

Pflichtenheft

SmartHome

**Eingebettete, intelligente Steuerung mit Webinterface für
SmartHome-Applikationen für intelligente Gebäude**

Gamze Söylev Öktem, Nils Kohlmeier, Jonas Wiese
Wladimir Streck, Justin Jagieniak

SoSe 2017

1 Allgemeines

1.1 Projektbeschreibung

In dem Pflichtenheft werden das Ziel des Projekts und die Anforderungen an das Endprodukt aus Sicht der Projektgruppe dokumentiert. Zentraler Bestandteil des Pflichtenhefts sind die User Stories, die aus der Projektplanungsphase entstanden sind. Die Diagramme dienen der Verdeutlichung, wie die Anforderungen umgesetzt werden sollen. Ferner liegt ein vorläufiger Projektplan vor, welcher im Laufe des Projektes noch wachsen oder verändert werden kann.

1.2 Hintergrund und Zielsetzung

Ziel des Projektes ist der Neubau einer SmartHome-Anwendung mit mehreren Sensoren und Aktoren sowie einer zentralen Steuereinheit. Die Konfiguration und Einsicht in die Daten soll durch ein Frontend in Form einer Weboberfläche bewerkstelligt werden.

Das Ziel des Semesterprojektes ist es, zunächst diese Anforderungen systematisch und möglichst vollständig zu erfassen, daraus einen Entwurf für die Software abzuleiten und sie selbst zu implementieren.

1.3 Gruppenbeschreibung

Table 1: Gruppenbeschreibung mit Rollenverteilung

Rolle	Name
Projektleitung Bildverarbeitung/OpenCV Frontend	Gamze Söylev Öktem
Softwareentwicklung HAL/Controller Frontend/Backend Entwicklung	Nils Kohlmeier
Softwareentwicklung Controller Datenmanagement MQTT	Jonas Wiese
Softwareentwicklung HAL/Controller Cloud Architekt Frontend	Wladimir Streck
Bildverarbeitung/OpenCV	Justin Jagieniak

2 Backlog als Userstories

2.1 Muss-Anforderungen

- Als Benutzer muss ich in das System über eine Schnittstelle so konfigurieren können, dass es automatisch auf bestimmte Eingaben reagiert um den Komfort zu erhöhen.
- Als Benutzer muss ich über eine Oberfläche gezielt Geräte An- und Ausschalten können, um Aktoren auch manuell steuern zu können.
- Als Smart-Home-Besitzer muss ich via Gesichtserkennung die Haustür öffnen, um die Tür ohne Schlüssel öffnen zu können.
- Das System muss in der Lage sein zu erkennen ob es brennt, um auf die Eingabe reagieren zu können.
- Das System muss in der Lage sein VOC-Daten zu erheben, um automatisch eine Aktion ausführen zu können z.B. Lüften bei zu hoher VOC-Belastung.
- Das System muss in der Lage sein Feuchtigkeit Daten zu erheben, um bei der Überschreitung von Grenzwerten eine Aktion ausführen zu können, z.B. Lüften oder heizen zur Senkung der Luftfeuchtigkeit.
- Das System muss in der Lage sein Feinstaub zu messen, um bei der Überschreitung von Grenzwerten eine Aktion ausführen zu können.
- Das System muss in der Lage sein Temperaturdaten zu erfassen, um die Heizung oder Klimaanlage steuern zu können.
- Das System muss ich in der Lage sein akustische Daten zu erfassen, um Aktionen durch Grenzwertüberschreitungen ausführen zu können.
- Das System muss in der Lage sein durch Bewegungsmelder Bewegungen im Raum festzustellen um eine Aktion auszuführen zu können.
- Das System muss in der Lage sein Feuchtigkeitswerte der Erde (Blumentopf) zu messen um eine Aktion bei Erreichen eines Grenzwertes durchzuführen.
- Das System muss in der Lage sein Wetterdaten von einem Dienstleister zu beziehen, um darauf reagieren zu können.

2.2 Optionale Anforderungen

- Als Benutzer möchte ich die Möglichkeit haben die Daten von unterwegs einzusehen, um eventuell Konfigurationen anpassen zu können.
- Als Benutzer möchte ich in der Lage sein Geräte von unterwegs An- und Auszuschalten, um auf unvorhersehbare Ereignisse reagieren zu könne.
- Als Benutzer möchte ich in der Lage sein Diagramme zu Sensor ansehen zu können, um einen besseren Überblick zu erlangen.

