

---

# KONZEPTION UND IMPLEMENTIERUNG VON SMARTHOME

---

Dokumentation  
vorgelegt am 19.01.2022  
WWI2019D  
Gruppe 2

Jan-Moritz Albrecht, Simon Fröhner, Tobias Hartmann, Marvin Kickel,  
Benjamin Leicht, Melissa Negele, Ion Tabyrca, Nicole Widmayer, Beeke Wiltfang

# Inhaltsverzeichnis

1.	Einleitung .....	1
2.	Zielsetzung des Projekts.....	3
3.	Projektorganisation.....	4
3.1.	Wahl der Projektleiter .....	4
3.2.	Lessons Learned aus einem vorangegangenen Projekt.....	5
3.3.	Projekt-Commitment.....	6
3.4.	Projektorganisation und Rollenfindung.....	7
3.5.	Projektmanagementmethode .....	9
3.6.	Ablaufdiagramm .....	10
3.7.	Gantt diagramm .....	11
3.8.	Kollaborationstools.....	11
4.	Anforderungsanalyse .....	16
4.1.	Funktionale Anforderungen .....	16
4.2.	Nichtfunktionale Anforderungen .....	16
4.3.	Liefer- und Abnahmebedingungen.....	17
4.4.	Systemmodelle .....	17
5.	Aufwandsschätzungen .....	34
5.1.	Methoden zur Aufwandsschätzung.....	34
5.1.1.	Planning Poker.....	34
5.1.2.	Drei-Zeiten-Methode.....	34
5.1.3.	Aufwandsschätzung Aufgabe 2.4.1 .....	35
5.1.4.	Aufwandsschätzung Aufgabe 2.4.2 .....	35
5.1.5.	Aufwandsschätzung Aufgabe 2.4.3 .....	36
6.	Ist-Aufwände.....	37
6.1.	Ist-Aufwand Aufgabe 2.4.1 .....	37
6.2.	Ist-Aufwand Aufgabe 2.4.2 .....	37
6.3.	Ist-Aufwand Aufgabe 2.4.3 .....	38
7.	IT-Architektur.....	39
7.1.	Raspberry Pi.....	39
7.2.	Hardwarearchitektur .....	40
7.3.	Softwarearchitektur .....	41
7.4.	SSH – Putty .....	42
7.5.	Shell .....	42

8. Sicherungskonzept .....	44
9. IT-Sicherheit .....	45
10. Implementierungsgrundlagen .....	46
10.1. Backendimplementierung .....	46
10.1.1. Devices .....	46
10.1.2. Hilfsmodule .....	48
10.1.3. Room .....	49
10.1.4. Perl-Skripte .....	50
10.2. Benutzerfreundliche Oberflächengestaltung .....	51
10.2.1. FTUI vs. FUIP .....	51
10.2.2. Visual Studio Code .....	52
10.2.3. Visuelle Darstellung .....	53
10.2.4. Hypertext Markup Language (HTML) .....	54
10.2.5. Cascading Style Sheets (CSS) .....	55
10.2.6. Layout .....	56
11. System-Test .....	60
11.1. Test 2.4.1 .....	60
11.2. Test 2.4.2 .....	61
11.3. Test 2.4.3 .....	62
12. Installationsanleitung .....	63
12.1. Schritt für Schritt Anleitung .....	63
12.2. Schnellinstallation .....	63
13. Abschließende Betrachtung .....	64
13.1. Lessons Learned .....	64
13.2. Fazit .....	66

# Abbildungsverzeichnis

Abbildung 1: What are the primary motivations for adopting the Smart Home technology .....	1
Abbildung 2: Projektablaufplan.....	10
Abbildung 3: Ausschnitt Conceptboard.....	13
Abbildung 4: Use Case Diagramm -a Weihnachtsbeleuchtung.....	17
Abbildung 5: Use Case Diagramm - Sturmweatherwarnung .....	18
Abbildung 6: Use Case Diagramm - Photovoltaik und Balkonkraftwerk Teil 1 .....	18
Abbildung 7: Use Case Diagramm - Photovoltaik und Balkonkraftwerk Teil 2 .....	19
Abbildung 8: Formel der Drei-Zeiten-Methode .....	34
Abbildung 9: Hardwarearchitektur .....	40
Abbildung 10: Softwarearchitektur .....	41
Abbildung 11: Device auf FHEM.....	47
Abbildung 12: FHEM Menüleiste.....	49
Abbildung 13: myUtilsTemplate Code.....	50
Abbildung 14: FTUI Beispiel.....	51
Abbildung 15: FUIP Beispiel .....	52
Abbildung 16: Beispiel für das Farbschema .....	54
Abbildung 17: Beispiel einer HTML-Datei .....	54
Abbildung 18: Beispiel eines HTML-Elements.....	55
Abbildung 19: Beispiel eines CSS Codes .....	55
Abbildung 20: Aufbau einer CSS-Datei .....	56
Abbildung 21: FTUI Gridster-Layout.....	57
Abbildung 22: FTUI Flex-Layout.....	57
Abbildung 23: FTUI Sheet-Layout .....	58
Abbildung 24: Umgesetztes Grund-Layot mit Gridster anhand der Sturmwarnung und Photovoltaik .....	58
Abbildung 25: Sheet-Layout innerhalb einer Kachel anhand der Weihnachtbeleuchtung.....	59

# Tabellenverzeichnis

Tabelle 1: Wahlergebnis des ersten Wahlgangs .....	4
Tabelle 2: Ergebnis des zweiten Wahlgangs .....	4
Tabelle 3: Gegenüberstellung Pros und Cons .....	5
Tabelle 4: Protokoll zur Akzeptanz des Commitments .....	6
Tabelle 5: Gantt diagramm .....	11
Tabelle 6: Aufwandsschätzung Aufgabe 2.4.1 .....	35
Tabelle 7: Aufwandsschätzung Aufgabe 2.4.2 .....	35
Tabelle 8: Aufwandsschätzung Aufgabe 2.4.3 .....	36
Tabelle 9: Ist-Aufwand Aufgabe 2.4.1 .....	37
Tabelle 10: Ist-Aufwand Aufgabe 2.4.2 .....	37
Tabelle 11: Ist-Aufwand Aufgabe 2.4.3 .....	38
Tabelle 12: Testprotokoll 2.4.1.....	60
Tabelle 13: Testprotokoll 2.4.2.....	61
Tabelle 14: Testprotokoll 2.4.3.....	62
Tabelle 15: Kontakt Support.....	63

# Abkürzungsverzeichnis

ARM	=	Acorn RISC Machines
CMD	=	command
CPAN	=	Comprehensive Perl Archive Network
CPU	=	Central Processing Unit
CSS	=	Cascading Style Sheets
CUL	=	CC1101 USB Lite (Radiofrequenz/ USB-Schnittstelle)
dd	=	disk dump
DHBW	=	Duale Hochschule Baden-Württemberg
FHEM	=	Freundliche Hausautomation und Energie-Messung
FTUI	=	FHEM Tablet UI
FUIP	=	FHEM User Interface Painter
HTML	=	Hypertext Markup Language
HTTP(S)	=	Hypertext Transfer Protocol (Secure)
IdD	=	Internet der Dinge
LED	=	Light Emitting Diode
MS	=	Microsoft
NAS	=	Network Attached Storage
OS	=	Operating System
SD	=	Secure Digital
SoC	=	System on a Chip
SSH	=	Secure Shell
TLS	=	Transport Layer Security
UI	=	User Interface
WLAN	=	Wireless Local Area Network

# 1. Einleitung

Smart Home und das Internet der Dinge (IdD) sind unwiderruflich miteinander verwoben. Das IdD ist derzeit einer der disruptivsten Trends, der sich auf zahlreiche traditionelle Branchen auswirkt. Die steigende Zahl der vernetzten Geräte und Sensoren kann über intelligente Algorithmen, wie z. B. maschinelle Lernwerkzeuge gesteuert werden und bildet zusammen das IdD. Die Nutzung des IdD im Kontext von Privathaushalten definiert ein Smart Home. Die Möglichkeiten, Prozesse im häuslichen Kontext zu automatisieren, sind nahezu unbegrenzt.

Laut einer Studie von Statista aus dem Jahre 2020 sind die primären Beweggründe zur Einführung von Smart-Home Technologien Sicherheit, Komfort, Zeitersparnis, Unterhaltung, Nachhaltigkeit und viele mehr (siehe Abbildung 1).

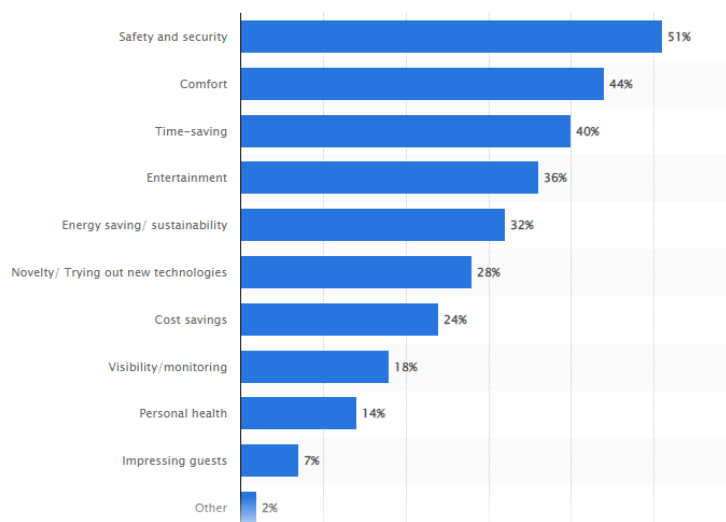


Abbildung 1: What are the primary motivations for adopting the Smart Home technology<sup>1</sup>

Diese Beweggründe und die unbegrenzten Möglichkeiten, Prozesse im häuslichen Kontext zu automatisieren, verleiten den Konsumierenden dazu, sich Smart-Home Technologien anzuschaffen. So sagt eine weitere Studie von Statista aus dem Jahr 2021, dass der globale Umsatz zwischen 2019 und 2026 um durchschnittlich 17,4% pro Jahr steigen soll. Es wird damit gerechnet, dass im Jahr 2026 global ein Umsatz von ca. 207,8 Mrd. US-Dollar generiert wird.<sup>2</sup> Betrachtet man diese Zahlen, so erkennt man, dass es für

<sup>1</sup> <https://www.statista.com/statistics/1113243/smart-home-technology-motivations-in-the-united-states-and-canada/>

<sup>2</sup> <https://www.statista.com/study/42112/smart-home-report/>

Unternehmen und angehende Mitarbeiter im IT-Sektor lukrativ sein könnte, sich mit dieser Thematik auseinanderzusetzen.

Im Rahmen des Wirtschaftsinformatik Studiums im Bereich Application Management an der DHBW Stuttgart, sollen sich Studierende mit dieser Thematik in Form eines mehrwöchigen Projekts auseinandersetzen.



## 2. Zielsetzung des Projekts

Die Zielsetzung des Projekts wurde in der initialen Anforderungsbeschreibung des Auftraggebers wie folgt beschrieben:

*„Ziel des Projektmoduls ist, die Themen Smart Home / Hausautomation konzeptionell näher zu beleuchten. Des Weiteren sollen mit dem Hausautomationsserver „FHEM“ ("Freundliche Hausautomation und Energiemessung", sprich "feem") entsprechende Automatisierungsszenarien technisch umgesetzt und dokumentiert werden. Als Laufzeitumgebung für FHEM wird vorab ein Raspberry Pi installiert, konfiguriert und ein Sicherheitskonzept erarbeitet. Abschließend werden die gemachten Erfahrungen sowie mögliche weitere Umsetzungsideen dokumentiert.“*

### 3. Projektorganisation

#### 3.1. Wahl der Projektleiter

Nachdem sich das Projektteam am 22. November 2021 gemeinsam mit der Zielsetzung des Projektes vertraut gemacht hatte, entschloss sich die Gruppe im Anschluss daran, eine Wahl der Projektleiter durchzuführen. Es sollten zwei Projektleiter auf der gleichen hierarchischen Ebene ernannt werden. Diese sollen beispielsweise Engpässe bei Abwesenheiten, etwa wegen Krankheit, unkompliziert und leicht abfangen können. Die Wahl wurde entsprechend der 5 Wahlgrundsätze aus Artikel 38 des Grundgesetzes durchgeführt.

Der erste Wahlgang brachte das folgende Ergebnis:

Beeke	Benjamin	Ion	Jan	Marvin	Melissa	Nicole	Simon	Tobias
1	2	7	4	1	0	0	1	2

**Tabelle 1: Wahlergebnis des ersten Wahlgangs**

Ion (7 Stimmen) nahm die Wahl zum Projektleiter an.

Jan (4 Stimmen) hat die Wahl zum Projektleiter abgelehnt.

Somit wurde ein zweiter Wahlgang zum Stechen zwischen Tobias und Benjamin durchgeführt.

Der zweite Wahlgang brachte das folgende Ergebnis:

Benjamin	Tobias
7	1

**Tabelle 2: Ergebnis des zweiten Wahlgangs**

Benjamin (7 Stimmen) nahm die Wahl zum Projektleiter an.

### 3.2. Lessons Learned aus einem vorangegangenen Projekt

Im Rahmen des Studiums wurde bereits im dritten Semester ein Projekt durchgeführt, bei dem die Erstellung einer Software das zentrale Ziel darstellte. Bevor wir mit dem neuen Projekt begannen, wollten wir das vergangene Projekt rekapitulieren. Die Erfahrungen des vorangegangenen Projektes sollen in das Neue einfließen und alle Projektmitglieder sowohl neu hinzugekommene als auch bestehende, über die positive und negative Erfahrung aufzugleisen.

Besonders interessant war, dass die Projektmitglieder im dritten Semester in unterschiedlichen Projektgruppen waren und daher von unterschiedlichen Erfahrungen erzählen konnten.

Die wichtigsten zusammengefassten Erfahrungen des vorangegangenen Projekts stehen im Folgenden:

Was lief gut?	Was kann verbessert werden?
<ul style="list-style-type: none"> <li>• Das Projektergebnis war gut</li> <li>• Effizienz der Gruppe war gut</li> <li>• Erst überlegen, dann machen!</li> <li>• Ausführliche Vorbereitungen</li> <li>• Gutes Zeitmanagement</li> <li>• Aufgabenverteilung via Kanban</li> <li>• Projektvorgehen (hybrid aus agiler und klassischer Vorgehensweise) hat gut funktioniert</li> </ul>	<ul style="list-style-type: none"> <li>• Kommunikation zur Projektleitung war schlecht</li> <li>• Anforderungen wurden teilweise zu detailliert diskutiert</li> <li>• Teilgruppen waren teilweise zu klein</li> <li>• Unterschiedliche Leistungs- und Lernbereitschaft</li> <li>• Zum Teil wurde zu viel Zeit auf unnötige Diskussionen verwendet</li> </ul>

Tabelle 3: Gegenüberstellung Pros und Cons

Aufgrund der Erfahrungen einigte sich das Projektteam auf die folgenden Stichpunkte, die bei diesem Projekt berücksichtigt werden sollten:

- Grundfunktionalität umsetzen, bevor über optionale Aufgaben diskutiert wird
- Sticker für anonymes Anmerken von Problemen (Einrichtung eines Kummerkastens)
- Nummerierung des Anhangs/der Dokumentation genauer prüfen
- Hybrides Projektvorgehensmodell beibehalten
- Aufgaben (mit Anpassungen im Vergleich zur Fallstudie) über Kanban dokumentieren
- Puffer in den Zeitplan mit einplanen

### 3.3. Projekt-Commitment

Beim vorangegangenen Projekt machten wir die Erfahrung, dass es unter den Teammitgliedern unterschiedliche Leistungs- und Lernbereitschaften geben kann. Dies ist bei neun Teammitgliedern mit jeweils eigenen Meinungen und Interessen selbstverständlich und zunächst nicht schlimm. Dies kann allerdings dazu führen, dass es im Laufe des Projekts zu Verstimmungen, Unzufriedenheiten und Diskussionen kommen kann, weil unterschiedliche Ansprüche, Motivationen und Interessen aufeinandertreffen. Um diesem negativen Effekt vorzubeugen, haben sich alle Teammitglieder über die Motivation und Leistungsbereitschaft Gedanken gemacht und darüber diskutiert und sich auf ein bestimmtes gemeinsames Level geeinigt. Dieses Commitment sollte schriftlich festgehalten und von jedem Teammitglied akzeptiert werden, um sich bei späteren Verstimmungen darauf zu berufen. Die Ergebnisse der Diskussion wurden durch die Projektleiter in einem „Commitment zur Leistungsbereitschaft“ aufbereitet und zusammengetragen. Dieses lautet folgendermaßen:

*„Das Interesse des Projektteams ist es, die volle Punktzahl zu erreichen. Mindestens alle im Vorlesungsplan stehenden Slots für das Projekt sollen vollständig genutzt werden. Alle Termine, die im Rahmen des Projektes notwendig sind, müssen besucht werden. Bei Nichterscheinen ist eine Entschuldigung im Voraus notwendig. Das Nichterscheinen befreit nicht von der Pflicht der Auskunft über den aktuellen Stand. Eine Auskunft wird von jedem erwartet, um Kapazitäten und Auslastungen zu erkennen. Es wird von allen Teammitgliedern der gleiche zeitliche Aufwand erwartet. Falls der Aufwand nicht im Gleichgewicht ist, sollte das gegenüber dem Team geäußert werden. In Anbetracht der Effizienz sollte der investierte Aufwand im gesunden Verhältnis zu den zur Verfügung stehenden 75h stehen.“*

Das obenstehende „Commitment zur Leistungsbereitschaft“ wurde allen Projektmitgliedern vorgestellt und jedes Projektmitglied stimmte diesem per Abstimmung über Microsoft Forms zu. Andernfalls wäre eine Anpassung des Commitments erforderlich, bis eine Version entsteht, der alle Mitglieder zustimmen könnten.

ID	Startzeit	Fertigstellungszeit	Name	Commitment
1	11.25.21 13:42:11	11.25.21 13:42:13	Simon Froehner	Ich stimme dem Vorschlag zu.
2	11.25.21 13:42:19	11.25.21 13:42:34	Benjamin Leicht	Ich stimme dem Vorschlag zu.
3	11.25.21 13:42:15	11.25.21 13:42:36	Marvin Kickel	Ich stimme dem Vorschlag zu.
4	11.25.21 13:42:41	11.25.21 13:42:45	Beeke Wiltfang	Ich stimme dem Vorschlag zu.
5	11.25.21 13:42:43	11.25.21 13:42:51	Jan-Moritz Albrecht	Ich stimme dem Vorschlag zu.
6	11.25.21 13:42:30	11.25.21 13:42:52	Ion Tabyrca	Ich stimme dem Vorschlag zu.
7	11.25.21 13:43:17	11.25.21 13:43:19	Tobias Hartmann	Ich stimme dem Vorschlag zu.
8	11.25.21 13:43:34	11.25.21 13:43:38	Melissa Negele	Ich stimme dem Vorschlag zu.
9	11.25.21 13:44:21	11.25.21 13:44:35	Nicole Widmayer	Ich stimme dem Vorschlag zu.

**Tabelle 4: Protokoll zur Akzeptanz des Commitments**

### 3.4. Projektorganisation und Rollenfindung

Aufgrund der gewählten agilen Projektmanagementmethode (siehe 3.5 Projektmanagementmethode) wurden den einzelnen Teammitgliedern keine festen Rollen zugeteilt, weil dies nicht dem agilen Ansatz gerecht werden würde. Andererseits sollte eine gewisse Einteilung erfolgen, da hinsichtlich der Komplexität mancher Bereiche nicht jeder auf jedem Gebiet tiefes Expertenwissen aufbauen kann, ohne dabei die Effizienzkomponente des „Commitment zur Leistungsbereitschaft“ zu verletzen.

Es wurde hinsichtlich der Rollenfindung, neben der Wahl der Projektleiter, lediglich eine Umfrage mit Mehrfachauswahl durchgeführt, bei der alle Teammitglieder gefragt wurden, in welchem Bereich man sich einbringen kann und möchte. Diese Bereiche stellen dabei keine starre Zuweisung dar, sondern sollen allen Projektmitgliedern transparent darstellen, wer in welchem Bereich Kompetenzen und Interessen hat, um zu jedem Thema möglichst schnell einen kompetenten Ansprechpartner zu erreichen.

