

Implementationsdokumentation

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, 20.01.2019

Inhaltsverzeichnis:

1.	EINLEITUNG.....	3
2.	PROZESSASSESSMENT	4
2.1	Thematischer Block MCI	4
2.2	Thematischer Block Architektur	4
2.3	Verhalten bei Problemen und Komplikationen	5
2.4	Einhaltung vom Projektplan.....	5
3.	ZIELERREICHUNGSGRAD	6
4.	FAZIT	7
5.	ABWEICHUNGEN MS2	8
6.	INSTALLATIONSANLEITUNG.....	13
7.	PROJEKTPLAN	15
	Meilenstein 3	15

1. Einleitung

Die Implementationsdokumentation, die im Rahmen des Moduls **Entwicklung Interaktiver Systeme** erstellt wurde, beinhaltet das Prozessassessment und den Zielerreichungsgrad für das Projekt. Zudem befinden sich die Installationsdokumentation, sowie die Änderungen die in diesem Meilenstein fortführend vom 2. Meilenstein bearbeitet wurden.

2. Prozessassessment

2.1 Thematischer Block MCI

Nach langer Überlegung wurde das System nach den Modellen des „Usage centered design“ von L. Lockwood und L. Constantine entwickelt. Das Buch „Human-Computer-Interaction“ von Jenny Preece war ein guter Leitfaden, die Modelle effektiv umzusetzen. Insbesondere die Konzentration auf die Role, Task- und Content Model, sowie das Navigationsmodell hat zeitlich gesehen eine effektive Wirkung auf das Projekt und die Arbeitsweise. Zudem waren die Modelle eine große Unterstützung, nicht lösungsorientiert zu denken, da beispielsweise User Interface Entscheidungen erst später im Implementierungsmodell getroffen werden und vorher der Fokus auf den Inhalt bzw. die Funktion gesetzt wurde. Es wurden Anforderungen und Ziele definiert, die an das System gerichtet sind und werden im nächsten Kapitel „Zielerreichungsgrad“ diskutiert, ob diese gut realisiert wurden.

Mit dem Ziel, frühzeitig Fehler oder Risiken zu bemerken und diese im Rahmen der Projektzeit beheben zu können, wurde anschließend eine Evaluationsphase durchgeführt. Die heuristische Evaluationsmethode hatte eine effektive Wirkung auf den zuvor modellierten Prototypen. Die Ergebnisse der Evaluationsmethode konnten überdacht und direkt in den modellierten Prototyp übernommen werden, sodass die Implementation gezielt und fokussiert begonnen werden konnte.

2.2 Thematischer Block Architektur

Im 1. Meilenstein war die Architektur noch nicht ganz fehlerfrei überdacht. Es waren auch andere Architektur Darstellungen möglich wie z.B. das Peer-to-Peer, doch ich hatte zuvor keine Erfahrung mit der Implementierung des Peer-to-Peers gesammelt. Daher wollte ich es nicht riskieren und habe mich für die Client-Server Architektur entschieden. Für das Rapid Prototyping war es aus zeitlichen Gründen schwierig eine mobile Anwendung zu programmieren. Daher habe ich es in einer Webansicht mittels node.js implementiert. Als Architekturstil war REST geplant, jedoch war diese in Zukunft für die Socket Programmierung nicht möglich. Um einem Client die Verbindung zu ermöglichen, musste man Sockets beidseits planen und implementieren. Alle Artefakte wurden iterativ in Frage gestellt und überdacht. Dies war ein sehr zeitintensiver Prozess. Ziel dieses Vorgehens war es, bei der Implementierung so wenig wie möglich anzupassen und zielgerichtet an der Umsetzung zu arbeiten. Besonderer Fokus wurde hierbei auf die Anwendungslogik gesetzt, die schrittweise spezifiziert wurde. Spätestens bei der Implementation stellten sich Probleme auf, wie z.B. der Verbindungsaufbau mit dem Raspberry Pi, der als Server dienen sollte. Die ersten Probleme gab es mit der Ausleihe eines Raspberry Pi's, da es vom moxdlab-Büro aus nicht mehr möglich war. Dieses Vorgehen hätte früher in Frage gestellt werden müssen. Das Equipment wurde beschaffen mit den nötigen Sensoren, jedoch gab es nach langer Recherche

Konfigurationsprobleme des Pi's. Somit war die Umsetzung der Anwendungslogik nur teils möglich. Der Fokus beim Server wurde auf die Implementation der Sockets und Testdaten gesetzt. Danach wurde sich auf die Implementation des Clients und Umsetzung der Anwendungslogik konzentriert.

