

Projektdokumentation

ENTWICKLUNGSPROJEKT INTERAKTIVE SYSTEME

ausgearbeitet von

Dilara Güven

vorgelegt an der

TECHNISCHE HOCHSCHULE KÖLN
CAMPUS GUMMERSBACH
FAKULTÄT FÜR INFORMATIK UND INGENIEURWISSENSCHAFTEN

im Studiengang

MEDIENINFORMATIK

Prof. Dr. Kristian Fischer
Prof. Dr. Gerhard Hartmann

Betreut von: Corinna Klein
 Sheree Saßmannshausen

Gummersbach, 16.12.2018

Inhaltsverzeichnis:

1.	EINLEITUNG.....	5
2.	ZIELHIERARCHIE	6
2.1	Ziel 1.....	6
2.2	Ziel 2.....	7
2.3	Ziel 3.....	7
2.4	Ziel 4.....	8
2.5	Ziel 5.....	8
2.6	Ziel 6.....	8
3.	PROOF OF CONCEPT - ÜBERARBEITET	9
3.1	Aufenthaltort korrekt prüfen.....	9
3.2	Sensoren prüfen und ansteuern können.....	9
3.3	Push-Benachrichtigungen an den User senden.....	10
3.4	Energieverbrauch berechnen	10
4.	RISIKEN UND MAßNAHMEN	11
4.1	Fehler bei der Ortsbestimmung.....	11
4.2	Unerreichbarer User	11
4.3	Störung bei Stromversorgung	11
5.	ARCHITEKTURDIAGRAMM.....	13
5.1	Server	13
5.2	Client	14
5.3	Datenhaltung	14
5.4	Sensoren.....	14
5.5	Kommunikation	14
6.	ANWENDUNGSLOGIK	16
6.1	Serverseitig	16
6.2	Clientseitig.....	16

7.	ANFORDERUNGEN	17
7.1	Identifizierung von Nutzungskontext.....	17
7.1.1	Private Haushalte	17
7.1.2	Firmen.....	17
7.1.3	Gastronomiebetriebe	17
7.2	Analyse der Anforderungen	18
7.2.1	Funktionale Anforderungen	18
7.2.2	Organisatorische Anforderungen	19
7.2.3	Qualitative Anforderungen	19
7.3	Priorisierung der Anforderungsanalyse.....	20
7.4	Fazit.....	21
8.	VORGEHENSMODELL – METHODISCHER MCI RAHMEN.....	22
8.1	ISO DIN 9241	22
8.2	User-Centered Design	22
8.2.1	Scenario Based Usability Engineering	23
8.2.2	Usability Engineering Lifecycle	23
8.3	Usage-Centered Design.....	23
8.4	Fazit Vorgehensmodell	24
9.	ESSENTIELLE MODELLE	25
9.1	Role Model	25
9.1.1	User Roles.....	25
9.1.2	User Role map	26
9.2	Task Model	28
9.2.1	Essential Use Case	28
9.2.2	Use Case Map	29
9.2.3	Concrete Use Case.....	30
9.3	Content Model.....	32
9.3.1	Content Model	33
9.3.2	Navigation Map	35
10.	IMPLEMENTIERUNGSMODELL – UI PROTOTYP	38
10.1	Fazit.....	43
11.	EVALUATION	44
11.1	Heuristische Evaluation	44
11.2	Fazit.....	46

12. WBA-MODELLIERUNG	47
13. DATENSTRUKTUR.....	50
14. ALLEINSTELLUNGSMERKMAL – ÜBERARBEITET	51
14.1 Push-Benachrichtigungen	51
14.2 GPS.....	51
14.3 Steuerung	51
14.4 Stromnetz	52
14.5 Selbstkonfiguration	52
14.6 Energieverbrauch	52
15. FAZIT PROJEKT	53
16. PROJEKTPLAN	54
16.1 Meilenstein 2	54
16.2 Meilenstein 3	55
17. LITERATURVERZEICHNIS	56

1. Einleitung

In der Projektdokumentation werden die im Konzept beschriebenen Aspekte vertieft und geplant, was implementiert werden muss. Die Projektdokumentation enthält bzw. umfasst alle Artefakte, die zur Entwicklung und Analyse des Systems hinsichtlich der Modelle und Methoden des Moduls Mensch Computer Interaktion und Web-basierte Anwendungen 2, benötigt werden.

Das zu entwickelnde System bietet die Möglichkeit von den Gefahren, die im Haus durch Geräte ausgelöst werden können, zu entkommen. Sobald eine Gefahr erkannt wird, soll das System den Client warnen und ihm entgegenkommen. Zudem wird auch dem Client der aufmerksame Umgang mit Haushaltsgeräten mitgeteilt, indem der Energieverbrauch der jeweiligen Haushaltsgeräte berechnet wird. Des Weiteren wird das zu entwickelnde System näher analysiert und bearbeitet.

2. Zielhierarchie

Strategische Ziele

Die strategische Ebene ist sehr abstrakt formuliert, d.h. nur der Zielpunkt wird genannt und nicht der Weg zum Ziel. Das System soll langfristig den betroffenen Stakeholdern/ Nutzern Sicherheit, Kontrolle und Überblick verschaffen.

Taktische Ziele

Bei der taktischen Ebene geht es um die Wege und Methoden, wie man das Ziel erreicht. Zuerst sollten die Erwartungen der Stakeholder berücksichtigt werden. Das System sollte Fehlermeldungen vermeiden und keine zusätzlichen Probleme bereiten, die durch die Nutzung der Applikation durch Dritte entstehen könnten. Den Nutzern sollte die Steuerung der Haushaltsgeräte einfach und schnell wie möglich gemacht werden, sodass kein Risiko eingegangen wird. Zudem sollte das zu entwickelnde System responsive-designed sein, um den User einen besseren Einblick zu verschaffen, sodass er die Steuerung der Geräte ohne Probleme durchführen kann.

Operative Ziele

Operative Ebene ist die Handlungsebene, d.h. es werden konkrete Handlungen/ Aktionen beschrieben, die die Zielsetzung der strategischen- und taktischen Ebene erreichen. Es sollten also die Maßnahmen bestimmt werden, die kurzfristig dafür benötigt werden, das Hauptziel zu erreichen. Dazu gehört das Festlegen eines Vorgehensmodells und die technischen Anforderungen des Systems. Es muss eine GPS-Abfrage entwickelt werden, die erkennt, wo sich der User befindet und wie weit die Distanz von zu Hause ist. Es müssen alle elektronischen Haushaltsgeräte aufgelistet sein, die zum Steuern bereitgestellt werden. Eigenschaften, die die Sensoren besitzen (wie z.B. der Energieverbrauch oder der aktuelle Status – ON/OFF), müssen definiert und klar ersichtlich für den Nutzer dargestellt werden. Daneben muss es noch eine Kommunikationsform ausgesucht und implementiert werden, die die wichtigsten Informationen an den User übermittelt. Diese Kommunikationsform soll vor zusätzlichem Aufwand schützen und die Steuerung in simpler Art und Weise ermöglichen.

Nachdem die Ziele aus dem Drei-Stufen-Schema nach Leontyev, 1978 beschrieben wurden, wird auf diese näher ins Detail eingegangen. Durch die im Konzept festgelegten Alleinstellungsmerkmale lassen sich Ziele ableiten, die in strategische, taktische und operative unterteilt werden.

2.1 Ziel 1

Strategisch:

Es soll der Aufenthaltsort richtig geprüft und ermittelt werden.

Taktisch:

Jugendliche sollen immer ab Verlassen des Hauses, ihren Standort freigeben vom mobilen Endgerät aus, sodass der Standort korrekt ermittelt werden kann.

Operativ:

Es muss eine GPS-Abfrage entwickelt werden, die erkennt, wo sich der User befindet und wie weit die Distanz von zu Hause ist.

2.2 Ziel 2

Strategisch:

Es sollen alle Haushaltsgeräte korrekt angezeigt werden, die nicht ausgeschaltet wurden.

Taktisch:

Nach dem Verbindungsaufbau soll der Nutzer die Haushaltsgeräte untereinander aufgelistet sehen können.

Operativ:

Das System überprüft welche Haushaltsgeräte registriert und nicht ausgeschaltet worden sind. Diese werden in der App aufgelistet.

2.3 Ziel 3

Strategisch:

Die Sensoren sollen geprüft und angesteuert werden können.

Taktisch:

Der Benutzer muss die Möglichkeit haben auf die im System registrierten Haushaltsgeräte zuzugreifen und zu steuern.

Operativ:

Die Sensoren werden auf dem Raspberry Pi registriert und von dem System aufgenommen. Es findet somit eine Selbstkonfiguration statt und werden immer aktuell gehalten. Mit „aktuell gehalten“ ist das Prüfen der Sensoren gemeint, d.h. es wird nach dem Status (ON/OFF) geprüft und die Möglichkeit gegeben umzuschalten.

2.4 Ziel 4

Strategisch:

Der Nutzer soll Push-Benachrichtigungen vom Server erhalten können. Es dürfen keine doppelten Benachrichtigungen empfangen werden. Die Benachrichtigung enthält eine Uhrzeit.

Taktisch:

Das System soll Push-Benachrichtigungen an mobile Endgeräte versenden, sodass der Nutzer über den aktuellen Stand jederzeit informiert wird.

Operativ:

Die Benachrichtigungsfunktion wird ermöglicht, sobald vom mobilen Endgerät die Ortsfreigabe aktiviert ist. Erst nach dem Überprüfen der Distanz des Aufenthaltsortes, können Push-Benachrichtigungen empfangen werden.

2.5 Ziel 5

Strategisch:

Es soll bei Notfällen alle Hausbeteiligte vor Gefahren informiert und gewarnt werden.

Taktisch:

Der Benutzer soll die Möglichkeit haben anderen Menschen (Familie/Freunde) einen Zugriff auf das System zu erlauben.

Operativ:

Der Benutzer bzw. Besitzer des Systems darf auch Zugriff entnehmen und neu vergeben.

2.6 Ziel 6

Strategisch:

Es soll der Energieverbrauch berechnet werden.

Taktisch:

Der Nutzer muss die Möglichkeit haben den Energieverbrauch von allen Geräten zu sehen.

Operativ:

Der Benutzer kann entscheiden, ob er den Energieverbrauch eines Gerätes einsehen möchte.

3. Proof of Concept - überarbeitet

3.1 Aufenthaltsort korrekt prüfen

Beschreibung: Um das System vollständig auszuführen und auf die Haushaltsgeräte zugreifen zu können, muss zuerst der Standort überprüft werden.

Exit: Der Aufenthaltsort des Stakeholders wird korrekt erfasst und im System aufgenommen.

Fail: Der ermittelte Standort wurde falsch ermittelt und das System stellt ggf. falsche Informationen dar.

Fallback: Diese Grundlage des Systems ist unverzichtbar. Diese Funktion war geplant mit einem GPS-Sensor durchzuführen. Jedoch wäre diese viel zu zeitaufwändig und als Alternative werden die GPS-Daten manuell eingegeben und überprüft.

3.2 Sensoren prüfen und ansteuern können

Beschreibung: Nachdem der Standort korrekt ermittelt wurde, soll auf das System zugegriffen werden können.

Exit: Die Geräte werden untereinander aufgelistet und zusätzlich wird der Status dessen angezeigt, d.h. ob das Gerät an oder aus ist. Der Prozess wird erfolgreich abgeschlossen, wenn das ausgewählte Gerät umgeschaltet worden ist.

Fail: Die Statusanzeige ist fehlerhaft bzw. reagiert nicht.

Fallback: Diese Grundlage des Systems ist unverzichtbar und ggf. findet eine erneute Abfrage an den Server statt.

3.3 Push-Benachrichtigungen an den User senden

Beschreibung: Nach der korrekten Standortabfrage schickt der Server Push-Benachrichtigungen bezüglich des Status der Geräte.

Exit: Fehlerfreie Standortermittlung und Ankunft der Push-Benachrichtigung

Fail: Inkorrekte Standortabfrage oder fehlerhafte Kommunikation zwischen Server und Client. Dadurch können auch keine Push-Benachrichtigung empfangen werden.

Fallback: Wiederholter Kommunikationsaufbau zwischen Server und Client. Push-Benachrichtigungen dürfen nicht mehrfach empfangen werden, sondern Redundanzen berücksichtigen.

3.4 Energieverbrauch berechnen

Beschreibung: Nach Bedarf soll auch der Energieverbrauch angezeigt werden.

Exit: Einwandfreier und realistischer Energieverbrauch wird angezeigt, nachdem der Algorithmus dafür richtig durchgeführt wurde.

Fail: Der Algorithmus der Berechnung ist fehlerhaft.

Fallback: Diese Grundlage des Systems ist unverzichtbar ggf. findet eine erneute Abfrage an den Server statt.

4. Risiken und Maßnahmen

Aus den festgelegten Alleinstellungsmerkmalen und Zielen werden die Risiken hergeleitet. Im Folgenden werden die Risiken identifiziert und entsprechende Maßnahmen geplant.