- Das System soll mit einem Passwort gegen Eindringlinge geschützt werden, um die Sicherheit zu erhöhen.
- Das System soll auf Spracheingaben reagieren, um die Bequemlichkeit zu erhöhen.
- Als Benutzer möchte ich in der Lage sein, neue Geräte über die Oberfläche hinzuzufügen, um die Flexibilität zu steigern.
- Als Benutzer möchte ich, dass die Daten der Oberfläche sich live aktualisieren, um aktuelle Daten zu sehen.
- Als Benutzer möchte ich anderen Nutzern Rechte zu Geräten geben können, um auch Besuchern die Steuerung des Hauses zu ermöglichen.
- Als Benutzer möchte ich mich registrieren können, um das Haus steuern zu können.

3 Konzept

3.1 Systemarchitektur

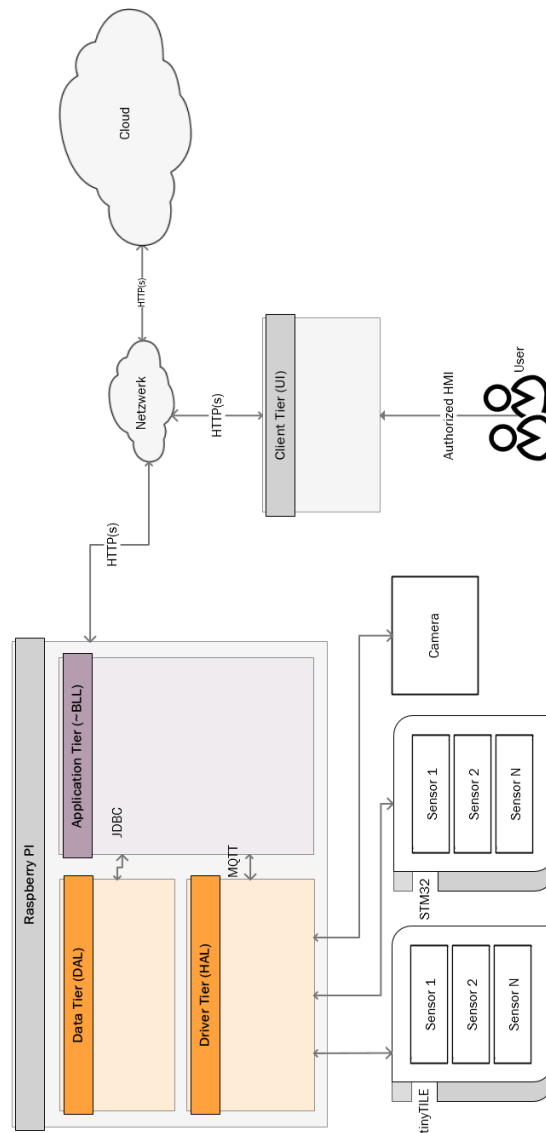
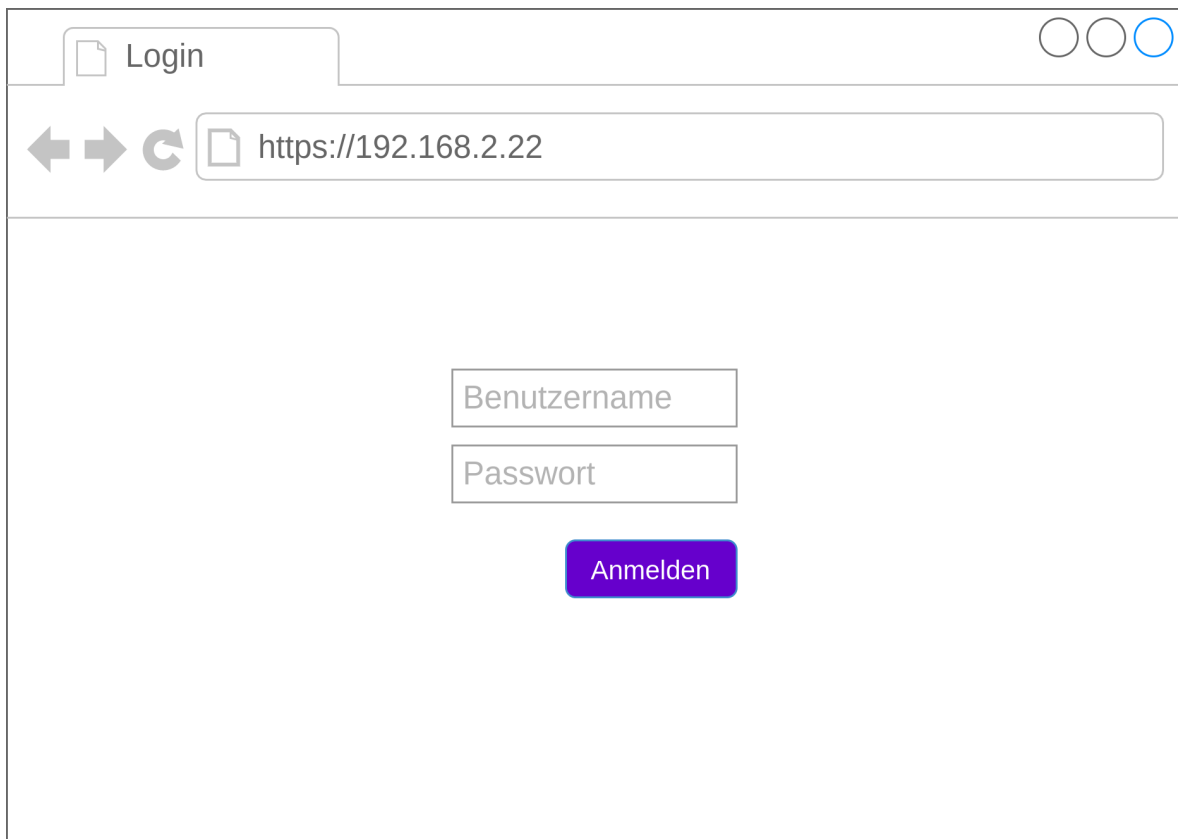