2.3 Verhalten bei Problemen und Komplikationen

Bei der Projektdurchführung war der Zeitfaktor ein Hauptproblem. Diesem Problem konnten durch wiederholtes Reflektieren des Vorgehens und immer wieder neuen Fokus setzen, entgegengewirkt werden. Technische Probleme wurden durch abgekapselte Problembehandlung, Recherche und Austausch zwischen Freunden nur teils gelöst. Konzeptionelle Probleme konnten ebenfalls durch Recherche und Fragen bei Beratungsterminen gelöst werden.

2.4 Einhaltung vom Projektplan

Zuallererst wurden bei der Erstellung des Projektplans nur grobe Aktivitäten festgelegt. Auf diese wurde beim 1.Meilenstein hingewiesen und in Zukunft wurden diese spezifiziert. Es wurden wöchentlich individuelle Deadlines festgelegt, damit frühzeitig Probleme erkannt und diese bei Beratungsgesprächen geklärt werden konnten. Der beigefügte Projektplan weist gegenüber der geplanten und tatsächlich durchgeführten Arbeitszeit einen Unterschied auf. Zu Anfang wurde viel Zeit für die Themenfindung und das Exposé investiert als ursprünglich geplant, da die Anwendungslogik der anderen zur Auswahl stehenden Themen nicht ausreichend war. Auch im Weiteren wurden Artefakte so verändert, dass man spätestens bei der Implementierung effektiv arbeiten kann. Dadurch mussten die in Meilenstein 2 angegebenen Mindestanforderungen eingehalten werden. Meilenstein 3 weicht dadurch sehr von der ursprünglich eingeplanten Zeit ab.

3. Zielerreichungsgrad

Die am Anfang gesetzten Ziele wurden mit dem Projektergebnis verglichen, um den Erreichungsgrad zu definieren. Die strategischen, taktischen und operativen Ziele, die sich auf die Anwendungslogik beziehen wurden konzipiert und mit Beachtung des PoC's versucht zu implementieren.

Im Großen und Ganzen konnte das Nutzungsproblem nicht gelöst werden, jedoch ist es nicht unmöglich gewesen dieses zu lösen, wenn es mehr Zeit zur Verfügung stehen würde. Einige Ziele wurden bei der Konzipierung des Systems berücksichtigt, jedoch aus zeitlichen Gründen im Prototypen nicht umgesetzt.

Ziel 1: Es soll der Aufenthaltsort richtig geprüft und ermittelt werden.

Zielerreichungsgrad: Dieses Ziel konnte nicht realisiert werden, da es technische Probleme mit dem Server gab und über eine lange Zeit keine Verbindung möglich war.

Ziel 2: Es sollen alle Haushaltsgeräte angezeigt werden, die nicht ausgeschaltet wurden.

Zielerreichungsgrad: Auch dieses Ziel war abhängig vom Server und konnten nicht richtig dargestellt werden.

Ziel 3: Die Sensoren sollen geprüft und angesteuert werden können.

Zielerreichungsgrad: Die Sensoren waren nur über den Raspberry anzusprechen und nicht über einen Client.

Ziel 4: Der Nutzer soll Push-Benachrichtigungen vom Server erhalten können. Es dürfen keine doppelten Benachrichtigungen empfangen werden. Die Benachrichtigung enthält eine Uhrzeit.

Zielerreichungsgrad: Die Push-Benachrichtigung erscheint beim Starten der App, so dass der User direkt eine Warnung erhält und weiß wie er zu handeln hat.

Ziel 5: Es sollen bei Nottfällen alle Hausbeteiligte vor Gefahren informiert und gewarnt werden.

Zielerreichungsgrad: Dies geschieht in Zukunft für alle die sich in der gleichen „SystemID“ befinden.