4.1 Fehler bei der Ortsbestimmung

Die Ortsbestimmung, die die GPS-Daten vom Client ermittelt, ist eine vom Alleinstellungsmerkmal wichtigste Funktion der App. Aufgrund der falschen oder nicht genauen Lokalisierung könnte es dazu kommen, dass der Aufenthaltsort nicht erkannt wird. Wenn die GPS-Abfrage nicht richtig durchgeführt wird, besteht das Risiko, dass falsche Informationen an den Nutzer übermittelt werden. Die GPS-Abfrage kann sowohl über Sensoren als auch über Spieldaten durchgeführt werden. Aufgrund der Komplexität wird die Abfrage mit Spieldaten erfolgen. Somit würden die Spieldaten als Maßnahme gelten.

4.2 Unerreichbarer User

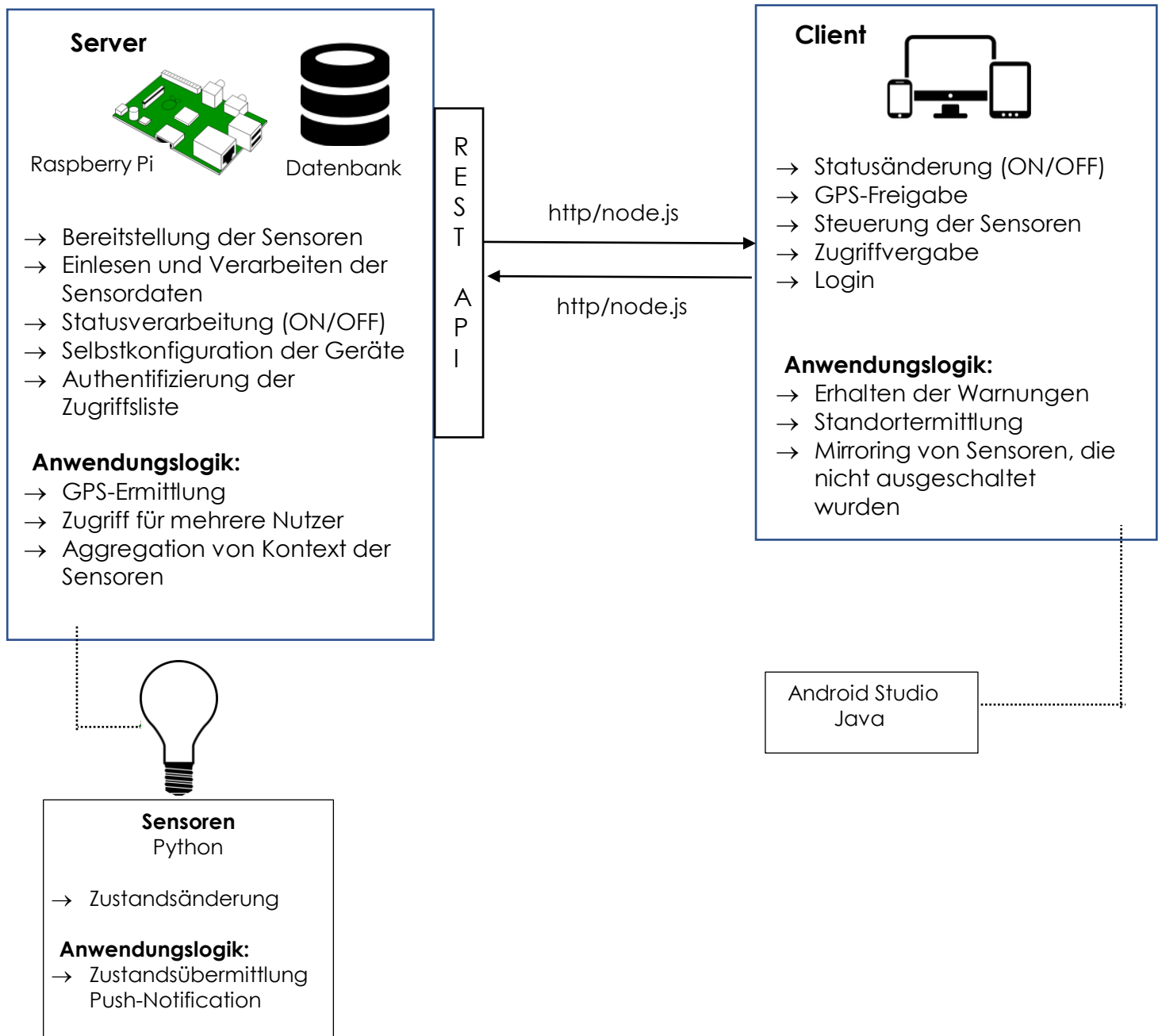
Die zu entwickelnde Anwendung bezieht sich hauptsächlich auf Jugendliche. Es wird oft unter Beweis gestellt (siehe Konzept), dass die meist Smartphone Nutzer die Jugendlichen sind. Die unterschiedlichen Nutzungsmöglichkeiten sind im Konzept ebenfalls beschrieben. Eines der Gründe bzw. Möglichkeiten der Nutzung, ist das Gaming und Chat. Jugendliche verbringen ihre meiste Zeit an Unterhaltung. Heutige häufig genutzte Spiele sind zum Beispiel MineCraft oder Fortnite etc. An Social Media sind die häufig genutzten Facebook, Twitter, Instagram und WhatsApp. Diese sind alle Energieverbrauchende Programme und daraus folgt der schnellere Akkuverbrauch des mobilen Endgerätes. Um diesen schnellen Verbrauch zu verhindern gibt es einige Möglichkeiten, wie zum Beispiel das Nutzen des Powerbank's. Als Powerbank bezeichnet man einen Akku mit Ladeanschlüssen für externe Geräte. Es handelt sich also um ein Gerät, welches Energie speichert, um damit weitere Geräte außer Ort aufzuladen. Eine weitere Möglichkeit, die für einige Stunden hilfreich ist, wäre das Aktivieren des Energiesparmodus auf dem mobilen Endgerät.

4.3 Störung bei Stromversorgung

Die meisten Haushalte sind stark vom Strom abhängig. Bei einem Stromausfall zum Beispiel funktionieren keine Geräte mehr im Haus und alle Kommunikationsmittel werden lahmgelegt. Stromausfälle oder Störungen an der Leitung können unterschiedliche Gründe haben. Es kann durchaus ein Stromausfall wegen eines Kurzschlusses geben, wenn ein Gerät unsachgemäß genutzt wurde oder wenn die Steckdose überbelastet wird. Durch Überspannung kann es zu Fehlfunktionen oder sogar zu Beschädigungen von elektrischen Geräten führen und die Stromversorgung beeinträchtigen. Als Maßnahme für Stromschläge hilft in meisten Fällen ein FI-Schutzschalter. Dabei wird die Stromzufuhr unterbrochen, um die Person vor einem Stromschlag zu

schützen. Es wird die Strommenge kontrolliert, die ein Gerät aufnimmt und wieder abgibt. Sobald der Schalter eine gefährliche Situation wahrnimmt, wird die Stromzufuhr unterbrochen. Der Stromausfall kann an dem Sicherungskasten überprüft werden. Auch der Wasserkontakt zu den Steckdosen sollte vermieden werden, da es auch Stromschläge u.Ä. als Auswirkung hat.

5. Architekturdiagramm



(Abbildung 1: Architekturdiagramm überarbeitet)

5.1 Server

Der Raspberry Pi ist die Serverkomponente des zu entwickelnden Systems und stellt die Hauptfunktion dar, die der Nutzer mit dem Endgerät anspricht. Der Pi

enthält alle wichtigen Sensoren der Haushaltsgeräte, die für die Anwendung eine wichtige Rolle spielen. Die Sensoren, die auf dem Pi gespeichert sind, werden mit Python implementiert. Python ist für dieses System eher geeignet, da die Sprache sowohl C als auch Java akzeptiert und kombinieren kann.

Nach einer GPS-Ermittlung zum mobilen Endgerät, wird eine Benachrichtigung gesendet. Die GPS-Abfrage findet manuell statt, d.h. es wurden für das Projekt Spieldaten angewandt. Nachdem diese Benachrichtigung wahrgenommen wurde, kann der Raspberry Pi reagieren und die Steuerung vom mobilen Endgerät verarbeiten. Somit ist der Miniserver eine Kommunikationsstelle und stellt genauso die Anwendungslogik dar.

Das bedeutet, die Distanz des Nutzers wird erfasst und per Push-Benachrichtigung werden Warnungen herausgeschickt, sodass der Nutzer darauf reagieren kann. Der Server an sich wird mit JavaScript implementiert und über Node.js auf die Sensoren in Python zugegriffen.

5.2 Client

Die mobile Anwendung soll in Java implementiert und mit Android App entwickelt werden. Die Kommunikation zwischen dem Miniserver und dem mobilen Endgerät findet synchron statt. Es können mehrere Endgerät-Nutzer auf den Server zugreifen. Das Front-End wird in Java implementiert für die Android App. Der Client kann Zugriff vergeben und löschen. Diese werden vom System gespeichert. Jeder neue Client muss sich über das Login-Verfahren registrieren um Teil des Systems zu werden. Für das Login wird eine Datenbank programmiert.

5.3 Datenhaltung

Es wird eine Datenbank programmiert, die eine Zugriffsliste auf das System repräsentiert. Um auf das System zugreifen zu können, muss die Person eine Berechtigung erhalten von einem Client aus und über das Login Verfahren sich anmelden. Ansonsten werden keine anderen Daten abgespeichert, da es sich nur um eine persistente Zustandsmitteilung der Sensoren handelt. Die Abspeicherung der Geräteabfrage würde zu viele Ressourcen fressen und das System erschweren.

5.4 Sensoren

Die Sensoren haben die Aufgabe, bei Zustandsänderung den Server zu informieren per Push-Notification. Die Sensorabfrage über den Server würde zu viel Aufwand bereiten und zu viel Zeit in Anspruch nehmen.

5.5 Kommunikation

Um den Server zu generieren und mit der Datenbank zu kommunizieren, bietet sich die serverseitige Plattform Node.js. Diese basiert auf JavaScript und bietet sehr detaillierte Funktionen an. Um einen Server zu erstellen, wird das Express Web-Framework verwendet. Via HTTP Nachrichten werden Instruktionen an den Server geschickt, die dort verarbeitet werden und eine Zustandsänderung

der Sensoren bewirken. Unter Umständen kommt ein Response in Form einer Push-Notification zurück.

6. Anwendungslogik

6.1 Serverseitig

Der Server bekommt von jedem Client den aktuellen Aufenthaltsort, bevor die Sensordaten geprüft werden. Die Anzahl der Clients ist unbegrenzt und je nach Bedarf kann diese Freigabe an das System verteilt werden, d.h. es besteht ein Zugriffsrecht erst wenn der Hauptuser es erlaubt. Das Anlegen mehrerer Clients wäre besonders für Familien oder Unternehmen und Restaurants geeignet (siehe Konzept „Stakeholder“). Ein weiterer wichtiger Punkt der Anwendungslogik des Servers ist das automatische Senden einer Benachrichtigung an den Client. Die Benachrichtigungen sollen den Nutzer vor Risiken warnen und darauf hinweisen, diese über das Gerät anzusteuern. Zudem ist der Server dazu fähig, den Zustand der Sensoren festzuhalten. Wenn der Sensor den Zustand von OFF zu ON oder andersrum ändert, wird eine Push-Notification an den Server versendet. Diese Zustände werden aggregiert und an den Client mitgeteilt.

6.2 Clientseitig

Das mobile Endgerät ist dazu fähig die Push-Benachrichtigungen automatisch zu erhalten, nachdem die Ortungsfreigabe aktiviert wurde. Somit kann auch der Aufenthaltsort ermittelt werden und der Nutzer wird vom System aufgenommen. Sobald alles durchgeführt wird, kann die App starten und es erscheint eine Liste von den Haushaltsgeräten. Aus dieser Liste ist zu entnehmen, welche Haushaltsgeräte nicht ausgeschaltet wurden im Haus.

7. Anforderungen

Um die Anforderungen ermitteln zu können, wurden im Konzept bestimmte Stakeholder und der Nutzungskontext identifiziert. Anforderungen sind Voraussetzungen, die zur Lösung einer Realisierung eines bestimmten Ziels gebraucht werden. Sie beziehen sich auf das System (=Lösungsraum). Allerdings ohne lösungsspezifische oder technologische Details bzw. Festlegungen. Der Grund dafür ist, dass wenn man technologische Festlegungen vornimmt, muss man prüfen, ob sich die validen Anforderungen noch realisieren lassen. Dadurch werden bei der Anforderungsanalyse die funktionalen Anforderungen (Systemaktivität und Funktionen), organisatorischen (Prozess und Kontrolle) und qualitativen Anforderungen (Leistung, Sicherheit und Standards) definiert.

