserver ist ein Serverprogramm zum Ausliefern von Webseiten

Apple iOS

Apple iOS ist ein Betriebssystem von Apple für mobile Geräte

Backend

Backend ist der Teil einer Webseite, den Benutzer erst nach Anmeldung sehen können

Bluetooth

Bluetooth ist ein Protokoll zur kabellosen Übertragung von Daten über kurze Distanzen

Entity Relationship Diagramm

Ein Entity Relationship Diagramm ist ein Diagramm zur Modellierung und Strukturierung von Beziehungen zwischen Datenbankentitäten

Framework

Ein Framework (dt.: Rahmenstruktur) ist ein Programmgerüst zur Bereitstellung von Rahmenfunktionen einer Anwendung

Frontend

Frontend ist der Teil einer Webseite, den Benutzer direkt sehen können

Header

Ein Header ist eine Zeile, die in der oberen Region einer Webseite zu finden ist

HTML

HTML ist eine Auszeichnungssprache zur Darstellung digitaler Dokumente

Java

Java ist eine objektorientierte Programmiersprache

Javascript

Javascript ist eine Programmiersprache zur Erweiterung von HTML und CSS

MMS

MMS (Multimedia Messaging Service) ist ein Dienst zum Austausch von multimedialen Nachrichten

Open Source

Open Source ist ein Entwicklungsmodell zur kostenfreien Lizenzierung und freien Verfügbarkeit des Programmcodes

Post-It

Post-It ist ein Markenname für Haftnotizen

Sidebar

Eine Sidebar ist ein seitliches Menü zur Bereitstellung von Funktionen

SMS

SMS (Short Message Service) ist ein Dienst zum Austausch von Textnachrichten

SMS Gateway

Ein SMS Gateway ist eine Schnittstelle zum Senden und Empfangen von SMS-Textnachrichten

Swipe

Swipe ist eine Geste zum Interagieren mit Touch-Oberflächen, bei der Nutzer mit ihren Fingern auf der Oberfläche in eine Richtung wischen

Tablet PC

Ein Tablet PC ist ein flacher kompakter Computer mit Touchscreen

Touchscreen

Ein Touchscreen ist ein Bildschirm mit sensibler Oberfläche zur Eingabe von Benutzeraktionen

UI

Ein UI (User Interface) ist ein Benutzerinterface zur Interaktion zwischen Benutzern und System