Ziel 6: Es soll der Energieverbrauch berechnet werden.

Zielerreichungsgrad: Der Energieverbrauch wird in der Geräteliste mit angezeigt.

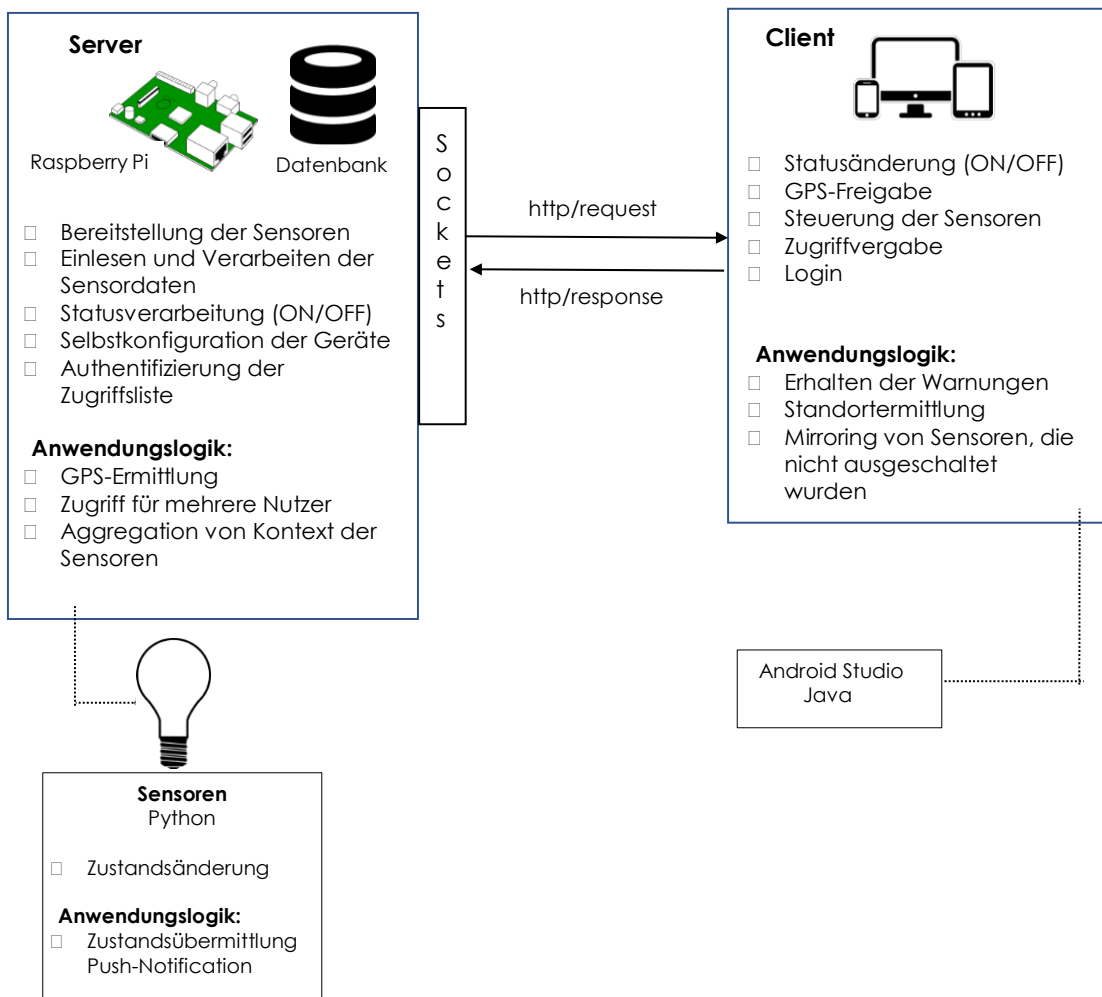
4. Fazit

Das Nutzungsproblem könnte durch das entwickelte System effektiver gelöst werden, wenn es mehr Zeit zur Verfügung stehen würde. Die Konzipierung des Systems mit „Usage centered design“ als Vorgehensmodell und die verwendete Evaluierungsmethode „Heuristische Evaluation“ konnte erfolgreich bearbeitet und umgesetzt werden, jedoch aus Zeitgründen nicht vollständig implementiert werden. Die Anwendungslogik der Komponenten wurden skizziert und erforderliche Architekturmerkmale, wie „Ressourcen“ modelliert. Bei der Implementierung musste aus zeit- und technischen ein Fokus auf den Client gesetzt werden, da es bis zur letzten Woche der Abgabe keine Kommunikation mit dem Server „Raspberry Pi“ möglich war. Das Problem lag daran, dass der Raspberry Pi nicht von der TH ausgeliehen werden konnte und dementsprechend nicht vorkonfiguriert war. Der Raspberry hatte eine statische IP-Adresse, was die Kommunikation zwischen dem Client abgeblockt hatte. Durch lange Recherche konnte dieses Problem leider nicht vorher gelöst werden. Trotzdem demonstriert der vertikale Prototyp durch die exemplarische und unvollständige Umsetzung der Anwendungslogik, dass diese insgesamt verwirklicht werden kann. Für die komplette Umsetzung des Systems müssen noch Methoden vervollständigt und die Sensoren und Datenbank eingebunden werden.

Während der Projektdurchführung wurden die theoretischen Konzepte der Fächer „Web basierte Anwendungen 2“ und „Mensch Computer Interaktion“ gelernt praktisch umzusetzen.

5. Abweichungen MS2

Abweichungen gab es hinsichtlich des Architekturdigramms, da es für die ersten Abgaben eine andere Anwendung geplant war, nämlich die Webansicht, anstatt die mobile Ansicht.



(Abbildung 1: Architekturdigramm überarbeitet)

Zudem gab es Abweichungen bei der Umsetzung des Prototypen, die für den User vereinfacht wurden. Die Anwendung startet mit einer Warnung. Aus einem ungelösten Grund wird der „OK“ Button transparent angezeigt, jedoch ist dieser unten rechts anzuklicken. Somit wird zum Login weitergeleitet.