7.1 Identifizierung von Nutzungskontext

Zur Identifizierung des Nutzungskontextes wurde sich an den im Konzept ermittelten Stakeholdern orientiert und diese nach ihren Anforderungen untersucht.

7.1.1 Private Haushalte

- Haushaltsgeräte steuern
- Berechtigte Person hat Zugriff auf das System
- Andere Benutzer einen Zugriff für das System vergeben können
- Einfach und schnelle Bedienung sowie Zielführung
- Einfaches Design
- Können jederzeit Status steuern und prüfen
- Haben Energieverbrauch im Überblick

7.1.2 Firmen

- Einfache und schnelle Bedienung der Anwendung
- Stellt Firmenzugriff bereit
- Haben Energieverbrauch im Überblick
- Können jederzeit Status steuern und prüfen

7.1.3 Gastronomiebetriebe

- Einfache und schnelle Bedienung der Anwendung
- Haben Energieverbrauch im Überblick
- Können jederzeit Status steuern und prüfen

7.2 Analyse der Anforderungen

Ausgehend von den Stakeholdern werden folgende Anforderungen in Benutzerinteraktionen und selbstständige Systemaktivitäten ermittelt und untergliedert:

7.2.1 Funktionale Anforderungen

- **F10 Aufenthaltsort Ermittlung**
Das System muss jeden Standort der Clients abrufen und ermitteln können.
- **F20 Abfrage der Haushaltsgeräte**
Das System muss fähig sein die Sensoren fehlerfrei zu überprüfen.
- **F30 Push-Benachrichtigung senden**
Das System muss die Nutzer über den Status der Haushaltsgeräte (ON) benachrichtigen, um eine Gefahr zu verhindern.
- **F40 Übersichtliche Darstellung der Auflistung der Geräte**
Das System sollte dem Nutzer einen übersichtlichen Einblick der Haushaltsgeräte verschaffen, sodass er in seinem bestimmten Nutzungskontext seine bestimmten Aufgaben (Haushaltsgeräte ausschalten) effektiv, effizient und zufriedenstellend erledigt.
- **F50 Zugriff mehrerer Nutzer**
Das System sollte die Möglichkeit bieten, mehrere zugehörige Nutzer aufzunehmen oder zu löschen.
- **F60 Energieverbrauch**
Das System muss allen Nutzern den Energieverbrauch der jeweiligen Haushaltsgeräte berechnen und darstellen können.

7.2.2 Organisatorische Anforderungen

- **O10 Datenschutz/-sicherheit**

Das System darf die Informationen über die Haushaltsgeräte nur verschlüsselt darstellen, so dass dritte Personen diese nicht einsehen können. Zudem soll das System die Daten zuverlässig und fehlerfrei verarbeiten.

- **O20 Geringe Akkuleistung**

Das System muss die Akkuleistung während der Nutzung der Anwendung gering halten, damit die Möglichkeit besteht, den Nutzer zu erreichen.

- **O30 Erreichbarkeit des mobilen Endgerätes**

Das System muss jederzeit erreichbar sein, da der Nutzer bei Komplikationen ansprechbar sein muss.

- **O40 Bedienung**

Das System sollte einfach und schnell zu bedienen sein. Außerdem sollte der Bedienprozess dynamisch gestaltet werden, damit die Nutzer flexibel bleiben können.

7.2.3 Qualitative Anforderungen

- **Q10 Darstellung**

Das System muss die angegebenen Informationen wie beispielsweise Art des Geräts, Energieverbrauch und Status (ON/OFF) dem Nutzer verständlich und überschaubar darstellen.

- **Q20 Anspruchslose Bedienung**

Das System muss über eine deutliche und leichte Benutzeroberfläche verfügen und dadurch einfach zu bedienen sein.

- **Q30 schnelle Zielerreichung**

Das Ziel ist es die Gefahren vor einem Brand im Haushalt zu verhindern. Das System muss dem Nutzer ermöglichen, dass er nur wenig Zeit zur Zielerreichung benötigt.

- **Q40 Systemstabilität**

Das System darf nicht die Möglichkeit haben, bei einem aktuellen Vorgang abzustürzen oder diese abubrechen.

- **Q50 Status Aktualisieren**

Das System muss jederzeit den Status der Haushaltsgeräte aktuell halten, um auch die anderen Nutzer zu informieren, dass eine Erneuerung vorgenommen wurde.

7.3 Priorisierung der Anforderungsanalyse

Die festgelegten Anforderungen des Systems müssen nach Wichtigkeit in Bezug auf das Projekt projiziert werden, damit leichter Entscheidungen bezüglich des Projektes getroffen werden können.

(Priorisierung: 5 sehr wichtig – 1 weniger wichtig)

Anforderungen	Priorität
F10 Aufenthaltsort Ermittlung	5
F20 Abfrage der Haushaltsgeräte	5
F30 Push-Benachrichtigung senden	5
F40 Übersichtliche Darstellung der Auflistung der Geräte	4
F50 Zugriff mehrerer Nutzer	5
F60 Energieverbrauch	5
O10 Datenschutz/-sicherheit	5
O20 Geringe Akkuleistung	3
O30 Erreichbarkeit des mobilen Endgerätes	5
O40 Bedienung	2
Q10 Darstellung	4
Q20 Anspruchslose Bedienung	4
Q30 schnelle Zielerreichung	5
Q40 Systemstabilität	4
Q50 Status Aktualisieren	4

(Abbildung 2: Prioritätstabelle)

7.4 Fazit

Auf der Basis der Anforderungsanalyse und dessen Prioritätensetzung hat sich ergeben, dass der Schwerpunkt auf den funktionalen Anforderungen liegt. Die Nutzer sollen zu jedem Zeitpunkt in Bezug auf den Standort erreicht werden, um dann die Sensoren Abfrage zu durchführen und diese dem Nutzer mitzuteilen. Zudem sollte auch ein Zugriff für mehrere Nutzer ermöglicht werden. Auch der Energieverbrauch sollte berechnet und dargestellt werden. Aspekte wie Datenschutz und Systemstabilität sollten auch beachtet werden.

8. Vorgehensmodell – Methodischer MCI Rahmen

In diesem Kapitel wird die Art und Weise menschlicher Interaktion und Kommunikation diskutiert, damit ein gebrauchstaugliches System entwickelt werden kann. Der methodische Rahmen erschließt sich aus dem Nutzungskontext bzw. dem Ziel der Applikation. Ergebnisse der Domänen- und der überarbeiteten Marktrecherche werden zusätzlich berücksichtigt.

Als Mensch-Computer-Interaktion verstehen wir ein interdisziplinäres, sowohl grundlagen- als auch anwendungsorientiertes Lehr- und Forschungsgebiet der Informatik. Der Fokus liegt hierbei auf der menschlichen Perspektive im Kontext der Nutzung interaktiver Systeme. Welches Vorgehensmodell das richtige für das zu entwickelnde System ist, wird im Folgenden diskutiert und beschrieben.

8.1 ISO DIN 9241

Die ISO DIN 9241 beschäftigt sich mit Anforderungen und beschreibt ergonomische Grundsätze. Diese können bei Leistungsbeschreibung, Gestaltung und Bewertung von Dialogsystemen angewandt werden. Die ISO Norm beschäftigt sich also mit Qualitätsaspekten und beinhaltet wichtige 7 Grundsätze der Dialoggestaltung.

Die ISO Norm ist sozusagen eine Richtlinie, an die sich die Entwickler einer Software richten können. Auch für die „Don't Forget“ App ist dies eine wichtige Methode, um die Anwendung gebrauchstauglicher und konsistenter zu machen. Zudem wird auch eine höhere Produktivität ermöglicht. Die Grundsätze sind wichtig, um den Nutzungskontext der Don't Forget App richtig zu ermitteln, die Anforderungen zu spezifizieren, Gestaltungslösungen durch die Empfehlungen zu entwickeln und diese zu evaluieren.

8.2 User-Centered Design

Der Begriff User-Centered Design wurde von Norman und Draper geprägt. Hierbei dienen die Merkmale der Benutzer, sowie die Aufgaben und Ziele, Nutzungskontext, Wissenstand, Fähig- und Fertigkeiten als sinnvoller Ausgangspunkt für die Konzeption und Entwicklung des interaktiven Systems. Bei diesem Modell handelt es sich nicht um eine konkrete Methode oder Arbeitstechnik der User, sondern eher um eine Grundhaltung in Bezug auf den Entwicklungsprozess für Artefakte. Diese Grundhaltung äußert sich darin, dass die User in alle Stufen des Entwicklungsprozesses einbezogen werden. Dies hat verschiedene Gründe wie beispielsweise die ungleiche Perspektive der Entwickler mit denen der Nutzer, das heißt der Nutzer muss befragt werden. Zudem hängt die wahrgenommene Gesamtqualität von Usability und der Produktqualität ab. Man erfährt also wie Nutzer handeln und wie man sie darin unterstützen kann.

8.2.1 Scenario Based Usability Engineering

Das Vorgehensmodell wurde von Rosson und Carrol 2002 vorgelegt. Der Fokus liegt hierbei auf dem Menschen und nicht auf dem System, d.h. es liegt eine benutzerzentrierte Perspektive vor. Es werden zuerst Szenarien erstellt, bevor man mit der Modellierung des Systems beginnt. Der Vorteil daran ist, dass es zuerst die Bedürfnisse und Merkmale der Benutzer erfasst werden. Es geht hierbei um ein iteratives Modell, das verschiedene Aktivitäten im Entwicklungsprozess spezifiziert, wie zum Beispiel die Anforderungsanalyse, das Design und die Entwicklung von Prototypen und deren Evaluation.

Auch dieses Vorgehensmodell eignet sich gut für das zu entwickelnde System, da im Rahmen dieses Projektes, sowie in WBA2 und anderen Modulen Problemszenarien erstellt und behandelt wurden.

8.2.2 Usability Engineering Lifecycle

Das von Deborah Mayhew erfundene Modell beschreibt wesentliche Prozess-Stufen und Aktivitäten im Zusammenhang mit der Herstellung von Gebrauchstauglichkeit für interaktive Systeme. Das Vorgehensmodell ist iterativ und skalierbar und besteht aus drei Bestandteilen wie die Anforderungsanalyse, das Design und die Installation. Dieses Modell ist so ausgelegt, dass Iterationen in allen qualitätsrelevanten Aktivitäten vorgesehen sind und es skalierbar auf die Komplexität des Projektes und der zu lösenden Entwicklungsaufgaben sind. Somit wird die Bedeutung von Usability und User Interfaces hervorgehoben. Dazu gehören behandelnde Themen bzw. Methoden wie User Profile, Style Guides, Analysis, Contextual Task und User Feedback.

Die Don't Forget Anwendung bzw. das User Interface wird mit Android Studio entwickelt. Android Studio gibt einiges schon vor was User Interface betrifft, wie zum Beispiel die größtenteils vorgefertigten Elemente wie Buttons etc. Um das User Interface zu verbessern, würde es viel Zeit in Anspruch nehmen, was im Rahmen dieses Projektes nicht zu verwirklichen wäre. Daher wird nicht weiter auf dieses Modell eingegangen.

8.3 Usage-Centered Design

Lockwood und Constantine schufen 1996 den Begriff usage-centered design und stellten damit den Verwendungszweck eines Systems in den Vordergrund. Die Idee dahinter war, gute Systeme und gute Werkzeuge für ihren Zweck zu entwerfen und zu erfüllen. Als Ausgangspunkt können die Merkmale der Benutzung für die Modellierung des interaktiven Systems gewählt werden. Somit wird klargestellt, dass die Aufmerksamkeit auf die Benutzung gerichtet ist und nicht nur auf dem Benutzer. Das Vorgehensmodell ist ein modellgetriebener Prozess und für einen erfolgreichen Entwurf sind mindestens drei Modelle notwendig wie zum Beispiel das: Role Model, Task Model und das Content Model.

Das User Role Model modelliert die Benutzer, das System und ihre Beziehung untereinander.

Beim Task Model werden Maßnahmen in strukturierter Form beschrieben, welche die Benutzer zur Erledigung ihrer Aufgabe ergreifen. Es werden sozusagen essential use cases erstellt, mit denen ermittelt wird, was der Benutzer beim System erreichen will. Das Task Modell steht im Zentrum des Diagramms aufgrund seiner zentralen Position im usage-centered design.

Das Content Model beinhaltet Artefakte, die das User Interface den Benutzern zur Verfügung stellt. Das bedeutet, es wird der Inhalt repräsentiert. Dabei hat der Inhalt und das User Interface einen höheren Wert als die Erscheinung und das Verhalten.