Figure 1: Zeigt die einfache Systemarchitektur des Projektes. Als zentrales Element gilt das Raspberry Pi. Hier läuft die Logik aller Applikationen, die Controller, Datenhaltung und HAL. Über die Treiber ist die Hardware zu erreichen. Eine Anbindung zur Cloud ist möglich und die Steuerung geschieht über ein Frontend im Browser.

Eine detaillierte Systemarchitektur befindet sich in den Anlagen. Dort ist zusätzlich zu erkennen, welche Technologie für die einzelnen *Tier* genutzt wird und welche Protokolle für die Kommunikation verwendet werden.

3.2 GUI Mockup

Soll die Konfiguration des Systems über die Weboberfläche geändert werden, so muss sich der Benutzer mit seinem Benutzernamen und seinem Passwort anmelden. In dem unten abgebildeten GUI-Mockup ist die Anmeldemaske zu sehen, in der der Benutzer seinen Benutzernamen und sein Passwort eingibt.



The image shows a web browser window with a single tab titled "Login". The address bar contains the URL "https://192.168.2.22". The main content area of the browser displays a login form centered on the page. The form consists of two text input fields: the top one is labeled "Benutzername" and the bottom one is labeled "Passwort". Below these fields is a blue button with the text "Anmelden" in white. The browser window has standard OS window controls (minimize, maximize, close) in the top right corner.

Figure 2: Der Login-Bildschirm des Frontend.

Das unten abgebildete GUI-Mockup stellt die Hauptseite der Weboberfläche dar. Auf dieser Seite befindet sich eine große Tabelle, die Sensoren, Bedingungen, Werte, Aktoren und verschiedene Steuerungselemente enthält. Jede Zeile in der Tabelle stellt eine Automatisierungs-Regel dar. Rechts neben der Tabellenüberschrift befindet ein Schalter mit dem alle Regel mit einem Klick ein- oder ausgeschaltet werden können. In der Spalte "Sensor" kann der Sensor ausgewählt werden, der in der jeweiligen Regel verwendet werden soll. Nachdem ein Sensor ausgewählt wurde, kann in der Spalte "Bedingung" eine der folgenden Bedingungen ausgewählt werden: Größer als, größer gleich, kleiner als, kleiner gleich, gleich und ungleich. Mit dieser Bedingung wird der Wert von den jeweiligen Sensoren mit dem in der Spalte "Wert" eingestellten Wert verglichen. Trifft die Bedingung zu, wird der in der Spalte "Aktor" eingestellte Aktor je nach eingestellter Aktion ein- oder ausgeschaltet. In der vorletzten Spalte können Regeln schnell deaktiviert werden, ohne die gesamte Regel löschen zu müssen. Rechts daneben befindet sich in jeder Zeile ein Button, mit dem die jeweilige Regel entfernt werden kann.