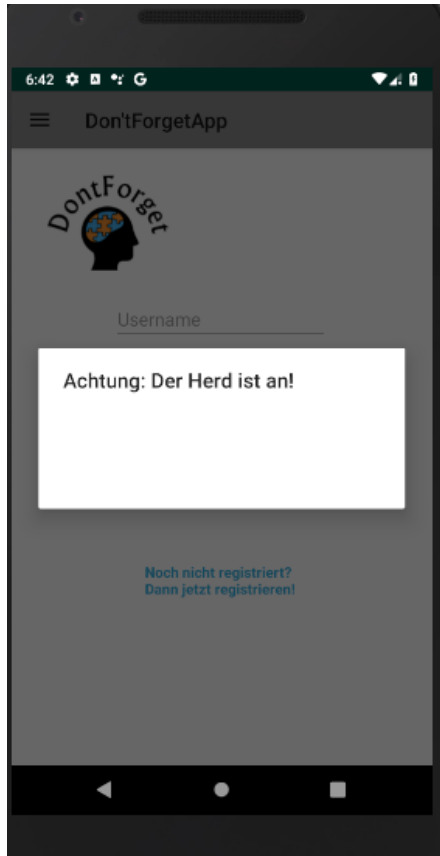


Abbildung 2: Warnung

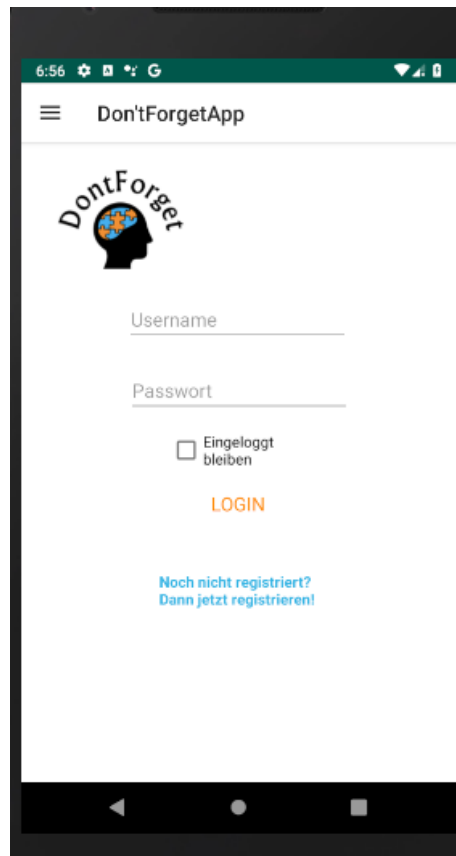


Abbildung 3: Login

Eine weitere Änderung war das Einführen der Navigationsbar, um die wichtigsten Funktionalitäten für den User ersichtlich zu machen. Unter dem Menüpunkt Profil kann der User seine privaten Daten bzw. sein Konto verwalten, wie z.B. sein Passwort oder sein Username ändern. Unter „Warnungen“ wie der Name schon sagt, werden alle Push-Notifications abgespeichert, die man einsehen kann. Unter dem Menüpunkt „Einstellungen“ kann der User Zugriff vergeben und entziehen. Beim Klicken des „Zugriff vergeben“ Buttons wird zur Registrieren-Seite weitergeleitet. Dies ist auch eine Änderung zum vorher im MS.2 geplanten Konzept. Das bedeutet jeder, der im System **registriert** ist, hat Zugriff zum System.

