NOVIA
UNIVERSITY OF APPLIED SCIENCES

FINAL REPORT

Vamiaeasy Smart Home Living

European Project Semester

SALtech
Senior Automated Living Technologies
solutions offered by vamia EASY

Authors:
Estefanio Kesto
Hugo Frauciel
Ties Kolthof
Tobias van Varik

Semester:
Autumn 2021

Supervisor:
Hans Lindén

This page is intentionally left blank

Table of Contents

Table of Contents 1
Table of Figures 5
Table of Tables 8
Acknowledgements 9
Abstract 10
1. Introduction 11
1.1 European Project Semester (EPS) 11
1.2 Vamiaeasy Simulation Room 12
1.2.1. Vamia School 12
1.2.2. The healthcare problem 12
1.2.3. Vamiaeasy 12
1.3. Team Members 13
1.3.1. Estefanio Kesto (United States) 13
1.3.2. Hugo Frauciel (France) 13
1.3.3. Ties Kolthof (Netherlands) 13
1.3.4. Tobias van Varik (Netherlands) 14
1.4. Teambuilding 14
1.4.1. Belbin Questionnaire 14
1.4.2. Sauna 16
1.4.3. Dinner & Lunch's 16
1.4.4. Teambuilding Class 16
1.5. Team Name 17
1.6. Team Logo 18
2. Project Management & Background Information 19
2.1. Purpose & Objective 19
2.2. RACI & Work Breakdown Structure (WBS) 20
2.2.1. RACI: 21
2.2.2. Explaining the roles of all the group members with respect to WBS 21
2.3. Project Scheduling 22
2.4. Risk Assessment 23
2.5. Budgeting Time and Money 24
2.6. Design Brief 26

2.7. Project goals 29

2.8. Stakeholders 30

2.9. System Requirements..31

3. Research 33

3.1. Internet of Things (IoT) 33

3.1.1. What is Internet of Things? 33

3.1.2. Why and How to use Internet of Things?..33

3.1.3. How IoT will be Implemented in this Project 33

3.2. IoT Devices...34

3.3. IoT Gateways 35

3.3.1. What is an IoT Gateway? 35

3.3.2. IoT Gateway Options: openHAB Vs. HomeAssistant 35

3.4. Home Assistant & openHAB...36

3.5. The ABB free@home System 37

3.5.1. General information...37

4. Bill of Materials 39

5. Visualization of the Vamiaeasy Showroom..43

5.1. Home By Me – 3D model..43

5.1.1. Realization of the 3D model 43

5.1.2. Limitations of this software 44

5.2. Showcasing the Vamiaeasy Room via Prezi..45

5.2.1. Description of the Prezi tool...45

5.2.2. Realization of the final Prezi – Vamiaeasy Showroom 45

6. Vamiaeasy's ABB free@home System 48

6.1. Getting Access to the free@home Configuration Environment 48

6.2. The free@home Configuration Environment 50

6.2.1 House setup...50

6.2.2. Devices 51

6.3. VAMIAeasy's free@home Configurations 53

7. Automating Motorized Cabinet..54

7.1. ESP32-Triggered Instant-on Relay Circuit 55

7.1.1. First automating cabinet circuit configurations (no success)............................55

7.1.2. Final automating cabinet circuit configuration (success) 56

7.2. Different Cabinet Automation Scenes 57

7.2.1. Preliminary Information 57

7.2.2. Utilizing Cabinet Controller as Originally Designed 58

7.2.3. Bringing Cabinet Completely Up/Down in a Single-Button-Click Action 58

7.2.4. Bringing Cabinet Completely Up/Down Wirelessly 58

7.3. Mitigating Faults and Errors 59

7.3.1. Buttons Pressed Synchronously 59

7.3.2. If Down Button is Held Down and then Up Button is Pressed Once 59

7.3.3. If WiFi is Disconnected from the ESP32 59

8. Google Nest 60

8.1. Configuring Google Nest 60

8.2. Connecting ABB free@home to the Google Nest 61

9. HomeAssistant (HA) 62

9.1. Home assistant installation guide 62

9.1.1. Step-by-Step Instructions for Installing HomeAssistant 62

9.2. Integration of SmartThings into HomeAssistant 64

9.3. Integration of Google Nest into HomeAssistant 65

9.4. Integration of Netatmo into HomeAssistant 66

9.5. Integration of free@home into HomeAssistant 67

9.6. Home Assistant Community Store (HACS) integration 70

9.7 Concluding HomeAssistant 73

10. OpenHAB 75

10.1. Installation & Configuration of openHAB 75

10.1.1. openHAB Installation 75

10.1.2. openHAB Configuration 76

10.2. Moving Forward with openHAB 79

10.2.1. Integrating OpenHAB into home assistant. 79

11. Project Management Evaluation 80

11.1. WBS 80

11.2. Risk assessment 81

11.3. Scheduling 82

11.4. Budgeting 82

12. Tracking Hours 84

Conclusion **85**

Recommendations for Prospective Group **86**

Making Non-Smart Roomba Vacuum into a Smart Vacuum 86

Adding Cabinet Up/Down into the free@home System 86

Utilize Existing Devices for Data Visualization 86

Contacting Maricare Health Sensors 87

Fixing the Casting of Homeassistant into the Google Nest Hub 87

Samsung Smart Fridge Live Images Integrated into Homeassistant 88

Z-Wave and Zigbee Integrations 88

References **89**

Table of Appendices **93**

Table of Figures

Figure 1: Vamiaeasy living room and kitchen....................12
Figure 2: Belbin chart of all the members of the team....................15
Figure 3: Brainstorming of team names....................17
Figure 4: The team logo....................18
Figure 5: WBS overview....................20
Figure 6: Brief snip of our Gantt Chart....................23
Figure 7: Earned Value Analysis....................26
Figure 8: HomeAssistant logo [16]....................36
Figure 9: openHAB Logo [15]....................36
Figure 10: HomeAssistant dashboard example [17]....................37
Figure 11: openHAB Dashboard example [19]....................37
Figure 12: Logo Home By Me....................43
Figure 13: 3D model of the Vamiaeasy showroom created via HomeByMe....................44
Figure 14: Logo Prezi....................45
Figure 15: Prezi home page of the Vamiaeasy room....................45
Figure 16: "List of devices" and "Plan" bubbles....................46
Figure 17: "Living Room" bubble....................46
Figure 18: "ABB" bubble....................47
Figure 19: "Lights" bubble....................47
Figure 20: Finding the router which a PC is connected to....................48
Figure 21: Router device list example....................49
Figure 22: Login page for System Access Point....................49
Figure 23: ABB free@home main menu....................50
Figure 24: ABB free@home house configuration....................50
Figure 25: Connecting ABB free@home devices....................51
Figure 26: Locates scenes/groups in the ABB free@home systems....................52