Unter der Tabelle "Automatische Steuerung" gibt es auf der rechten Seite zwei Buttons: "Regel hinzufügen" und "Speichern".

Weiter unten auf der Seite befinden sich zwei Tabellen. Die linke Tabelle enthält die aktuellen Daten zu allen angeschlossenen Sensoren. Die rechte Tabelle lässt den Benutzer alle angeschlossenen Aktoren manuell steuern.

Page 1

http://192.168.2.1

Automatische Steuerung EIN

#	Sensor	Bedingung	Wert	Aktor	Aktion	Aktiviert
1	Helligkeit	Kleiner als	200	Licht	Ein	<input checked="" type="checkbox"/>
2	Bewegungsmelder	Fester Wert	1	Licht	Ein	<input checked="" type="checkbox"/>
3	Luftfeuchte	Größer als	70	Fenster	Auf	<input type="checkbox"/>

Regel entfernen Regel entfernen Regel entfernen

Regel hinzufügen Speichern

Aktuelle Werte

Sensor	Wert
Helligkeit	430
Bewegungsmelder	0
Luftfeuchte	52

Manuelle Steuerung

Aktor	Steuerung
Licht	EIN
Fenster	EIN

Figure 3: Ein Beispiel für einen Konfigurationsbildschirm mit Datenanzeige.

3.3 Datenbank-Layout

Das Datenbank-Layout in Figure 4 beschreibt den Aufbau der Datenbank. Mithilfe der Datenbank können Automatisierungen abgebildet werden. Eine Automatisierung besteht aus Conditions und Actions. Conditions stehen in Verbindung mit einem ConditionType. Dieser definiert die Vergleichsoperation die mit der Condition ausgeführt wird. Außerdem steht die Condition noch in Beziehung mit einem Thing. Die Beziehung definiert den Sensor, der in der Condition im Zusammenhang mit dem ConditionType ausgewertet werden soll. Actions wiederum definieren alle Aktionen die durch Automatisierungen ausgeführt werden

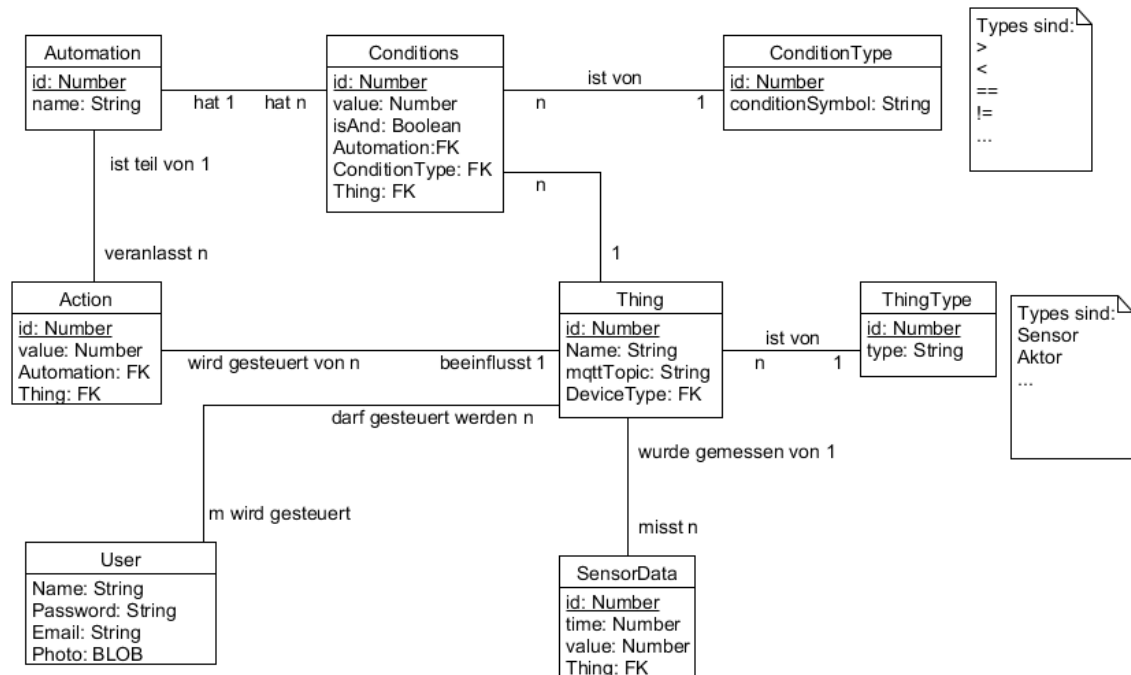


Figure 4: ER-Modell des Datenbank-Layouts.