Es wurden dafür vom Projektleiterteam die folgenden Projektbereiche vorab festgelegt:

- Projektmanagement
- Hardware (Raspberry Pi + Komponenten)
- Entwicklung Frontend (FTUI/FUIP)
- Entwicklung Backend/Logik (FHEM)
- Anforderungsmanagement
- Tutorial/Dokumentation
- Qualitätssicherung/ IT-Architekt

Diese Vorgehensweise wurde gewählt, da zu Beginn des Projekts nur schwer geschätzt werden konnte, wie viele Kapazitäten in welcher Rolle erforderlich sein könnten. Außerdem wurde sich dadurch erhofft, dass sich die Teammitglieder untereinander besser unterstützen können und es so zu weniger zeitlichem Verzug durch Abwesenheiten kommt.

Es wurde sich darauf geeinigt, dass sich jeder unabhängig von den angegebenen favorisierten Projektbereichen an der Implementierung des im Backend und/oder Frontend beteiligt. Das heißt, jedes Teammitglied ist dazu verpflichtet, mindestens ein funktionierendes Inkrement inkl. Dokumentation zu liefern.

Die Ergebnisse der Abstimmung über Microsoft Forms waren die Folgenden:

- Hardware (Raspberry Pi + Komponenten)
  - Simon Fröhner
  - Jan-Moritz Albrecht
  - Marvin Kickel
- Entwicklung Frontend (FTUI/FUIP)
  - Simon Fröhner
  - Nicole Widmayer
  - Beeke Wiltfang
  - Tobias Hartmann
  - Melissa Negele
  - Marvin Kickel
  - Ion Tabyrca
- Entwicklung Backend/Logik (FHEM)
  - Simon Fröhner
  - Beeke Wiltfang
  - Tobias Hartmann
  - Melissa Negele
  - Marvin Kickel
  - Ion Tabyrca
  - Benjamin Leicht
  - Jan-Moritz Albrecht
- Anforderungsmanagement
  - Nicole Widmayer
  - Melissa Negele
  - Marvin Kickel
  - Ion Tabyrca
  - Benjamin Leicht
  - Jan-Moritz Albrecht
- Tutorial - /Dokumentation
  - Simon Fröhner
  - Nicole Widmayer
  - Beeke Wiltfang
  - Benjamin Leicht
- Qualitätsbeauftragter/ IT-Architekt
  - Jan-Moritz Albrecht

Alle Projektbereiche wurden dem Aufwand entsprechend angemessen besetzt, sodass keine Nachsteuerung durch die Projektleiter erforderlich war. Die Liste diente als Nachschlagewerk, damit sich bei der Umsetzung der Anforderungen die Teams sowie die Ansprechpartner für die verschiedenen Bereiche schnellstmöglich gefunden werden können.

### 3.5. Projektmanagementmethode

Bei der Projektmanagementmethode hatte man die Wahl zwischen einem agilen Framework wie Scrum und einer klassischen Wasserfallmethode. Scrum ist ein Vorgehensmodell des Projekts- und Produktmanagement, insbesondere zur agilen Softwareentwicklung. Die Wasserfallmethode ist ein lineares Vorgehensmodell für das Projektmanagement, bei dem die Anforderungen zu Beginn des Projektes gesammelt werden. Anschließend wird ein sequenzieller Projektplan entworfen, der darauf abzielt, diese Anforderungen zu erfüllen. Unter Berücksichtigung der Tatsache, dass die Anforderungen zwar in der Aufgabenstellung feststanden, allerdings noch nicht durch uns als Auftragnehmer spezifiziert wurden, hatte man keine Planungssicherheit und konnte demnach keine klassische Wasserfallmethode einsetzen. Außerdem haben sich die Anforderungen im Laufe des Projektes geändert und es kamen optionale Anforderungen hinzu, welche mithilfe einer klassischen Wasserfallmethode nicht organisiert werden könnten.

Die Gruppe hat sich dazu entschieden einen agilen Ansatz zu wählen, bei dem man flexibler auf sich ändernde Anforderungen reagieren konnte und Kompetenzen projektübergreifend einsetzen konnte, sodass sich alle Mitarbeiter in allen Projektphasen einbringen konnten. Es wurde eine Abwandlung des agilen Frameworks Scrum ausgearbeitet. Die Länge eines Sprints betrug eine Woche. Innerhalb eines Sprints sollte das Team ein Inkrement erarbeiten. Dieses bestand nicht nur aus dem Code, sondern auch aus der dazugehörigen Schritt-für-Schritt Anleitung sowie der Dokumentation. Der Fortschritt des Projektes wurde durch sogenannte „Dailys“ geprüft. Aufgrund der hohen Auslastung in der Hochschule, war es uns nicht möglich tägliche Dailys zu veranstalten, sondern diese auf jeden Mittwoch und Freitag zu beschränken. Dabei hat jedes Projektmitglied die Möglichkeit gehabt, dem Team mitzuteilen, woran man gearbeitet hat, woran man arbeiten wird und an welcher Stelle es Probleme gibt. So konnte man Ressourcen effizient einsetzen, Aufgaben gezielt zuteilen und Impediments (Hindernisse) beseitigen. Jeden Montag fand das Sprint Review, die Retrospektive und das Sprint Planning statt. Dieser Termin diente dazu, das Ergebnis (Inkrement), welches innerhalb des Sprints erarbeitet wurde vor der gesamten Gruppe zu präsentieren. Während der Retrospektive hatte das Team die Möglichkeit über die Zusammenarbeit innerhalb der Gruppe zu sprechen. Dabei konnten positive als auch negative Aspekte angesprochen und diskutiert werden. Im Sprint Planning erarbeitete man die bevorstehenden To-Dos und teilte diese den jeweiligen Projektmitgliedern zu.

### 3.6. Ablaufdiagramm

Um die logischen Abläufe und Abhängigkeiten frühzeitig zu erkennen wurde ein Projektablaufplan erstellt, der die aufeinanderfolgenden Schritte aufzeigt. Dadurch kann jederzeit schnell erkannt werden, was die nächsten Schritte sind und welche Tätigkeiten abgeschlossen werden müssen, um den nächsten Projektschritt abzuschließen.

Der Projektablaufplan findet sich im Folgenden:

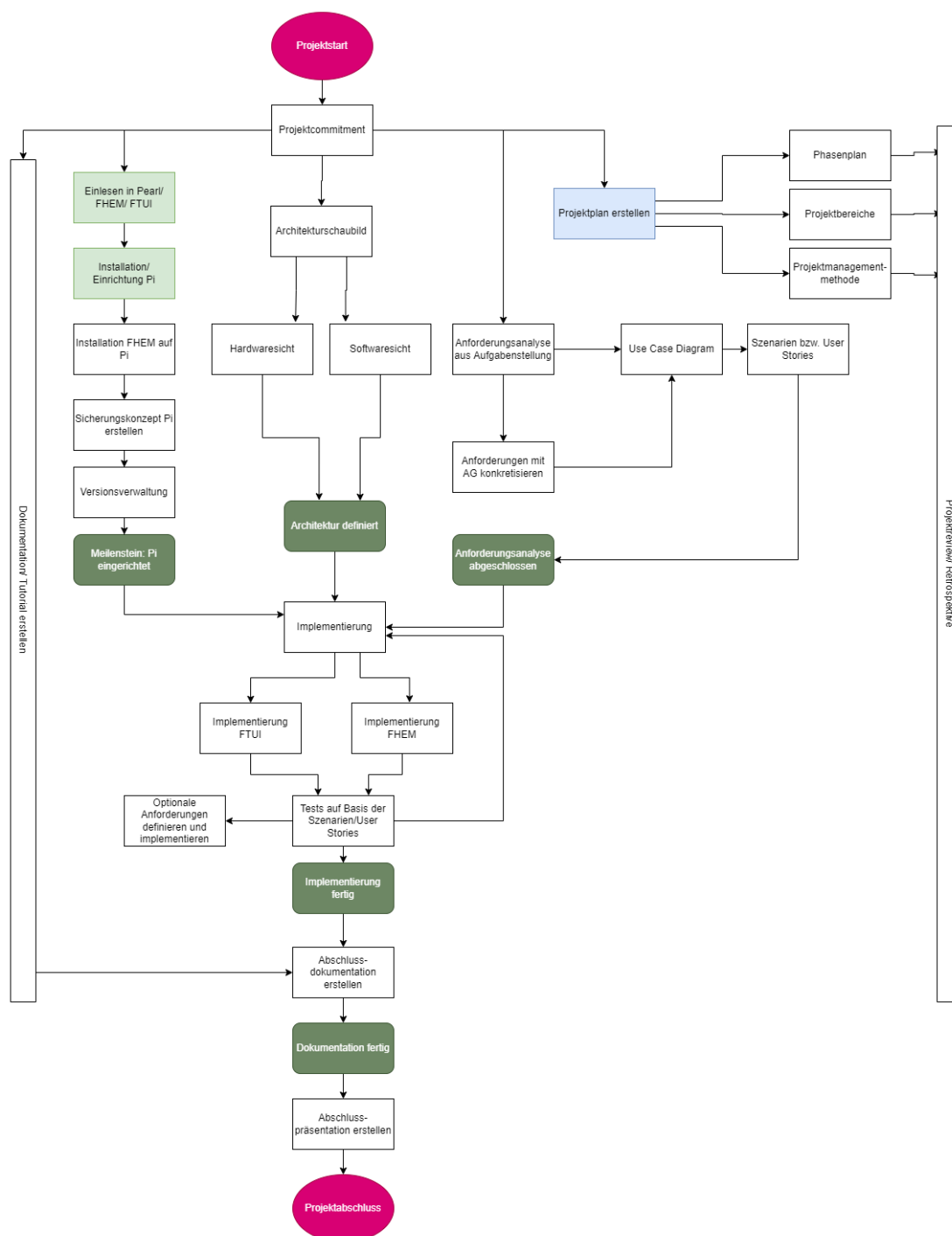


Abbildung 2: Projektablaufplan





### 3.8.1. Microsoft Teams

Microsoft Teams (abgekürzt MS Teams) ist eine von Microsoft entwickelte Kollaborationssoftware, die Chat, Besprechungen, Notizen, To-Dos und Anhänge kombiniert.

Im Rahmen unseres Projektes stellt Teams die zentrale Plattform zur Kommunikation dar. MS Teams wurde genutzt, um Besprechungen abzuhalten. Es wurde weiter ein Chat für das gesamte Team und für diverse, spezialisierte Teams eingerichtet. Außerdem wurde die private Chatfunktion genutzt. Darüber hinaus wurde MS Teams als zentraler Cloudspeicher zur Ablage, Versionierung und zur gemeinsamen Bearbeitung von Dateien genutzt. MS Teams ermöglichte beispielsweise das gemeinsame Bearbeiten der Projektdokumentation. Außerdem wurde die in MS Teams integrierte Funktion Microsoft Forms verwendet, um Umfragen und Abstimmungen durchzuführen. MS Teams stellte ein Kanban-Board zur Verfügung, über das man das Product Backlog und das Sprint Backlog realisieren und organisieren konnte. Außerdem konnte in MS Teams ein Conceptboard zum kollaborativen Arbeiten integriert werden (siehe 0).

Zu Beginn des Projektes wurde sichergestellt, dass jedes Teammitglied Microsoft Teams auf dem Rechner und Smartphone installiert, eingerichtet und Benachrichtigungen aktiviert hat. Dies diente dazu, dass jedes Teammitglied bei erforderlichen Abstimmungsbedarf schnellstmöglich, an jedem Ort, erreichbar ist.

### 3.8.2. Git

Git bzw. Github ist ein dezentrales Tool zu Verwaltung von Versionen. Da wir im Rahmen des Projektes gleichzeitig Oberflächen mit HTML, CSS und JavaScript entwickelt haben, war eine Verwaltung über ein dezentrales Versionsverwaltungssystem unabdingbar. Das Repository wurde zum Entwickeln an den privaten Rechnern eingerichtet. Zum Testen wurde das Git-Repository dann sowohl in den FHEM-Umgebungen der privaten Raspberry Pi's eingerichtet als auch in der „produktiven“ Umgebung. Das Repository konnte über Putty auf den Raspberry Pi's eingerichtet werden und der aktuelle Stand konnte regelmäßig auf dem Raspberry Pi in dem jeweiligen Verzeichnis aktualisiert werden.

Darüber hinaus wurde Github dazu verwendet die Config-Datei von FHEM des produktiven Raspberry Pi's automatisiert regelmäßig wegzusichern. Dies wurde als temporäres Sicherungskonzept genutzt, für den Fall, dass der Raspberry Pi wiederhergestellt werden muss.

### 3.8.3. Conceptboard

Conceptboard ist ein kollaboratives Online-Whiteboard, das weltweit von Teams als gemeinsamer, visueller Knotenpunkt für funktionsübergreifende Projekte wie Produktentwicklung, Design-Zusammenarbeit, agiles Projektmanagement und vieles mehr genutzt wird. Dieses wurde in MS Teams integriert, damit es für alle Teammitglieder schneller erreichbar ist.

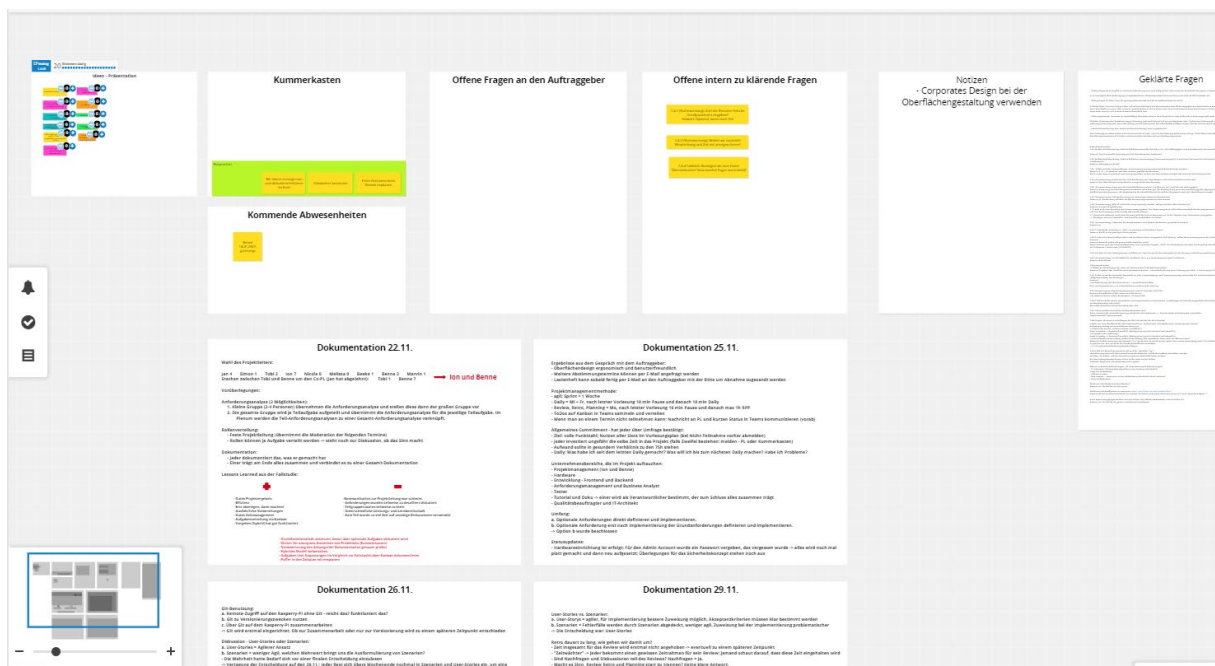


Abbildung 3: Ausschnitt Conceptboard

Das Conceptboard wurde im Rahmen des Projektes zu verschiedenen Zwecken verwendet.

Als erstes zu nennen sind die Protokolle, die dort über die gemeinsamen Besprechungen erstellt und abgelegt wurden, damit diese zu jederzeit an einer zentralen Stelle für alle Teammitglieder einsehbar sind.

Weiter wurden dort Bereiche angelegt, um intern zu klärende Fragen und Fragen, die mit dem Auftraggeber zu klären sind, zu sammeln und zu dokumentieren. So konnte sichergestellt werden, dass Fragen nicht doppelt gestellt wurden, da diese gesammelt und dem Auftraggeber über die Projektmanager gestellt wurden. Die Antworten auf die gestellten Fragen wurden auf dem Conceptboard so dokumentiert, dass diese jederzeit einsehbar waren.

Zusätzlich wurde das Conceptboard genutzt, um zur Implementierung, Dokumentation oder der Abschlusspräsentation Ideen zu sammeln und zu diskutieren.

Der vierte und letzte Anwendungsfall war die Einrichtung eines anonymen Kummerkastens. Dazu wurde ein Bereich eingerichtet, der für alle Projektmitglieder gedacht ist und welcher anonyme Kritik bspw. an der Vorgehensweise während den Terminen oder an der Projektleitung ermöglicht. Der anonyme Charakter der Kritik soll dazu führen, dass mehr Kritik eingebracht wird und diese ehrlicher und zielgerichteter ist. Die Kritik im Kummerkasten wurde vom Projektleitungsduo regelmäßig betrachtet und im Rahmen der Retrospektive wurden zusammen mit dem Team Lösungen zu den kritisierten Punkten erarbeitet. Die bereits besprochene Kritik, die schon zu Änderungen an der Umgangs- oder Herangehensweise geführt hat, wurde archiviert, um sichtbar zu machen, welche Optimierungen das Projektteam bereits erlangt hat und wie sich das Team im Laufe des Projektes weiterentwickelt.

### 3.8.4. Lokale FHEM Installationen

Um ein Verständnis für FHEM zu bekommen und erste Funktionalitäten mit Dummys umzusetzen, wurde FHEM von jedem Teammitglied auf dem privaten Rechner unter dem Betriebssystem Windows oder MacOS installiert. Dies wurde genutzt um in der Anfangsphase des Projektes, als die Anforderungen noch nicht vollständig definiert waren, praktisches Wissen anzueignen ohne auf die Installation und Einrichtung des „produktiven“ Raspberry Pi's angewiesen zu sein. Nachdem sich alle Teammitglieder mit FHEM auseinandergesetzt haben und die Anforderungsanalyse abgeschlossen war, wurden die Funktionalitäten in den lokalen FHEM Umgebungen implementiert und nachfolgend auf die privaten Raspberry Pi's zum Testen übertragen.

### 3.8.5. Private Pis

Zum Entwickeln und zum Testen der implementierten Funktionalitäten wurden von einigen Projektmitgliedern private Raspberry Pi's zur Verfügung gestellt. Diese beliefen sich auf eine Anzahl von 5 privaten Raspberry Pi's. So waren wir als Gruppe in der Lage unabhängig voneinander gleichzeitig zu entwickeln und zu testen. Nach erfolgreicher Umsetzung auf den privaten Raspberry Pi's wurden die Schritt-für-Schritt Anleitungen zur Einrichtung auf dem produktiven Raspberry Pi geschrieben und von anderen Teammitgliedern umgesetzt und getestet.

### 3.8.6. Apple Kalender

Um die regelmäßigen Termine wie die Dailys, die Sprint Reviews, Retrospektiven und Sprint Plannings nicht zu verpassen, wurde ein kollaborativer Online-Kalender eingerichtet. Da die Hochschul-Accounts unter Teams eingeschränkt waren und wir somit nicht die Möglichkeit hatten die Termine über diese zu managen, mussten wir nach einer Alternative suchen. Da der größte Teil des Teams im privaten Bereich mobile Endgeräte von Apple verwendet, hat es sich angeboten einen Online-Kalender über Apple „Kalender“ einzurichten. Diese Termine wurden direkt zu Beginn des Projektes eingestellt und konnte bei Änderung des Vorlesungsplans ohne viel Abstimmungsaufwand abgeändert werden.

## 4. Anforderungsanalyse

Bei der Anforderungsanalyse hat sich die Gruppe in kleine Gruppen aufgeteilt. Die Aufteilung erfolgte nach der Aufgabenstellung. Dabei haben sich die Projektmitglieder zunächst einmal mit den Anforderungen, welche aus dem Text hervorgingen, beschäftigt und Use-Case Diagramme und die dazugehörigen User-Stories definiert. Die Anforderungen unterteilen sich in funktionale und nichtfunktionale Anforderungen.