Figure 27: ABB dashboard panel 53

Figure 28: Motorized cabinet with up & down controller 54

Figure 29: ESP32-Triggered instant-on relay circuit schematic......56

Figure 30: Blackbox circuit for automation of cabinet 57

Figure 31: Google Nest Logo [33] 60

Figure 32: Getting Started with Google Home Assistant 60

Figure 33: Balena etcher home page......62

Figure 34: SmartThings Token Generator 64

Figure 35: SmartThings icon in HomeAssistant integrations 65

Figure 36: Confirmation of successful SmartThings integration 65

Figure 37: Configuring Google cast into HomeAssistant 66

Figure 38: Locating configuration tab in HomeAssistant 66

Figure 39: HomeAssistant dashboard with the Netatmo data 66

Figure 40: Terminal & SSH addon for free@home integration into HomeAssistant 67

Figure 41: Configuration credentials for free@home integration into HomeAssistant 67

Figure 42: Winscp file transfer screenshot. 68

Figure 43: winSCP password and connection 68

Figure 44: Exhanging filed to raspberry pi through WinSCP 69

Figure 45: Confirmation of free@home HomeAssistant integration 69

Figure 46: Inputting free@home information for HomeAssistant integration 69

Figure 47: HACS device activiation......70

Figure 48: HACS integration prompt 70

Figure 49: Entering device activiation code for HACS integration......71

Figure 50: Authorizing HACS 71

Figure 51: Confirmation of properly integrated HACS......71

Figure 52: The first iteration of the HomeAssistant dashboard 72

Figure 53: The HomeAssistant dashboard for the iPad 72

Figure 54: The HomeAssistant dashboard for the google nest hub......73

Figure 55: image of how the display looks in the room 73

Figure 56: Logging into Raspberry Pi..76

Figure 57: openHAB Welcome Screen...76

Figure 58: Getting to openHAB WiFi set up 77

Figure 59: WiFi set up 77

Figure 60: openHAB homepage as well as help and info commands 78

Figure 61: Earned value analysis chart 82

Figure 62: Excel sheets 84

Figure 63: Summary of total hours worked per person 84

Figure 64: iRobot Roomba 606 Buttons...86

Table of Tables

Table 1: Planned Value……………………………………………………………………25

Table 2: Design Brief for the Vamiaeasy EPS Project...26

Table 3: Requirements Sorted into Different Priorities 32

Table 4: ABB free@home devices..40

Table 5: Smart Devices 41

Table 6: No Smart Devices..42

Acknowledgements

The Vamiaeasy EPS group would like to acknowledge Yrkeshögskolan Novia for accepting us into their EPS program and giving our team the opportunity to work in collaboration with Vamia on this respective project. We would like to give an explicit mention to the individuals working directly with our team throughout the durartion of this EPS program.

- Hans Linden: Acting as our project mentor and working diligently/patiently with our team for the duration of the project.
- Roger Nylund: Providing the autumn 2021 EPS students with the team building activities and inviting us to his house for a phenomenally cooked dinner.
- Mikael Ehrs: Sharing his wealth of knowledge on project management for the autumn 2021 EPS group.
- Hanna Kuusisto: Giving the EPS group a fundamental understanding on cultural differences and the way on life in Finland. Additionally, guidance on how to write academic papers.
- Anu Ylikahri: Working with the Vamiaeasy EPS group on behalf of Vamia and providing us with the necessary resources and information in order for our team to accomplish what is being asked of us.

Additionally, we would like to express our gratitude to the tutors and Katja Bonäs who helped all the exchange students during the travels from their home countries to Vaasa. Without their help it would have been a difficult and frustrating process getting settled here in Vaasa.

It has been an absolute pleasure here at Yrkeshögskolan Novia and in the welcoming city of Vaasa.

Sincerely,
The Vamiaeasy EPS Group

Abstract

A simulated home environment which students at the Hansa campus of the Vamia vocational school have created is known as Vamiaeasy. And it should be exactly that, an all-around *easy* to use automated home environment. Many of us can agree that seniors are not so comfortable when interacting with technology, so creating a convincingly intuitive platform this automated living environment for seniors is the first step in making such a solution realizable in the real-world.

This EPS group was able to realize the vision of the Vamiaeasy living environment by creating an intuitive user-interface. An application was created through HomeAssistant which integrated many devices that once used different applications. Now instead of having to go from one application to another, the user of this living environment is now able to control their home through one application created by the Vamiaeasy EPS group.

The team also created some visualization renders of the Vamiaeasy room and automating devices which once were not envisioned to be automated. The visualization tools did not previously exist and were created through the homebyme and prezi services. Furthermore, a motorized cabinet which did not impress an audience in demonstration purposes was innovated further to leave a good impression when being demonstrated to people. This report will outline the teams time here in Vaasa and dive into the intricate details of their progress to realize the objectives mentioned thereof.