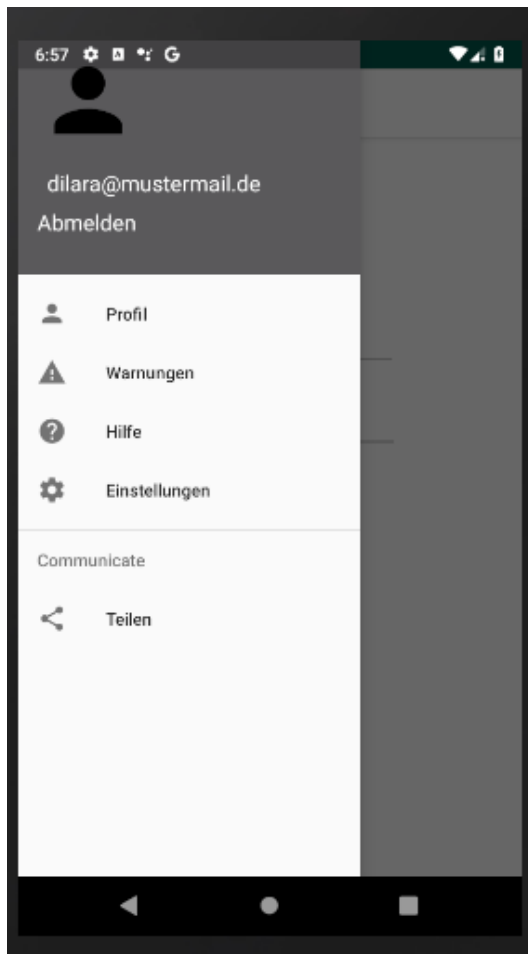


Abbildung 4: Navigation Toolbar

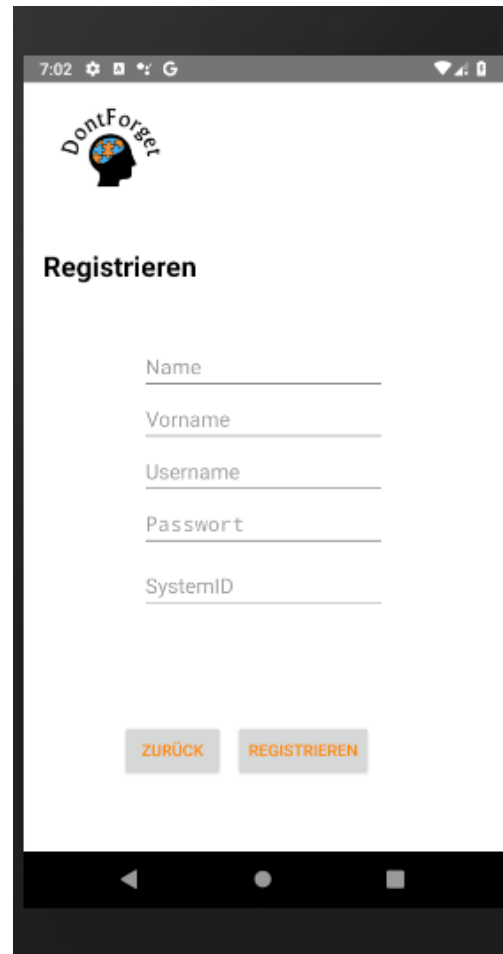


Abbildung 5: Registrieren

Nachdem erfolgreich registriert wurde, gelangt man zur Zugriffsliste. Hierfür wurde eine Datenbank in MySQL implementiert, die aus Zeitgründen nicht mehr eingebunden werden konnte. Es können User hinzugefügt und Zugriff entnommen bzw. gelöscht werden. Die Selbstkonfiguration der Geräte ist eines der Alleinstellungsmerkmale und wurde umgesetzt in einer Geräteliste. Man muss ein Gerät hinzufügen mit den Parametern Art, Stromverbrauch und das Datum wird automatisch übergeben. Erst dann kann es vom System aufgenommen werden. Nachdem diese eingegeben worden sind, erscheinen die Geräte in der „Haushaltsgeräte“-Liste.