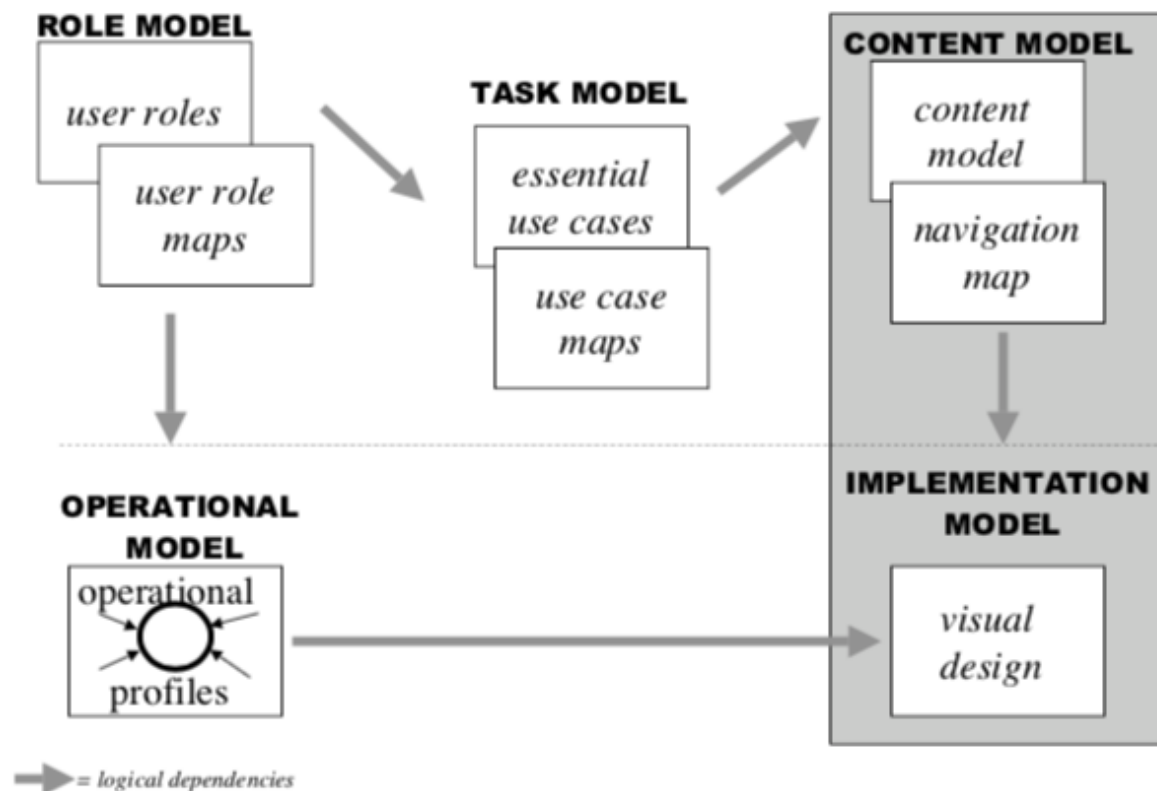
In diesem Modell werden die verschiedenen Benutzer analysiert und erfasst. Daraufhin kann der Fokus auf die Benutzung des zu entwickelnden Systems gerichtet werden.

8.4 Fazit Vorgehensmodell

Schlussfolgernd eignet sich das Usage-Centered Design, entwickelt von L. Constantine und L. Lockwood, weil der Verwendungszweck eines Systems in den Vordergrund gestellt wird und nicht der Nutzer. Im Folgenden werden die zum Usage-Centered design gehörigen Modelle, wie das Rollenmodell (role model), das Aufgabenmodell (task model), Inhaltsmodell (content model) und Implementationsmodell (implementation model) behandelt. Dieses Modell eignet sich besser, denn die Funktionalität, Sicherheit und Zuverlässigkeit des Systems ist die Hauptaufgabe der Anwendung an den User.

9. Essentielle Modelle

Essentielle Modelle sollen den Entwicklern helfen, den Verwendungszweck der Software besser zu verstehen. Daher ist es ein modellgetriebener Prozess, der sich darauf konzentriert, die Arbeit und die Ziele der Arbeit der Benutzer zu verstehen. Größtenteils beschäftigen essentielle Modelle sich mit der Frage „Was will der Benutzer warum erreichen?“. Für einen erfolgreichen Entwurf sind mindestens drei Modelle notwendig, nämlich das role model, task model und content model, die im Folgenden behandelt werden.



(Abbildung 3: logischen Beziehungen essentielle Modelle)

9.1 Role Model

Im role model sollen die verschiedenen Arten der Benutzer und ihre Rollen erfasst und verstanden werden, sowie ihre Beziehung untereinander. Das role model mit dem task model bilden ein Modell der zu lösenden Probleme ab.

9.1.1 User Roles

Zunächst werden die einzelnen User Roles nach den Aspekten: Bedürfnisse, Interesse, Erwartung, Verhalten, Verantwortlichkeiten in Brainstorming sessions untersucht und die wichtigsten Aspekte für die Beschreibung der User Roles gewählt.

- **Allgemeiner App-Nutzer**

Allgemeiner Nutzer bedeutet, dass die App regelmäßig und täglich in Benutzung ist. Das System soll schnell und unkompliziert sein. Der Nutzer ist an schneller und flexibler Zielerreichung interessiert, also zum Beispiel bei der Suche nach dem Energieverbrauch oder An- und Ausschalten der Haushaltsgeräte. Eine spezielle Anforderung wäre die Gewährleistung der Sicherheit des Systems vom ermittelten Standort aus. Zudem ist der allgemeine App Nutzer, derjenige der mit dem System vertraut ist und besitzt eventuell auch ein hohes technisches Verständnis.

- **Unregelmäßiger App-Nutzer**

Der unregelmäßige App-Nutzer ist jemand, der die Anwendung nur nutzt, wenn er sie auch braucht, d.h. wenn er Push-Benachrichtigungen erhält und gefordert wird die Haushaltsgeräte auszuschalten. Der aktuelle Energieverbrauch hat für den Nutzer keinen großen Wert. Das System soll übersichtlich und unkompliziert sein. Man könnte davon ausgehen, dass der unregelmäßige Nutzer ein geringeres technisches Verständnis hat, da er das System nur bei Bedarf nutzt und wenig Wert auf die Einzelheiten legt.

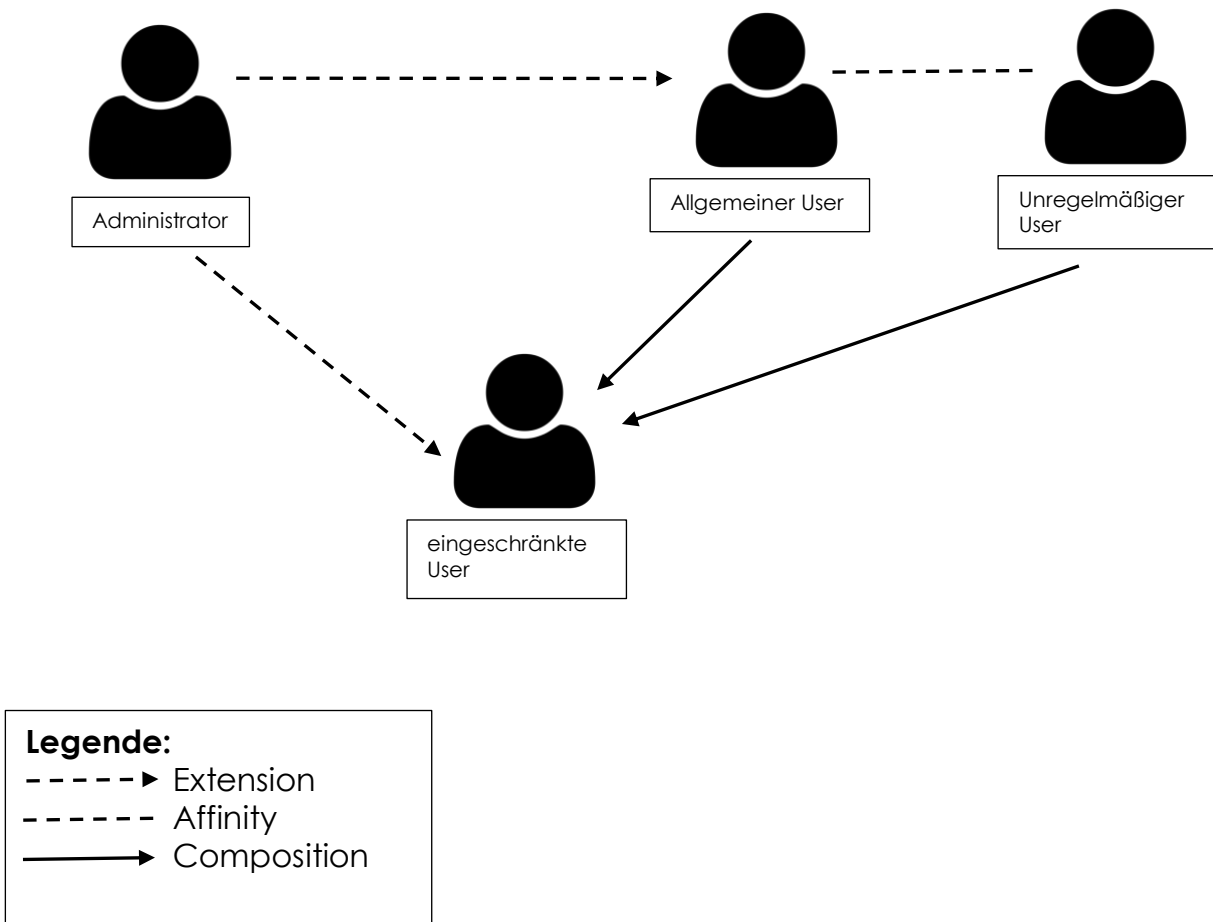
- **Eingeschränkte (vergessliche) Menschen**

Die App wird regelmäßig benutzt, da die Person mit einem geminderten Wahrnehmungsvermögen (bestimmter Vergesslichkeitsgrad), daran erinnert wird mit dem System zu interagieren. Die Erinnerung findet statt, sobald er das Haus verlässt und die Geräte überprüft wurden. Durch das häufige Nutzen der App, ist er mit dem System auch vertrauter als eine Person, die es unregelmäßig nutzt. Das System soll unkompliziert und übersichtlich sein, um die Gebrauchstauglichkeit zu unterstützen.

- **Administrator**

Die Entwickler haben ebenfalls die Möglichkeit die App zu nutzen und zählen auch zu den Benutzern. Ihre Anforderungen an das System ist die fehlerfreie Funktion, sowie der schnellen Diagnose bei Problemen und dessen Lösung. Auch Konflikte mit anderen Nutzern soll vermieden werden. Für Verbesserungsvorschläge oder Kritik kann eine Feedback Funktion mit eingeführt werden. Falls es eine Firma gibt, soll auch Verantwortung gegenüber der Firma und Mitarbeiter getragen werden.

9.1.2 User Role map



(Abbildung 4: User Role map)

Ein Admin kann sowohl ein allgemeiner User als auch eingeschränkter User sein. Allgemeiner User und unregelmäßiger User sind sich ähnlich, daher verwandt. Auch diese Nutzerrollen können sich erweitern zu eingeschränkten Usern. Die User Rollen sind sich in einigen Punkten ähnlich, jedoch unterscheiden sich wegen der jeweiligen Anforderungen und Funktionen.

9.2 Task Model

Im Folgenden werden Maßnahmen in strukturierter Form beschrieben, welche die Benutzer zur Erledigung ihrer Aufgaben ergreifen.

9.2.1 Essential Use Case

Ein essentieller Anwendungsfall ist eine strukturierte Erzählung, ausgedrückt in einer Sprache der Anwendungsdomäne und der Benutzer, die eine spezifizierte, verallgemeinerte, abstrakte, technologiefreie und implementierungsunabhängige Beschreibung einer Aufgabe oder Interaktion umfasst. Essential Use Case sorgt dafür, dass die Sicht der Nutzer in Bezug auf dem System mit dem Zweck oder Absicht aussagekräftig definiert werden. Darüber hinaus erfüllt es die Anforderungen der Nutzer, um ein benutzergerechtes System zu liefern und leitet sich daraus ab.