1. Introduction

1.1 European Project Semester (EPS)

The European Project Semester is a collaborative learning program within the European Union for undergraduate students studying within the sciences, technology, engineering, and mathematics. There are 19 host institutions across 13 countries that make up this program today. This program is project-based and many times sponsored by companies in industry which gives students the opportunity to apply their theoretical studies into the real working world. [1]

For the duration of the EPS program, students work in teams of three to six students which are multinational and interdisciplinary. The key elements of the EPS project are bulleted below:

- Semester lasting a minimum of 15 weeks
- Working in the English language
- Courses including
  - Teambuilding
  - Project Management
  - Local language of respective country
- Focus being respective project which is a minimum of 20 ECTS
- Project supervision focusing on project productivity as well as product in itself
- Continuous assessment which includes interim and a final report

The students are presented the respective projects at the beginning of the semester in which they later choose their preferences. This year the EPS administrators at Yrkeshögskolan Novia had given their students three project options: Vamiaeasy, Health Care Sensors, and Aquaponics. All the projects mentioned thereof are utilizing big data, internet of things, automation, and other aspects of software/mechanical/electrical engineering.

1.2 Vamiaeasy Simulation Room

1.2.1. Vamia School

Vamia is a vocational school in Vaasa. The school offers vocational education to young adults who are 16-19 years old. It offers education in healthcare, energy, technology, business, welfare, tourism, and general education. Vamia has created an automated living environment known as Vamiaeasy. The Vamiaeasy room is a simulated apartment of the future for elderly or disabled persons.

1.2.2. The healthcare problem

Vamia easy is a room that is built to teach students the health care of the future. The number of elderly people will increase globally. According to the United Nations, the population of elderly people has doubled since 1980 and will increase rapidly to 2,1 billion people who are older than 60 years [2]. This form of healthcare which Vamia is demonstrating provides technological benefits for the coming elderly.

Unfortunately, the future provides not enough nurses. Due to the aging of western societies, fewer young nurses must take care of more elderly. So, the question is: How do western societies take care of the aging problem?

The answer is technology. Companies have developed different solutions for the elderly so they can keep living in their own homes instead of retirement houses. Examples of technologies are sensors and robots. Fall sensors can notice whether an elder person has fallen or not. Robots can help elderly persons in their daily routine [3].

1.2.3. Vamiaeasy

Vamia has collected technologies in the Vamiaeasy room. Multiple smart devices have been set up by technical students of Vamia in the guidance experts. Smart devices can communicate via WiFi or bluetooth. The whole list of devices can be found in chapter 4. The purpose of these devices is education and demonstration. Vamiaeasy is being used to help the students develop their skills so elderly people can live with all the conveniences of the 21st century. The Vamiaeasy room consists of different rooms: the living room, the kitchen, the bathroom, and the bedroom. Smart devices are divided into these different rooms.

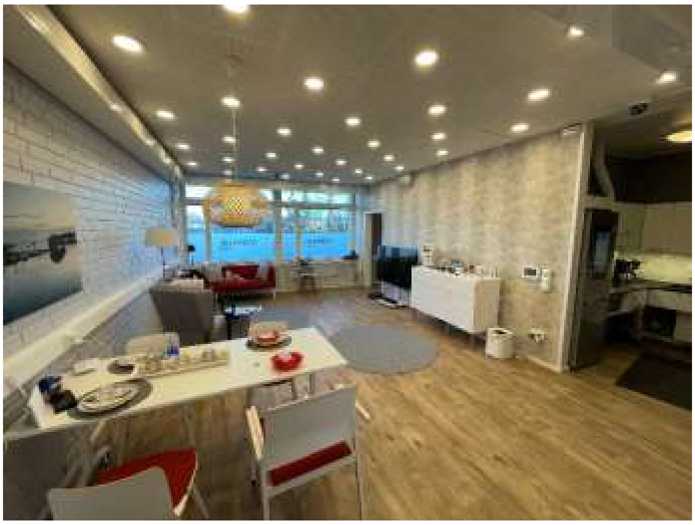


Figure 1: Vamiaeasy living room and kitchen

1.3. Team Members

We have included an introduction to each member which makes up the Vamiaeasy EPS team. Each team member will mention their home institution, field of study, strong skills, why they chose the EPS program, and any other miscellaneous information they would like to include about themselves.

1.3.1. Estefanio Kesto (United States)

I am an electrical engineering student at Michigan Technological University which is located in Houghton, Michigan. My degree concentration is in photonics – the physical science of light detection, manipulation, and generation. During my time at Michigan Technological University, I have gained a variety of engineering skills such as circuit analysis, signal analysis, control systems, digital logic, hardware programming, and photonic systems.

I chose to join the EPS program mainly because it fulfils my senior design requirement back in the states which is a huge incentive. In addition to this, I wanted to gain a fundamental understanding of the education systems which are outside of the United States and wanted to work with a group of international students to increase my cross-cultural awareness.

1.3.2. Hugo Frauciel (France)

I am a general engineering student at the ENIT (National Engineering School of Tarbes). I have knowledge in the fields of Mechanical Engineering, Industrial Engineering and Civil Engineering. My main competence is civil engineering thanks to two 6 months internships in a general contracting company and in steel construction.

I joined an EPS program in order to work in a team on a interdisciplinary project. In addition, it allows me to be aware of interculturality, to have a new industrial experience, to develop my language and communication skills. Other EPS were offered but I chose Finland to discover a new culture and enjoy the beautiful Nordic landscapes.

1.3.3. Ties Kolthof (Netherlands)

I am a Mechatronical engineering student at Saxion University of Applied Sciences in the Netherlands. I have knowledge of mechanical engineering, electrical engineering and some computer science and programming. I have done a Minor in Robotics and Vision also at Saxion. My main skills are in combining all these different engineering parts into one and knowing a bit about all of them.

I joined the EPS because I wanted to work in a team that was multidisciplinary and multicultural. I find it really interesting to work with people from all over the world with different backgrounds and experiences. And why I came to Finland is because I always wanted to visit the Nordic countries and they have a big emphasis on renewable energy.