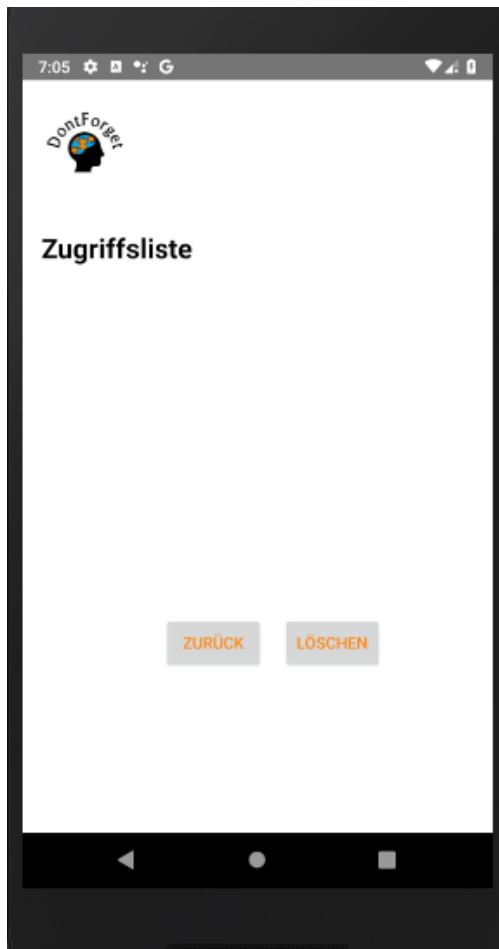


Abbildung 6: Zugriffsliste

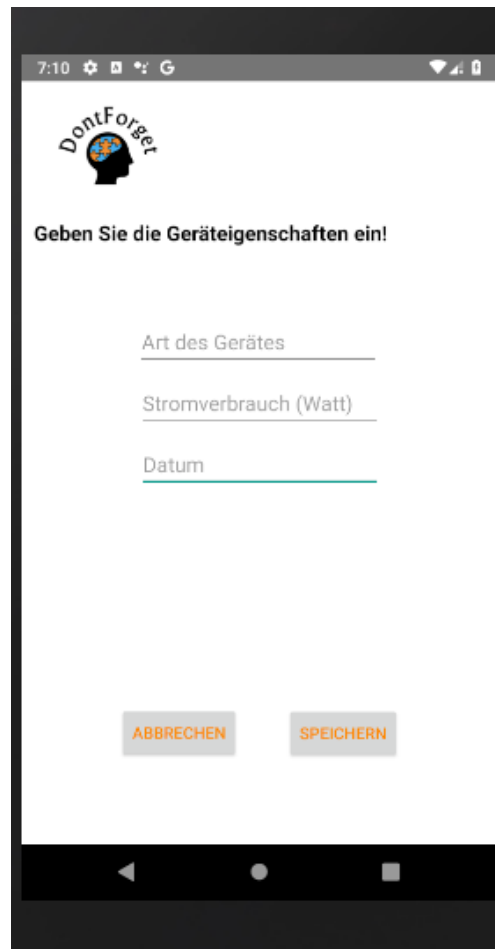


Abbildung 7: Gerät hinzufügen



Abbildung 8: Haushaltsgeräte mit Zustand



Abbildung 9: Geräteliste

Alle im System registrierten Geräte werden in „Geräteliste“ abgespeichert (Abb.9). Diese können ausgewählt und auch aus der Liste gelöscht werden. Der Zustand der Haushaltsgeräte wird durch das Toogle Button verändert. Die Implementierung konnte aus Zeitgründen auch nicht umgesetzt werden. Die Anwendung hat eine relativ simple Übersicht und weist auf keine Begriffskomplikationen. Das Konzept der Anwendung wurde teils verändert wie zum Beispiel das Registrieren und Zugriff vergeben wurde in eine Aktivität zusammengeführt, anstatt in zwei unterschiedliche. Somit hat jeder Zugriff, der auch im System registriert wurde. Auch der Parameter „SystemID“ war nicht vorher überdacht, denn jeder der Zugriff bekommt darf nur eine ID auswählen, in dessen Nähe er sich befindet. Diese ist auch wiederum mit GPS-Daten zu erfassen, was in diesem Projektumfang nicht mehr zu realisieren war.

6. Installationsanleitung

Datenbank

Die Datenbank wurde in MySQL / Java implementiert. Der Ordner beinhaltet alle Klassen, die für die Datenbank nötig sind.

Link:

https://github.com/Dilara9402/EISWS1819Gueven/tree/master/Meilenstein3/Code/Datenbank/DB_EIS/src

Server

Da der Server von Anfang an ein großes Problem war, wegen dem Verbindungsaufbau, war es bis zur letzten Woche keine Verbindung zwischen Client- und Server möglich.

Der Servercode befindet sich unter diesem Link:

<https://github.com/Dilara9402/EISWS1819Gueven/tree/master/Meilenstein3/Code/EIS/Server/src>