Anhang

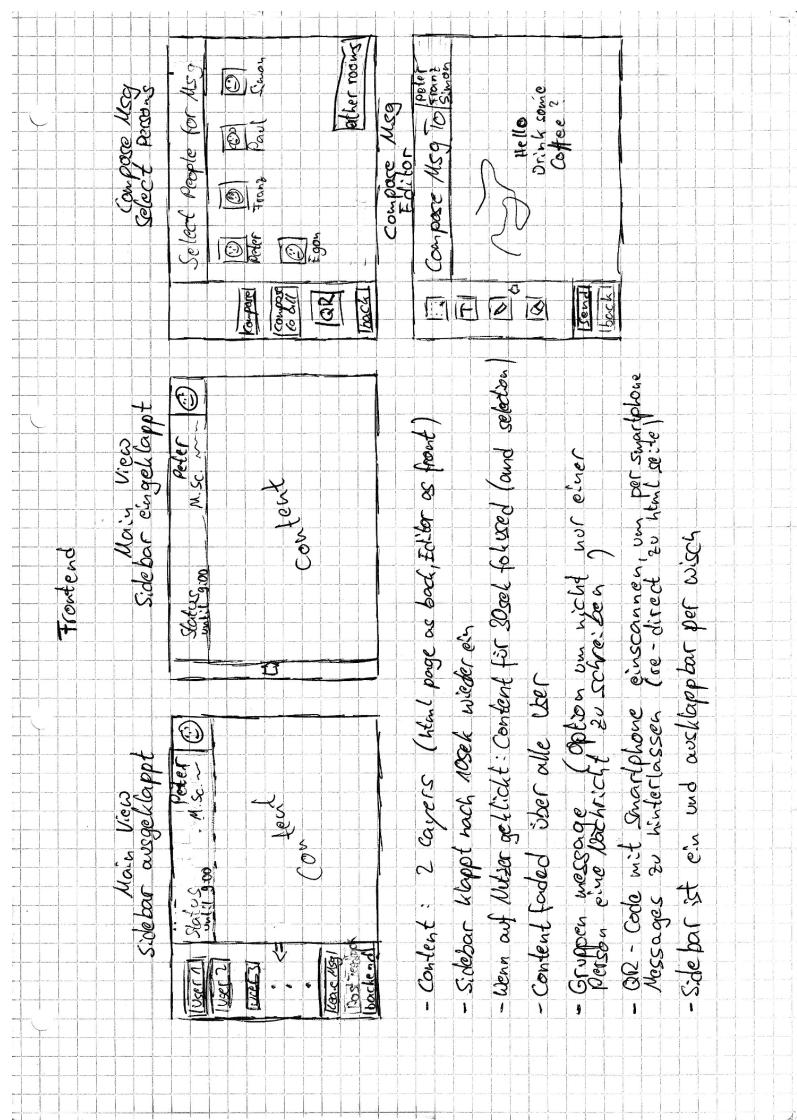


Abbildung 1: Paper Prototype - Frontend

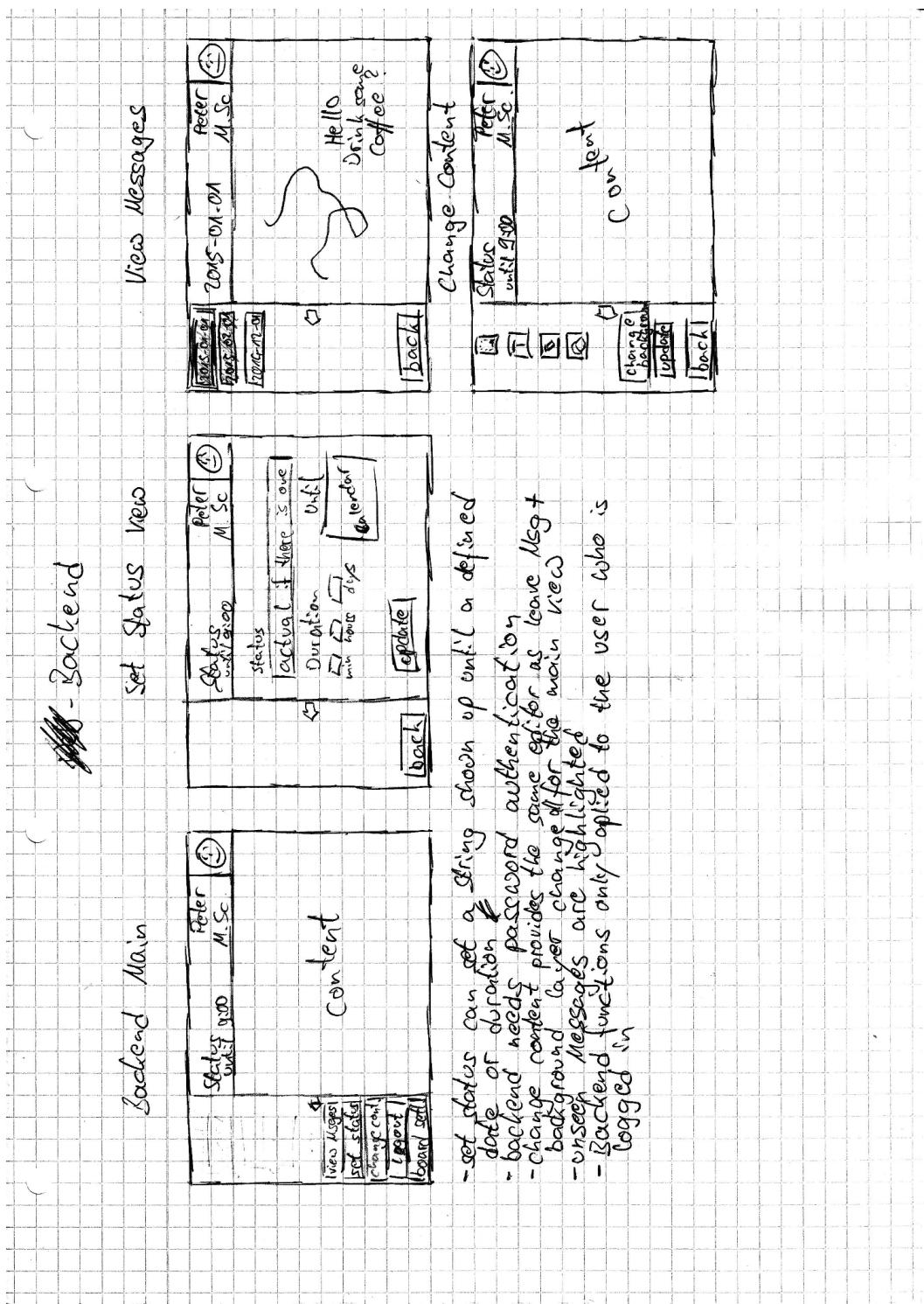


Abbildung 2: Paper Prototype - User Backend

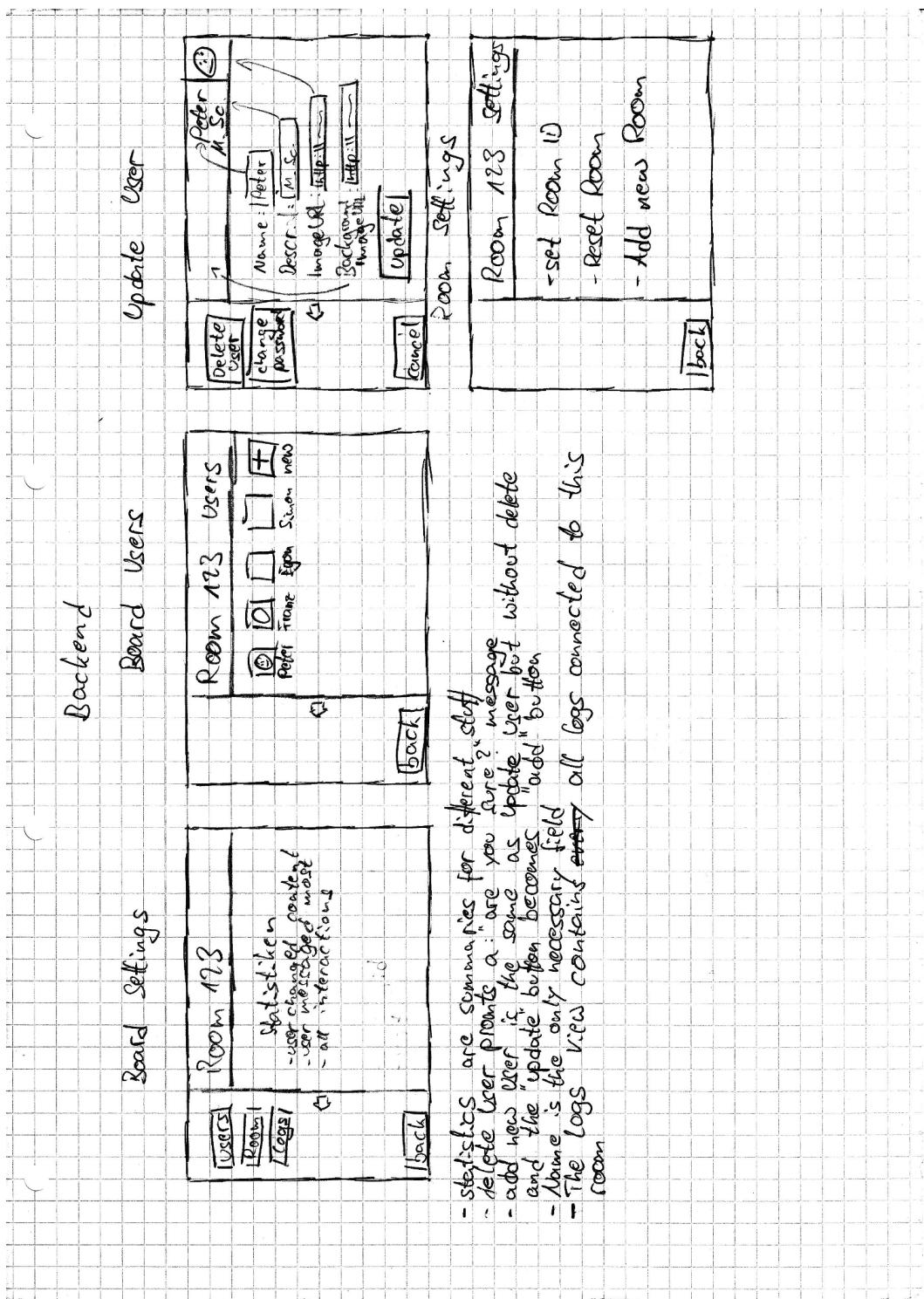


Abbildung 3: Paper Prototype - Admin Backend

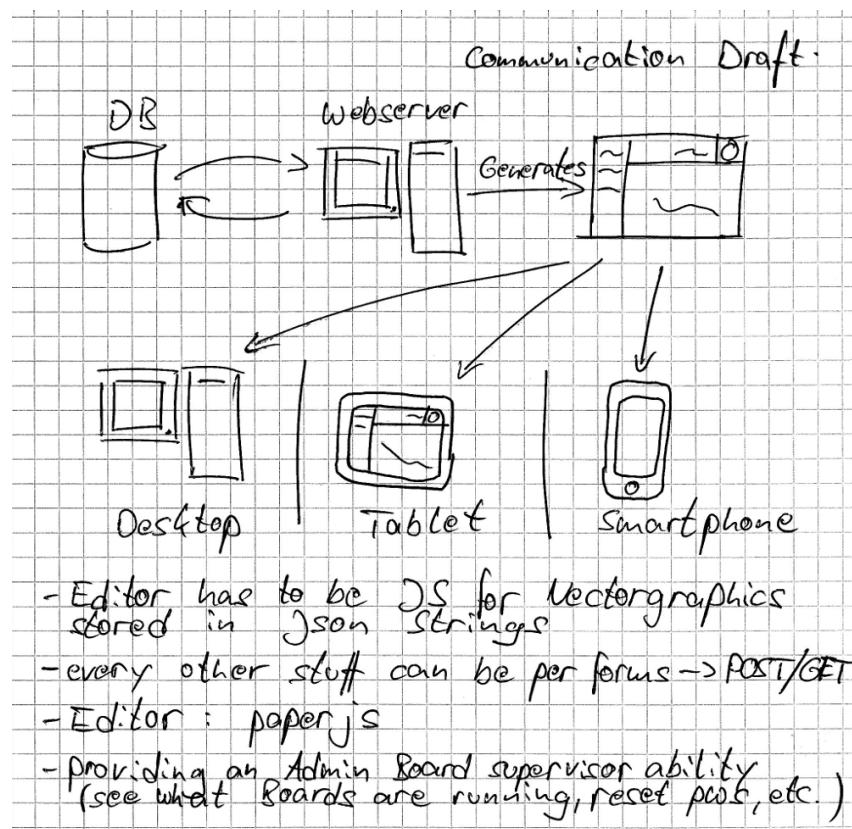


Abbildung 4: Entwurf - Systemaufbau

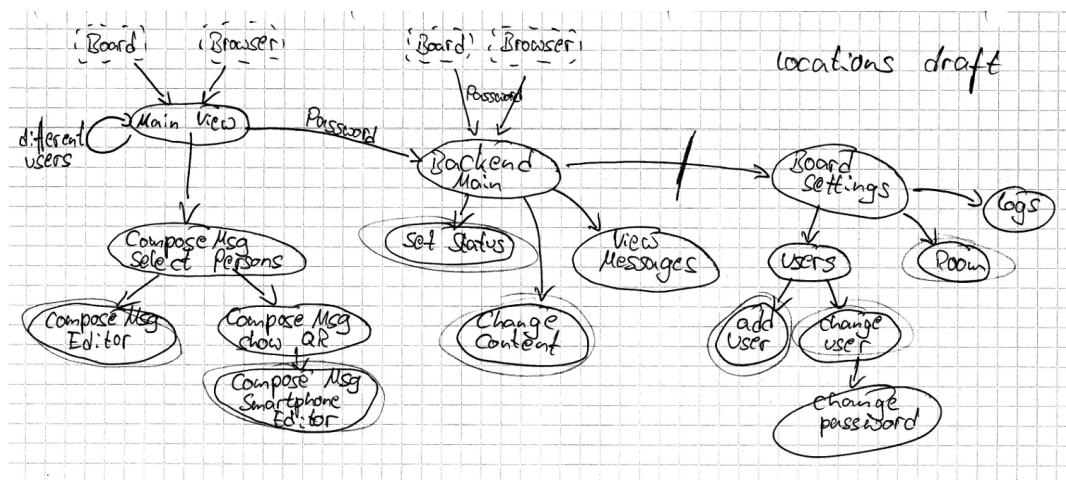


Abbildung 5: Entwurf - Funktionen

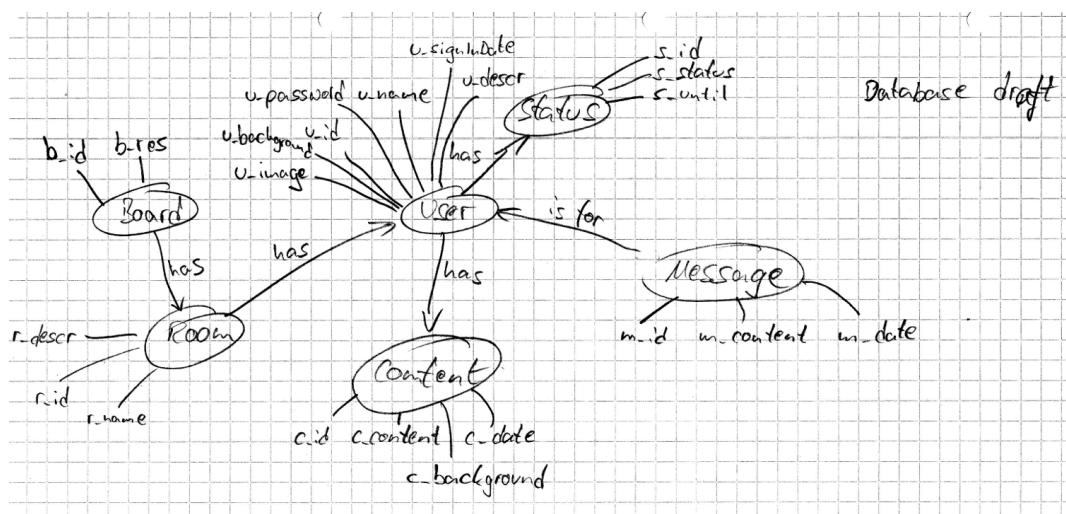


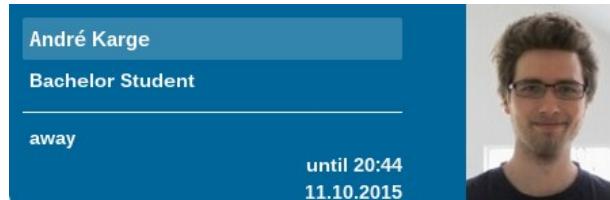
Abbildung 6: Entwurf - Datenbank

Bauhausboards - Usage

Main View

The main view is used to present personal data like images, gifs, drawings, text, a html background page and the user status. It keeps switching