### 4.1. Funktionale Anforderungen

Funktionale Anforderungen sind vom Auftraggeber gewünschte Funktionalitäten bzw. das vom Auftraggeber gewünschte Verhalten eines Systems bzw. Produktes. Durch die funktionalen Anforderungen wird beschrieben, was das Endprodukt machen oder können soll. Die Summe aus den verschiedenen Anwendungsfällen stellt die Gesamtfunktionalität des Systems dar. Nachfolgend wird eine Übersicht aller funktionalen Anforderungen aufgelistet, welche in den Systemmodellen (siehe 4.4) detailliert beschrieben werden.

Aufgabenstellung 2.4.1	Umsetzung Weihnachtsbeleuchtung
Aufgabenstellung 2.4.2	Umsetzung Sturmwarnung
Aufgabenstellung 2.4.3	Photovoltaik / Balkonkraftwerk: Optimierung der Ausnutzung des erzeugten Stroms
Aufgabenstellung 2.4.4	Erstellung einer benutzerfreundlichen Oberfläche per FTUI / FUIP zur Visualisierung des SmartHome
Aufgabenstellung 2.4.5	Optionale Anforderungen

### 4.2. Nichtfunktionale Anforderungen

Die Smart-Home Anwendung soll über eine benutzerfreundliche Oberfläche per FTUI / FUIP dargestellt werden. Die Elemente sollen intuitiv bedienbar und anschaulich gestaltet werden. Das System soll in regelmäßigen Abständen eine Sicherung auf einen dezentralen „NAS-Server“ speichern und soll vor unberechtigten Zugriffen von außen geschützt sein. Die Verfügbarkeit und Schnelligkeit des Systems sind abhängig von dem Netzwerk, in dem das System integriert ist. Es wird allerdings bei der Implementierung angestrebt eine hohe Verfügbarkeit und Schnelligkeit zu erreichen.

### 4.3. Liefer- und Abnahmebedingungen

Am 19. Januar 2022 ist das Projekt mit den dazugehörigen Dokumenten dem Auftraggeber zu übergeben. Die Abschlusspräsentation des Projektes folgt am 26. Januar 2022. Nach erfolgreicher Abnahme gilt das Projekt als abgeschlossen.

### 4.4. Systemmodelle

Für das Erfassen der Anforderungen wurden Systemmodelle erstellt, welche die Entwickelnden bei der Implementierung unterstützen sollen. Bei den Systemmodellen wird zwischen den Use-Case-Diagrammen und den User Stories unterschieden. Diese werden nachfolgend aufgeführt.

#### 4.4.1. Use-Case-Diagramm

Für dieses Projekt wurden je Aufgabenstellung Use-Case-Diagramme erstellt, um die Übersichtlichkeit zu bewahren.

##### Aufgabenstellung 2.4.1 Weihnachtsbeleuchtung

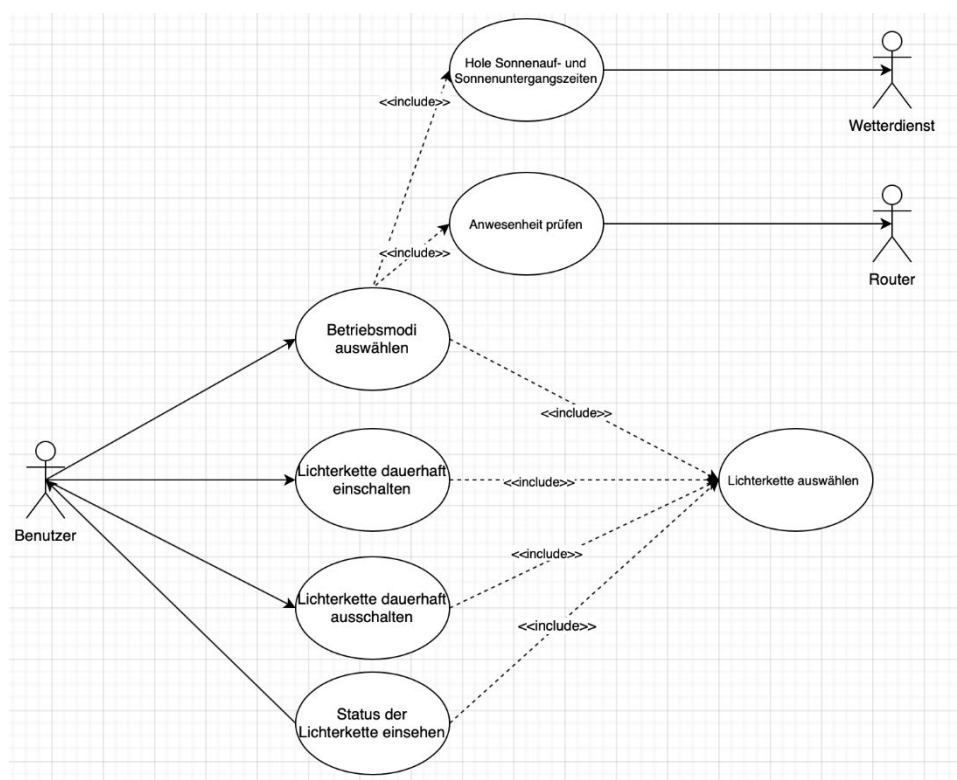


Abbildung 4: Use Case Diagram -a Weihnachtsbeleuchtung

### Aufgabenstellung 2.4.2 Sturmweatherwarnung

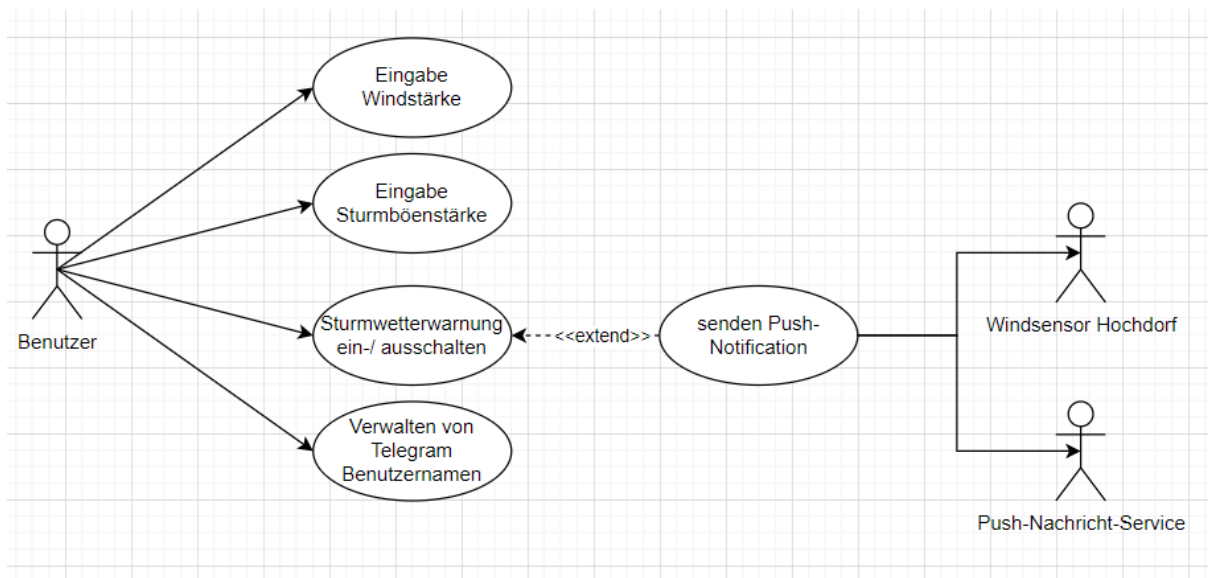


Abbildung 5: Use Case Diagramm - Sturmweatherwarnung

### Aufgabenstellung 2.4.3 Photovoltaik / Balkonkraftwerk

Aufgabenteil 2.4.3.1, 2.4.3.2 und 2.4.3.4:

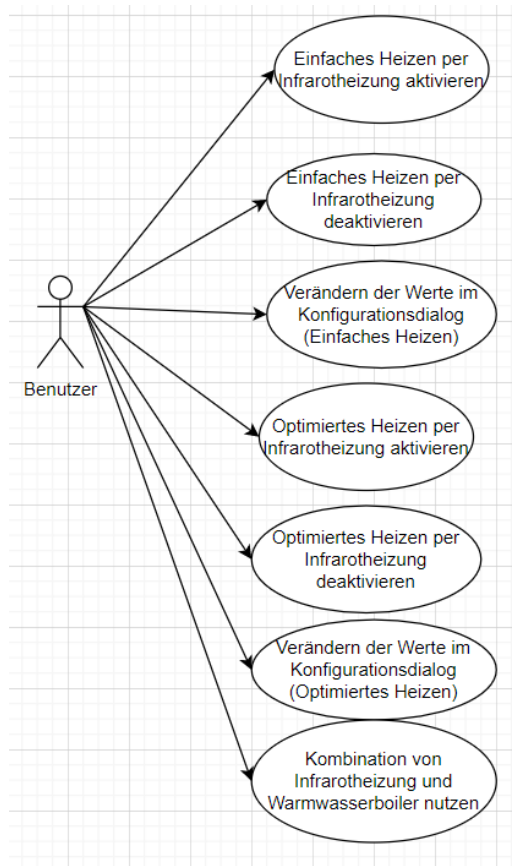


Abbildung 6: Use Case Diagramm - Photovoltaik und Balkonkraftwerk Teil 1



Aufgabenteil 2.4.3.3 und 2.4.3.5:

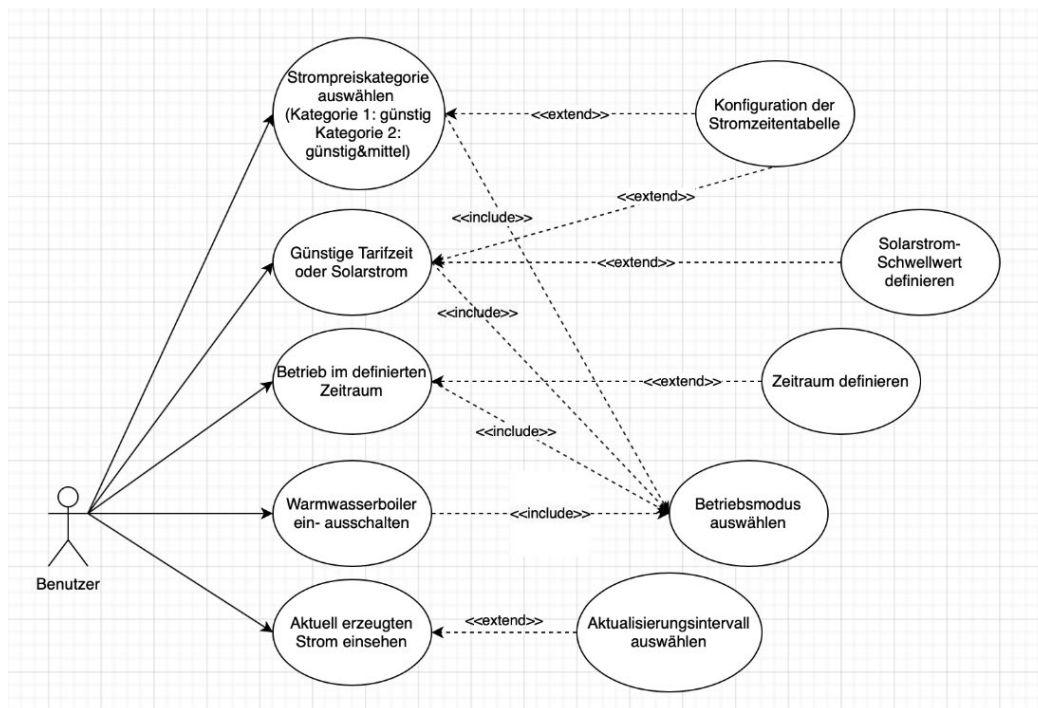


Abbildung 7: Use Case Diagramm - Photovoltaik und Balkonkraftwerk Teil 2

#### Aufgabenstellung 2.4.5 Optionale Anforderungen

Es wurde im Rahmen des Projektes eine optionale Anforderung umgesetzt. Diese umfasst die Umsetzung eines Müllplans. Dieser zeigt dem Nutzenden des Smart-Home-Systems, wann und welcher Müll abgeholt wird. Für die optionale Anforderung wurde kein Use-Case-Diagramm erstellt, da es dem Projektteam freistand diese umzusetzen und deswegen nicht an feste Anforderungen des Auftraggebers gebunden waren.

## 4.4.2. User-Stories

Für dieses Projekt wurden User-Stories formuliert. Diese werden je Aufgabenstellung dargestellt.

### 4.4.2.1. Aufgabenstellung 2.4.1 Weihnachtsbeleuchtung:

#### **Epic:**

Fest definierte Betriebszeit

#### **User Story:**

Als Benutzer möchte ich den Betriebsmodus „Fest definierte Betriebszeit“ auswählen, damit die Lichterketten zu einer gewünschten Betriebszeit in Betrieb sind.

#### **Akzeptanzkriterien:**

- Der Betriebsmodus schaltet die Lichterkette per Funksteckdose um 17 Uhr ein.
- Der Betriebsmodus schaltet die Lichterkette per Funksteckdose um 7 Uhr aus.
- Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen.
- Sobald die Lichterkette manuell bedient wird, entfällt der zuvor ausgewählte Betriebsmodus.

#### **Epic:**

Betriebszeit abhängig von Sonnenaufgang/Sonnenuntergang

#### **User-Story:**

Als Benutzer möchte ich den Betriebsmodus „Sonnenaufgang/Sonnenuntergang“ auswählen, damit am "Tag" keine Lichterkette in Betrieb ist.

#### **Akzeptanzkriterien:**

- Der Betriebsmodus schaltet per Funksteckdose die Lichterkette bei Sonnenaufgang aus.
- Der Betriebsmodus schaltet per Funksteckdose die Lichterkette bei Sonnenuntergang ein.
- Der Betriebsmodus ruft die Sonnenaufgangs- und Sonnenuntergangszeiten ab.
- Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen.
- Sobald die Lichterkette manuell bedient wird, entfällt der zuvor ausgewählte Betriebsmodus.

**Epic:**

Betriebszeit abhängig von Sonnenaufgang/Sonnenuntergang und der Uhrzeit an Werktagen

**User-Story:**

Als Benutzer möchte ich den Betriebsmodus „Sonnenaufgang/Sonnenuntergang und Uhrzeit an Werktagen“ auswählen, damit die Lichterketten unabhängig vom Sonnenaufgang an Werktagen spätestens ausgehen, wenn jeder das Haus verlassen hat.

**Akzeptanzkriterien:**

- Der Betriebsmodus schaltet per Funksteckdose die Lichterkette unabhängig von Sonnenaufgang an Werktagen (Montag - Freitag) spätestens um 7:30 Uhr aus.
- Der Betriebsmodus schaltet per Funksteckdose die Lichterkette bei Sonnenaufgang aus.
- Der Betriebsmodus schaltet per Funksteckdose die Lichterkette bei Sonnenuntergang ein.
- Der Betriebsmodus ruft die Sonnenaufgangs- und Sonnenuntergangszeiten ab.
- Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen.
- Sobald die Lichterkette manuell bedient wird, entfällt der zuvor ausgewählte Betriebsmodus.

**Epic:**

Betriebszeit abhängig von der Anwesenheitserkennung

**User-Story:**

Als Benutzer möchte ich den Betriebsmodus „Anwesenheitserkennung“ auswählen, um bei Abwesenheit zu verhindern, dass unnötig Lichterketten an sind.

**Akzeptanzkriterien:**

- Die Anwesenheit der Familienmitglieder wird über die WLAN-Verbindung des jeweiligen iPhones überprüft.
- Der Betriebsmodus schaltet per Funksteckdose die Lichterkette an Werktagen bei Anwesenheit zwischen 17 und 7 Uhr ein.
- Der Betriebsmodus schaltet per Funksteckdose die Lichterkette an Werktagen bei Abwesenheit zwischen 17 und 7 Uhr aus.
- Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen.
- Sobald die Lichterkette manuell bedient wird, entfällt der zuvor ausgewählte Betriebsmodus.

**Epic:**

Manueller Betrieb

**User-Story:**

Als Benutzer möchte ich die Lichterketten in FHEM unabhängig voneinander manuell ein- und ausschalten, um zu entscheiden, wann ich diese in Betrieb nehmen möchte.

**Akzeptanzkriterien:**

- Der Benutzer kann die Lichterkette per Funksteckdose manuell einschalten.
- Der Benutzer kann die Lichterkette per Funksteckdose manuell ausschalten.
- Sobald die Lichterkette manuell bedient wird, entfällt der zuvor ausgewählte Betriebsmodus.
- Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen.

**Epic:**

Statuseinsicht

**User-Story:**

Als Benutzer möchte ich den Status der Steckdosen in FHEM einsehen, um die Funktionsfähigkeit der Steckdosen zu überprüfen.

**Akzeptanzkriterien:**

- Der Benutzer kann den Status der Funksteckdose einsehen.
- Zwischen FHEM und CUL (Funksteckdose) besteht eine Verbindung.

**User-Story:**

Als Benutzer möchte ich den Status der Lichterketten in FHEM einsehen, um zu überprüfen, ob sie an- oder ausgeschaltet sind.

**Akzeptanzkriterien:**

- Der Benutzer kann den Status der Lichterkette per Funksteckdose einsehen.

#### 4.4.2.2. Aufgabenstellung 2.4.2 Umsetzung Sturmwarnung

**User-Story:**

Der Benutzer möchte die Windgeschwindigkeit auf der FTUI Oberfläche einsehen können, um die aktuelle Wetterlage zu erfahren.

**Akzeptanzkriterien:**

- Der Benutzer kann die Windgeschwindigkeit als Fließkommazahl einsehen.
- Zwischen FHEM und Windsensor besteht eine Verbindung.

**User-Story:**

Der Benutzer möchte die Windböe auf der FTUI Oberfläche einsehen können, um die aktuelle Wetterlage zu erfahren.

**Akzeptanzkriterien:**

- Der Benutzer kann die Windböe als Fließkommazahl einsehen.
- Zwischen FHEM und Windsensor besteht eine Verbindung.

**User-Story:**

Der Benutzer möchte den Schwellwert der Windgeschwindigkeit auf der FTUI Oberfläche eingeben, um eine Warnung bei schlechtem Wetter zu bekommen.

**Akzeptanzkriterien:**

- Der Benutzer muss die Windgeschwindigkeit als Fließkommazahl eingeben.

**User-Story:**

Der Benutzer möchte den Schwellwert der Windböen-Stärke auf der FTUI Oberfläche eingeben, um eine Warnung bei schlechtem Wetter zu bekommen.

**Akzeptanzkriterien:**

- Der Benutzer muss die Windböen-Stärke als Fließkommazahl eingeben.

**User-Story:**

Der Benutzer möchte die Sturmweatherwarnung einschalten, um bei Unwetter eine Push-Notification zu bekommen.

**Akzeptanzkriterien:**

- Die Sturmweatherwarnung kann aktiviert werden.

**User-Story:**

Der Benutzer möchte die Sturmweatherwarnung ausschalten, um keine Push-Notification mehr zu bekommen.

**Akzeptanzkriterien:**

- Die Sturmweatherwarnung ist aktiv.

**User-Story:**

Der Benutzer möchte seine Telegram Benutzernamen auf der FTUI Oberfläche einsehen, um zu wissen an welches Handy die Push-Notification gesendet wird

**Akzeptanzkriterien:**

- Der Benutzer kann die eingegebenen Telegram Benutzernamen einsehen.

**User-Story:**

Der Benutzer möchte Telegram Benutzernamen hinterlegen, um Warnungen zu erhalten

**Akzeptanzkriterien:**

- Der Benutzer kann Telegram Benutzernamen eingeben.

**User-Story:**

Der Benutzer möchte Telegram Benutzernamen löschen, um keine Warnungen zu erhalten

**Akzeptanzkriterien:**

- Der Benutzer kann Telegram Benutzernamen löschen.

**User-Story:**

Der Benutzer möchte, dass das FHEM beim Erreichen eines der Schwellenwerte eine Push-Notification an die Handynummern sendet.

**Akzeptanzkriterien:**

- Zwischen FHEM und Push-Notification-Server besteht eine Verbindung.
- Zwischen FHEM und Windsensor besteht eine Verbindung.
- Zwischen FHEM und Windsensor besteht eine Verbindung.
- Der gewünschte Telegram Benutzername ist hinterlegt.