Für den Verbindungsaufbau muss die IP-Adresse eine dynamische sein. Der Raspberry hingegen war mit einer statischen IP-Adresse vorkonfiguriert, was die Verbindung von einem Client geblockt hat und nach einer bestimmten Wartezeit ein Timeout erschien.

Nachdem diese geändert worden ist, konnte eine Verbindung hergestellt werden. Für die Ausführung muss im Terminal eingegeben werden:

Javac Server.java – für das Kompilieren
Java Server – für die Ausführung

Ein erfolgreicher: **Server is running**

Client

Die für den Client verwendete Software: **Android Studio v.3.3**

Benötigt wird eine Standard Installation von Android Studio.

Link: <https://developer.android.com/studio/index.html>

Die App kann nur mit einem Emulator gestartet werden. Dieser wird separat installiert je nach Betriebssystem. Für den Client gibt es zwei unterschiedliche Ordner.

Der Ordner unter EIS im GitHub wurde nur für den Verbindungsaufbau getestet und implementiert.

Link:

<https://github.com/Dilara9402/EISWS1819Gueven/tree/master/Meilenstein3/Code/EIS/Client/src>

Der eigentliche Client ist als APK im folgenden Link zu finden:

<https://github.com/Dilara9402/EISWS1819Gueven/tree/master/Meilenstein3/Code/apk/debug>

7. Projektplan

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>
44. KW – ab 29.10.18		Implementation			
48.KW- 26.11.18			Client	40h	60h
44.KW- 30.10.18			Server	60h	90h
1.KW – 05.01.19			Datenbank	40h	60h
1.KW- 01.01.19			Android Studio	30h	70h
Ab 2. KW – 10.01.19		Implementations- dokumentation		12h	15h
2.KW- 13.01.19			Abweichungen	5h	7h
2.KW – 13.01.19			Prozessassessment	6h	8h
3. KW – 16.01.19			Fazit	2h	4h
3.KW – 19.01.19			Zielerreichungsgrad	5h	8h
20.01.2019	3. MS				
				200h	322h

(Abbildung 17: Projektplan Meilenstein 3)