1.3.4. Tobias van Varik (Netherlands)

I have gained experience of project management during my studies. My field of study is industrial engineering and management. I have knowledge about project management, designing industrial processes, and business-related subjects. My home university of university is located in ‘s Hertogenbosch.

The EPS program has been an excellent choice for me. It was my wish to work with different cultures and to meet people with different point of views. My choice is based on the country and not on the study. I want to experience the Scandinavian (yes, I consider Finland as Scandinavia) way of living and how they respect nature. In comparison with my own country, it is quite the opposite.

1.4. Teambuilding

This section will discuss the teambuilding activities which the Vamiaeasy EPS team engaged in for the duration of our project. Some of these activities which will be mentioned were done in our own time and in the teambuilding course.

1.4.1. Belbin Questionnaire

The Belbin questionnaire is used to find the behavioural strength and weaknesses of each individual team member on a respective project. Upon completing the questionnaire, each member will have specific ratings on the type of role they mostly coincide with. The team roles and their descriptions which a Belbin questionnaire rates are mentioned below [4].

- Resource Investigator: Being proactive in finding new ideas for their projects
- Team worker: Holds the team together and identifies work to be complete
- Coordinator: Focuses on team’s objectives and delegates work to team members
- Plant: Highly creative and good at solving complex problems with a unique approach
- Monitor: Judges and evaluates team’s work and opinions
- Shaper: Bring the drive necessary to complete the end-goal and mitigate loss of focus
- Implementer: Plans a realistic strategy which is efficiently carried out
- Finisher: Combs through the fine details of end tasks to ensure high standards of quality

This form of rating allows each team member to learn about their peers’ personalities, skills, abilities, values, motivations, and influences. Our teams’ ratings may be found below.

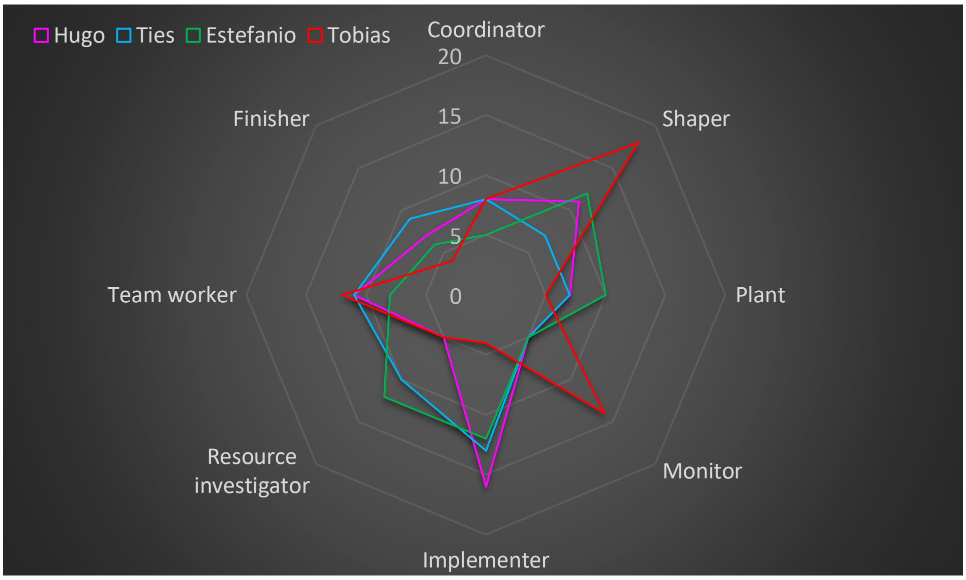


Figure 2: Belbin chart of all the members of the team

The results of Belbin questionnaire are displayed in Figure 2. It highlights the strong and the weak points of the project members. The Belbin questionnaire has revealed noteworthy subjects. Every project member is a team worker. This means that every group member has experience in working in project groups.

There are no weak spots in this group when all the group members are combined. The only score with less than ten points is the finisher. This could be a problem at the end of the project. The preference roles are presented in the next lines:

Hugo's roles: Implementer, Shaper, Team worker.
Ties's roles: Investigator, Team worker, finisher.
Estefanio's roles: Resource investigator, Implementer, Plant.
Tobias's roles: Shaper, monitor, team worker.

Estefanio's qualities can be used at the beginning of the project. His proactive attitude is useful for this project. He finds new ideas and can come up with the best approach. For this reason, Estefanio had been the group leader before the midterm. Before the midterm had been passed, Hugo assisted Estefanio as secretary. Hugo is a shaper and an implementer. These roles fit with the last stages of the project.

After the midterm, Ties has been appointed as the group leader. He is the best finisher in our group. These skills could be used to finalize the project. Tobias acted as Ties' secretary following the midterm. He has got the role to monitor and shape the ideas and results. This critical mindset is necessary to build a good product.

1.4.2. Sauna

Saunas are a part of the Finnish culture and play an important role in the national identity of Fins [5]. Accordingly, our team has been engaging in this cultural aspect of the Finnish way of life. The sauna sessions allow us to relieve and take our minds off the normal everyday stresses; it is a chance to step away from all technology and truly engage in deep and meaningful conversations with each other. This sauna subsection is included within our teambuilding because it is probably the strongest teambuilding activity which we have engaged in. During these sauna sessions we have been exposed to each other's cultural differences and parental backgrounds. Through these sauna sessions we have been able to build a close friendship within our direct EPS group.

1.4.3. Dinner & Lunch's

Almost every school day our team gathers to enjoy a nice break for lunch on Novia's campus. Teambuilding dinners or lunch's act as a great teambuilding medium to promote team member engagement. We are able to share ideas, thoughts, laughter, and conversion which in turn creates a tighter team bond and may lead to better performances within our group.

The dinner which Roger hosted for the whole EPS group at a lake house was quite a memorable experience for us all. During this visit to the lake house, we had the opportunity to try, as what Roger calls it, a *true* Finnish sauna. This sauna was a wood burning sauna which was set directly right next to the sea. While the EPS students were enjoying the sauna next to the sea, Roger was preparing a phenomenal salmon dinner with some side dishes. Upon the sauna session being complete we went into the lake house to enjoy Rogers great salmon dinner with all the EPS students.