1) Status (ON/OFF) der Haushaltsgeräte suchen

User Intention	System Responsibility
0. Status der Geräte finden	1. Repräsentation der Haushaltsgeräte in einer Liste 2. Statusanzeige

Tabelle 1: **StatusSuchen**

2) Gerät mit Energieverbrauch suchen

User Intention	System Responsibility
0. Energieverbrauch finden	1. bietet eine Repräsentation der Haushaltsgeräte in einer Liste an 2. zeigt eine Übersicht der Geräte mit den Eigenschaften an 3. Eigenschaften zeigen den Energieverbrauch

Tabelle 2: **EnergieverbrauchSuchen**

3) Zugriff der Nutzer einsehen

User Intention	System Responsibility
0. Ich möchte wissen, wer Zugriff auf die Haushaltsgeräte hat	1. Repräsentation auf der Hauptseite 2. Funktion spezifizieren 3. Repräsentation der Zugriffsliste

Tabelle 3: **NutzerZugriff**

4) Zugriff der Nutzer vergeben

User Intention	System Responsibility
0. Ich möchte jemandem einen Zugriff auf die Haushaltsgeräte geben	1. Repräsentation der Hauptseite 2. Überprüfung der Berechtigung des Nutzers, der Zugriff vergibt 3. bietet eine Funktion, das den Zugriff erlaubt 4. Bestätigung der Zugriff Freigabe

Tabelle 4: **NutzerZugriffFreigabe**

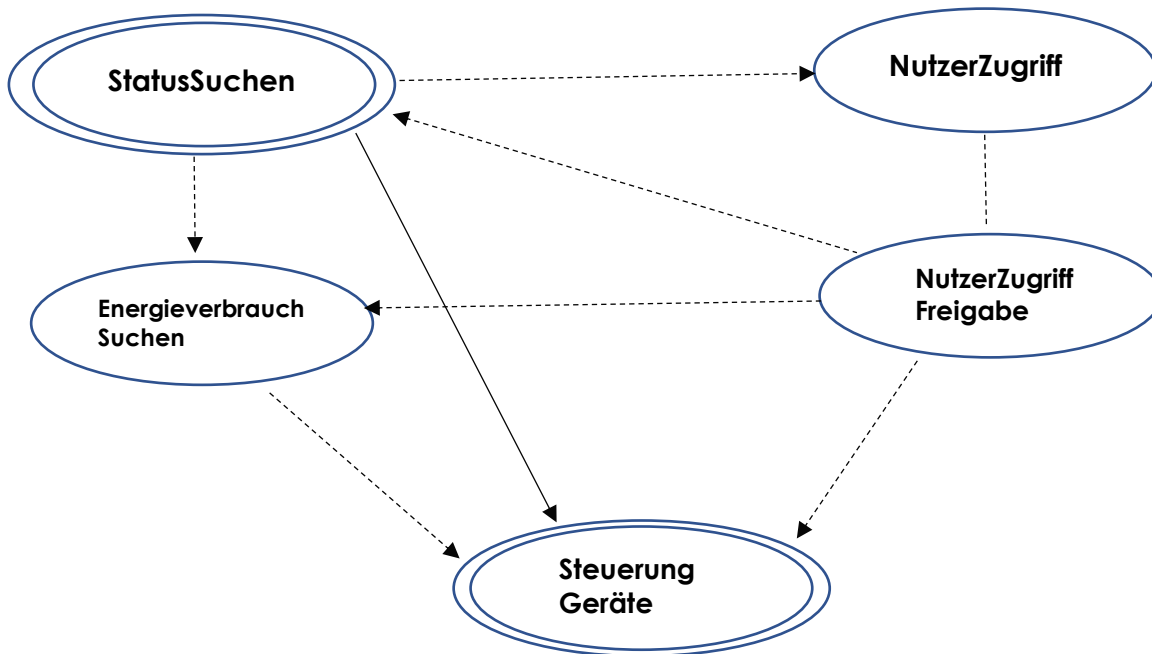
5) Steuerung der Haushaltsgeräte

User Intention	System Responsibility
0. Haushaltsgeräte an-/ausschalten	1. bietet eine Repräsentation der Haushaltsgeräte (mit ON/OFF) in einer Liste an 2. Bestätigung der Statusänderung

Tabelle 5: **SteuerungGeräte**

9.2.2 Use Case Map

Die Use Case Map sorgt dafür, dass die erstellten Essential Use Cases deutlich vereinfacht werden und diesen Sinn für den Benutzer macht. Die Use Case Map leitet sich ab aus den Essential Use Cases.



(Abbildung 5: Use Case Map)

Legende:

- - - - -> Extension
- - - - - Affinity
- > Composition

9.2.3 Concrete Use Case

Das concrete use case ist eine ausführliche Beschreibung einer Aufgabe, die keine unwichtige Aufgabenszenarien personalisiert. Es wird die Verwendung des Systems auf einer allgemeineren Ebene beschrieben. Das concrete use case wird definiert aus Interaktionsstile- und Modi, sowie der technologischen

Ebene. Im Folgenden werden, die im essential use case behandelten Punkte in das concrete use case übertragen.

1) Status (ON/OFF) der Haushaltsgeräte suchen

User Intention	System Responsibility
0. Status der Geräte finden	1. System stellt die Hauptseite dar 2. System fordert Interaktion 3. Darstellung der Haushaltsgeräte in einer Liste

Tabelle 1: **StatusSuchen in concrete use case**

2) Gerät mit Energieverbrauch suchen

User Intention	System Responsibility
0. Energieverbrauch finden	1. bietet eine Darstellung der Haushaltsgeräte in einer Liste an 2. System bietet die Funktion zum Einsehen der Eigenschaften an 3. Eigenschaften zeigen den berechneten Energieverbrauch

Tabelle 2: **EnergieverbrauchSuchen in concrete use case**

3) Zugriff der Nutzer einsehen

User Intention	System Responsibility
----------------	-----------------------

0. Ich möchte wissen, wer Zugriff auf die Haushaltsgeräte hat	1. Darstellung der Hauptseite 2. Das System fordert den Benutzer dazu auf eine Funktion auszuwählen 3. System bietet eine Auflistung der Zugriffsliste
---	--

Tabelle 3: **NutzerZugriff in concrete use case**

4) Zugriff der Nutzer vergeben

User Intention	System Responsibility
0. Ich möchte jemandem einen Zugriff auf die Haushaltsgeräte geben	1. Darstellung der Hauptseite 2. Überprüfung der Berechtigung des Nutzers, der Zugriff vergibt 3. bietet eine Funktion, das den Zugriff erlaubt 4. Bestätigung der Zugriff Freigabe

Tabelle 4: **NutzerZugriffFreigabe in concrete use case**

5) Steuerung der Haushaltsgeräte

User Intention	System Responsibility
0. Haushaltsgeräte an-/ausschalten	1. bietet eine Darstellung der Haushaltsgeräte (mit ON/OFF) in einer Liste an 2. System fordert Benutzer auf Interaktion 2. Bestätigung der Statusänderung

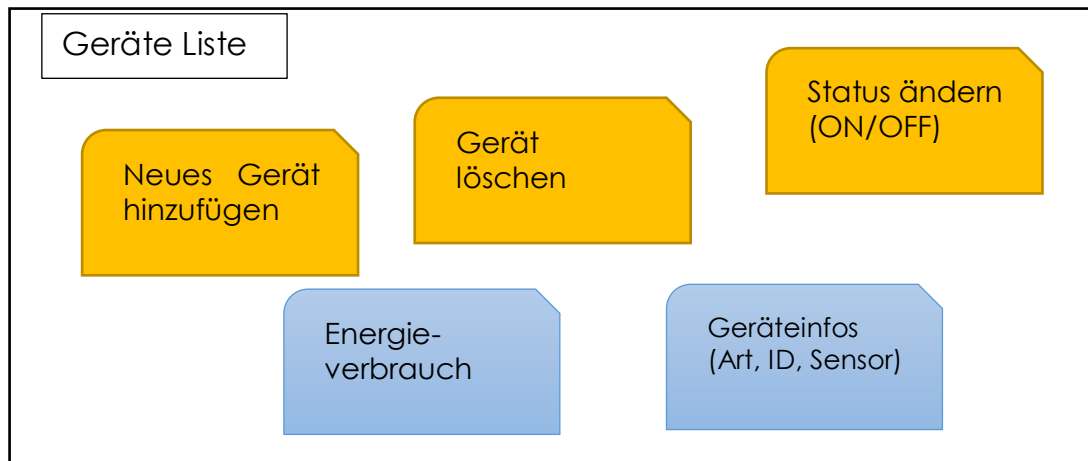
Tabelle 5: **SteuerungGeräte in concrete use case**

9.3 Content Model

Das content model handelt von einer abstrakten Repräsentation der Inhalte der verschiedenen Interaktionsräume. Die Aufgabendurchführung der Benutzer findet in diesen Interaktionsräumen statt. Die einfachste und beste

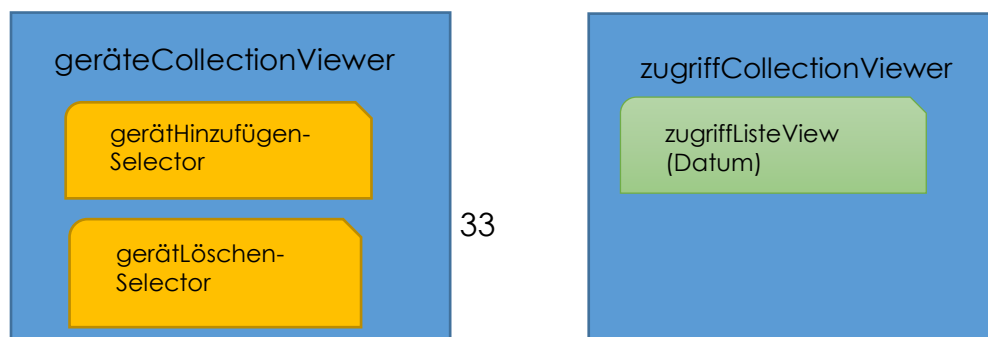
Methode nach Constantine und Lockwood ist es, das Modell mit Hilfe von Papier und post-it Klebzetteln zu realisieren. Das Blatt Papier repräsentiert den Interaktionsraum und die post-it's die Werkzeuge und Materialien. Der Sinn dieser Methode liegt darin, dass es den Designern die bildliche Darstellung der Anwendung außenvorgehalten wird und sie sich besser auf die Funktionen und Materialien konzentrieren sollen. Nachdem dieser Prozess beendet ist, dürfen Entscheidungen über das Design getroffen werden. Im Folgenden werden die Artefakte sinnvoll in Einheiten organisiert und vernetzt.

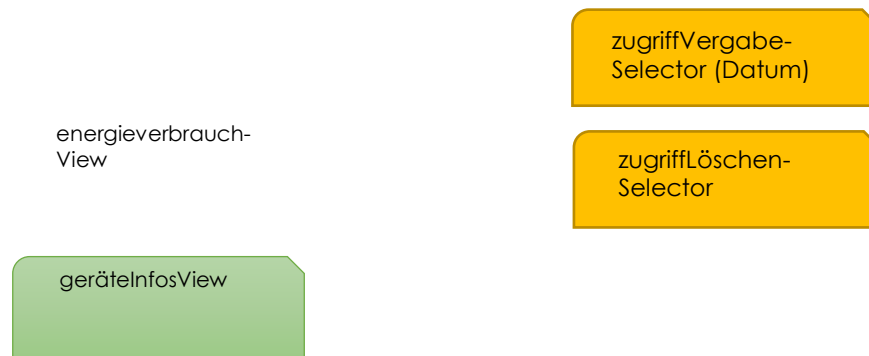
9.3.1 Content Model



(Abbildung 6: content model Geräte Liste)

Als Darstellungsart wurde das "Post-it" Modell nach Lockwood gewählt, da dieses zunächst die genaue Wahl des GUI Widget noch offen lässt und dadurch den Design Prozess vereinfacht und noch flexibel gestaltet. Die kalten Farben, wie **blau und grün** stehen für passive Elemente wie **repräsentativer Inhalt (Views), Informationen, Daten** etc. Warme Farben wie **orange, gelb oder pink** stehen **für funktionale Elemente wie Selector und Controller**. Diese Funktionen (Tools) sind die aktiven, vom Benutzer steuerbaren Elemente. In der oben dargestellten Abbildung werden die wichtigsten Funktionen und die Views einer Geräteliste aufgezeigt. Der Benutzer hat die Möglichkeit ein neues Gerät im System zu konfigurieren und zu löschen, sowie die Steuerung dieser Geräte. Auch das Einsehen von Informationen wie zum Beispiel der Energieverbrauch und die Art des Gerätes ist möglich.