#### 4.4.2.3. Aufgabenstellung 2.4.3 Photovoltaik / Balkonkraftwerk: Optimierung der Ausnutzung des erzeugten Stroms

**Epic:**

Einfaches Heizen per Infrarotheizung mit erzeugtem Solarstrom

**User-Story:**

Der Benutzer möchte das einfache Heizen per Infrarotheizung aktivieren, um Heizen zu können.

**Akzeptanzkriterien:**

- Das Balkonkraftwerk muss mehr als der hinterlegte Wert an Strom produzieren.
- Der Überhitzungsschutz muss aus sein.
- Das einfache Heizen muss aktivierbar sein.

**User-Story:**

Der Benutzer möchte das einfache Heizen per Infrarotheizung deaktivieren, um das Heizen abzuschalten.

**Akzeptanzkriterien:**

- Das einfache Heizen muss deaktivierbar sein.

**User-Story:**

Der Benutzer möchte die Werte im Konfigurationsdialog verändern, um die Initialwerte zu verändern.

**Akzeptanzkriterien:**

- Der Benutzer darf nur numerische Werte eingeben, um den Initialwert zu verändern.
- Der Benutzer kann die Initialwerte einsehen.

**Epic:**

Optimiertes Heizen per Infrarotheizung mit erzeugtem Solarstrom

**User-Story:**

Der Benutzer möchte das optimierte Heizen per Infrarotheizung aktivieren, um Strom zu sparen.

**Akzeptanzkriterien:**

- Das optimierte Heizen muss aktivierbar sein.
- Die Verbindung zwischen Temperatur-Sensor und FHEM muss aktiv sein.

**User-Story:**

Der Benutzer möchte das optimierte Heizen per Infrarotheizung deaktivieren, um selbständig zu entscheiden, wann das optimierte Heizen in Betrieb genommen wird.

**Akzeptanzkriterien:**

- Das optimierte Heizen muss deaktivierbar sein.

**User-Story:**

Der Benutzer möchte die Werte im Konfigurationsdialog verändern, um die Initialwerte zu verändern.

**Akzeptanzkriterien:**

- Der Benutzer darf nur numerische Werte eingeben, um den Initialwert zu verändern.
- Der Benutzer kann die Initialwerte einsehen.

**Epic:**

Optimierter Betrieb eines elektrischen Warmwasserboilers mit erzeugtem Solarstrom

a) Manuelle Bedienung:

**User-Story:**

Als Benutzer möchte ich den Warmwasserboiler über die FTUI manuell ein- und ausschalten können, um selbständig zu entscheiden, wann dieser in Betrieb sein soll.

**Akzeptanzkriterien:**

- Der Benutzer kann den Warmwasserboiler manuell einschalten.
- Der Benutzer kann den Warmwasserboiler manuell ausschalten.
- Der Benutzer kann den Betriebsmodus durch die Inbetriebnahme eines anderen Betriebsmodus deaktivieren.

b) Zeitraum:

**User-Story:**

Als Benutzer möchte ich den Warmwasserboiler für einen fest definierten Zeitraum in Betrieb nehmen, um ihn für den benötigten Benutzungszeitraum nicht manuell ein- bzw. ausschalten zu müssen.

**Akzeptanzkriterien:**

- Der Benutzer kann den Betriebsmodus auswählen.
- Der Benutzer muss den Zeitraum zur Inbetriebnahme definieren/festlegen.
- Der Warmwasserboiler wird zum definierten Endzeitpunkt automatisch ausgeschaltet.
- Der Warmwasserboiler wird zum definierten Startzeitpunkt automatisch eingeschaltet.
- Der Benutzer kann den Betriebsmodus durch die Inbetriebnahme eines anderen Betriebsmodus deaktivieren.
- Nach Beendigung des definierten Zeitraumes ist der Betriebsmodus nicht mehr aktiv.

c) Stromtarifabhängig:

**User-Story:**

Als Benutzer möchte ich den Warmwasserboiler nur zu gewünschten Stromtarifen in Betrieb nehmen, um Stromkosten zu sparen.

**Akzeptanzkriterien:**

- Der Benutzer kann den Betriebsmodus auswählen.
- Der Benutzer muss den definierten Stromzeiten in der Stromzeitentabelle eines der folgenden Preiskategorien zuweisen (günstiger/mittlerer/teuerster Preis pro KWh).
- Der Benutzer muss die Stromzeiträume definieren können.
- Die Stromzeitenwerte müssen initial, wie in der Aufgabenstellung, hinterlegt sein.
- Der Benutzer kann bei Inbetriebnahme des Betriebsmodus zwischen den Tarifzeiten "günstigster Strompreis" und "günstigster/mittlerer Strompreis" auswählen.
- Der Benutzer kann den Betriebsmodus durch die Inbetriebnahme eines anderen Betriebsmodus deaktivieren.
- Der Warmwasserboiler schält sich automatisch ein, wenn die ausgewählte Tarifzeit durch die Angaben in der Stromzeitentabelle eintritt.
- Der Warmwasserboiler schält sich automatisch außerhalb der ausgewählten Tarifzeit aus (siehe Tabelle).

d) Stromtarif- und Solarstromabhängigkeiten:

**User-Story:**

Als Benutzer möchte ich den Warmwasserboiler nur zu günstigen Stromtarifzeiten oder mit Solarstrom in Betrieb nehmen können, um Stromkosten zu sparen.

**Akzeptanzkriterien:**

- Der Benutzer kann den Betriebsmodus auswählen.
- Der Benutzer muss die Stromzeiträume in (Stunden/Minuten) definieren können.
- Die Stromzeitenwerte müssen initial, wie in der Aufgabenstellung, hinterlegt sein.
- Der Benutzer muss den definierten Stromzeiten in der Stromzeitentabelle eines der folgenden Preiskategorien zuweisen (günstiger/mittlerer/teuerster Preis pro KWh).
- Der Warmwasserboiler schaltet sich automatisch nach der Stromzeitentabelle bei "günstiger Stromzeit" oder beim Erreichen des definierten Schwellenwerts für Solarstrom ein.
- Der Warmwasserboiler schaltet sich automatisch nach der Stromzeitentabelle bei nicht "günstiger Stromzeit" oder bei Unterschreiten des Solarstromschwellwerts aus.
- Der Benutzer muss den Schwellenwert für den Solarstrom als Einschaltpunkt für den Warmwasserboiler definieren.
- Im Intervall von 5 Minuten wird der erzeugte Solarstrom geprüft und mit dem Vergleichswert zur Einschaltung verglichen.
- Der Initialwert für die Einschaltung des Warmwasserboiler beträgt > "400" Watt Solarstrom.
- Der Benutzer kann den Betriebsmodus durch die Inbetriebnahme eines anderen Betriebsmodus deaktivieren.

**Epic:**

Kombination zwischen Infrarotheizung und Warmwasserboiler:

**User-Story:**

Als Benutzer möchte ich, dass der Boiler und die Heizung mit einer Priorisierung Solarenergie beziehen.

**Akzeptanzkriterien:**

- Wenn die Heizung in der Überhitzungsschutzphase ist, wird der Boiler gespeist werden.
- Zwischen 5 und 8 Uhr (Duschzeit) fließt Energie in den Boiler, solange niemand die Heizung nutzt. Heizung kann mittels Taster auf Oberfläche für 2 Minuten aktiviert werden. (Quick Shower)

**Epic:**

Visualisierung des durch Balkonkraftwerk erzeugten Stroms auf der FTUI:

a) Statistische Auswertung:

**User-Story:**

Als Benutzer möchte ich eine statistische Auswertung des erzeugten Stroms auf der FTUI-Oberfläche einsehen, um die Produktivität meines Balkonkraftwerks zu überprüfen.

**Akzeptanzkriterien:**

- Der Benutzer soll die Daten in einer übersichtlichen Darstellung einsehen können.
- Das Aktualisierungsintervall für den aktuell erzeugten Strom soll in Minuten konfigurierbar sein.
- Das Aktualisierungsintervall soll initial auf 1 Minute hinterlegt sein.
- Folgende Daten sollen in der Darstellung ersichtlich werden: Aktuell produzierter Strom, Aktueller Tag/Vorhergehender Tag, Aktuelle Woche/Vorhergehende Woche, Aktueller Monat/Vorhergehender Monat, Aktuelles Jahr/Vorhergehendes Jahr.
- Die Daten von "Aktuell produzierter Strom" und "aktueller Tag" werden in dem definierten Aktualisierungsintervall aktualisiert.

b) Visuelle Darstellung: Balkendiagramm:

**User-Story:**

Als Benutzer möchte ich eine visuelle Darstellung in Form eines Balkendiagramms des erzeugten Stroms auf der FTUI-Oberfläche einsehen, um die Produktivität meines Balkonkraftwerks graphisch abgleichen zu können.

**Akzeptanzkriterien:**

- Der Benutzer soll die Daten in einer übersichtlichen Darstellung in Form eines Balkendiagramms einsehen können.
- Jeder Balken stellt einen einzelnen Tag dar, ausgenommen "Aktuelles Jahr" und "Vorhergehendes Jahr". Hier stellt jeder Monat einen Balken dar.
- Folgende Daten sollen in der Darstellung ersichtlich werden: Aktuelle Woche/Vorhergehende Woche, Aktueller Monat/Vorhergehender Monat, Aktuelles Jahr/Vorhergehendes Jahr.

## 5. Aufwandsschätzungen

### 5.1. Methoden zur Aufwandsschätzung

Es stehen diverse Methoden zur Schätzung von Aufwänden zur Verfügung. Im Rahmen des Projektes wurden die Aufwände zunächst mittels Planning Poker und der gewichteten Drei-Zeiten-Methode geschätzt. Durch die Kombination der Methoden erhoffte man sich auf der einen Seite eine präzisere Schätzung der Aufwände, um eine bessere Planbarkeit zu erlangen, auf der anderen Seite bestand ein Interesse des Projektteams innerhalb eines universitären Umfelds unterschiedliche Schätzungsmethoden kennenzulernen. Die Schätzungen konnten den tatsächlich aufgewendeten Aufwänden im Anschluss gegenübergestellt und mit diesen verglichen werden.

Es wurden lediglich zu den Funktionalen Anforderungen Aufwandsschätzungen sowie Erfassungen der Ist-Aufwände angestellt. Die Erstellung des Frontends, die in 2.4.4 beschrieben wird, wird nicht gesondert ausgewiesen, sondern wird bei den Teilaufgaben entsprechend der jeweiligen Funktionalitäten berücksichtigt. Die geschätzten Aufwände werden im Folgenden jeweils in Stunden angegeben.

#### 5.1.1. Planning Poker

Bei Planning Poker geben die Mitglieder der Gruppe Schätzungen ab, indem sie nummerierte Karten verdeckt auf den Tisch legen, anstatt sie laut auszusprechen. Der Zahlenbereich der Karten entspricht der Fibonacci-Folge. Die Karten werden gleichzeitig aufgedeckt, und die Schätzungen werden anschließend diskutiert. Indem die Zahlen auf diese Weise versteckt werden, kann die Gruppe die kognitive Verzerrung der Verankerung vermeiden, bei der die erste laut gesprochene Zahl einen Präzedenzfall für die nachfolgenden Schätzungen schafft.

#### 5.1.2. Drei-Zeiten-Methode

Bei der gewichteten Drei-Zeiten-Methode wird der Best Case (BC), der Worst Case (WC) und der Most Likely Case (LC) geschätzt. Der Most Likely Case wird mit dem Faktor 4 multipliziert. Die sich aus diesem und dem Best Case sowie Worst Case ergebende Summe wird anschließend durch 6 dividiert. Dieser Wert stellt die Aufwandsschätzung (A) mithilfe der Drei-Zeiten-Methode dar.

$$A_i = \frac{(BC_i + 4 \times LC_i + WC_i)}{6}$$

Abbildung 8: Formel der Drei-Zeiten-Methode



### 5.1.3. Aufwandsschätzung Aufgabe 2.4.1

User Stories	Planning Poker	Best Case	Most Likely	Worst Case	Drei- Zeiten- Methode
Als Benutzer möchte ich den Betriebsmodus „Fest definierte Betriebszeit“ auswählen, damit die Lichterketten zu einer gewünschten Betriebszeit in Betrieb sind.	5,8	8,0	13,0	21,0	13,5
Als Benutzer möchte ich den Betriebsmodus „Sonnenaufgang/Sonnenuntergang“ auswählen, damit tagsüber keine Lichterkette in Betrieb ist.	8,4	3,0	8,0	13,0	8,0
Als Benutzer möchte ich den Betriebsmodus „Sonnenaufgang/Sonnenuntergang und Uhrzeit an Werktagen“ auswählen, damit die Lichterketten unabhängig vom Sonnenaufgang an Werktagen spätestens ausgehen, wenn jeder das Haus verlassen hat.	3,0	2,0	5,0	8,0	5,0
Als Benutzer möchte ich den Betriebsmodus „Anwesenheitserkennung“ auswählen, um bei Abwesenheit zu verhindern, das unnötig Lichterketten an sind.	13,2	5,0	8,0	13,0	8,3
Als Benutzer möchte ich die Lichterketten in FHEM unabhängig voneinander manuell ein- und ausschalten, um zu entscheiden, wann ich diese in Betrieb nehmen möchte.	2,4	1,0	3,0	8,0	3,5
Als Benutzer möchte ich den Status der Lichterketten in FHEM einsehen, um zu überprüfen ob sie an- oder ausgeschaltet sind.	4,0	1,0	3,0	8,0	3,5
<b>Gesamtaufwand</b>	<b>36,8</b>				<b>41,8</b>

Tabelle 6: Aufwandsschätzung Aufgabe 2.4.1

### 5.1.4. Aufwandsschätzung Aufgabe 2.4.2

User Stories	Planning Poker	Best Case	Most Likely	Worst Case	Drei- Zeiten- Methode
Der Benutzer möchte die Windgeschwindigkeit auf der FTUI Oberfläche einsehen können, um die aktuelle Wetterlage zu erfahren	8,0	5,0	8,0	13,0	8,3
Der Benutzer möchte die Windböe auf der FTUI Oberfläche einsehen können, um die aktuelle Wetterlage zu erfahren	1,0	1,0	2,0	5,0	2,3
Der Benutzer möchte den Schwellwert der Windgeschwindigkeit auf der FTUI Oberfläche eingeben, um eine Warnung zu bekommen bei schlechtem Wetter	3,0	3,0	5,0	8,0	5,2
Der Benutzer möchte den Schwellwert der Windböen-Stärke auf der FTUI Oberfläche eingeben, um eine Warnung zu bekommen bei schlechtem Wetter	1,0	1,0	3,0	8,0	3,5
Der Benutzer möchte die Sturmweatherwarnung einschalten, um bei Unwetter eine Push-Notification zu bekommen	2,0	1,0	2,0	3,0	2,0
Der Benutzer möchte die Sturmweatherwarnung ausschalten, um keine Push-Notification mehr zu bekommen	1,0	1,0	2,0	3,0	2,0
Der Benutzer möchte seine Telegram Benutzernamen auf der FTUI Oberfläche einsehen, um zu wissen an welches Handy die Push-Notification gesendet wird	4,0	2,0	3,0	8,0	3,7
Der Benutzer möchte Telegram Benutzernamen hinterlegen, um Warnungen zu erhalten	4,0	2,0	5,0	8,0	5,0
Der Benutzer möchte Telegram Benutzernamen löschen, um keine Warnungen zu erhalten	2,0	2,0	3,0	5,0	3,2
Der Benutzer möchte, dass das FHEM beim Erreichen eines der Schwellenwerte eine Push-Notification über Telegram sendet	8,0	3,0	5,0	8,0	5,2
<b>Gesamtaufwand</b>	<b>34,0</b>				<b>40,3</b>

Tabelle 7: Aufwandsschätzung Aufgabe 2.4.2

### 5.1.5. Aufwandsschätzung Aufgabe 2.4.3

Bei der Aufwandsschätzung für die Aufgabe 2.4.2 hat sich das Projektmanagement aus diversen Gründen dazu entschieden lediglich die Drei-Zeiten-Methode einzusetzen. Es hat sich im Laufe des Projekts herausgestellt, dass die Drei-Zeiten-Methode nicht nur präziser, sondern auch aufwandsärmer war. So brauchte man für die Schätzung beim Planning Pokers mindestens 3 Personen, für die Schätzung der Drei-Zeiten-Methode reichte lediglich ein Schätzer.

User Stories	Best Case	Most Likely	Worst Case	Drei-Zeiten-Methode
Der Benutzer möchte das einfache Heizen per Infrarotheizung aktivieren, um Heizen zu können.	1,0	2,0	5,0	2,3
Der Benutzer möchte das einfache Heizen per Infrarotheizung deaktivieren, um das Heizen abzuschalten.	1,0	2,0	3,0	2,0
Der Benutzer möchte die Werte im Konfigurationsdialog verändern, um die initialwerte zu verändern.	2,0	3,0	8,0	3,7
Der Benutzer möchte das optimierte Heizen per Infrarotheizung aktivieren, um Strom zu sparen.	1,0	2,0	3,0	2,0
Der Benutzer möchte das optimierte Heizen per Infrarotheizung deaktivieren, um selbständig zu entscheiden, wann das optimierte Heizen in Betrieb genommen wird.	1,0	2,0	3,0	2,0
Der Benutzer möchte die Werte im Konfigurationsdialog verändern, um die initialwerte zu verändern.	1,0	2,0	3,0	2,0
Als Benutzer möchte ich den Warmwasserboiler über die FTUI manuell ein- und ausschalten können, um selbständig zu entscheiden wann dieser in Betrieb sein soll.	3,0	5,0	8,0	5,2
Als Benutzer möchte ich den Warmwasserboiler für einen fest definierten Zeitraum in Betrieb nehmen, um ihn für den benötigten Benutzungszeitraum nicht manuell ein- bzw. ausschalten zu müssen.	2,0	3,0	5,0	3,2
Als Benutzer möchte ich den Warmwasserboiler nur zu gewünschten Stromtarifen in Betrieb nehmen, um Stromkosten zu sparen.	5,0	8,0	13,0	8,3
Als Benutzer möchte ich den Warmwasserboiler nur zu günstigen Stromtarifzeiten oder mit Solarstrom in Betrieb nehmen können, um Stromkosten zu sparen.	3,0	5,0	8,0	5,2
Als Benutzer möchte ich, dass der Boiler und die Heizung mit einer Priorisierung Solarenergie beziehen.	5,0	8,0	13,0	8,3
Als Benutzer möchte ich eine statistische Auswertung des erzeugten Stroms auf der FTUI-Oberfläche einsehen, um die Produktivität meines Balkonkraftwerks zu überprüfen.	3,0	5,0	8,0	5,2
Als Benutzer möchte ich eine visuelle Darstellung in Form eines Balkendiagramms des erzeugten Stroms auf der FTUI-Oberfläche einsehen, um die Produktivität meines Balkonkraftwerks graphisch abgleichen zu können.	3,0	5,0	8,0	5,2
<b>Gesamtaufwand</b>				<b>54,5</b>

**Tabelle 8: Aufwandsschätzung Aufgabe 2.4.3**

## 6. Ist-Aufwände

### 6.1. Ist-Aufwand Aufgabe 2.4.1

Die Analyse von Planning Poker und der Drei-Zeiten-Methode zur Aufgabe 2.4.1 findet sich in der folgenden Tabelle.

Teilaufgabe	Aufwandsschätzung		Ist-Aufwand	Abweichung absolut		Abweichung prozentual	
	Planning Poker	Drei-Zeiten-Methode		Planning Poker	Drei-Zeiten-Methode	Planning Poker	Drei-Zeiten-Methode
2.4.1	36,8	41,8	41,5	4,7	-0,3	12,8%	-0,7%

Tabelle 9: Ist-Aufwand Aufgabe 2.4.1

Die Analyse der Aufwandsschätzungen ergab, dass die Schätzung der Drei-Zeiten-Methode mit einer Abweichung von nur 0,7 % sehr nahe an dem Ist-Aufwand war. Es zeigte sich auch, dass die Schätzung, die sich auf Basis des Planning Poker ergab mit einer Abweichung von knapp 13 % solide war.

### 6.2. Ist-Aufwand Aufgabe 2.4.2

Die Analyse von Planning Poker und der Drei-Zeiten-Methode zur Aufgabe 2.4.2 findet sich in der folgenden Tabelle.