Another scheduled dinner was hosted by Hugo and Pierre in the form of a pizza party. They invited their respective groups, being the health care sensors and Vamiaeasy group. During this dinner we engaged in friendly discussions and social drinking. We appreciate Hugo and Pierre for setting this gathering up.

1.4.4. Teambuilding Class

In the team building class, which was under the guidance of Roger, we engaged in many teambuilding activities. Of these, the most memorable was a day where we had to complete multiple tasks within our respective EPS groups. Some of the tasks were building a tall newspaper tower, completing a trivial map, carrying a ball on a binder ring connected to four strings to a separate corner of the room, and drawing a cat using a pen connected to four strings. This form of teambuilding forced us to really work as a team to complete these respective tasks and was quite an enjoyable time.

In addition to these teambuilding activities, we were tasked with creating a team name and logo, which in essence is a teambuilding activity in itself. The team's name and logo are further discussed in the coming sections.

1.5. Team Name

Creating a team logo and name is a requirement set by the administrators of the EPS program [6]. A team name and logo allow a brand to express its personality which sets themselves apart from other competitors. In our instance for the EPS program, coming up with a team name and logo was a sense of teambuilding which allowed us to learn how creative our respective team members can be.

After a brainstorming session within our group, we agreed to set SALtech as our official team. SALtech is an abbreviation for "(S)enior (A)utomated (L)iving (tech)nologies" which we found to be catchy and briefly describes the solutions we are looking to provide for our prospective customers. Other potential team names which we came up with can be found in Figure 3.

1. SeniorAuto
2. ALS (Automated Living Solutions)
3. Auto Senior Living
4. Senior Living Technologies
5. Living Autonomy
6. SeniorTech

Figure 3: Brainstorming of team names

None of the team names shown in Figure 3 truly resonated with our team. We then combined the names of 1 and 4 shown in Figure 3 where we all had the "yes, this is it" moment and came up with the catchy abbreviation SALtech.

1.6. Team Logo

For the team logo there were a few ideas. One was with designing a man in a rocking chair holding a tablet. But eventually the team decided on using a cat for usage in the main logo. This is mainly because in the introduction video of the Vamiaeasy room there is also a cat used.

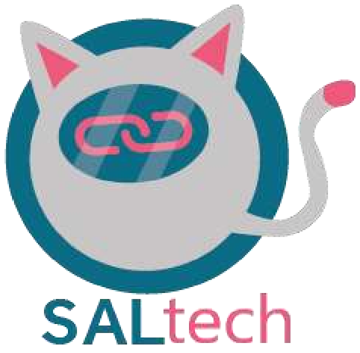


Figure 4: The team logo

So, the final logo as you can see in Figure 4, uses the colours of Vamia. A nice detail in the logo also can be seen in the eye of the cat. The impression of connectivity is given by these almost linked chains. Giving the sense the project involving internet and connectivity. Then on the bottom of the log is the name of our project team visible. With the first three letters in a different colour than the latter four.

2. Project Management & Background Information

2.1. Purpose & Objective

Vamia has developed an automated living environment for the elderly. The main use of this room will be to demonstrate to Vamia healthcare students and instructors how healthcare is moving towards automation which will allow them to become familiar with these future technologies. This automated living environment has been designed and created by their students in collaboration with industry leaders. This form of automated living is to provide solutions that will allow individuals to live in the comfort of their own homes for as long as possible.

Currently, Vamia students have already installed smart devices and hardware for the automation of the living environment in a real-world apartment. One of the main problems with these smart devices is that they use different platforms to control them, resulting in a multitude of mobile/computer applications. Our EPS team is tasked with integrating the smart devices within this simulation room to limit the number of mobile/computer applications used to control the automated living environment. The solutions that our EPS team comes up with will be used by the instructors and healthcare students at Vamia.

The instructors use such an environment to cast certain scenarios of an elderly patient which will allow healthcare students to put their skills learned in the classroom to use in a real-life situation. In addition to being used to cast scenarios for instructional purposes, it is also used to demonstrate the types of solutions which can be used for assisted living purposes. This demonstrative barrier-free living solution will be able to showcase how large of an impact automated living can have in somebody's life.

Not only does this type of demonstration give the next generation of healthcare workers a taste of how future senior and disabled home healthcare is going to be automated, but it may also be used to incentive smart device manufacturers to add more compatibility to their respective devices.

2.2. RACI & Work Breakdown Structure (WBS)

The Vamiaeasy EPS team has broken the project into five main categories: Integration, Google Connection, User Interface, Project Management, and Documentation. These different tasks have been displayed in Figure 5 below. The entirety of the WBS can be found in Appendix 6 on page 94.

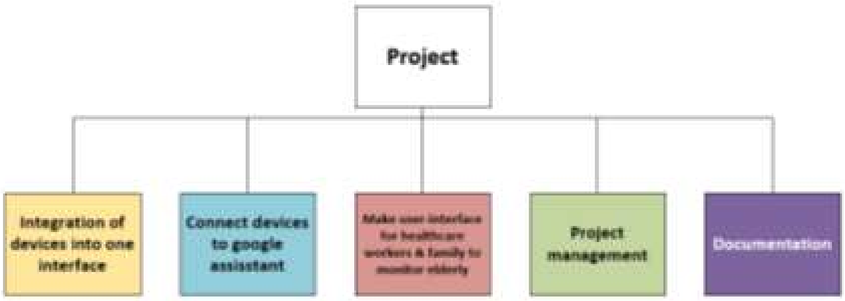

Project
Integration of devices into one interface
Connect devices to google assistant
Make user interface for healthcare workers & family to monitor elderly
Project management
Documentation

Figure 5: WBS overview

We have been tasked to integrate multiple smart home devices and sensors into one user-interface so there are not different applications controlling an automated living environment. To first achieve this, we must find a platform to integrate and control the smart home devices into. We chose to use a raspberry pi with different IoT hubs installed on them: openHAB and/or HomeAssistant. We chose these two different IoT hubs because we would like to test which one will be the best for our purposes. In addition, an intuitive interface needs to be designed according to the stakeholders' wishes.