(Abbildung 7: Content model erweitert)

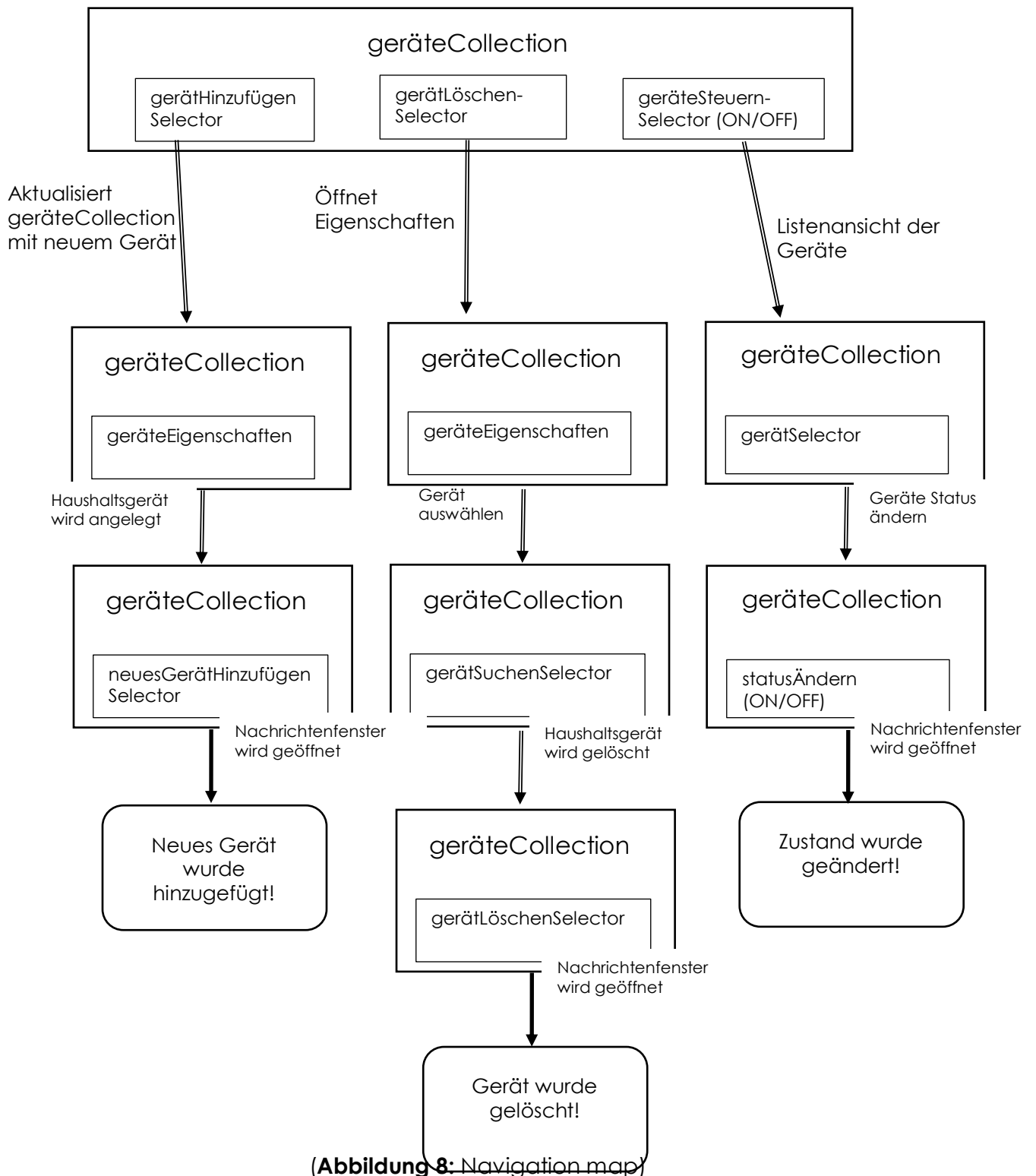
Das erweiterte content model beinhaltet alle Funktionen und Inhalte, dass das System bieten soll. Mit geräteCollectionView sind alle relevanten Funktionen zu betrachten. Der Benutzer muss die Möglichkeit haben ein Gerät hinzufügen, zu löschen und auch diese zu steuern. Dabei können die Informationen der Haushaltsgeräte eingesehen werden. Zusätzlich wird auch der Energieverbrauch zur Auswahl gestellt. Unter dem Menüpunkt zugriffEinstellungSelector kann Zugriff für andere Nutzer vergeben oder entnommen werden. Die Auswahl dieser Funktion führt zu zugriffCollectionView. Dort ist eine Liste zu finden, indem alle Benutzer mit dem Datum eingetragen sind. Das content model dient somit zur vereinfachten Funktionsübersicht der Anwendung.

9.3.2 Navigation Map

Navigation Maps bieten einen Überblick über die Komplexität des Systems. Sie sind auch relevant für das Dokumentieren bestehender Systeme. Durch Navigation Maps können use cases überprüft und korrigiert werden. Die Navigation Map modelliert die Beziehungen zwischen verschiedenen Interaktionsräumen und zeigt Wege auf, wie die Benutzer durch die Interaktionsräume navigieren kann, also von einem Kontext zum nächsten gelangen. Die Navigation Map wird dafür eingesetzt, um Konflikte aufzudecken, da jede Änderung des Kontextes die Benutzer auffordert umzudenken. So löst der Benutzer bei der Implementierung einen Zielkonflikt aus. Das User Interface soll einen simplen und effizienten Workflow bieten. Das Ziel ist es die Aufgaben, die im use case beschrieben sind mit diesem Modell zu erfüllen.

Wichtig zu erwähnen ist, dass bei den Interaktionsräumen der Kontext relativ simpel/ schlicht gehalten werden muss, da die Anzahl der Interaktionsräume zunehmen. Jedoch wenn man die Interaktionsräume geringhält, würde es auch die Komplexität erhöhen. Es muss sozusagen ein Mittelmaß gefunden werden.

Im Folgenden wird eine projektspezifische Navigation Map erstellt, die alle Interaktionsräume und spezielle Funktionen enthält.



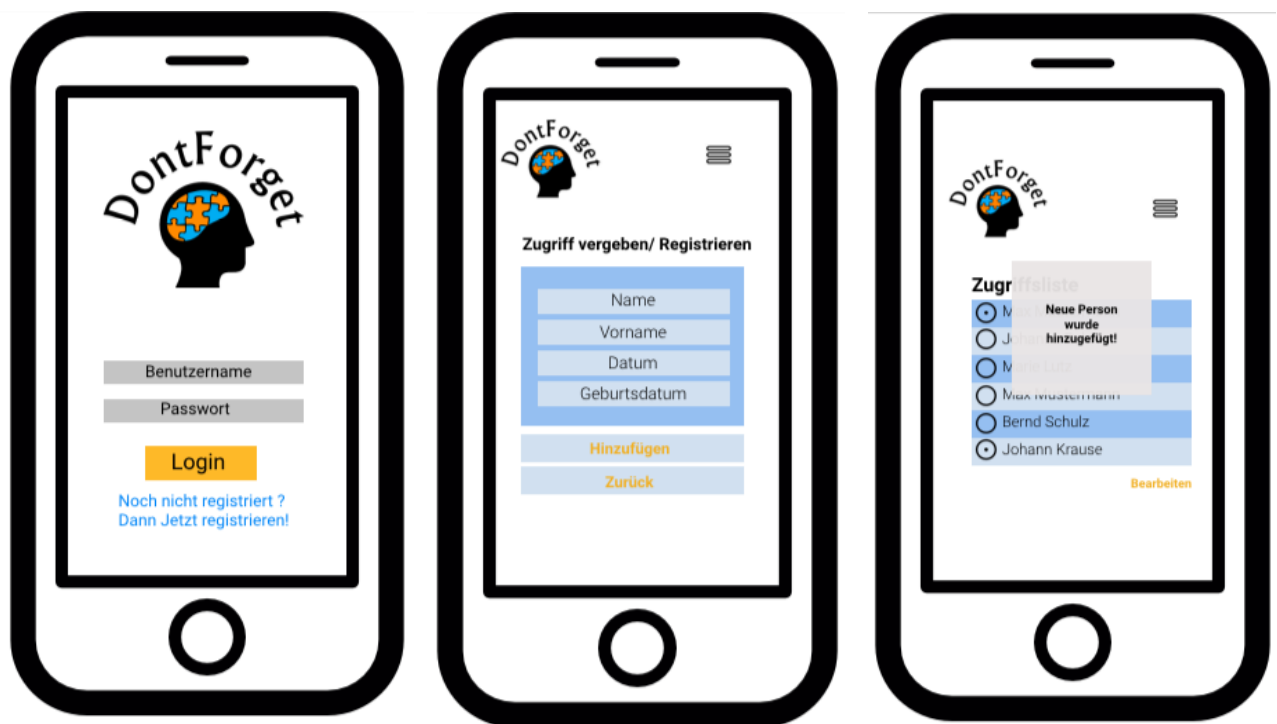
Die erstellte Navigation Map repräsentiert grob die Funktionsdarstellung und den Verlauf des Systems.

Jeder Interaktionsraum wird im Modell von einem Rechteck repräsentiert. Pfeile zwischen den Rechtecken zeigen mögliche Verbindungen und Transitionen auf. Die Pfeile werden noch unterschieden in Aktion und Auswahl. Die doppelt gestrichelten Pfeile stehen für eine Auswahl, die getroffen werden kann und die geradlinigen Pfeile sind für eine Aktion zuständig. Die abgerundeten Kästen stellen Benachrichtigungen bzw. eine Nachricht oder ein Dialogfenster dar. Die Bedingungen sind als Label an den Pfeilen repräsentiert.

10. Implementierungsmodell – UI Prototyp

Als Implementierungsmodell soll der Prototyp des zu entwickelnden Systems dargestellt werden. Der Prototyp besteht aus ...

Mit diesen Darstellungen kann die erste Evaluierung stattfinden. Zudem wird die Implementierung der gesamten Benutzeroberfläche durch die detaillierte Darstellung erleichtert. Die Design Entscheidungen wurden mit Hilfe des Essential Use Case Model, Use Case Map, Anforderungen, Content Model und Navigation Map erstellt. Es wurden jedoch nicht nur die Essential Use Cases dargestellt, sondern auch weitere Wireframes wie die Anmeldung, Registrierung und Bestätigung bei Änderungen. Auf die Einstellungen der Anwendung wurde nicht eingegangen, da diese auf der Android Studio API basiert und die genauen Vorgaben für das Design beinhalten. Aufgrund dessen können eigentlich keine Design Entscheidungen getroffen werden.



(Abbildung 9: Anmeldung und Registrierung)

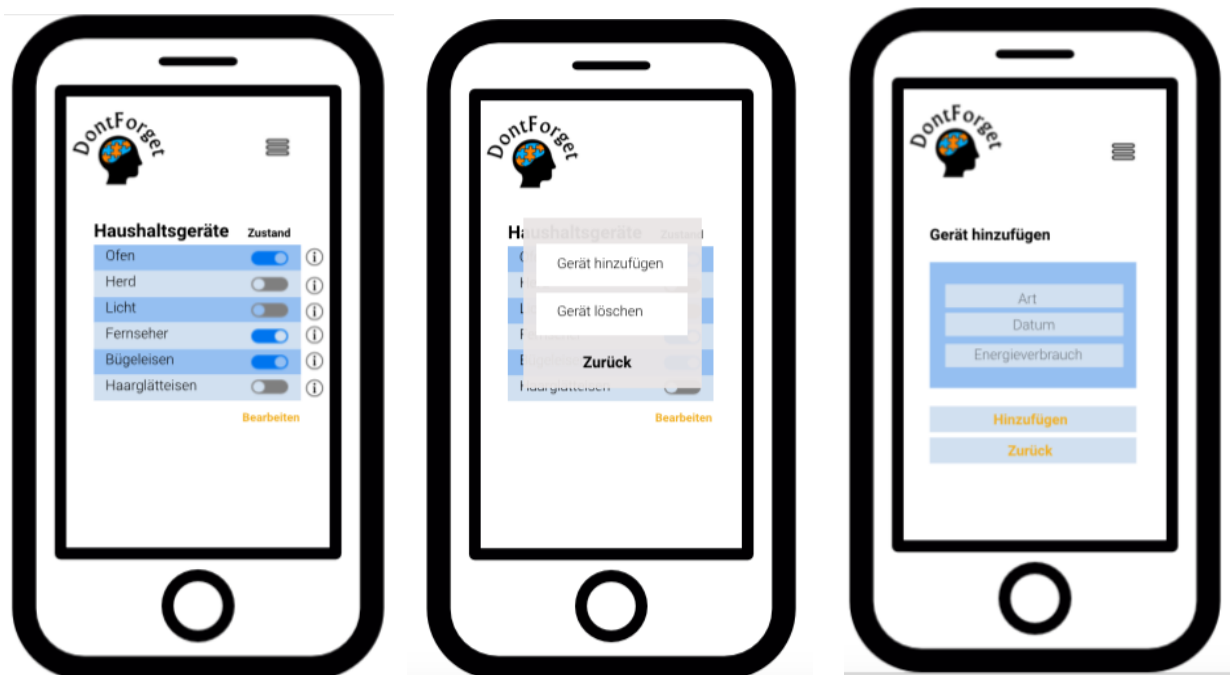
Mockup: Anmeldung und Registrierung

Die Abbildung 9 stellt die Anmeldung und Registrierung der Don't Forget Anwendung dar. Diese Funktion erschien mir als wichtig, da somit nicht jedes Individuum auf das System zugreifen und manipulieren kann. Der Nutzer muss im System registriert sein und sobald er sich registriert hat er Zugriff. Der Hauptuser kann auch Zugriff vergeben, was später beschrieben wird. Die Anmeldung erfolgt einmalig nach Installation der App. Zum „Registrieren“-Wireframe wird der Nutzer über die Anweisung unter dem Login Button weitergeleitet. Es erscheint das Zugriffsformular, weil in beiden Fällen eine Person neu hinzugefügt wird im System. Dieser Wireframe besteht aus einem

Registrierungsformular, bei dem die erforderlichen Daten der Person eingegeben werden müssen, wie z.B. der Name, Vorname, Datum der Registrierung/ Zugriffsvergabe, Geburtsdatum. Damit es zu keinen Konflikten in Bezug auf Redundanzen kommt, wird der Person eine ID zugeteilt vom System. Dieser Prozess findet intern statt und wird durch die „Eigenschaften“-Funktion eingesehen. Mit dem „Zurück“-Button kann der Prozess abgebrochen werden und es erscheint wieder der erste Wireframe. Nachdem die Registrierung oder die Zugriffsvergabe stattgefunden hat, wird weitergeleitet zur Geräteliste mit dessen Statusanzeige.