between all users assigned to the room. On the left side is a swipeable sidebar to manually switch between the users or to use a board function.

The Message function is to create a message for users in the room. The Feedback function is to write some feedback for the usability of the board or feature requests. The User Settings function is a backend for the users assigned to the room.



On the top right there is the header area. It contains the user name, the user description, the user image and if set the user status.



On the bottom right the latest twitter tweet of the user is displayed

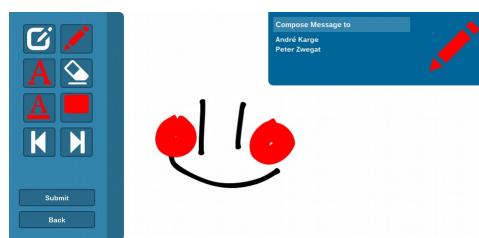


Messages

If the Message function is clicked a popup shows up to select the users to write a message for. It is possible to select or deselect the users by clicking on their images or to compose a message to all users.



Then the Editor shows up and the users can draw a sketch or write a message. If the message is finished it only has finally to be submitted. The receivers will get a notification mail that somebody just wrote a message for them.



Editor

There are 8 main functions for the editor (the selected tool is shown in the header):

1. Selector

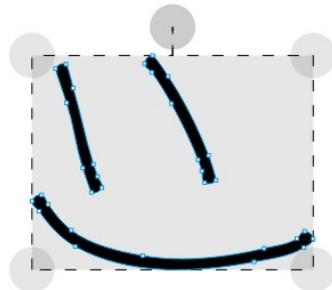


The Selector is to select objects in the editor screen. If an object is clicked it will be selected. If the user makes a swipe gesture a selection area is being made and all objects intersecting this area will be selected after the user stops swiping.

If objects are selected a boundary for all objects and a popup shows up.

The circles on the corners of the bounding box are to scale and the circle on the top edge to rotate the selected objects.

In the popup the trash bin is to remove the selected items, the arrows are to layer them up or down and the last button is to make a copy of the selection.



2. Pen



The Pen is to draw dots or lines. The lines got the current selected color and current selected stroke width.

3. Text



The Text tool is to write a text on the board. If this tool is selected the user has to click on the board where the text should appear and then an input window shows up to write the text which then shows up on the board.

4. Eraser

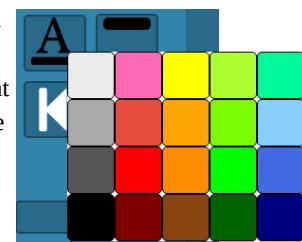


This tool is just an eraser. If you swipe over the board while this tool is selected all objects which were hit will be removed.

5. Color Picker



If the user pushes on this tool a popup with all available colors will show up. If the color is changed while a selection is present then the color of all selected objects will be changed to the new color.



To indicate which color is selected the pen icon, the text icon, the color selector icon and the stroke selector will be drawn in this color.

6. Stroke Weight Picker



This tool is to select the stroke width of the pen tool. The button indicates which stroke width is selected.