Project management tools are then used to run the project efficiently. These project management tools are derived from the course "project management for EPS."

Documenting all the progress made on the respective tasks mentioned in the preceding paragraphs is an integral part of this project. This will allow for the next EPS group who is working on this project to pick up where we left off with ease. The project management task allows our team to ensure we deliver what is being asked of us to our upmost capabilities. Our team assigned the roles of these respective tasks based on the background knowledge of everybody within our team. These roles may be found in the responsibility matrix.

2.2.1. RACI:

The responsibility matrix is a tool to bring structure and clarity to the roles that every person plays in a project. It determines responsibility and objectives that can be done by an individual [7]. The responsibility matrix has 4 distinct roles that it defines:

- Responsible: This person is doing a specific task in the project and is making the decisions.
- Accountable: This person is accountable for the work that has been done. So, the owner of the project. This person must approve of the done tasks.
- Consulted: Person who is consulted for information or that can give input to the project.
- Informed: These are stakeholders that need to be informed on what is going on. Or they need to be kept in the picture.

In our project, we divided our responsibility matrix into a few sub-matrices. The following matrixes have been defined: Connect devices to google assistant, integration of devices into 1 interface, make a user interface for healthcare workers and family to monitor the elderly, project management, documentation. These different categories were divided among the team members according to their study background and may be found in the appendices.

2.2.2. Explaining the roles of all the group members with respect to WBS

- Integration of devices into one interface: This part of the project has been considered the toughest part of the project. Ties and Estefanio have the most (limited) knowledge among the group. Tobias and Hugo will be kept up to date.
- Connect devices to google assistant: The devices will be connected to google assistant by Hugo. Information is easily gained on the internet. The ABB device will be connected by Tobias. Tobias and Hugo will be consulted by Ties and Estefanio.
- Make user-interface for healthcare workers & families to monitor elderly: The interface will be designed by Ties.
- Project management: The tools for the project management program are being created by Tobias and Hugo. The project management tools are familiar to Tobias. Tobias will be assisted by Hugo. The other group members need to be informed.
- Documentation: Every group member will document their findings. In the end, all the documents need to be compiled by Estefanio. He is a native English speaker and makes the fewest language mistakes. Everybody takes a part in the midterm/end presentation. Feedback will have been given before the end presentation starts.

The work breakdown structure and RACI tables can be found in Appendix 6 (page 94) and Appendix 7 (page 94), respectfully.

2.3. Project Scheduling

Thanks to the creation of a Gannt chart, we created a general schedule for our project. Each colour is representative of the different stages of the project:

- 🟦 Start Project: Code of conduct, Teambuilding, General research on ABB
- 🟦 Project Plan: Defining project, RACI, WBS, Make stakeholder management
- 🟪 System Requirements: Analysing problem, Defining requirements
- 🟧 Functional Design: Choosing operating system pi, Designing flow of data
- 🟨 Realisation: Connecting maximum of devices to google home
- 🟩 Final Report: Working on final document, Working on presentation

We also added three other legends:

- 🟩 The weekly meetings on the last line of the table
- 🟥 The important dates: deadline
- 🟦 The presentation for the end of the project

Before creating our final Gantt chart, we introduced our WBS. The tasks we had determined were not optimal and did not fit in the right way in this Gantt chart. So, we reviewed all the tasks to be done in order to better structure the project planning over the whole semester.

In addition, we have added arrows to indicate the prerequisites for certain tasks. This makes it possible to determine which tasks should be carried out first or which ones can be started much earlier. In Figure 6 shown below, an overview of the Gantt chart is shown. The legends of the beginning of this section (2.3) could be used to identify the stage of the project.

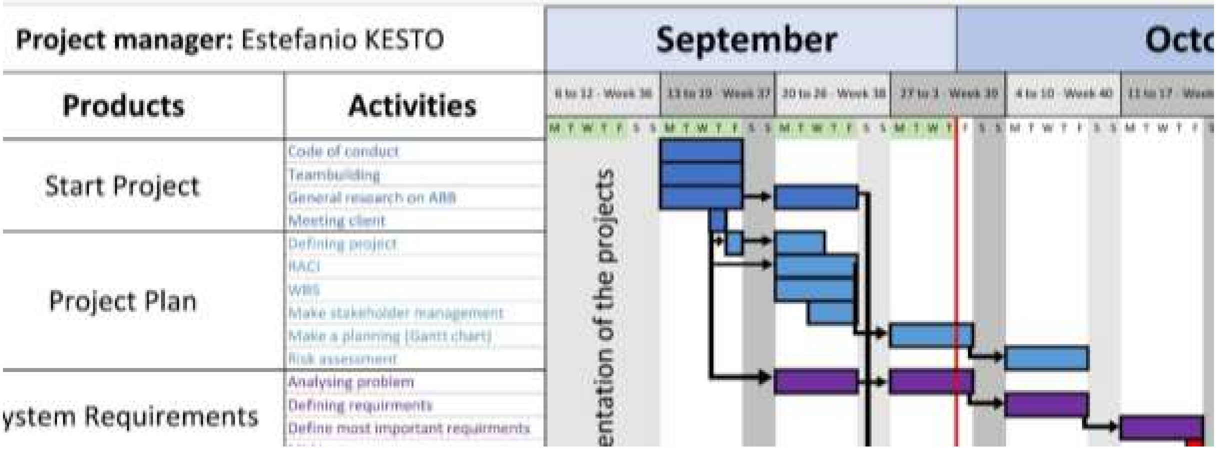


Figure 6: Brief snip of our Gantt Chart

This Gannt chart in the Figure 6 has been composed by the project group. This figure just displays an example the beginning of the Gannt chart. The whole Gannt chart is displayed in Appendix 1 on page 94.