Mockup: Startseite Geräteliste

Die Startseite beinhaltet die Auflistung der Haushaltsgeräte mit dem aktuellen Zustand. Diese können bearbeitet werden, indem man auf „Bearbeiten“ klickt. Es kann danach ein Gerät hinzugefügt oder gelöscht werden. Bei der Auswahl „Gerät hinzufügen“ erscheint wieder ein Formular, das ausgefüllt werden muss mit der Art des Gerätes, das Datum, und der Energieverbrauch in Watt/Volt. Auch dieser Prozess kann mit dem „Zurück“-Button abgebrochen werden.

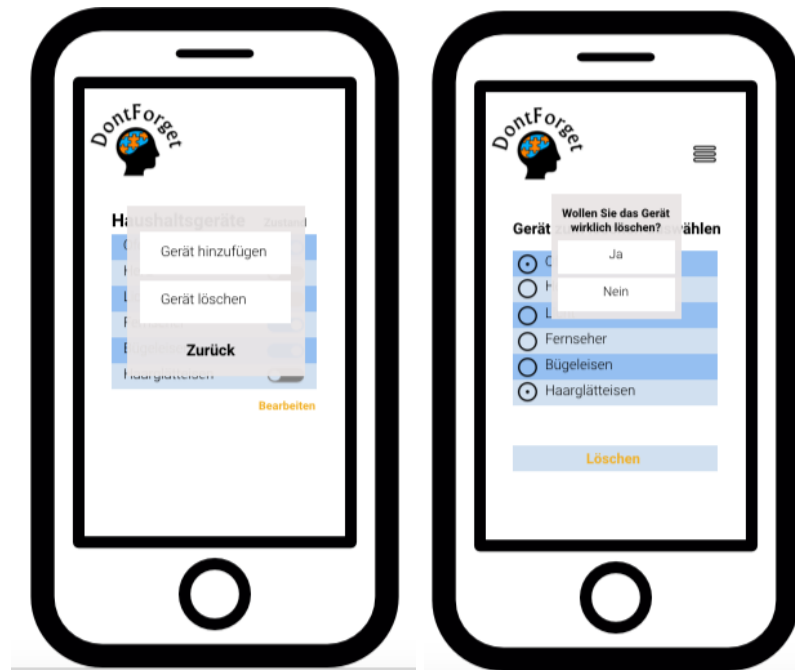


(Abbildung 10: Gerät hinzufügen)



Als letztes gibt es noch eine Bestätigung beim Hinzufügen eines Gerätes.
(Abbildung 10: Gerät hinzufügen Bestätigung)

Wenn ein Gerät aus der Liste gelöscht werden soll, dann erscheint nach dem Bearbeiten wieder die Auswahl von Gerät löschen oder hinzufügen. Wenn es auf Gerät löschen geklickt wurde, dann kann es aus der Geräteliste ausgewählt werden mit dem linken Button. Der Kreis mit dem Punkt in der Mitte stellt das „ausgewählte“ Gerät dar. Nach dem „Löschen“ Button erscheint ein Pop-Up Fenster, ob das Gerät wirklich gelöscht werden soll aus der Liste. Auch dieser Prozess wird mit einer Bestätigung oder einem Abbruch durchgeführt.



(Abbildung 11: Gerät löschen)

Mockup: Zugriffsliste

Die Zugriffsliste ist unter dem Hamburger-Menü zu finden. In diesem werden alle Personen, die Zugriff auf das System haben untereinander aufgelistet und können auch unter „Bearbeiten“ verändert werden. Es erscheint ein Pop-Up Fenster mit der Auswahl „Zugriff vergeben“ und „Zugriff entnehmen“. Bei „Zugriff vergeben“ erscheint wieder das Registrierungsformular oder ein Formular, in dem der Benutzername und das Passwort gefordert wird. Bei „Zugriff entnehmen“ folgt die Zugriffsliste mit den Auswahlbuttons, die in Abbildung 11 zu sehen sind. Auch dieser Prozess wird mit einer Bestätigung beendet.



(Abbildung 12: Zugriffsliste)

10.1 Fazit

Im Großen und Ganzen wurde versucht ein leicht verständliches Design zu erstellen. Einige Mockups wurden aus Zeitgründen nicht erstellt, wie zum Beispiel die einzelnen Schritte des Energieverbrauchs oder das Löschen eines Zugriffs. Bei jedem Mockup ist es möglich über das Logo auf die Startseite zu gelangen. Die aktuelle Seite, in der sich der User befindet, ist immer in der Überschrift zu erkennen.

11. Evaluation

Im Folgenden wurde entschieden, einen analytischen Ansatz zur Evaluation zu wählen, da zum derzeitigen Stand des Projektes noch kein funktionierender Prototyp vorhanden ist und dadurch ein empirischer Ansatz schwieriger durchzuführen ist. Die Evaluation wird während des Entwicklungsprozesses durchgeführt und somit wird eine formative Evaluation durchgeführt anstatt summative. Im Grunde genommen wird die Entwicklung von Gestaltungslösungen zu den Benutzungsschnittstellen mittels Prototypen und deren Evaluation so oft iteriert, bis das Ziel des Projektes erreicht wird. Um viele Defekte der Usability zu finden und bevor eine Evaluation mit echten Benutzern stattfindet, die ggf. den „Ruf“ des zu entwickelnden Systems zu beeinflussen, ist die Heuristische Evaluation gut geeignet. Viele Entwickler empfinden Evaluationsmethoden zu komplex und zeitaufwändig, doch dies wird mit der heuristischen Evaluationsmethode verändert. Die Methode von Nielsen und Molich 1990 eignet sich besonders, da es zum jetzigen Zeitpunkt noch kein fertiger Prototyp vorhanden ist und somit eine Evaluation mit echten Nutzern sich schwieriger gestaltet als eine mit Experten. Die Heuristische Evaluationsmethode empfiehlt sich als gute Usability Inspektionstechnik für gutes User Interface Design.

11.1 Heuristische Evaluation

Zunächst werden Aufgaben formuliert, die die Evaluatoren mit der Systemrepräsentation lösen müssen. Als Evaluatoren dienen ein Domänenexperte und ein Entwickler. Bei jedem Interaktionsschritt werden die Gutachter die Gebrauchstauglichkeit analysieren und auf Probleme hinweisen. Es wird zusätzlich noch die Einhaltung der 10 Heuristiken untersucht.

Aufgaben

1. Login

Es soll sich im System angemeldet werden oder ggf. registriert werden.

2. Status ändern und ein Gerät hinzufügen

Auf der Startseite soll ein Gerät ihren Zustand wechseln und ein neues Gerät hinzugefügt werden mit den Eigenschaften: Art, Datum und Energieverbrauch.

3. Zugriff an eine Person geben

Eine neue Person aus dem Bekanntenkreis soll Zugriff auf das System bekommen.

Ergebnis

1. Visibility of system status (Sichtbarkeit des Systemstatus)

Beobachtung: Es wurde gelobt, dass die Aktualisierung der Daten visualisiert wurde bzw. der Nutzer darüber mit einer Bestätigung benachrichtigt wurde.

Maßnahmen: /

2. Aesthetic and minimalist design (Ästhetisches und minimalistisches Design)

Beobachtung: Die Benennung der aktuellen Seite wurde benannt und es gab keine Probleme beim Finden der Aufgabenlösung. Der Energieverbrauch könnte anders dargestellt werden. Nicht beide Evaluatoren sind auf dem gleichen Weg auf den Energieverbrauch gekommen. Beide hatten unterschiedliche Vorgehensweisen.

Maßnahme: Der Energieverbrauch könnte noch mit ins Menü reingebracht werden, sodass die Hauptfunktionen der App zum Vorschein kommen.

3. User control and freedom (Benutzersteuerung und Freiheit)

Beobachtung: Das System unterstützt die Zurück-Funktion des mobilen Endgerätes, wobei durch das Logo immer die Weiterleitung der Startseite geboten wird.

Maßnahme: Ein (Home)Button für die Startseite wäre nützlich.

4. Consistency and standards (Konsistenz und Standards)

Beobachtung: Es werden einheitliche Symbole benutzt, die mit den Interaktionen übereinstimmen.

5. Error prevention (Fehlerverhinderung)

Beobachtung: Es wurde angemerkt, dass keine Vorgaben bei Formularen angegeben wurden, wie zum Beispiel die Art der Geräte oder generelle Energieverbrauchsangaben etc.

Maßnahme: In Zukunft werden Dropdown-Menüs zur Auswahl ergänzt.

6. Recognition rather than recall (Anerkennung statt Rückruf)

Beobachtung: Es gibt ein Hamburger-Menü, das ersichtlich ist. Eingabefelder und Buttons sind auch erkennbar.

7. Flexibility and efficiency of use (Flexibilität und Effizienz der Nutzung)

Beobachtung: Das Registrieren beim Zugriff vergeben war verwirrend und müsste getrennt werden.

Maßnahme: Es wird in Zukunft ein Mockup für Registrieren erstellt, sodass es leichter nachzuvollziehen ist.

8. Recognition, diagnosis and recovery from errors (Erkennung, Diagnose und Erholung von Fehlern)

Beobachtung: Alle Darstellungen waren verständlich genug.

9. Help and documentation (Hilfe und Dokumentation)

Beobachtung: Aktuell gibt es noch keine Dokumentation zur Benutzung.

Maßnahme: Ein FAQ sollte angelegt werden, wo gängige Fragen erklärt werden.

10. Match between system and real world (Ähnlichkeit zwischen System und reale Welt)

Beobachtung: Das System benutzt Begriffe, welche in der Realität benutzt werden. Es werden keine unverständlichen Systemfachbegriffe benutzt.

11.2 Fazit

Durch die Durchführung der heuristischen Evaluationsmethode, hat sich herausgestellt, dass die Mockups zum größten Teil gut erstellt wurden. Doch einige Mockups müssen über kleine, aber wichtige Punkte überarbeitet und verbessert werden.

12. WBA-Modellierung

In der Tabelle wird eine Übersicht der Ressourcen des Systems abgebildet und die HTTP-Methoden, die von diesen unterstützt werden. Die Ressourcen wurden bei der Spezifikation der REST-API identifiziert nach den Vorgaben von Rest und HTTP.

Ressource	Methode	URI	Semantik	content-type (request)	content-type (response)	Exit	Error
Nutzer	GET	/nutzer/ {nutzerID}	Die Informationen des bestimmten Nutzers werden abgerufen (z.B. Name)	/	Application JSON	200: OK- Anfrage erfolgreich bearbeitet und Ergebnis in der Antwort übertragen	404: (not Found) – angeforderten Ressourcen wurden nicht gefunden
	POST	/nutzer	neuer Benutzer wird erzeugt – URI wird zurückgegeben	Application JSON	JSON URL: http://dontforget.com/nutzer/1001	200: OK 201: Created – angeforderte Ressource wurde erstellt	405: Method Not Allowed – Daten können nicht an den Webserver gesendet werden 500: Internal Server Error
	DELETE	/nutzer/ {nutzerID}	Nutzer wird aus der Liste entfernt	Application JSON	Application JSON	200: OK 204: <u>no</u> Content – Antwort enthält keine Daten, die zurückgegeben werden	403: (Forbidden) – keine Berechtigung

(Abbildung 13: Ressource Nutzer)

Auf die Ressource „Nutzer“ kann man drei der HTTP-Methoden anwenden. Mit GET wird der Nutzer mit einer ID repräsentiert, bei dem der Name mitgeliefert wird. Es wird durch die Registrierung ein neues Userprofil mit allen Eigenschaften angelegt, die aufgerufen, geändert und gelöscht werden kann. Mit POST wird ein neuer Nutzer übergeben in einem URL. Das System soll auch die Möglichkeit bieten einen Nutzer zu entfernen, um das System nicht zu beanspruchen.