7. Undo and Redo



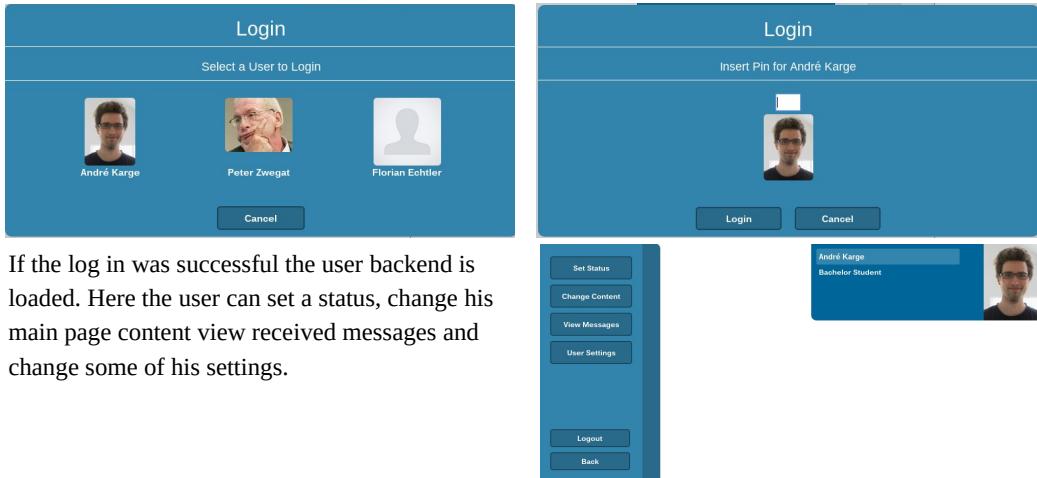
The left button is to undo a interaction and the right one is to redo it.

8. Image Drag & Drop

When a user is using the editor on a desktop pc he is able to drag and drop an image, gif or an image url into the editor. The image then is uploaded and included in the editor.

User Backend

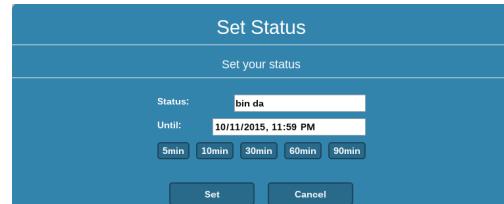
If the User Settings button is pushed a popup shows up to indicate which user wants to log in. By pushing on the user image this user gets selected and a pin input prompt shows up. Then the user has to enter his 4 digit long pin to authenticate himself.



If the log in was successful the user backend is loaded. Here the user can set a status, change his main page content view received messages and change some of his settings.

Status

If the Set Status button is pushed a popup to enter a status appears. The user can enter a short status text and a time until the status is active.



Change Content

If the Change Content button is pushed an editor with the current main view content is loaded. The user now can change this content as he likes. When the user is finished with his changes he has to upload the changes by pushing the change button in the sidebar.

Some users want to present a website as a background layer (e.g. a background image, a site to present the results of a exam or a blog). This is possible by clicking the background button in the sidebar and enter the url in the input mask.



View Messages

On this site all messages for the user are loaded into the sidebar. It is possible to navigate through all of them by clicking the arrow buttons in the sidebar. If a message was not yet seen its button is displayed with a red border.

When the user clicks on a message its content is drawn on the board. The header shows the date when the message was created and a button to mark the message as unseen.



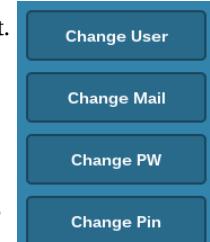
User Settings

To change the user settings the user has to enter his password in the popup prompt.

On Change User the user can change his name, description, twitter account and profile picture url.

On Change Mail he can change his mail address.

On Change PW he can change his password and on Change Pin he can change his pin number.