2.4. Risk Assessment

Each group member created a list of potential risks to carry out the risk analysis for our project. These lists were then put together and a total of 24 different risks were found. We decided to score these risks according to two criteria: the probability of it happening and the consequences it would have on our project. These two criteria were on a scale of 5 with 1 most least probable and 5 being most probable.

Once all the risks were scored, we applied the following formula to rank them from highest to lowest priority: the probability score multiplied by the consequence score. Finally, we obtained the ranking that you can find in Appendix 2 on page 94.

To reduce the potential risks of our project, the method would be to find a solution to each of these risks. The solution could be done after the problem has occurred, but in some cases, it can be done beforehand to avoid it. So, the team decided to find a solution to our 3 most important risks.

- Problem 1: The room is always occupied by Vamia students or teachers.
  - Solution: Make a list of all the devices and have as much information as possible on them (model, serial number…) so that you can work on them without having to be physically present in the room
- Problem 2: The password for free@home is not available.
  - Solution: If the school is not able to retrieve this password, we will have to reset the entire system and set it up all over again. To counter this risk, we will try to give the school an ultimate date upon which they need to find the password. If they have not found it by that date, the project team will proceed with resetting the system and start over again. This is done so the team does not waste too much time waiting for the password.
- Problem 3: Covid-19 restrictions
  - Solution: Provide a cloud space with as much information as possible to work independently. Also use platforms for group work like Zoom.

The risk assessment could be found in Appendix 2 on page 94.

2.5. Budgeting Time and Money

To realize the earned value analysis (EVA) of our project, we used all the tasks of our Gantt chart. We first set an hourly rate for each task, considering the social and health insurance expenses of the staff. Then we completed the planned value, actual cost, and earned value by following our Gantt chart as closely as possible, assigning achievable and suitable schedules to each task.

- Planned Value (PV): This is the first element of Earned Value Management. Planned value is the value assigned to a job within an allotted time. It is the value to be achieved throughout the project. It is important to calculate this planned value before you do the work to have a baseline. Once all the tasks are planned, we can obtain a final budget for our project.
- Actual Cost (AC): This is the second element of earned value management. The total actual cost is the total time spent on each task to date. In other words, it calculates the total amount of our project to date. It is very simple to calculate this cost, just multiply the time spent on each task by its hourly rate.
- Earned Value (EV): This is the third and final element of earned value management. It is the value of the work we have done to date. When the project is finished, we will know the exact value of the work we have done, in other words the value produced by the project. This value is the most important, it indicates what you have earned from the money you have spent so far.

The planned value could be found in Appendix 3 on page 94.

The actual cost could be found in Appendix 4 on page 94.

The earned value could be found in Appendix 5 on page 94.

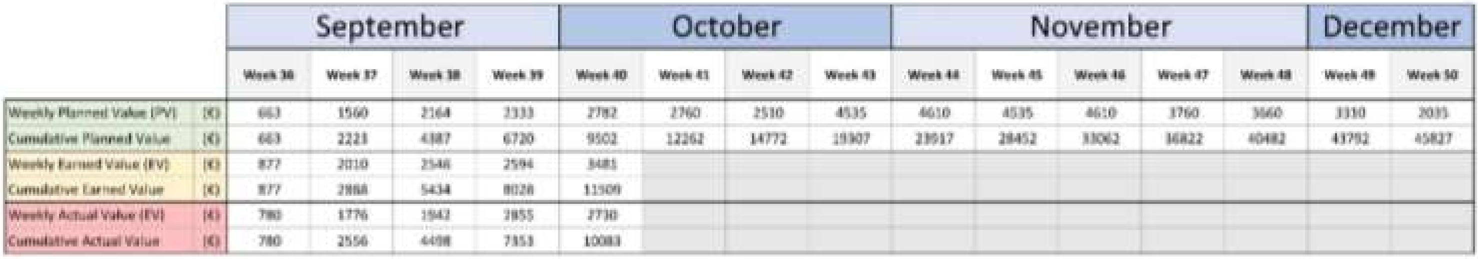
The financial results of the project can be found in the table below this paragraph. The planned and actual data are based on two estimations:

- <u>Planned worked hours (Q (hours))</u>: How many hours will be spent on the task? This depends on the different tasks. The tasks have already been discussed in paragraph 2.2. The schedule is displayed in a Gantt chart in chapter 2.3. The hours are divided as planned or as actual hours. The planned hours are used for the planned value and the actual hours for the actual cost.
- <u>Costs for an hour work (€/hour)</u>: Not every task costs the same amount of money. One task could be very difficult to execute, but the other task could be the opposite. Different competencies are needed for different tasks. These competencies could be found on the job market, but these competencies differ in salary. This is the reason, an estimation of euros an hour has been made for every individual task.

The following formula has been used for the results in the table:

Value = €/hour × Q (hours)

Table 1: planned value



		September				October				November					December	
		Week 36	Week 37	Week 38	Week 39	Week 40	Week 41	Week 42	Week 43	Week 44	Week 45	Week 46	Week 47	Week 48	Week 49	Week 50
Weekly Planned Value (PV)	(€)	663	1560	2164	2333	2782	2760	2510	4535	4610	4535	4610	3760	3660	3310	2035
Cumulative Planned Value	(€)	663	2223	4387	6720	9502	12262	14772	19307	23917	28452	33062	36822	40482	43792	45827
Weekly Earned Value (EV)	(€)	877	2010	2546	2594	3481										
Cumulative Earned Value	(€)	877	2888	5434	8028	11509										
Weekly Actual Value (EV)	(€)	780	1776	1942	2855	2730										
Cumulative Actual Value	(€)	780	2556	4498	7353	10083										

Sometimes, work has already been put in, but the has not been finished by the project group. This progress should be noted without considering it as a completed task. This is the reason another method is used to monitor the progress without under or overestimating it.