Teilaufgabe	Aufwandsschätzung		Ist-Aufwand	Abweichung absolut		Abweichung prozentual	
	Planning Poker	Drei-Zeiten-Methode		Planning Poker	Drei-Zeiten-Methode	Planning Poker	Drei-Zeiten-Methode
2.4.2	34	40,3	51	17	10,7	50,0%	26,6%

Tabelle 10: Ist-Aufwand Aufgabe 2.4.2

Die Analyse der Aufwandsschätzungen ergab, dass die Schätzung der Drei-Zeiten-Methode mit einer Abweichung von etwa 27 % näher an dem Ist-Aufwand war als die Schätzung, die sich auf Basis des Planning Poker ergab. Diese lag mit einer Abweichung von 50 % weit weg vom Ist-Aufwand.

### 6.3. Ist-Aufwand Aufgabe 2.4.3

Aufgrund der vorangegangenen Erfahrungswerte wurde für die Aufgabe 2.4.3 keine Schätzung mit der Planning Poker-Methode durchgeführt. Das Planning Poker war deutlich zeitaufwendiger, da das gesamte Team an der Schätzung teilgenommen hat. Dennoch war die Schätzungen mit der Planning Poker-Methode in allen unseren Fällen unpräziser als die Schätzungen mit der Drei-Zeiten-Methode.

Die Analyse der Drei-Zeiten-Methode zur Aufgabe 2.4.3 findet sich in der folgenden Tabelle.

Teilaufgabe	Aufwandsschätzung	Ist-Aufwand	Abweichung absolut	Abweichung prozentual
	Drei-Zeiten-Methode		Drei-Zeiten-Methode	Drei-Zeiten-Methode
2.4.3	54,5	151,2	96,7	177,4%

Tabelle 11: Ist-Aufwand Aufgabe 2.4.3

Die Analyse der Aufwandsschätzungen ergab, dass die Schätzung der Drei-Zeiten-Methode mit einer Abweichung von etwa 177 % für diese Teilaufgabe eine relative schlechte Schätzung darstellt. Die tatsächlichen Aufwände lagen weit über den vorgenommenen Schätzungen anhand der Drei-Zeiten-Methode. Zum Zeitpunkt der Schätzung war den Personen, die die Aufwandsschätzungen durchgeführt haben, nicht bekannt, wie zeitaufwendig die Implementierung sein würde. Bei der Verwendung von Planning Poker wären mehr Personen an der Schätzung beteiligt und es könnte daher sein, dass der Aufwand dadurch höher, also näher am tatsächlichen Aufwand, eingeschätzt worden wäre. Allerdings kann darüber nur spekuliert werden und die Erfahrungswerte der vorangegangenen Schätzungen zeigten, dass Planning Poker deutlich aufwendiger war. Daher könnte argumentiert werden, dass trotz der starken Abweichung, die Entscheidung für die Drei-Zeiten-Methode die richtige Entscheidung war.

## 7. IT-Architektur

Die Architektur des Gesamtsystem besteht aus der Architektur der Hardware, bestehend aus Raspberry Pi und angeschlossenen Geräten, sowie der Software, bestehend aus Betriebssystem, Programmplattform FHEM und diversen Modulen.

### 7.1. Raspberry Pi

Der Raspberry Pi ist ein Computer auf einer Platine von der Größe einer Kreditkarte. Die CPU entspricht der ARM-Architektur mit genau einem Chip und ist daher ein SoC (System on a Chip) PC, wie etwa auch die neuen M1 Chips von Apple oder die altbekannten Chips aus Handys. Beispielhaft zu nennen ist die Snapdragon Produktfamilie von Qualcomm, welche in vielen Android Geräten verbaut wurde und wird. Der Pi ist eins zu eins vergleichbar mit einem normalen Computer in der Funktionalität. Wie auch bei Desktop-Computern lassen sich verschiedene Betriebssysteme aufspielen und betreiben. Die vergleichsweise schwächere Leistungsfähigkeit und die fehlende Möglichkeit des Anschlusses einer Grafikkarte empfehlen kein technisch möglich ist. Der produzierte Overhead von Windowssystemen ist schlichtweg zu groß. Es empfehlen sich auf Linux basierende und unixoide Betriebssysteme, wie das hauseigene Debian Derivat Raspbian/Raspberry Pi OS, oder Ubuntu, Archlinux oder FreeBSD. Der geringe Strombedarf, das kompakte Format und der Fakt, dass das System nicht dauerhaft an einen Bildschirm angeschlossen sein muss, prädestinieren das System zum Serverbetrieb im 24/7 Dienst. Der Computer ist inzwischen in der 4 Generation verfügbar, die Produktlinie wurde um eine noch kleinere Version erweitert. Wikipedia beschreibt "das Motiv hinter der Entwicklung eines preisgünstigen Rechners war die sinkende Anzahl an Informatikstudenten an der Universität Cambridge sowie die jedes Jahr geringeren Programmierkenntnisse der Studienanfänger"<sup>3</sup>, was wohl auch heute noch als korrekt angesehen werden kann. Der Raspberry Pi eignet sich neben dem Betrieb als Mini-Server für daheim auch zum Betrieb in der Industrie, wo er im Rahmen des Konzeptes Industrie 4.0 an Maschinen verbaut werden kann, um IdD Komponente zu sein. Seiner ursprünglich bedachten Aufgabe als Lern- und Probiercomputer kann der Pi ebenfalls gerecht werden. GPIO (General Purpose Input/Output) Pins ermöglichen das Experimentieren mit LEDs und Co. Gerade aber auch der erste Kontakt mit linuxbasierten oder unixoiden Betriebssystemen werden dem Gedanken try-and-error gerecht.

---

<sup>3</sup> [https://de.wikipedia.org/wiki/Raspberry\\_Pi](https://de.wikipedia.org/wiki/Raspberry_Pi)

## 7.2. Hardwarearchitektur

Zu Beginn der Projektarbeiten wurde während der Anforderungsanalyse ein Schaubild der Hardwarearchitektur angefertigt.

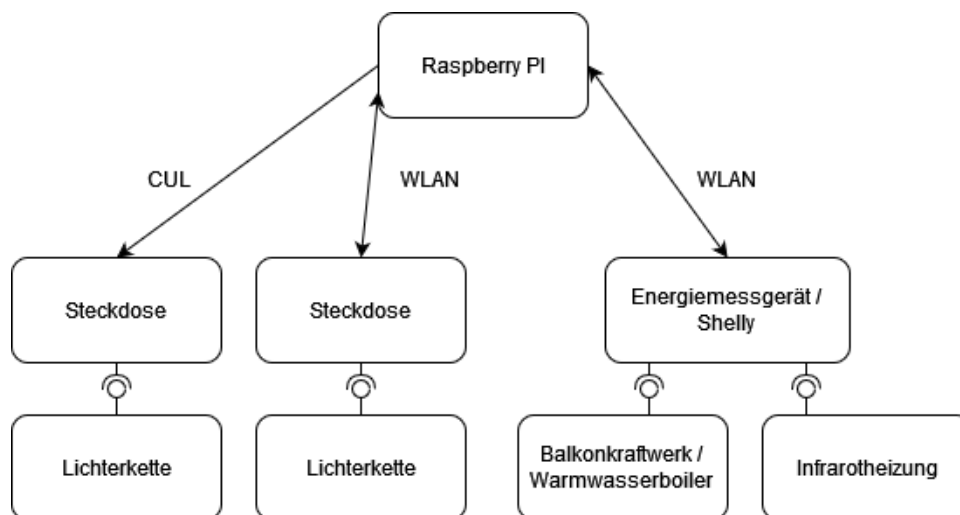


Abbildung 9: Hardwarearchitektur

Aufgezeichnet sind die zentralen Einheiten des Raspberry Pi's, sowie die schnittstellenimplementierenden Geräte. An die Schnittstellen angeschlossene Geräte sind in der untersten Reihe ersichtlich. Die Begebenheit, dass die Kommunikation zwischen Pi und schnittstellenimplementierenden Geräten über verschiedene Medien realisiert wird, ist via den Kommunikationspfeilen ersichtlich. Diese zeigen ebenfalls an, ob der Kommunikationskanal bidirektional, also Statusabfragemöglichkeit und Feedback, realisiert ist, oder nur unidirektional ist. In diesem Fall werden Befehle abgeschickt, eine Bestätigung des Empfangens ist allerdings nicht möglich. Zustandsdarstellungen in der Software sind somit antizipiert.

### 7.3. Softwarearchitektur

Ebenfalls zu Beginn der Projektarbeiten, während der Anforderungsanalyse und dem Beginn der Implementierungsphase, wurde ein Schaubild der Softwarearchitektur angefertigt.

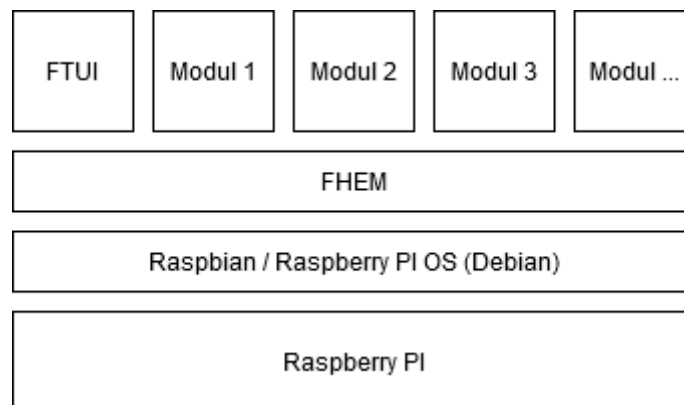


Abbildung 10: Softwarearchitektur

Im Schichtendiagramm sind die Softwareschichten dargestellt. Zur Referenz wurde als unterste Schicht die Hardware des Raspberry Pi's dargestellt. Auf diesem wird das Betriebssystem Raspberry Pi OS, ehemals Raspbian, betrieben. Das linuxbasierte Debian Derivat ist für den Betrieb auf dem Mini-PC mit ARM-Chip Architektur ausgelegt. Es eignet sich speziell für die Nutzung als Server im 24/7 Betrieb. Hervorragende Eigenschaften der gängigen Linux-Derivate Debian und Ubuntu wurden hierbei übernommen. Auf der Betriebssystemschiicht wird die Softwareplattform FHEM betrieben. Diese nutzt die, Sprache Perl, welche in letzten Jahren stark an Bedeutung verloren hat. Perl besticht durch Typunsicherheit, unleserliche Syntax und einer Identitätskrise, ob nun objektorientiert oder nicht, wird allerdings als Sprache noch weiterentwickelt. Perl wurde 1987 für die NSA von Larry Wall entwickelt und genoss ihre besten Tage in den Jahren des WEB 1.0 als Internetsprache. Mit Perl wird das Programm FHEM betrieben, welches eine Plattform mit Basisfunktionalität realisiert, auf welcher die darauf liegenden Module betrieben werden. Die Module müssen ggf. in den Update Prozess von FHEM integriert werden. Dabei werden sie von einem Content Delivriery Network bezogen. Dies ist vergleichbar mit der Funktionalität von apt oder CPAN. CPAN ist das Content Delivery Network für Perl Module. Die Module selbst realisieren in der Regel eine gewisse Funktionalität und können mit anderen Modulen dafür kommunizieren. Auch das Frontend FTUI ist gerade ein Modul neben anderen.

## 7.4. SSH – Putty

SSH (Secure Shell) ist ein Netzwerkprotokoll, welches Nutzern ermöglicht, eine gesicherte Verbindung innerhalb eines unsicheren Netzes, wie zum Beispiel dem Internet, zwischen zwei Computern aufzubauen. Der Zugriff auf den entfernten Rechner erfolgt dabei meist über die Kommandozeile. Damit ist es möglich, Befehle auf dem entfernten Gerät auszuführen. Typische Einsatzgebiete von SSH sind beispielsweise folgende:

- Verwaltung von Servern, auf die man lokal keinen Zugriff hat
- Sichere Übermittlung von Dateien
- Sicheres Erstellen von Backups
- Verbindung zwischen zwei Rechnern mit Ende-zu-Ende-Verschlüsselung
- Fernwartung von anderen Computern

SSH basiert auf der Client-Server-Architektur, das bedeutet, dass ein SSH-Client sich immer mit einem SSH-Server verbindet. Ein populärer SSH-Client ist PuTTY, eine Open-Source-Software, die das Herstellen von Verbindungen über SSH ermöglicht. Auch lassen sich damit Befehle über die Kommandozeile auf dem entfernten Rechner ausführen. PuTTY ist auf Rechnern mit Windows, Linux und MacOS als Betriebssystem einsetzbar.

PuTTY kann auch dazu benutzt werden, um einen Raspberry Pi über den eigenen PC zu steuern. Der Pi muss sich dazu lediglich im selben Netzwerk wie der PC befinden. Außerdem muss SSH aktiviert werden. Danach muss in PuTTY der Hostname angegeben werden und man kann sich nun mittels User und Passwort mit dem Pi verbinden und Befehle ausführen.

## 7.5. Shell

Eine Shell ist die Schnittstelle zum Benutzer unter linuxbasierten und unixoiden Betriebssystemen. Das Gegenstück sind die PowerShell und die CMD unter Windows. In den Shells lassen sich Befehle in Textform gegen das Betriebssystem absetzen. In der Regel kommen die Shells in der Bedienung ohne Maus aus und werden allein durch Tastatureingaben bedient. Es existiert in der Regel ein Standardsatz an Befehlen des Betriebssystems, wie unter Linux etwa `cd` für change directory, `pwd` für print working directory oder `ls` für list. `ls` zeigt die Inhalte eines Verzeichnisses, `cd` wechselt dieses, und `pwd` zeigt den Pfad des aktuellen Verzeichnisses an. Die Befehle werden in der Regel durch Argumente spezifiziert. Die Befehle, die einen Programmaufruf darstellen, konsumieren die Argumente und führen ihren Programmcode mit den zusätzlichen Daten aus. Ein `mkdir scripts` legt etwa ein Verzeichnis mit dem Namen `scripts` an. Rückgaben von Ausgabewerten der Befehle werden ebenfalls in die Shell gedruckt. Der Standardsatz des



Betriebssystems wird durch Programme erweitert, welche beispielsweise über den Standardpaketverwaltungsdienst von Debian installiert werden können. Beim Ausführen von Skripten kann ausgewählt werden, mit welcher Shell diese ausgeführt werden sollen. Dies wird explizit beim Aufrufen mit angegeben. So wird durch `sh install.sh` das Skript `install.sh` durch die Bourne Shell (sh) ausgeführt. Ein `bash install.sh` würde ein Ausführen mit der Bash Shell bezwecken, ein `ksh install.sh` das Ausführen mit der Korn Shell. Implizit bezeichnet das Shebang oder Hashbang Zeichen in der ersten Zeile eines Skriptes `#!/` implizit mit welcher Shell das Skript ausgeführt werden soll. Der besondere Kommentar verweist dabei auf den Pfad, in dem die Shell zu finden ist, etwa `#!/bin/sh`. Es ist ebenfalls möglich Interpreter und Laufzeitumgebungen aus den Umgebungsvariablen zu nutzen. Hierbei wird auf das Programm `/bin/env` verwiesen und als Argument die Sprache verwendet. `#!/bin/env python` sucht in den Umgebungsvariablen nach dem Pfadeintrag für das Programm Python und führt das Skript damit aus. Zu Bedenken ist, dass gleiche Programme in gleichen Versionen für unterschiedliche Shells sich unterschiedlich verhalten können. Es bestehen etwa große Unterschiede darin wie die Sprachen `sed` und `awk` in den unterschiedlichen Shells zu bedienen sind.

Mit Shells ist ein Computer, sofern entsprechende Rechte vorhanden sind, umfänglich steuerbar. Herunterfahren und Neustarten sind ebenso möglich wie das Installieren von Programmen.

## 8. Sicherungskonzept

Ein Sicherungskonzept sollte in jedem IT-System vorhanden sein, auch wenn die Hoffnung stets ist, dass es niemals gebraucht wird. Es können allerdings in der Realität viele unterschiedliche Fälle auftreten, in denen das System beschädigt wird oder komplett verloren geht. Sollte dies passieren, muss die Möglichkeit bestehen, einen möglichst aktuellen Stand des Systems ohne großen Aufwand wieder aufsetzen zu können. Dazu wird in diesem Projekt das von Linux bereitgestellte Programm `dd` (disk dump), in Verbindung mit Dropbox, dem Programm `libfuse2` und Cron-Deamon verwendet. Dropbox ist ein Filehosting-Dienst des Unternehmens Dropbox Inc. und wird verwendet, um die Sicherungen örtlich getrennt vom System in der Cloud zu speichern. Um die Dropbox in dem auf dem Raspberry Pi laufenden Linux-Betriebssystem als Verzeichnis einzubinden, wird das Programm `libfuse2` verwendet. Nach der Einrichtung kann die Dropbox verwendet werden, als wäre sie ein lokal gespeichertes Verzeichnis. Das Programm `dd` ist ein Unix-Kommando, das zum blockorientierten Kopieren oder Konvertieren beliebiger Dateien dient. Es wird in einem Skript verwendet, um die gesamte SD-Karte des Raspberry Pi zu kopieren und in der Dropbox zu speichern. Der Cron-Deamon dient der zeitbasierten Ausführung von Prozessen in Unix. Darüber können sogenannte Cronjobs angelegt werden, die immer zu einem festgelegten Zeitpunkt, in diesem Fall einmal täglich um 0:30 Uhr, ausgeführt werden. Dieser Cronjob startet das Backup-Skript, das, wie bereits erwähnt, `dd` nutzt um die SD-Karte in die Dropbox zu sichern. Dabei wird das erzeugte Backup immer mit dem aktuellen Datum versehen. Da der Speicherplatz in der Dropbox nicht unendlich ist wurde festgelegt, dass nur die Backups der letzten vier Wochen vorgehalten werden. Dementsprechend wird am Ende des Skripts geprüft, ob ein Backup von vier Wochen (27 Tagen) vor dem aktuellen Datum vorliegt und dieses gegebenenfalls gelöscht. Um das Backup wieder einzuspielen, wird ein Linux-System mit einem SD-Kartenleser benötigt. Das Backup kann dann, ebenfalls mit dem Programm `dd`, von der Dropbox direkt auf die SD-Karte gespielt werden.

## 9. IT-Sicherheit

IT-Sicherheit spielt eine wichtige Rolle im kommerziellen IT-Betrieb. Nur schleppend transferieren sich die Wertevorstellungen auch in den privaten Sektor. Gerade hier sind lascher Umgang mit Informationen noch weit verbreitet. Sparten der IT entwickelten Geschäftsmodelle mit dem Ziel Informationen von Nutzern aus dem privaten Sektor abzuziehen und zu monetarisieren. Nennenswert sind Konzerne mit fragwürdigen Moralvorstellungen, wie Meta, mit den Produkten Facebook und Instagram, Google und Co, sowie auch deren asiatische Äquivalente wie Samsung und Co. Die Plattform FHEM ist an sich nicht sonderlich kommunikativ, sie schickt keine Daten an ehemalige Entwickler oder in Clouds. Identifizierter Hebelpunkt ist die Einrichtung von HTTPS als TLS. HTTPS sichert die Kommunikation über Internetprotokolle ab und verschlüsselt den Inhalt der versendeten Pakete. Diese sind somit für außenstehende nicht lesbar, sofern diese nicht den Entschlüsselungsschlüssel besitzen. Dem Benutzer des Systems wird zudem empfohlen ein sicheres Password für sein WLAN und den Nutzer Pi auf dem Raspberry zu wählen.

Explizit zu erwähnen ist der Verzicht auf die WLAN-Steckdose. Zur Einrichtung dieser ist es unumgänglich einen Account beim Hersteller in China einzurichten und der Übermittlung von Nutzungsdaten an dessen Cloud zuzustimmen. Es ist kritisch zu betrachten, dass die Hochschule dies nicht überprüft hat und entsprechende Hardware an die Studierenden ausgegeben wurde. Wie bereits erwähnt stellen Nutzer oft freiwillig Ihre Daten den Unternehmen zur Verfügung. Dieser Umstand berechtigt die DHBW allerdings nicht davon auszugehen, dass gelebte Praxis einen aktiven Verstoß der IT-Sicherheit begründen darf, welchen Studierende zu akzeptieren haben. Von der Nutzung solcher Geräte ist abzusehen. Es ist positiv anzumerken, dass der Dozent nach Hinweis den Verzicht der Steckdose.