Fragestellung BauhausBoards (Besitzer)

1. Nachrichten empfangen

(die Nachrichten der Nutzer durchgehen)

- a) Worum ging es in den Nachrichten?
- b) Wer hat die Nachricht geschrieben, falls bekannt?
- c) Gab es viele anonyme Nachrichten, die nicht einem bestimmten Gast zugewiesen werden können?
- d) Wie haben Gäste ihre Identität in Nachrichten gekennzeichnet?
- e) Wie hätte man dem Nutzer, außer Bauhausboards, eine Nachricht hinterlassen können?
- f) War es anders, über das Board eine Nachricht zu erhalten? (besser/schlechter/egal)
- g) Wie lang hat es durchschnittlich gedauert, bis der Nutzer eine Nachricht gelesen hat?
- h) Hat der Nutzer die Funktion "View Messages" im Backend genutzt oder eher nur die Nachricht per Mail eingesehen?
- i) Was war gut/schlecht am Nachrichtensystem?
- j) Was könnte man am Nachrichtensystem noch verbessern?

2. Nachrichten schreiben

- a) Hat der Nutzer einem anderen Nutzer eine Nachricht hinterlassen? (Nachricht im Backend suchen)
- b) Worum ging es in der Nachricht?
- c) Ist bekannt, ob der Empfänger die Nachricht gelesen hat?
- d) Wäre eine bidirektionale Kommunikation besser? (Empfänger hat Möglichkeit zu antworten)

3. Status

- a) Wurde das Feature überhaupt genutzt? (Wenn nein – warum nicht?)
- b) Inwiefern wurde der Status geändert?
- c) Wurde der Status am Board, am Desktop oder am Smartphone geändert?
- d) Wurde die "Not Available" Funktion genutzt?
- e) Wurde die N/A Funktion in Kombination mit einem Statustext verwendet?
- f) Was waren die üblichen Statustexte?

- g) Warum wurde der Status nicht geändert? (vergessen?, keine Lust?, zu aufwändig?)
- h) Wann wurde der Status üblicherweise geändert?
- i) Würde man Status häufiger verwenden, wenn es vordefinierte Statustexte geben würde?
- j) Was war gut/schlecht am Statussystem?
- k) Was würde man am Statussystem noch verbessern?

4. Content

(Die Contents der Nutzer durchgehen)

- a) Wurde das Feature überhaupt genutzt?
- b) Warum / Warum nicht?
- c) Inwiefern wurde der Content geändert? (warum geändert / warum nicht)
- d) Was wurde angezeigt (Für welche Informationen wurde es genutzt)
- e) Wann wurde der Content üblicherweise geändert?
- f) Gab es Leute, die auf den Content eingegangen sind? (etwa in einer Nachricht oder persönlich?)
- g) Wurde das Background-Feature genutzt?
- h) Wenn ja – wofür? Wenn nein – warum nicht?
- i) Wurde für den Background extra eine Seite geschrieben?
- j) Pro/Contra/Verbesserungen am Background-Feature
- k) Wurde das Twitter-Feature genutzt?
- l) Wenn nein – warum nicht?
- m) Wenn ja – für den eigenen Twitteraccount oder einem anderen?
- n) Pro/Contra/Verbesserungen am Twitter-Feature
- o) Was war gut/schlecht am Contentsystem?
- p) Was würde man am Contentsystem noch verbessern?

5. Editor

- a) Wurden eigene Sketche gezeichnet?
- b) Was wurde gezeichnet?
- c) Wurden Bilder/Gifs eingefügt?
- d) Pro/Contra/Verbesserungen am Editor

6. Board Allgemein

- a) Backend genutzt? Pro/Contra
- b) Meinung zum Pin als schnelle Authentisierung
- c) Meinung zum Passwort für Backend Authentisierung
- d) Vorteile/Nachteile zu Post-Its/Whiteboards/Pinnwänden
- e) Boardgröße: zu groß/zu klein?
- f) Könnte sich der Nutzer vorstellen regulär Bauhausboards zu benutzen?
- g) Allgemeine Verbesserungsvorschläge