sollen. Jede Aktion hat einen Wert, auf den ein bestimmtes Thing gesetzt werden soll. Eine Automatisierung veranlasst also eine Aktion mit einem Thing, wenn eine bestimmte Bedingungen erfüllt sind. Alle von einem Thing gemessenen Daten werden in der SensorData Tabelle gespeichert. Der ThingType definiert die Art des Things. Um eine Userverwaltung zu ermöglichen werden in der User-Tabelle alle Userdaten gehalten. Um Gästen den Zugriff auf bestimmte Geräte, nicht aber auf das gesamte System zu ermöglichen, besteht der User in einer Verbindung zu einem Thing. So kann für jeden User genau konfiguriert werden, welche Geräte er steuern darf, bzw. Sensoren er einsehen kann.

3.4 Use-Case-Diagramme

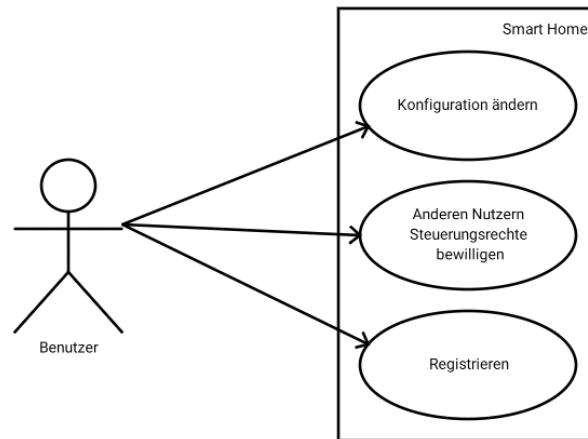


Figure 5: Der Benutzer muss sich auf der Oberfläche registrieren können. Als registrierter Benutzer muss er die Konfigurationen im Smart Home ändern und anderen Nutzern Steuerungsrechte bewilligen können.

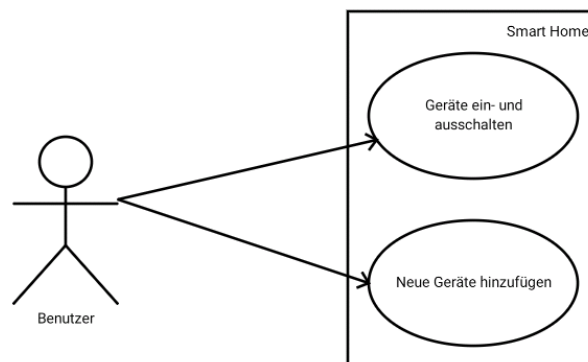


Figure 6: Der Benutzer kann in seinem Zuhause Geräte ein- und ausschalten. Dafür ist es auch sinnvoll, dass er neue Geräte hinzufügen kann. Zum Beispiel eine Klimaanlage oder eine Heizungssteuerung.

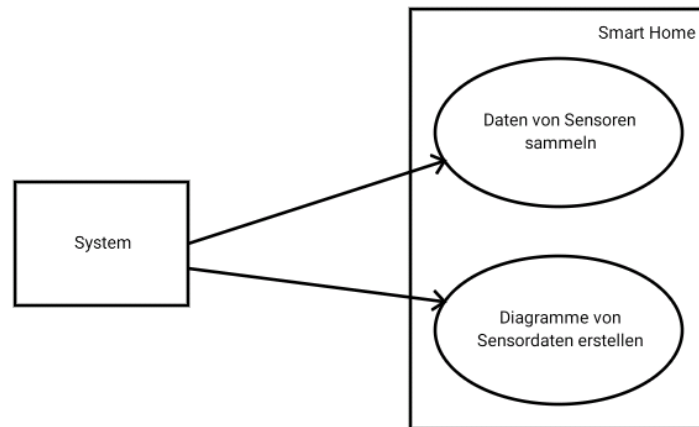


Figure 7: Das System muss Daten von Sensoren zum Beispiel Rauchmelder, Bewegungsmelder oder Feuchtigkeitsmesser sammeln. Mit den gesammelten Daten kann es Diagramme auf der Oberfläche erstellen können.