Mobiles Endgerät	GET	/mobilgerät	Mobilgerät Informationen werden ausgegeben (z.B. Art, Benutzer, IP-Adresse, Aufenthaltsort)	/	Application JSON, GPS-Standort	200: OK	404: (Not Found) – angeforderten Ressourcen wurden nicht gefunden
	POST	/mobilgerät	Neues Gerät wird erzeugt – URI wird zurückgegeben	Application JSON	Application JSON URL: http://dontforget.com/mobilgerät/50	200: OK 201: Created – angeforderte Ressource wurde erstellt	500: Internal Server Error
	PUT	/mobilgerät/{mobilgerätID}	Gerät Infos werden aktualisiert (z.B. ID oder IP-Adresse)	Application JSON	Application JSON	200: OK 204: No Content – Antwort enthält keine Daten, die zurückgegeben werden	409: merge conflict – bei gleichzeitiger Aktualisierung der Lieferanteninfos 510: (Not Extended) – Anfrage enthält nicht alle Infos, die der Server zwingend erwartet
	DELETE	/mobilgerät/{mobilgerätID}	Gerät wird entfernt	Application JSON	/	200: OK 204: No Content – Antwort enthält keine Daten, die zurückgegeben werden	403: (Forbidden) – keine Berechtigung

(Abbildung 14: Ressource mobiles Endgerät)

Die Ressource „mobiles Endgerät“ liefert alle HTTP-Methoden mit den Informationen z.B. der Art des mobilen Endgerätes, dem Benutzer, der IP-Adresse und dem Aufenthaltsort. Der Aufenthaltsort wird durch eine GPS-Abfrage ermittelt. Zuerst war der Gedanke den Aufenthaltsort als eigene Ressource zu betrachten. Die Ressource würde nur eine GET-Methode liefern und daher hat es mehr Sinn gemacht diesen mit dem mobilen Endgerät einzubinden. Die Funktionen der restlichen Methoden sind in der Tabelle aufgelistet.

Haushaltsgerät	GET	/hausgerät / {hausgerätID}	Geräte Information wird abgerufen (z.B. Art, Sensor, Energieverbrauch, Status (ON/OFF))	/	Application JSON	200: OK	404: (Not Found) – angeforderten Ressourcen wurden nicht gefunden
	POST	/hausgerät	Entität erstellen, die einer bekannten Entität untergeordnet ist (neues Gerät wird erzeugt – URI wird zurückgegeben)	Application JSON	Application JSON URL: http://dontforget.com/ hausgerät/300	200: OK 201: Created – angeforderte Ressource wurde erstellt	500: Internal Server Error
	DELETE	/hausgerät / {hausgerätID}	Gerät wird aus der Liste entfernt	Application JSON	Application JSON	200: OK 204: No Content – Antwort enthält keine Daten, die zurückgegeben werden	403: (Forbidden) – keine Berechtigung

(Abbildung 15: Ressource Haushaltsgerät)

Die Ressource „Haushaltsgerät“ beinhaltet die Informationen: Art des Gerätes, den dazugehörigen Sensor, den Energieverbrauch und den Status des Gerätes. Auch hier war am Anfang der Gedanke die Sensoren zu trennen und als eigene Ressource einzuteilen. Es sollte aber jeder Sensor ein Haushaltsgerät repräsentieren. Deswegen passt die Kombination zueinander und es kann direkt der Bezug auf die Sensoren implementiert werden.

13. Datenstruktur

Als Daten werden vom Server nur die User abgespeichert. Es wird eine Datenbank für das Login-Verfahren implementiert. Alle User müssen sich im System registrieren, um Zugriff zu haben auf die Haushaltsgeräte und den Energieverbrauch einzusehen. Der Energieverbrauch wird über einen Algorithmus bestimmt. Durch das Login-Verfahren wird der Benutzer authentifiziert und kann Teil der Zugriffsliste sein.

Als Entität gibt es nur den User, der mit dem System interagiert und Zugriff an andere vergeben kann.

```
CREATE TABLE User (  
  UserID smallint NOT NULL,  
  Username VARCHAR (255) NOT NULL,  
  Passwort VARCHAR (255) NOT NULL,  
  SysName VARCHAR (255) NOT NULL  
);
```

14. Alleinstellungsmerkmal – überarbeitet

Alleinstellungsmerkmale heben sich im zu entwickelnden System in einigen Punkten der Konkurrenz ab. Aus der Domänen- und Marktrecherche können Alleinstellungsmerkmale abgeleitet werden. Die durch die Recherche gewonnenen Informationen sollen helfen, die Funktionalitäten des zu entwickelnden Systems herauszuarbeiten und zu planen. Zusätzlich helfen sie bei der Entwicklung eines grundlegenden Konzeptes für das System. Im Folgenden werden die heraus entwickelten Alleinstellungsmerkmale vorgestellt, die von MS1 überarbeitet wurden.

14.1 Push-Benachrichtigungen

Push-Benachrichtigungen sind wichtige Bestandteile des zu entwickelnden Systems. Sie geben Informationen an den User, die eine Interaktion fordern. Bei nicht Eintreffen der Benachrichtigung, wird es erneut der Server angesprochen und die Information abgeschickt. Dabei wird auch auf die Redundanz Rücksicht genommen, sodass keine doppelten Nachrichten eintreffen. Um überhaupt Benachrichtigungen im System zu erhalten, muss der User eingeloggt sein, Zugriff auf das System haben und seine GPS-Daten freigeben. Push-Benachrichtigungen enthalten die Zeit, den Inhalt (um welches Gerät es sich handelt) und den Status der Haushaltsgeräte. Somit ist der User jederzeit erreichbar und kann über Wichtiges benachrichtigt und gewarnt werden. Die Push-Benachrichtigung ist in anderen Anwendungen der im Konzept genannten Marktrecherche auch vorhanden. Die dort angewandte Push-Benachrichtigung erfüllt jedoch einen anderen Zweck, nämlich nur die Informationsübermittlung zum Beispiel, wenn das Wasser der Kaffeemaschine nachgefüllt werden muss oder die Tabs der Geschirrspülmaschine fehlen etc. Die Benachrichtigung der Don't Forget Anwendung hebt sich von anderen Anwendungen ab, da der User vor Gefahren gewarnt wird, die vermieden werden können.

14.2 GPS

Auch GPS-Ermittlung ist einer der wichtigsten Bestandteile des zu entwickelnden Systems. Ein mit dem Raspberry Pi verbundener GPS-Sensor ermöglicht die Ermittlung des aktuellen Aufenthaltsortes. Der Sensor ist in Verbindung mit dem Ortungsdienst des mobilen Endgeräts und muss mit ihm kommunizieren, um den Standort festlegen zu können. Nur nach Festlegung des Standorts, kann die Überprüfung der Haushaltsgeräte und der Informationsaustausch stattfinden. Es wird auch eine Benachrichtigung gesendet, wenn der Ortungsdienst ab einer bestimmten Entfernung nicht angeschaltet ist. Zur eigenen Sicherheit müsste diese dann aktiviert werden. In diesem Projekt werden aus zeittechnischen Gründen, Spieldaten (manuelle GPS-Eingabe zum mobilen Endgerät) verwendet.

14.3 Steuerung

Die Hauptfunktion, nämlich die Steuerung der Haushaltsgeräte, ist die Lösung des Nutzungsproblems. Das zukünftige System soll dem User Sicherheit geben. Nach Erhalt der Push-Benachrichtigung sollen die Haushaltsgeräte, die vergessen wurden auszuschalten, gesteuert werden können. Genauso wie die Benachrichtigung des zu viel verbrauchenden PCs oder TVs. Auf diese ist Zugriff, wenn der User eingeloggt ist mit seinem Benutzernamen und Passwort. Somit kann die Sicherheit zum Teil gewährleistet werden. Die mit dem Server verbundenen Haushaltsgeräte (Sensoren) werden über WLAN ermittelt und an den User mitgeteilt.

14.4 Stromnetz

Anders als bei den konkurrierenden Anwendungen, läuft dieses System ohne das WLAN-Netz. Es sollen die Angriffe von außen vermieden und um eine Komponente erleichtert werden. Die Haushaltsgeräte werden ausschließlich an Steckdosen mit Strom versorgt, sodass keine fremden Menschen sich in das System hacken können.

14.5 Selbstkonfiguration

Hierbei werden die Haushaltsgeräte an den Steckdosen verbunden und manuell im System registriert mit den Parametern Art, Datum und Energieverbrauch. Somit wurde der WLAN Kommunikationskanal vermieden und das System vereinfacht. Konkurrenten verbinden die Geräte über das WLAN, was die Komplexität steigert. Doch das zu entwickelnde System wird über das Stromnetz verbunden und ausgeführt.

14.6 Energieverbrauch

Anders als bei den Konkurrenten wird zusätzlich noch der Energieverbrauch durch einen Algorithmus berechnet und dem Nutzer die Möglichkeit gegeben diesen einzusehen. Die Energiecheck-Anwendung von MS1 in der Marktrecherche ist an manuellen Funktionen gebunden, wie z.B. das Einscannen der Zählerstände etc. Doch bei der Don't-Forget App wird nur am Anfang der generelle Energieverbrauch in Volt/Watt angegeben. Die Berechnung wird dann aktiviert, wenn das Gerät im aktiven Zustand ist. Der aktuelle Verbrauch wird dann über den Informations-Button festgestellt.

15. Fazit Projekt

Mein Ziel war es alle Punkte vollständig und projektspezifisch zu begründen und zu erarbeiten. Dies ist mir bei einigen Punkten schwergefallen, jedoch habe ich größtenteils mein Vorgehen klar definiert und beschrieben. Bei den Punkten „Datenstrukturen“ war ich mir ziemlich unsicher, da ich nur eine Datenbank für die User implementieren möchte. Es wäre kein ER-Diagramm dafür geeignet. Aus zeittechnischen Gründen konnte ich den Punkt „Anwendungslogik“ nicht detailliert genug beschreiben. In diesem Meilenstein habe ich mich ausführlich mit den Benutzermodellen konfrontiert und zugleich Punkte aus MS1 überarbeitet und korrigiert. Im Projektplan wird deutlich, dass ich die eingeplante Zeit ziemlich unterschätzt habe und der ganze Prozess sehr zeitaufwändig wurde, denn jedes Thema musste intensiv überdacht und zielführend bearbeitet werden.

16. Projektplan

16.1 Meilenstein 2

<i>Datum/KW</i>	<i>Aktivität</i>	<i>1. Unteraktivität</i>	<i>2. Unteraktivität</i>	<i>Workload geplant</i>	<i>Workload tatsächlich</i>
21.11.18 47. KW	überarbeitet	Zielhierarchie		2h	2h
21.11.18 47. KW	überarbeitet	Risiken		2h	3h
26.11.18 48. KW		Anwendungslogik		2h	2h
29.11.18 48. KW		Anforderungen		3h	4h
3.12.18 49. KW	überarbeitet	Architekturdiagramm		4h	5h
6.12.18 49. KW		Vorgehensmodell – Usage centered design		5h	6h
7.12.18 49. KW			Role Model	4h	5h
9.12.18 49. KW			Task Model	6h	7h
9.12.18 49. KW			Content Model	6h	8h
9.12.18 49. KW	überarbeitet	Proof of Concept		2h	2h
10.12.18 50. KW		Navigationsmodell		5h	8h
11.12.18 50. KW		Implementationsmodell – UI Prototyp		8h	11h
12.12.18 50. KW		Evaluation		8h	9h
14.12.18 50. KW		Datenstruktur		2h	2h
13.12.18 50. KW		WBA-Modellierung			
9.12.18 49. KW			REST-Tabelle	6h	6h
12.12.18 50. KW	überarbeitet	Alleinstellungsmerkmal		3h	3h
16.12.2018	2. MS				
				68h	83h

(Abbildung 16: Projektplan Meilenstein 2)

16.2 Meilenstein 3

<i>Datum/KW</i>	<i>Aktivität</i>	<i>1. Unteraktivität</i>	<i>2. Unteraktivität</i>	<i>Workload geplant</i>	<i>Workload tatsächlich</i>
51. KW – 56.KW		Implementation		200h	h
51. KW – 56.KW		Implementations- dokumentation		12h	h
51. KW – 56.KW		Fazit		4h	h
51. KW – 56.KW		Prozessassessment		6h	h
20.01.2019	3. MS				
				222h	

(Abbildung 17: Projektplan Meilenstein 3)

17. Literaturverzeichnis

1. **Rombach, Rainer.** Mein Stromverbrauch. [Online] 9. 8 2013. [Zitat vom: 2. 11 2018.] <http://www.rainerrombach.de/apps/mein-stromverbrauch/>.
2. **Preece, Jenny.** *Human-Computer-Interaction.* s.l. : Addison-Wesley-Publishing Company, 1994. 0-201-62769-8.
3. **Plaßmann, Prof.Dr. Gerhard.** *Handbuch der MCI.* 14.April 2008.
4. **gmbh, mobile solutions.** bluesource. [Online] 2018. [Zitat vom: 9. 12 2018.] <https://www.bluesource.at/blog/detail/wissenswertes-ueber-push-notifications/>.
5. **ICONS.** [Online] [Zitat vom: 11. 12 2018.] <https://icons8.de/icon/set/info/ios>.
6. **Material Design.** [Online] [Zitat vom: 10. 12 2018.] <https://material.io/design/layout/understanding-layout.html#>.