## 10. Implementierungsgrundlagen

### 10.1. Backendimplementierung

#### 10.1.1. Devices

Der Grundbaustein von FHEM sind die sogenannten „Devices“. Jedes Device bildet typischerweise einen realen Gegenstand wie zum Beispiel Funksteckdosen, Glühbirnen oder Messgeräte ab. Devices können aber auch nicht physische Gegenstände abbilden, indem sie beispielsweise benutzt werden, um Variablen abzuspeichern oder bestimmte Funktionalitäten zu erfüllen. Devices werden in FHEM angelegt, indem man folgenden Befehl in das Kommandofeld eingibt:

```
define „Name des Devices“ „Typ des Devices“
```

Dabei hat jedes Device genau einen „Typ“. Diese Typen sind mit Perl-Modulen verknüpft, in der bestimmte Routinen und Eigenschaften definiert sind. Um im Folgenden den Aufbau und verschiedene Funktionalitäten eines Devices genauer erläutern zu können, soll dazu ein Device vom Typ „Dummy“ benutzt werden. Ein Dummy-Device benötigt kein korrespondierendes physisches Gerät und kann beispielsweise dazu benutzt werden, den Programmablauf zu steuern oder Informationen userseitig zur Verfügung zu stellen.

Jedes Device in FHEM hat einen Zustand, der sich ändern kann. So ist beispielsweise eine Glühbirne an oder eine Funksteckdose hat Spannung. Ein Zustand muss dabei nicht exakt eine Variable beinhalten (z.B. Glühbirne an oder aus), sondern kann durch mehrere Variablen beschrieben werden. Dieser Zustand wird in FHEM durch insgesamt drei Größen beschrieben, nämlich durch Internals, Readings und Attribute.

Die Internals enthalten grundlegende Informationen zum Device, die beim Erstellen des Devices erzeugt werden. Diese werden nur in den seltensten Fällen direkt vom Benutzer bearbeitet. Mindestens enthalten die Internals dabei folgende Größen:

- *NAME* ist der vom Benutzer vergebene Name
- *TYPE* ist der Typ des FHEM-Devices
- *STATE* ist der global angezeigte Status des Geräts. Wenn das Gerät noch keinen Status hat, wird „??“ angezeigt

Darüber hinaus können die Internals noch viele weitere Größen beinhalten, wie zum Beispiel die Seriennummer des Geräts, die aktuelle Firmware-Version des Geräts, die IP-Adresse des Geräts etc.

Daten, welche von einem Gerät gelesen und in FHEM für den Benutzer verständlich bereitgestellt werden, werden Readings genannt. Sie geben den Status des Geräts wieder und erzeugen Events innerhalb von

FHEM, auf die die anderen Geräte reagieren können. Jedes Reading besitzt einen eindeutigen Zeitstempel in der Form YYYY-MM-DD HH:MM:SS. Üblicherweise werden in den Readings Messwerte des jeweiligen Geräts erfasst und geben somit beispielsweise die aktuelle Temperatur, Luftfeuchtigkeit, Windgeschwindigkeit, elektrische Spannung oder die elektrische Leistung an.

Mit Hilfe von Attributen kann der Benutzer das Verhalten von Geräten zur Laufzeit individuell anpassen. Bei den Attributen wird zwischen globalen, lokalen und individuellen Attributen unterschieden. Globale Attribute können von jedem Gerät benutzt werden. Mit Hilfe des Befehls „list global“ können diese angezeigt werden.

Lokale Attribute können nur von individuellen Geräteklassen oder einzelnen Geräten benutzt werden. Welche das sind, wird in den jeweiligen Perl-Modulen des Geräts festgelegt. Hinweise zur Funktion dieser Attribute kann man also in der jeweiligen Modulbeschreibung finden.

Neben den von FHEM und den jeweils verwendeten Modulen bereitgestellten Attributen besteht die Möglichkeit, eigene Attribute zu definieren. Dies kann wiederum entweder global für alle Geräte oder lokal für ein bestimmtes Gerät geschehen.

Im Gegensatz zu Attributen, werden als Readings, wie bereits beschrieben, Daten bezeichnet, die von einem Gerät gelesen und in FHEM bereitgestellt werden. Readings ändern sich im laufenden Betrieb und werden nicht in der Konfiguration gespeichert.

In folgender Abbildung ist der Zustand eines Dummy-Devices zu sehen, welches den Betriebsmodus einer Glühbirne, die über eine Funksteckdose gesteuert wird, repräsentiert.

DeviceOverview		
funkLampModus		Manuell
set funkLampModus <div>switch</div> <div>festeZeit</div>		
Internals		
FUUUID	61b9e964-f33f-c687-3da6-bf88158338cba53b	
NAME	funkLampModus	
NR	29	
STATE	Manuell	
TYPE	dummy	
Readings		
state	Manuell	2022-01-13 10:20:28
attr funkLampModus <div>room</div> <div>241_Weihnachtsbeleuchtung</div>		
Attributes		
room	241_Weihnachtsbeleuchtung	deleteattr
setList	switch:festeZeit,Sonne,SZeit,Anwesenheit,Manuell	deleteattr
Probably associated with		
doif_funk_general	cmd_5	DOIF
doif_funk_support	deactivated	DOIF
Select icon   Extend devStateIcon   Raw definition   Delete this device (funkLampModus)   Device specific help		

Abbildung 11: Device auf FHEM

### 10.1.2. Hilfsmodule

Hilfsmodule werden in FHEM genutzt, um bestimmte Funktionalitäten zu erfüllen und Problemstellungen zu lösen, ohne selbst Perl-Code programmieren zu müssen. Hilfsmodule werden in FHEM auch als Devices mit dem entsprechenden Typ dargestellt. Im Folgenden sollen drei wichtige Hilfsmodule vorgestellt werden, die bei der Implementierung eine wesentliche Rolle gespielt haben. Das erste Hilfsmodul ist ein Device vom Typ DOIF. Mit diesem Modul ist es möglich, Anweisungen ereignis- und zeitgesteuert in Abhängigkeit definierter Bedingungen auszuführen. Dadurch können komplexere Problemstellungen innerhalb eines DOIF-Moduls gelöst werden, ohne Perl programmieren zu müssen. Die Syntax eines DOIFs lautet folgendermaßen:

```
define <name> DOIF (<Bedingung>) (<Befehle>) DOELSEIF (<Bedingung>)
(<Befehle>) DOELSEIF ... DOELSE (<Befehle>)
```

Ereignisse, Zeittrigger, Readings oder der Status werden dabei stets in eckigen Klammern angegeben. Sie führen zur Triggerung des Moduls und damit zur Ausführung der definierten Befehle. Die Angaben werden immer von links nach rechts abgearbeitet und logische Abfragen werden in den Bedingungen meist mit Hilfe von and/or-Operatoren erstellt. Jeder Ausführungszweig eines DOIFs (DOIF/DOELSEIF/DOELSE) stellt einen eigenen Zustand dar (cmd1, cmd2, etc.), der im state des Devices gespeichert wird. Das Modul merkt sich den zuletzt ausgeführten Ausführungszweig und wiederholt diesen standardmäßig nicht. Ein Ausführungszweig wird erst dann wieder ausgeführt, wenn zwischenzeitlich ein anderer Ausführungszweig ausgeführt wurde, also ein Statuswechsel innerhalb des Moduls stattgefunden hat. Das Verhalten des Moduls kann durch verschiedene Attribute verändert werden. So gibt es beispielsweise das Attribut „do“, das entweder den Wert „always“ oder „resetwait“ annehmen kann. Mit Hilfe von „do always“ kann ein Befehl mehrfach nacheinander ausgeführt werden, wenn die Bedingung eintritt und kein Statuswechsel stattgefunden hat. Mit Hilfe von „do resetwait“ wird ein sogenannter „wait-Timer“ zurückgesetzt, wenn die gleiche Bedingung wiederholt wahr wird. Damit können Ereignisse ausgelöst werden, wenn etwas innerhalb einer Zeitspanne nicht passiert. Der wait-Timer kann mit Hilfe des Attributs „wait“ gestartet werden. Dieser verzögert die Ausführung von Befehlen um einen bestimmten Wert. Sollen Verzögerungen innerhalb von Befehlsfolgen stattfinden, so müssen diese Befehle in eigene Klammern gesetzt werden.

Das nächste Hilfsmodul ist ein Device vom Typ at. Es ermöglicht Befehle einmalig oder regelmäßig zu einem späteren Zeitpunkt auszuführen. Es wird folgendermaßen definiert:

```
define <name> at [<timespec>|<datespec>] <command>
```

Schließlich soll noch das Hilfsmodul *HTTPMOD* genannt werden. Damit lassen sich Informationen aus Websites auslesen und in Readings eines Devices speichern. Um die HTTP-Antwort zu parsen, werden häufig reguläre Ausdrücke benutzt. Außerdem kann das Modul auch Informationen an Devices senden. Um dies zu ermöglichen, kann ein Set-Attribut konfiguriert werden. Ein Device vom Typ HTTPMOD wird folgendermaßen definiert:

```
define <name> HTTPMOD <URL> <Interval>
```

Das Modul spricht nach jedem Intervall, das in Sekunden angegeben wird, die URL an und gibt dann die Antwort zurück. Wenn man die Antwort manuell mit einem get-Befehl abfragen will, kann das Intervall auf 0 gesetzt werden.

### 10.1.3. Room

Das Attribut „room“ ist ein globales Attribut mit besonderer Bedeutung. Es wird benutzt, um Geräte zu filtern und zu gruppieren und damit zu strukturieren. Prinzipiell werden alle Räume auf der FHEM-Oberfläche in der linken Spalte angezeigt. Geräte, welche kein room-Attribut besitzen, werden im Raum „unsorted“ angezeigt. Weiterhin gibt es den Raum namens „Everything“, der jedes einzelne Gerät beinhaltet. Um ein Gerät einem bestimmten Raum zuzuordnen, muss das Attribut mit dem entsprechenden Namen des Raums gesetzt werden. Die Syntax lautet folgendermaßen:

```
attr <devicename> room <roomname>
```

Die folgende Abbildung zeigt die Strukturierung der Geräte nach der Implementierungsphase:



Abbildung 12: FHEM Menüleiste

### 10.1.4. Perl-Skripte

Wie bereits beschrieben sind Devices das Grundgerüst von FHEM. Jedes Device hat einen bestimmten Typ und jeder Typ stellt ein Modul dar, das in Perl programmiert wurde. So ist beispielsweise die gesamte Logik des Dummy-Moduls in einem Perlskript namens „98\_dummy.pm“ enthalten.

Die Speicherung von Perl-Code unmittelbar in Eventhandlern, Timehandlern oder anderen Geräten wird mit zunehmender Zahl von Geräten und Logiken unübersichtlich. Um Logikbausteine zentral nutzen zu können, kann es daher sinnvoll sein, eigene Module bzw. Dateien anzulegen. FHEM enthält hierzu eine Vorlage myUtilsTemplate.pm, die zur Erzeugung eigener Programmdateien genutzt werden kann. Um dies konkret umzusetzen, navigiert man in FHEM in der linken Spalte zu „Edit files“. Dort öffnet man die eben genannte Vorlage und kann den Inhalt nun durch das eigene Skript ersetzen. Zuletzt muss im Textfeld neben „Save as“ der Name des eigenen Skripts eingesetzt werden und man kann mit einem Klick auf „Save as“ speichern. Die Funktionalität des Skripts lässt sich nun global nutzen.



```

1 #####
2 # $Id: myUtilsTemplate.pm 21509 2020-03-25 11:20:51Z rudolfkoenig $
3 #
4 # Save this file as 99_myUtils.pm, and create your own functions in the new
5 # file. They are then available in every Perl expression.
6
7 package main;
8
9 use strict;
10 use warnings;
11
12 sub
13 myUtils_Initialize($$)
14 {
15     my ($hash) = @_;
16 }
17
18 # Enter you functions below _this_ line.
19
20
21 1;
  
```

Abbildung 13: myUtilsTemplate Code



## 10.2. Benutzerfreundliche Oberflächengestaltung

### 10.2.1. FTUI vs. FUIP

FHEM Tablet Ui (FTUI) ist ein Frontend-Framework. Mithilfe des Frameworks können Geräte aus FHEM gesteuert und überwacht werden. Das basiert auf HTML, CSS und JavaScript. Um FTUI in den Betrieb zu nehmen ist eine FHEM-Installation nötig, welche in der Schritt-für-Schritt-Anleitung zu finden ist.



Abbildung 14: FTUI Beispiel

FHEM User Interface Painter (FUIP) ist ein Modul mit dem FTUI Seiten durch einfaches „zusammenklicken“ erstellt werden können. Dafür muss das FUIP Modul installiert werden. Um in dem Modul Änderungen vornehmen zu können, muss auf das Zahnradchen des Widgets gedrückt werden. Der zugehörige HTML-Code wird dabei im Hintergrund erstellt. Um die verschiedenen Räume und deren Devices in FUIP anzeigen zu können, müssen diese im Vorfeld in FHEM angelegt werden.

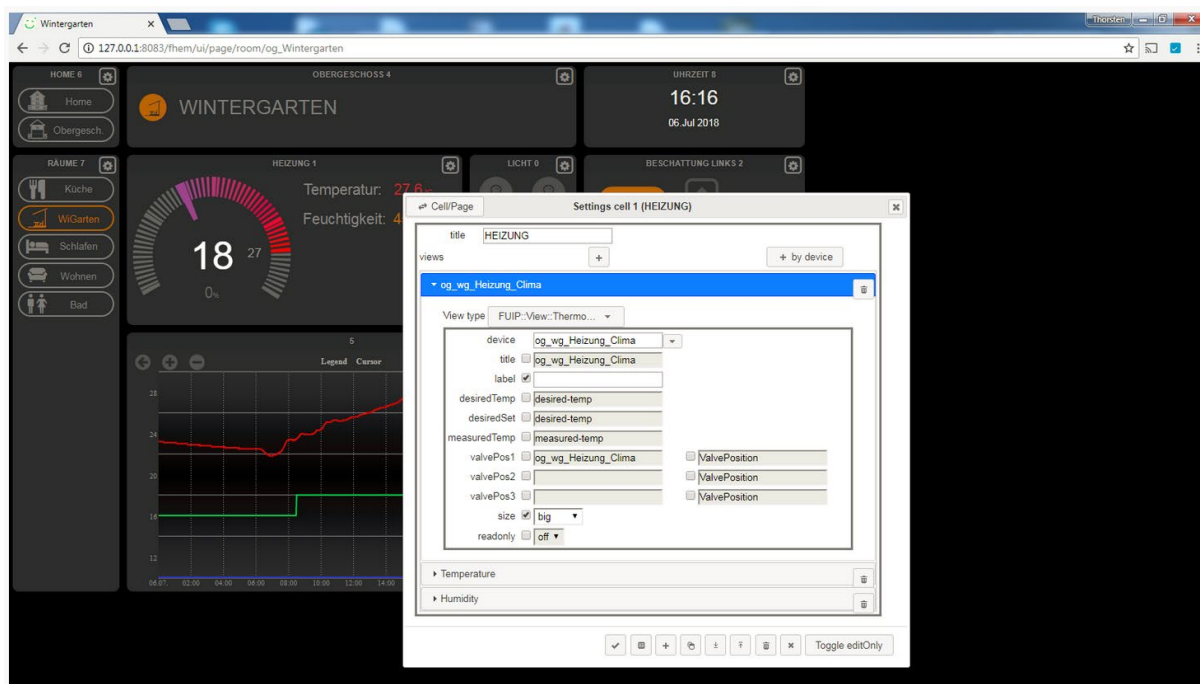


Abbildung 15: FIUP Beispiel

Da FIUP auf FTUI aufbaut und es anfangs leichter war HTML Dateien, aufgrund von Vorwissen aus vorherigen Vorlesungen und Praxiseinsätzen, zu nutzen, wurde sich für das FTUI Framework entschieden. Des Weiteren war eine frühzeitige Frontend-Gestaltung, ohne das Anlegen von Räumen und Devices, mit FTUI möglich. Das Frontend war demnach in seiner Entwicklung und Gestaltung vom Backend losgelöst, was zusätzlich für die Nutzung von FTUI und nicht FIUP sprach.

### 10.2.2. Visual Studio Code

Bei Visual Studio Code handelt es sich um einen kostenlosen Quelltext-Editor von Microsoft. Visual Studio Code ist für die Betriebssysteme Windows, macOS und Linux verfügbar. Der Quelltext-Editor unterstützt Sprachen wie Java, HTML, CSS und Perl. Dabei arbeitet Visual Studio Code auf Basis von Quelltextdateien, Ordnern und Textdateien. Der kostenfreie Quelltext-Editor ist Industrie-Standard und uns aus früheren Projekten schon bekannt, weshalb wir uns aus Gründen der Einfachheit, des Grundverständnisses und Zusammenarbeit dafür entschieden haben. Mithilfe des Editors können wir die HTML- und CSS-Dateien für die benötigten benutzerfreundlichen Oberflächen gestalten und anpassen.

### 10.2.3. Visuelle Darstellung

Im Rahmen der Visuellen Darstellung der Funktionen für die Hausautomation ergab sich die Aufgabe einer benutzerfreundlichen Darstellung, als auch Bedienung, der einzelnen Funktionalitäten. Eine Umsetzung in Form von einer Single-Page wäre nicht benutzerfreundlich gewesen, da die Übersichtlichkeit darunter leiden würde. Die Anzahl der Funktionalitäten mit der Menge an einzelnen Anforderungen hätte den Umfang einer Single-Page gesprengt. Daraus ergab sich in einer gemeinsamen Absprache eine Darstellung mithilfe einer Menüleiste und eine Darstellung für Funktionalitäten auf mehreren Pages verteilt.

Die Menüleiste soll stets zu sehen sein, um zwischen den Hauptfunktionalitäten wechseln zu können und in Abhängigkeit der Hauptfunktionalität die Bedienung der dazugehörigen einzelnen Anforderungen zu ermöglichen. Dies hatte zur Folge das jede Page einen einzelnen, fest definierten Bereich, aus Gründen der Gleichheit, für eine Menüleiste aufzuweisen hat. Dies ermöglicht den einzelnen Bedienmöglichkeit einer Hauptfunktionalität mehr Platz zur Darstellung und damit einer einhergehenden Übersichtlichkeit.

Im Falle der Hauptfunktionalität Photovoltaik/Balkonkraftwerk haben sich eine größere Menge an Einzelanforderungen für die Oberfläche ergeben. Diese Anforderungen hätten sowohl die gewonnene Übersichtlichkeit als auch den Platzbedarf einer einzelnen Page überschritten. In Absprache wurde daher ein paar Bedienmöglichkeiten durch den Benutzer ausgelagert. Die Auslagerung solcher Anforderungen wurde dabei durch ein Pop-Up, ein zusätzliches Fenster, realisiert. Dies erhielt die Übersichtlichkeit und sparte zusätzlich Platz auf der eigentlichen Page.

Der Grundsatz für die Darstellung aller Funktionalitäten besagt, dass über kurze Befehle durch den Bediener, wie einem Button, bedient werden können soll. Nur wenn dies aufgrund der Anforderung nicht möglich ist, soll der Grundsatz der Gestaltung nicht eingehalten werden.

Des Weiteren wurden die Farben der Oberfläche visuell angepasst. Die Intention war ein helleres Frontend welches mit Farbakzenten ausgestattet wurde. Für Felder, welche eine Aktion beinhalten, wurde eine auffallendere Farbe ausgewählt. Als Beispiel ist hier die Photovoltaikseite zu sehen.