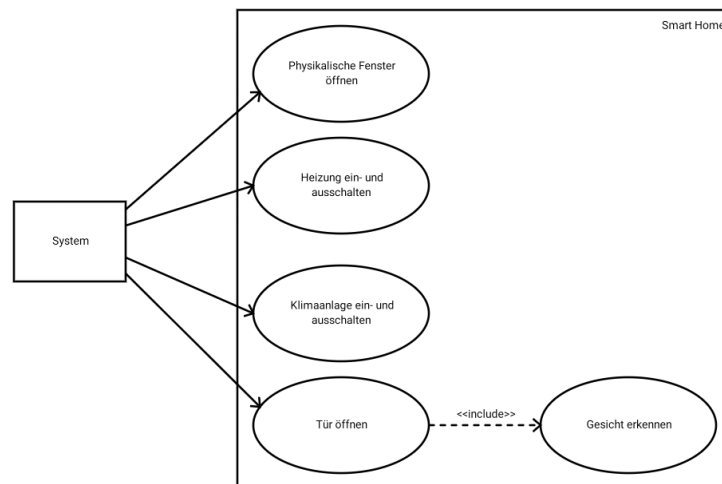


Figure 8: Um die Temperatur zu regeln muss das System in der Lage sein die Heizung ein- und ausschalten, die Fenster im Haus öffnen und die Klimaanlage ein- und ausschalten zu können. Zum Regeln der Feuchtigkeit muss das System ebenso die Fenster öffnen und die Klimaanlage ein- und ausschalten können. Wenn jemand einen abgesicherten Raum oder die Haustür öffnen möchte, muss das System die Tür per Gesichtserkennung öffnen können.

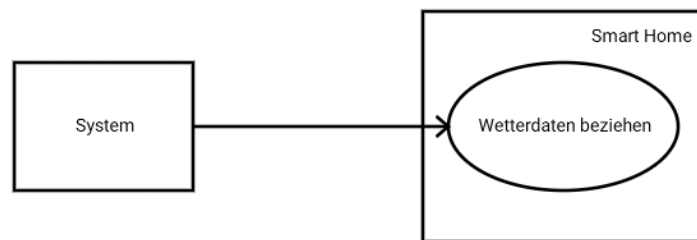


Figure 9: Das System muss Wetterdaten beziehen, um die Temperatur oder Feuchtigkeit im Haus zu regeln oder den Nutzer über das Wetter zu informieren.

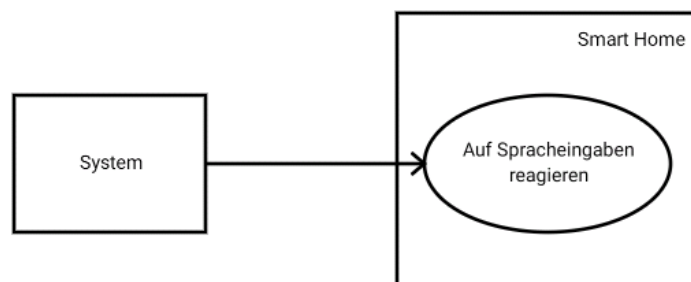


Figure 10: Das System soll auf Spracheingaben reagieren, um den Nutzer eine einfache Steuerung des Smart Homes oder das Beschaffen von Informationen (zum Beispiel Wetterdaten oder Innentemperatur) zu ermöglichen.

4 Projektplanung

Dieses Arbeitspaket umfasst die Erstellung des Pflichtenheftes zu dem Projekt.

Aufwand	20 h
Personen	Gamze Söylev Öktem, Nils Kohlmeier, Jonas Wiese, Wladimir Streck, Justin Jagieniak
Arbeitsmittel	Literatur, Internet
Ergebnisse	Pflichtenheft

4.1 Informationssammlung/-zusammenstellung

Dieses Arbeitspaket umfasst die Recherche nach Verfahren für SmartHome-Anwendungen, ihren Implementierungsmöglichkeiten auf Mikrocontrollern und Grundsaltungen zur Datenaufnahme sowie dem Cloud-Service.

Aufwand	48 h
Personen	Gamze Söylev Öktem, Nils Kohlmeier, Jonas Wiese, Wladimir Streck, Justin Jagieniak
Arbeitsmittel	Literatur, Internet
Ergebnisse	Stichpunktliste mit Realisierungsmöglichkeiten, Ordner mit relevanten Unterlagen, Grobkonzept für Hard- und Software

4.2 Bedienkonzept

Dieses Arbeitspaket umfasst den Entwurf des Bedienkonzeptes der SmartHome-Anwendung. Es wird festgelegt welche Bedienelemente dem Benutzer bereitgestellt werden und wie das System auf eingaben reagieren soll.

Aufwand	8 h
Personen	Gamze Söylev Öktem, Nils Kohlmeier, Wladimir Streck
Ergebnisse	Liste der Bedienelemente, stichpunktartige Beschreibung der Use-Cases und ein GUI Mockup

4.3 Hardware-Entwurf

Dieses Arbeitspaket umfasst den Entwurf der Hardware für das gesamte System.

Aufwand	5 h
Personen	Wladimir Streck, Nils Kohlmeier
Ergebnisse	Systemarchitektur und Softwarearchitektur

4.4 Software-Entwurf

Diese Arbeitspaket umfasst den Entwurf und der Implementierung der Software. Die Software wird in einzelne Module aufgeteilt und die Schnittstellen zwischen den Modulen werden definiert.

Aufwand	130 h
Personen	Gamze Söylev Öktem, Nils Kohlmeier, Jonas Wiese, Wladimir Streck, Justin Jagieniak
Ergebnisse	Programmelemente für die SmartHome-Anwendung.

4.5 Hardware-Aufbau

Dieses Arbeitspaket umfasst die Erstellung des Platinen-Layouts/Verbindung auf dem Breadboard.

Aufwand	10 h
Personen	Nils Kohlmeier
Arbeitsmittel	Datenblätter und Hardware
Ergebnisse	Schaltplan

4.6 Test des Gerätes

Dieses Arbeitspaket umfasst den Test der entwickelten Hard- und Software.

Aufwand	24 h
Personen	Gamze Söylev Öktem, Nils Kohlmeier, Jonas Wiese, Wladimir Streck, Justin Jagieniak
Ergebnisse	Dokumentation der Testergebnisse, Liste mit Mängeln

4.7 Fehlerbehebung

Dieses Arbeitspaket umfasst die Behebung der in 4.6 festgestellten Mängel.

Aufwand	40 h
Personen	Gamze Söylev Öktem, Nils Kohlmeier, Jonas Wiese, Wladimir Streck, Justin Jagieniak
Arbeitsmittel	Mängelliste, Entwicklungsumgebung für Mikrocontroller
Ergebnisse	Fehlerbereinigte Hard-/Software, Liste der verbleibenden Mängel

5 Meilensteine

Für die Abarbeitung des Projektes sind 16 Stunden pro Woche geplant.

Pflichtenheft fertig 15 Mai - 29 Mai

Zu diesem Meilenstein ist das Pflichtenheft fertiggestellt.

Entwurf fertig 23 Mai - 29 Mai

Zu diesem Meilenstein ist der Entwurf von Hard- und Software beendet.

Implementierung 29 Mai - 23 Juni

Zu diesem Meilenstein ist die gesamte Implementierung aller Teilkomponenten fertig zu stellen.

HAL

Feinstaubsensor + Treiber 5 Juni

Licht + Treiber 5 Juni

Feuer + Treiber 19 Juni

Lampe + Treiber 5 Juni

Audio + Treiber 5 Juni

Backend

JPA + Datenbank 5 Juni

REST Interface 5 Juni

Manuell 19 Juni

Automatisierung 23 Juni

(Alexa 23 Juni)

GUI

GUI Manuell 19 Juni

Gui Automatisierung 23 Juni

Gesichtserkennung 23 Juni

Zusammenführung 21 Juni - 25 Juni

Zu diesem Meilenstein werden alle Komponenten des Systems im Team zusammengeführt.

Funktionstest 23 Juni - 26 Juni

Zu diesem Meilenstein ist die Hard- und Software fertig aufgebaut und zum ersten mal getestet.

Präsentation 3 Juli

Zu diesem Meilenstein sind alle Arbeiten zu dem Projekt abgeschlossen und die Präsentation wird gehalten.