Abbildung 16: Beispiel für das Farbschema

#### 10.2.4. Hypertext Markup Language (HTML)

Bei HTML handelt es sich um eine Auszeichnungssprache die Webseiten strukturiert. Dabei besteht eine HTML-Datei aus mehreren Elementen, die den Inhalt strukturieren. Das Grundgerüst einer HTML-Datei sieht dabei wie folgt aus:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Meine Testseite</title>
  </head>
  <body>
  </body>
</html>
```

Abbildung 17: Beispiel einer HTML-Datei

- `<!DOCTYPE html>` → Dokumententyp: wurde früher benötigt, um den Browser die zu befolgenden Regeln mitzuteilen. Heutzutage wird dies vernachlässigt, aber dennoch zur korrekten Anzeige der HTML-Dateien benötigt.
- `<html>` → Hierbei handelt es sich um das root-Element, welches den ganzen Inhalt umhüllt.
- `<head>` → In den Head kommen jegliche Informationen, die nicht auf der Webseite angezeigt werden. Exemplarisch hierfür ist die Seitenbeschreibung und die CSS.
- `<title>` → Der Titel der Webseite, welcher im Tab angezeigt wird.

- `<body>` → In den Body kommen jegliche Informationen, die auf der Webseite angezeigt werden sollen. Dazu gehört der Text, die Bilder, die Videos und Audiodateien.

Jedes HTML-Element hat dabei denselben Aufbau. Es beginnt immer mit dem öffnenden Tag, welcher die Struktur des folgenden Inhaltes vorgibt. Dieser muss am Ende durch einen schließenden Tag wieder geschlossen werden.



Abbildung 18: Beispiel eines HTML-Elements

### 10.2.5. Cascading Style Sheets (CSS)

CSS wird verwendet, um HTML Webseiten zu gestalten und anzupassen. Dabei handelt es sich um eine Stylesheet-Sprache, mit der die definierten Elemente im HTML-Code, z.B. mit einer Wunschfarbe, gestaltet werden können.

```
p {
  color: red;
}
```

Abbildung 19: Beispiel eines CSS Codes

Damit die Webseiten gestalterisch so aussehen, wie gewünscht, muss die CSS-Datei im HTML-Dokument im `<head>` Tag verknüpft werden. Ohne diese Verknüpfung wird die CSS-Datei vom Browser nicht berücksichtigt.

Folgende Zeile muss hinzugefügt werden:

```
<link href="styles/style.css" rel="stylesheet" type="text/css">
```

Hierbei muss der Pfad und der Name in href immer angepasst werden.

Genau wie bei einer HTML-Datei gibt es bei einer CSS-Datei einen bestimmten Aufbau, der wie folgt aussieht.

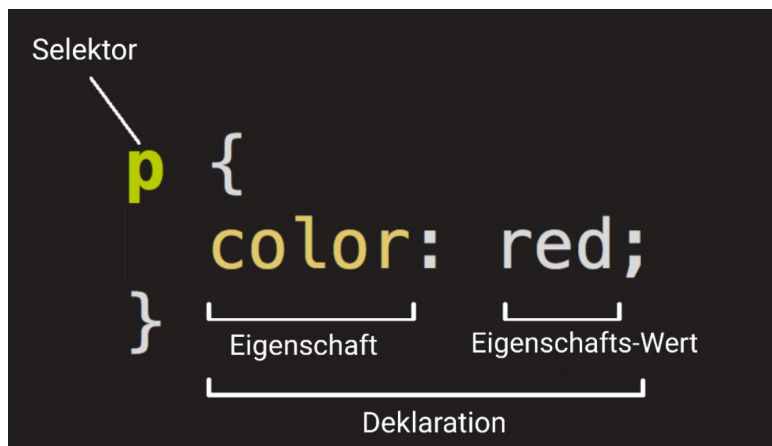


Abbildung 20: Aufbau einer CSS-Datei

Dabei wird die ganze Struktur als Regel bezeichnet.

Selektor: Durch ihn wird das HTML-Element, welches verändert werden soll, ausgewählt.

Deklaration: Darunter wird verstanden, welche Eigenschaften das Element annehmen soll.

Eigenschaft: Hiermit wird die Eigenschaft deklariert, welche ausgeführt werden soll.

Eigenschafts-Wert: Neben der Eigenschaft wird der Wert der Eigenschaft gesetzt.

### 10.2.6. Layout

Beim Grundlayout der Pages gibt es bei FTUI die Layout-Optionen Gridster, Flex und Sheet. Dies entscheidet über die Anordnung der einzelnen Widgets bzw. einzelnen Kacheln.

Das Gridster-Layout ist aufgeteilt in Reihen und Spalten. Es kann sich daher als Schachbrettmuster vorgestellt werden. Die Anzahl an Reihen und Spalten ist dabei frei wählbar. Jede Kachel, die dabei erstellt wird, kann wiederum in sich eine Kachel tragen. Die einzelnen Kacheln können auf der Benutzeroberfläche, wenn man die Funktion nicht sperrt, per Drag & Drop beliebig verschoben werden. Die Größe der Kacheln wird automatisch an den verfügbaren Platz auf dem Display angepasst. Sie kann aber auch durch einen gewünschten Festwert festgelegt werden. Grundsätzlich hat das Gridster-Layout einen Standardwert von 4x4. Die Darstellung einer Kachel erfolgt dabei über ein HTML-Listenelement, wo die Position mithilfe der

Reihen- und Spaltennummer und die Größe mit der x- und y-Achse angegeben werden kann. Links ist einmal das Standardlayout und rechts das Standardlayout mit unterschiedlichen Kachelgrößen dargestellt.

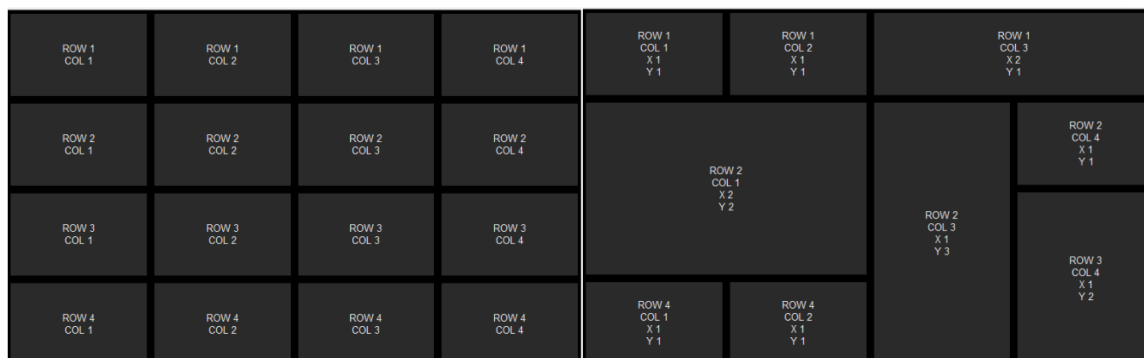


Abbildung 21: FTUI Gridster-Layout

Das Flex-Layout basiert auf der CSS Flex-Technologie. Dieses arbeitet mit dem hbox/vbox-Layout, welches mit dem vbox-Element vertikal, also übereinander, und mit dem hbox-Element horizontal, also nebeneinander, anordnet. Die einzelnen Kacheln bei dem Layout passen sich automatisch an die verfügbare Displaygröße an. Demnach werden nur die einzelnen Boxen definiert, nicht aber deren genaue Position oder Größe. Es kann demnach nur eine grobe Platzierung (oben, zentriert, unten), eine Mindest- und Maximalgröße, sowie eine vergleichende Platzeinnahme durch einen Wachstumsfaktor (Wachstumsfaktor 0 bis 9 möglich) des verfügbaren platzten, angegeben werden. Auf den Bildern wird ein horizontales Flexbox-Layout bei unterschiedlichen Bildschirmgrößen aufgezeigt.

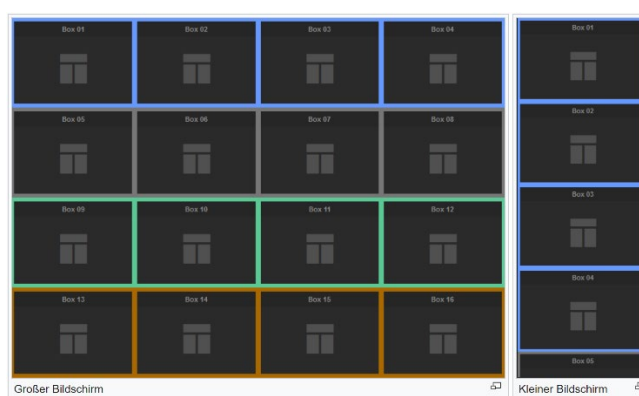


Abbildung 22: FTUI Flex-Layout

Das Sheet-Layout stellt eine Darstellung in tabellenähnlicher Form dar. Die Inhalte innerhalb des Sheet-Layouts werden horizontal und vertikal mittig platziert. Die Größe passt sich dabei stets der Bildschirmgröße an. Jedes Sheet ist dabei als ein neues „Blatt“ zu sehen, welche man mit einer beliebigen Anzahl an Reihen(row) und Spalten(cell) füllen kann. Der Aufbau eines Sheet-Layouts entspricht demnach dem sheet/row/cell Schema. Mithilfe von CSS können die Reihen und Spalten in ihrer Größe und der

Position verändert werden. Jedoch muss ein Sheet-Layout in jeder Reihe immer gleich viele Spalten bzw. jede Spalte muss immer gleich viele Reihen besitzen. Möchte man diese Struktur durchbrechen, so muss ein Sheet-Layout verschachtelt werden. Demnach wird ein Sheet-Element in ein bereits bestehendes cell-Element integriert und der Aufbau beginnt von vorne mit sheet/row/cell. Das Sheet-Layout ist aufgrund seiner starren Eigenschaft lediglich zur Positionierung von Elementen innerhalb eines Gridster- oder Flex-Layouts geeignet. Auf dem linken Bild ist ein Sheet-Layout als gleichmäßige Tabellenform von 4x4 und auf dem rechten Bild eine Verschachteltes Sheet-Layout.

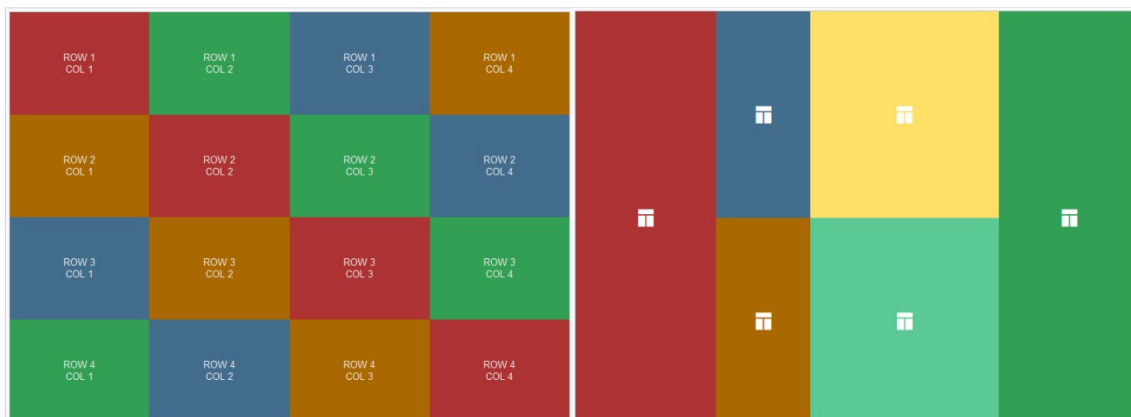


Abbildung 23: FTUI Sheet-Layout

In Abwägung der einzelnen Optionen haben wir uns für das Gridster-Layout entschieden, da es mehr Spielraummöglichkeiten bietet und in seiner Benutzung einfacher, flexibler, konstanter und vorhersehbarer ist. Dies erleichtert die Umsetzung und die Gestaltung der Benutzeroberfläche. Unser Layout sieht demnach immer gleich aus egal was für eine Bildschirmgröße vorhanden ist. Dies kann bei einem kleinen Bildschirm zwar einen Nachteil darstellen, die positiven Aspekte des Gridster-Layouts überwiegen aber gegenüber dem Flex-Layout. Im Bild sind daher 2 unterschiedliche Gridster-Layouts, links, die der Hauptfunktionalität Sturmwarnung und rechts, die der Hauptfunktionalität Photovoltaik/Balkonkraftwerk, als grafische Darstellung zu sehen.

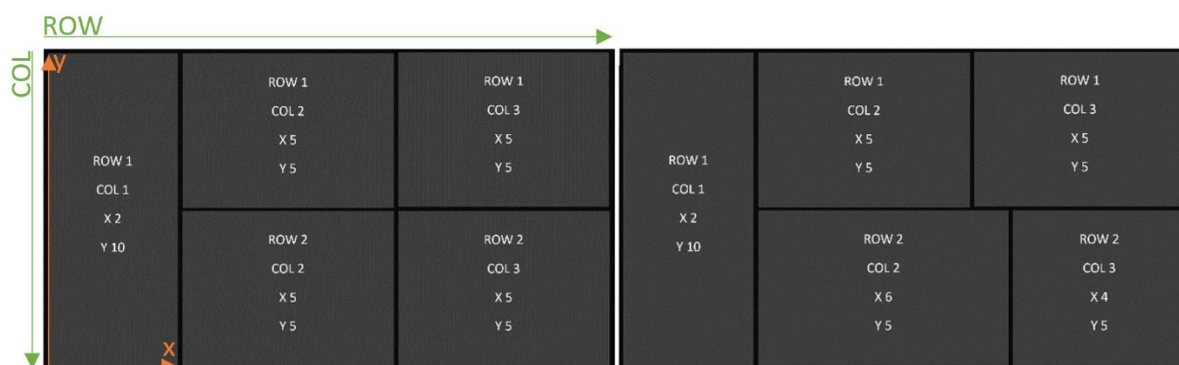


Abbildung 24: Umgesetztes Grund-Layout mit Gridster anhand der Sturmwarnung und Photovoltaik



Der 4x4 Standard hat nicht optimal zu unseren Vorstellungen der Visuellen Darstellung gepasst, weshalb wir uns aus Gründen der Einfachheit für einen fest definierter Rahmenbereich von 12X10 (ROWxCOL) entschieden haben. Dies war nach unserer Meinung die beste Aufteilung für das Anlegen und Anpassen der einzelnen Widgets.

Zusätzlich haben wir innerhalb der Kacheln das Sheet-Layout angewendet. Aufgrund der horizontalen und vertikalen Vermittlung hat dies das Platzieren der einzelnen Elemente vereinfacht. Auch bei komplizierteren Verschachtelungen war das Sheet-Layout gut zu nutzen. Im folgenden Bild ist das Sheet-Layout der Kachel Funksteckdose der Funktionalität Weihnachtsbeleuchtung grafisch dargestellt.



**Abbildung 25: Sheet-Layout innerhalb einer Kachel anhand der Weihnachtbeleuchtung**

## 11. System-Test

Um die Funktionalität des entwickelten Systems zu testen, wurden System-Tests mithilfe der User Stories und den dazugehörigen Akzeptanzkriterien getestet. Dabei wurden die Akzeptanzkriterien parallel zu dem System geprüft und dokumentiert. Die Akzeptanzkriterien wurden farblich hinterlegt. Grün bedeutet, dass das Akzeptanzkriterium erfüllt wurde und erfolgreich getestet wurde. Gelb bedeutet, dass das Akzeptanzkriterium nicht oder nur teilweise erfüllt wurde. Bei Nichterfüllung des Akzeptanzkriteriums wurden Kommentare zu alternativen Lösungen gemacht.

### 11.1. Test 2.4.1

User Stories	Akzeptanzkriterien
Als Benutzer möchte ich den Betriebsmodus „Fest definierte Betriebszeit“ auswählen, damit die Lichterketten zu einer gewünschten Betriebszeit in Betrieb sind.	Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen
	Der Betriebsmodus schält die Lichterkette per Funksteckdose um 17 Uhr ein
	Der Betriebsmodus schält die Lichterkette per Funksteckdose um 7 Uhr aus
	Sobald die Lichterkette manuell bedient wird entfällt der zuvor ausgewählte Betriebsmodus.
Als Benutzer möchte ich den Betriebsmodus „Sonnenaufgang/Sonnenuntergang“ auswählen, damit tagsüber keine Lichterkette in Betrieb ist.	Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen
	Der Betriebsmodus schaltet per Funksteckdose die Lichterkette bei Sonnenaufgang aus.
	Der Betriebsmodus schaltet per Funksteckdose die Lichterkette bei Sonnenuntergang ein.
	Der Betriebsmodus ruft die Sonnenaufgangs- und Sonnenuntergangszeiten ab.
	Sobald die Lichterkette manuell bedient wird entfällt der zuvor ausgewählte Betriebsmodus.
Als Benutzer möchte ich den Betriebsmodus „Sonnenaufgang/Sonnenuntergang und Uhrzeit an Werktagen“ auswählen, damit die Lichterketten unabhängig vom Sonnenaufgang an Werktagen spätestens ausgehen, wenn jeder das Haus verlassen hat.	Der Betriebsmodus schaltet per Funksteckdose die Lichterkette unabhängig von Sonnenaufgang an Werktagen (Montag - Freitag) spätestens um 7:30 Uhr aus.
	Der Betriebsmodus schaltet per Funksteckdose die Lichterkette bei Sonnenaufgang aus.
	Der Betriebsmodus schaltet per Funksteckdose die Lichterkette bei Sonnenuntergang ein.
	Der Betriebsmodus ruft die Sonnenaufgangs- und Sonnenuntergangszeiten ab.
	Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen
	Sobald die Lichterkette manuell bedient wird entfällt der zuvor ausgewählte Betriebsmodus.
Als Benutzer möchte ich den Betriebsmodus „Anwesenheitserkennung“ auswählen, um bei Abwesenheit zu verhindern, dass unnötig Lichterketten an sind.	Die Anwesenheit der Familienmitglieder wird über die WLAN-Verbindung des jeweiligen iPhones überprüft
	Der Betriebsmodus schaltet per Funksteckdose die Lichterkette an Werktagen bei Anwesenheit zwischen 17 und 7 Uhr ein
	Der Betriebsmodus schaltet per Funksteckdose die Lichterkette an Werktagen bei Abwesenheit zwischen 17 und 7 Uhr aus
	Prüfung auf Werktag jeden Tag um 17 Uhr
	Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen
	Sobald die Lichterkette manuell bedient wird entfällt der zuvor ausgewählte Betriebsmodus.
Als Benutzer möchte ich die Lichterketten in FHEM unabhängig voneinander manuell ein- und ausschalten, um zu entscheiden, wann ich diese in Betrieb nehmen möchte.	Der Benutzer kann die Lichterkette per Funksteckdose manuell einschalten.
	Der Benutzer kann die Lichterkette per Funksteckdose manuell ausschalten.
	Sobald die Lichterkette manuell bedient wird entfällt der zuvor ausgewählte Betriebsmodus.
	Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen
Als Benutzer möchte ich den Status der Lichterketten in FHEM einsehen, um zu überprüfen ob sie an- oder ausgeschaltet sind.	Der Benutzer kann den Status der Lichterkette per Funksteckdose einsehen.

Tabelle 12: Testprotokoll 2.4.1

Auf den ersten Blick scheint es so, als wäre bei Aufgabenstellung 2.4.1 nur die Hälfte der Akzeptanzkriterien erfüllt worden. Das liegt daran, dass es jede User-Story jeweils zwei Akzeptanzkriterien besitzt, die nicht erfüllt werden konnten.

Das erste Akzeptanzkriterium lautet „Sobald die Lichterkette manuell bedient wird, entfällt der zuvor ausgewählte Betriebsmodus“. Nun ist es allerdings so, dass die Lichterkette unabhängig vom Betriebsmodus aktiviert bzw. deaktiviert werden kann. Der Betriebsmodus „Manueller Betrieb“ muss also

nicht zwingend ausgewählt sein, um die Lichterkette manuell zu bedienen. Das Projektteam hat sich auf diese Lösung geeinigt, da ein bedingungsabhängiges Disable des Schalters nicht umzusetzen war.

Das zweite Akzeptanzkriterium, das nicht erfüllt werden konnte, lautet „Der Benutzer muss zuerst die zu steuernde(n) Lichterkette(n) auswählen“. Ursprünglich sollten zwei Lichterketten, die einmal per Funksteckdose und einmal per WLAN-Steckdose angeschlossen waren, unabhängig voneinander bedient werden. Während der Implementierungsphase ist uns allerdings aufgefallen, dass die WLAN-Steckdose große Sicherheitslücken aufweist. Um diese Sicherheitslücke zu schließen, hätte man eine neue Firmware installieren müssen. Dies war allerdings nicht möglich, da die WLAN-Steckdose einen Chip besitzt, der sich nicht flashen lässt. Das Projektteam hat sich daraufhin mit dem Auftraggeber darauf geeinigt, die Funktionalität im Zusammenhang mit der WLAN-Steckdose wegzulassen.

## 11.2. Test 2.4.2

User Stories	Akzeptanzkriterien
Der Benutzer möchte die Windgeschwindigkeit auf der FTUI Oberfläche einsehen können, um die aktuelle Wetterlage zu erfahren	Der Benutzer kann die Windgeschwindigkeit als Fließkommazahl einsehen. Zwischen FHEM und Windsensor besteht eine Verbindung.
Der Benutzer möchte die Windböe auf der FTUI Oberfläche einsehen können, um die aktuelle Wetterlage zu erfahren	Der Benutzer kann die Windböe als Fließkommazahl einsehen. Zwischen FHEM und Windsensor besteht eine Verbindung.
Der Benutzer möchte den Schwellwert der Windgeschwindigkeit auf der FTUI Oberfläche eingeben, um eine Warnung zu bekommen bei schlechtem Wetter	Der Benutzer muss die Windgeschwindigkeit als Fließkommazahl eingeben.
Der Benutzer möchte den Schwellwert der Windböen-Stärke auf der FTUI Oberfläche eingeben, um eine Warnung zu bekommen bei schlechtem Wetter	Der Benutzer muss die Windböen-Stärke als Fließkommazahl eingeben.
Der Benutzer möchte die Sturmweatherwarnung einschalten, um bei Unwetter eine Push-Notification zu bekommen	Die Sturmweatherwarnung kann aktiviert werden.
Der Benutzer möchte die Sturmweatherwarnung ausschalten, um keine Push-Notification mehr zu bekommen	Die Sturmweatherwarnung ist aktiv.
Der Benutzer möchte seine Telegram Benutzernamen auf der FTUI Oberfläche einsehen, um zu wissen an welches Handy die Push-Notification gesendet wird	Der Benutzer kann die eingegebenen Telegram Benutzernamen einsehen.
Der Benutzer möchte Telegram Benutzernamen hinterlegen, um Warnungen zu erhalten	Der Benutzer kann Telegram Benutzernamen eingeben.
Der Benutzer möchte Telegram Benutzernamen löschen, um keine Warnungen zu erhalten	Der Benutzer kann Telegram Benutzernamen löschen.
Der Benutzer möchte, dass das FHEM beim Erreichen eines der Schwellenwerte eine Push-Notification über Telegram sendet	Zwischen FHEM und Push-Notification-Server besteht eine Verbindung. Die Push-Notification Nachricht ist hinterlegt. Der gewünschte Telegram Benutzername ist hinterlegt. Zwischen FHEM und Windsensor besteht eine Verbindung.

Tabelle 13: Testprotokoll 2.4.2

Bei der Aufgabenstellung 2.4.2 wurden alle Tests erfolgreich durchgeführt und alle User-Stories bzw. Akzeptanzkriterien der Aufgabenstellung 2.4.2 konnten erfüllt werden.

### 11.3. Test 2.4.3

User Stories	Akzeptanzkriterien
Der Benutzer möchte das einfache Heizen per Infrarotheizung aktivieren, um Heizen zu können.	Das Balkonkraftwerk muss mehr als der hinterlegte Wert an Strom produzieren. Der Überhitzungsschutz muss aus sein. Das einfache Heizen muss aktivierbar sein. Das einfache Heizen muss deaktivierbar sein.
Der Benutzer möchte das einfache Heizen per Infrarotheizung deaktivieren, um das Heizen abzuschalten.	
Der Benutzer möchte die Werte im Konfigurationsdialog verändern, um die Initialwerte zu verändern.	Der Benutzer darf nur numerische Werte eingeben, um den Initialwert zu verändern. Der Benutzer kann die initial Werte einsehen. Das optimierte Heizen muss aktivierbar sein.
Der Benutzer möchte das optimierte Heizen per Infrarotheizung aktivieren, um Strom zu sparen.	Die Verbindung zwischen Temperatur-Sensor und FHEM muss aktiv sein. Das optimierte Heizen muss deaktivierbar sein.
Der Benutzer möchte das optimierte Heizen per Infrarotheizung deaktivieren, um selbständig zu entscheiden, wann das optimierte Heizen in Betrieb genommen wird.	
Der Benutzer möchte die Werte im Konfigurationsdialog verändern, um die Initialwerte zu verändern.	Der Benutzer darf nur numerische Werte eingeben, um den Initialwert zu verändern. Der Benutzer kann die initial Werte einsehen. Der Benutzer kann den Warmwasserboiler manuell einschalten
Als Benutzer möchte ich den Warmwasserboiler über die FTUI manuell ein- und ausschalten können, um selbständig zu entscheiden wann dieser in Betrieb sein soll.	Der Benutzer kann den Warmwasserboiler manuell ausschalten Der Benutzer kann den Betriebsmodus durch die Inbetriebnahme eines anderen Betriebsmodus deaktivieren Der Benutzer kann den Betriebsmodus auswählen
Als Benutzer möchte ich den Warmwasserboiler für einen fest definierten Zeitraum in Betrieb nehmen, um ihn für den benötigten Benutzungszeitraum nicht manuell ein- bzw. ausschalten zu müssen.	Der Benutzer muss den Zeitraum zur Inbetriebnahme definieren/festlegen Der Warmwasserboiler wird zum definierten Endzeitpunkt automatisch ausgeschaltet Der Warmwasserboiler wird zum definierten Startzeitpunkt automatisch eingeschaltet Der Benutzer kann den Betriebsmodus durch die Inbetriebnahme eines anderen Betriebsmodus deaktivieren Nach Beendigung des definierten Zeitraumes ist der Betriebsmodus nicht mehr aktiv Der Benutzer kann den Betriebsmodus auswählen
Als Benutzer möchte ich den Warmwasserboiler nur zu gewünschten Stromtarifen in Betrieb nehmen, um Stromkosten zu sparen.	Der Benutzer muss den definierten Stromzeiten in der Stromzeitentabelle eines der folgenden Preiskategorien zuweisen (günstiger/mittlerer/teurerster Preis pro kWh) Der Benutzer muss die Stromzeiträume definieren können. Die Stromzeitenwerte müssen initial, wie in der Aufgabenstellung, hinterlegt sein Der Benutzer kann bei Inbetriebnahme des Betriebsmodus zwischen den Tarifzeiten "günstigster Strompreis" und "günstigster/mittlerer Strompreis" auswählen. Der Benutzer kann den Betriebsmodus durch die Inbetriebnahme eines anderen Betriebsmodus deaktivieren Der Warmwasserboiler schält sich automatisch ein, wenn die ausgewählte Tarifzeit durch die Angaben in der Stromzeitentabelle eintritt Der Warmwasserboiler schält sich automatisch außerhalb der ausgewählten Tarifzeit aus (siehe Tabelle) Der Benutzer kann den Betriebsmodus auswählen
Als Benutzer möchte ich den Warmwasserboiler nur zu günstigen Stromtarifzeiten oder mit Solarstrom in Betrieb nehmen können, um Stromkosten zu sparen.	Der Benutzer muss die Stromzeiträume in (Stunden/Minuten) definieren können Die Stromzeitenwerte müssen initial, wie in der Aufgabenstellung, hinterlegt sein Der Benutzer muss den definierten Stromzeiten in der Stromzeitentabelle eines der folgenden Preiskategorien zuweisen (günstiger/mittlerer/teurerster Preis pro kWh) Der Warmwasserboiler schält sich automatisch nach der Stromzeitentabelle bei "günstiger Stromzeit" oder beim Erreichen des definierten Schwellenwerts für Solarstrom ein Der Warmwasserboiler schält sich automatisch nach der Stromzeitentabelle bei nicht "günstiger Stromzeit" oder bei Unterschreiten des Solarstromschwellenwerts aus Der Benutzer muss den Schwellenwert für den Solarstrom als Einschaltzeitpunkt für den Warmwasserboiler definieren Im Intervall von 5 Minuten wird der erzeugte Solarstrom geprüft und mit dem Vergleichswert zur Einschaltung verglichen Der Initialwert für die Einschaltung des Warmwasserboiler beträgt > "400" Watt Solarstrom Der Benutzer kann den Betriebsmodus durch die Inbetriebnahme eines anderen Betriebsmodus deaktivieren
Als Benutzer möchte ich, dass der Boiler und die Heizung mit einer Priorisierung Solarenergie beziehen.	Wenn die Heizung in der Überhitzungsschutzphase ist wird der Boiler gespeist werden.
	Zwischen 5 und 8 Uhr (Duschzeit) fließt Energie in den Boiler, solange niemand die Heizung nutzt. Boiler kann mittels Taster auf Oberfläche für 2 Minuten aktiviert werden. (Quick Shower)
Als Benutzer möchte ich eine statistische Auswertung des erzeugten Stroms auf der FTUI-Oberfläche einsehen, um die Produktivität meines Balkonkraftwerks zu überprüfen.	Der Benutzer soll die Daten in einer übersichtlichen Darstellung einsehen können.
	Das Aktualisierungsintervall für den aktuell erzeugten Strom soll in Minuten konfigurierbar sein Das Aktualisierungsintervall soll initial auf 1 Minute hinterlegt sein
	Folgende Daten sollen in der Darstellung ersichtlich werden: Aktuell produzierter Strom, Aktueller Tag/Vorhergehender Tag, Aktuelle Woche/Vorhergehende Woche, Aktueller Monat/Vorhergehender Monat, Aktuelles Jahr/Vorhergehendes Jahr Die Daten von "Aktuell produzierter Strom" und "aktueller Tag" werden in dem definierten Aktualisierungsintervall aktualisiert
Als Benutzer möchte ich eine visuelle Darstellung in Form eines Balkendiagramms des erzeugten Stroms auf der FTUI-Oberfläche einsehen, um die Produktivität meines Balkonkraftwerks graphisch abgleichen zu können.	Der Benutzer soll die Daten in einer übersichtlichen Darstellung in Form eines Balkendiagramms einsehen können.

Tabelle 14: Testprotokoll 2.4.3

Wie man sieht, konnten die meisten User-Stories bzw. Akzeptanzkriterien der Aufgabenstellung 2.4.3 erfüllt werden. Die Akzeptanzkriterien der User-Story „Als Benutzer möchte ich, dass der Boiler und die Heizung mit einer Priorisierung Solarenergie beziehen“ wurden nicht erfüllt, da eine andere Logik implementiert wurde. Die User-Story wird aber trotzdem erfüllt. Der Grundgedanke ist folgender:

Wenn nicht genügend Solarstrom zur Verfügung steht, um Infrarotheizung und Wasserboiler gleichzeitig zu betreiben, soll nur die Infrarotheizung betrieben werden.

Die Priorisierung liegt hierbei auf der Infrarotheizung. Genau diese Funktionalität wurde umgesetzt.

Die Akzeptanzkriterien der User-Story „Als Benutzer möchte ich eine statistische Auswertung des erzeugten Stroms auf der FTUI-Oberfläche einsehen, um die Produktivität meines Balkonkraftwerks zu überprüfen“ wurden nur teilweise erfüllt. Der Benutzer kann die Daten zwar einsehen, jedoch werden diese nur einmal täglich aktualisiert. Aus diesem Grund kann der Benutzer weder das Aktualisierungsintervall konfigurieren noch die Daten des aktuellen Tags einsehen, da immer nur der vorhergehende Tag angezeigt werden kann.

## 12. Installationsanleitung

Bei der Installation kann man sich zwischen zwei verschiedenen Alternativen entscheiden. Die erste Alternative wäre die Schritt-für-Schritt-Anleitung, mit der man das Smart-Home-System nachbauen kann. Die zweite Alternative ist ein Installationsskript, mit dem man Teile der Installation automatisch durchführen kann.

### 12.1. Schritt für Schritt Anleitung

Der Vorteil dieser Schritt-für-Schritt-Anleitung ist, dass man einen Lerneffekt erzielt und dass man das nachgebaute System je nach Belieben an einigen Stellen konfigurieren kann. So kann man beispielsweise Raumbezeichnungen, Attribute oder Bot-Namen konfigurieren. Die Schritt-für-Schritt-Anleitung ist unter nachfolgendem Link zu finden:

<https://github.com/doenisf/HomeAutomationProjektGruppe2/tree/main/Schritt-F%C3%BCr-Schritt-Anleitung>

### 12.2. Schnellinstallation

Der Vorteil der Schnellinstallation ist, dass man die Installation deutlich schneller durchführen kann als mit der Schritt-für-Schritt-Anleitung. Es können keine Fehler beim Abtippen der Befehle oder beim Durchführen der Schritte entstehen. Auch der Zeitaufwand ist vergleichsweise gering gegenüber der manuellen Installation. Der Nachteil ist, dass man wenig Konfigurationsmöglichkeiten während des Installationsprozesses hat. Das FHEM-System lässt sich allerdings natürlich nachträglich noch konfigurieren. Den Link zum Schnellinstallationsskript findet man hier:

<https://github.com/doenisf/HomeAutomationProjektGruppe2/tree/main/Schnellinstalltionsskript>

Bei Fragen zur Schritt-für-Schritt-Anleitung oder zum Schnellinstallationsskript stehen wir jederzeit unter folgenden Kontaktdaten zur Verfügung:

Benjamin Leicht	<a href="mailto:wi19134@lehre.dhbw-stuttgart.de">wi19134@lehre.dhbw-stuttgart.de</a>
Ion Tabyrca	<a href="mailto:wi19069@lehre.dhbw-stuttgart.de">wi19069@lehre.dhbw-stuttgart.de</a>

Tabelle 15: Kontakt Support

## 13. Abschließende Betrachtung

Im Rahmen der abschließenden Betrachtung werden im Unterkapitel Lessons Learned zunächst die, im Laufe des Projektes gesammelten, Erfahrungen des gesamten Projektteams dargestellt. Anschließend wird das Thema Smart Home im Ausblick behandelt. Im Fazit wird ein Resümee zur Aufgabenstellung, zum Projekt und zum Ablauf gezogen.

### 13.1. Lessons Learned

Im Rahmen des letzten Teammeetings wurde kurz vor Projektabgabe ein Lessons Learned mit dem gesamten Team durchgeführt. Dabei wurden positive und negative Aspekte gesammelt und zusammengefasst. Das Rekapitulieren zum Ende des Projekts soll den Projektmitgliedern helfen die Arbeit bei künftigen Projekten zu erleichtern.

Es wurden die folgenden positiven Aspekte gesammelt:

- Hohe Einsatzbereitschaft, hohes Engagement der Beteiligten
- Von Anfang an konsequent am Projekt gearbeitet
- Projektmanagementmethode mit regelmäßiger Absprache und Fortschrittskontrolle
- Jeder hat sich an Implementierung und an der Doku beteiligt
- MS Teams, und weitere Kollaborationstools haben für Zusammenarbeit gut funktioniert
- Demokratische Abstimmungen, alle Projektmitglieder waren zu jederzeit auf einer hierarchischen Ebene
- Problemlose, schnelle, konstruktive Absprache mit Auftraggeber
- Teambuildingevent der gesamten Gruppe in Präsenz hat zu guten Klima geführt
- Bildung von kleinen Expertengruppen; sehr gute und rasche Kommunikation
- Allgemeine Transparenz des Projekts, z.B. durch Apple-Kalender, Conceptboard mit Abwesenheiten, öffentliches Kanbanboard und Kummerkasten
- Sowohl bei der Anforderungsanalyse als auch bei der Implementierungsphase haben sich Expertenteams selbstständig gefunden
- Projektergebnis
- Realitätsnahes Projekt hinsichtlich des Zeitdrucks und der Aufgabenstellung
- Arbeiten mit Linux als OS ist interessant und kann relevant für Arbeitsleben sein
- Zusammenarbeit im Team hat Spaß gemacht
- Conceptboard war gut; Kummerkasten, Abwesenheiten,
- Dokumentation ist nicht hinten runtergefallen; es wurde konsequent daran gearbeitet
- Teammitglied war 1,5 Wochen krank und es hat nichts ausgemacht, da dies vom restlichen Team kompensiert werden konnte
- Zeitmanagement war in der Summe gut/ok (sind in der Summe gut durchgekommen)

- Debatten waren sehr konstruktiv und auch zeitlich in einem effektiven Rahmen
- Projektablauf war gut, haben alles gemacht, was man machen sollte
- Mindset des Teams war gut
- Projektcommitment des Teams war gut, da so einige Entscheidungen schneller getroffen werden konnten
- Beteiligung war großartig (niemand der keinen Bock hatte)
- 2 Projektleiter festzulegen war gut, da es sich um ein Teilzeitprojekt neben Studium handelte und so sichergestellt werden konnte, dass immer ein Projektleiter bei Terminen verfügbar war
- Lessons Learned wurde zum Ende des Projekts mit dem gesamten Team durchgeführt, um für weitere Projekte zu lernen

Außerdem wurden die folgenden negativen Aspekte festgestellt:

- Aufwand wurde zu Beginn teilweise falsch eingeschätzt
- Aufgabenstellung zu zeitaufwendig; entspricht nicht dem dafür vorgesehenen Aufwand
- Aufgabenstellung widersprüchlich geschrieben; könnte präziser sein
- Keine Vorortmeetings; mehr Zusammenhalt im Team und bessere Absprachen
- Fürs Testen wurde zu wenig Zeit eingeplant und es gab keine klare Vorgehensweise oder einen standardisierten Testablauf. Daher sollte das Testen höher priorisiert werden, auch zeitlich sollte dies ernsthafter angegangen werden, da man Erfahrungen machen muss, um die Implementierungsqualität durch Tests zu steigern
- Entwickler Tests könnten besser sein, z.B. durch einen kompletten Entwicklertest nochmal nach dem Entwickeln, um "Vergessenes" minimieren
- Dailys waren manchmal länger als 10 Minuten
- FHEM ist nicht anwenderfreundlich
- Projekt hat sehr geschlaucht, teilweise allein gefühlt; Aufgaben werden an Teammitglieder abgeschoben
- FHEM, Perl scheint nicht „state of the art“ zu sein
- Smart Home als Thema ist interessant, aber nicht ganz studiennah. Etwas näheres am Studienplan wäre besser. Es könnten auch die Studenten gefragt werden, was implementiert werden könnte.
- Im Projektteam war die Erwartungshaltung da, dass wir die gesamte Freizeit opfern; Erwartungshaltung an uns, dass jeder immer an dem Projekt arbeitet, war nicht realistisch und erzeugte Druck
- Beim Implementieren zu nachlässige Behandlung der Schritt für Schritt Anleitung
- Zu wenig Teambuilding
- Kanbanboard wurde von einigen Personen nur nachlässig und nach Aufforderung gepflegt
- Teilweise ungenutzte Zeit in den Dailys

- Motivationsmanagement zum Ende; Motivation sollte konstanter hochgehalten werden
- Manche Themen wurden zu lange mit herumgeschleppt; es wurden Entscheidungen zu spät getroffen
- MS Teams ist an vielen Ecken toll, aber auch userunfreundlich und wenig intuitiv
- Conceptboard war nicht userfreundlich, evtl. wäre ein anderes virtuelles Whiteboard besser

## 13.2. Fazit

Nach dem 8-wöchigen Projekt hat das Projektteam deutlich dazugelernt. Es war nicht nur auf der fachlichen Ebene, sondern auch auf der Sozialkompetenzebene eine Herausforderung. Die Thematik des Smart Home hat den ein oder anderen Projektteilnehmer zwar schon in seinem Privatleben tangiert, das Entwickeln eines Perl-basierten System mit FHEM war allerdings für alle Projektteilnehmer Neuland. So musste sich jedes Projektmitglied Wissen zu FHEM, Perl und FTUI von Grund auf aneignen. Diese Umstände führten dazu, dass es auf der einen Seite verzweifelnde Momente gab, in denen man nicht wusste, wie man weiter machen soll, auf der anderen Seite war das Erfolgserlebnis umso größer, wenn eine Anforderung umgesetzt werden konnte.

Die Projektteilnehmer konnten das bisherige Wissen im Bereich des Smart Homes ausbauen. Das Verständnis bezüglich der Implementierung eines eigenen Smart Home konnte verbessert werden. Es kann nun besser abgeschätzt werden welche Funktionen mit einem Smart Home realisiert werden können und welche Aufwände dazu erforderlich sind.

Des Weiteren wurden sehr wichtige Grundlagen der Anwendung FHEM autodidaktisch erlernt. Diese ermöglichen es den Projektteilnehmern in Zukunft, beispielsweise im Rahmen der Implementierung eines privaten Smart Homes, die verwendeten Module zu optimieren. Außerdem könnte das Smart Home mithilfe der Anbindung von weiteren FHEM-Modulen erweitert werden.