- <u>Percentage</u>: Percentage of the task has already been finished.
- <u>Planned value</u>: The planned value results are based on the calculations of the planned value table.

Actual value = % of the task × planned value

The data out of the table is displayed in Figure 7 shown below. The graph shows us that the project is ahead of the planning halfway. This planning has been made by the project group (see Gantt chart in chapter 2.3).

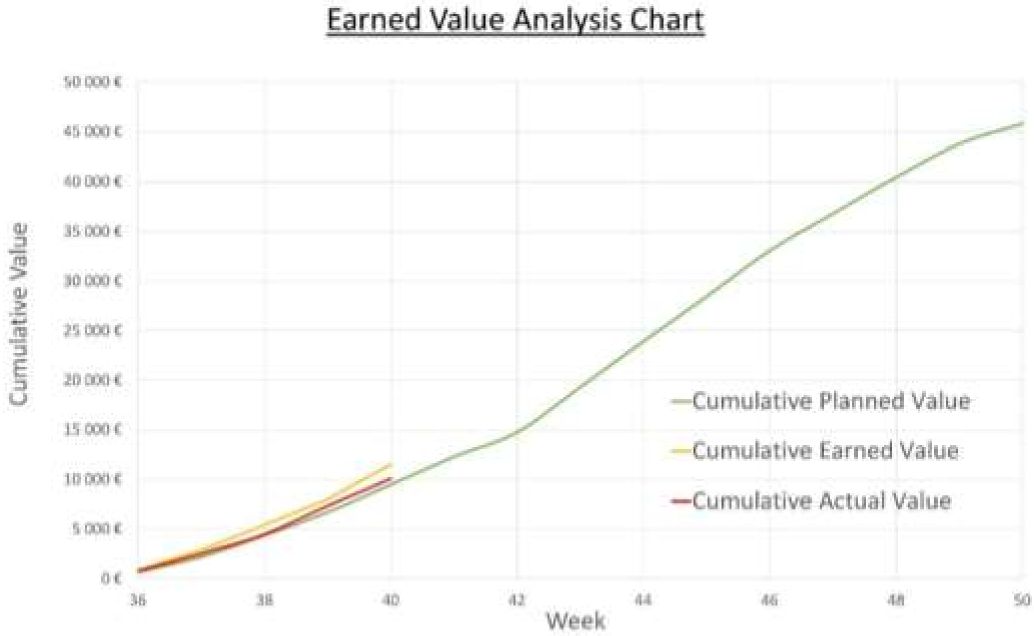


Figure 7: Earned Value Analysis

2.6. Design Brief

The Vamiaeasy EPS team decided to create a design brief which summarizes a bulk of what this project entails. This design brief was not created out of a requirement from the project management class; rather, we believed it was a good method to help us get the summarize the key components for our project.

For those who are unfamiliar with what a design brief is we will explain. This is another type of project management tool which outlines the specifics of a respective project. There are not set-in-stone rules as to what must be included, which at times make a design brief frustrating to create but it still quite advantageous. A design brief, at a minimum, should cover the project overview, scope, and user/audience information [8]. Our brief is found below.

Table 2: Design Brief for the Vamiaeasy EPS Project

Project Description	**What is the problem or opportunity?**
	• Problem: Too many apps controlling devices within automated home

	• Problem: Usability of devices is not intuitive for elderly/nurses to use • Problem/Opportunity: Integration of devices utilizing different servers/platforms • Opportunity: Minimize the number of apps controlling devices (ideally one app, but may not be possible) • Opportunity: Ease the usability of devices for elderly/nurses
	Describe the problem / opportunity in a few sentences as you would in an elevator pitch.
	• Problem: The Vamiaeasy concept is beneficial for elderly living, but currently is not intuitive for the elderly and healthcare workers to use. • Problem: The integration of the devices into one user-interface are causing complications due to the constraints of the device manufacturers • Opportunity: We search for solutions to make the user-interface intuitive to use for non-technical individuals • Opportunity: We will integrate as many devices as possible into a minimum number of user-interfaces.
Scope	**What is within the scope of the project and what is outside it?**
	• Within: Ease the usability of devices • Within: Minimizing apps being used to control devices • Within: Usability for able-bodied people • Outside: Usability for technically incompetent users • Outside: integrate all devices into **one** user-interface • Outside: low and mid-level programming (which may be required) • Outside: No privacy regulation
Constraints	**What constraints must you work within?**
	• Not having full physical access to automated living environment • Minimized interaction with the customer (Vamia instructors/students) • No available budget to make project related purchases • Possible restriction to internal servers at Vamia

	• Limited group member knowledge of software programming
Target Users	**Who are you designing for?**
	• Vamia healthcare Instructors • Vamia students • Health care workers • Elderly • Family of elderly using facility
	Be specific as you can – who are they and what is important for you to know about them?
	• Vamia Instructors → Instructors of future healthcare workers. What are their current frustrations in the automated home at Vamia? • Vamia students → If the system is fully integrated the Vamia students are also able to make full use of the system. This is by practicing healthcare using a smart home to monitor the elderly. • Health care workers → They are able to work with tablets, phones and other smart devices. However, they do not have the knowledge about software. So, they cannot fix any IoT related problems. • Elderly → The elderly have problems with their health. Physical disabilities are quite normal for elderly people (deaf, blind, mute and immovability) • Family of elderly using facility → The family of the elderly could also be given access to the system. So, they receive some information about how their family is doing in the smart home.
Exploration questions	**What key questions will you need to answer through your research?**
	• Is it possible to integrate the third-party devices into the current ABB system? o If not, how should we go about integrating these third-party devices? • Which IoT hub is the best for our devices and uses? • How do we connect these devices to the IoT hub of our choice? • What is the end-user looking for in the “intuitive” user-interface?

	Are there ecosystem or regulatory concerns (device, server, platform, location, or personal restrictions)?
	• Sensors working on different platforms. • Different platforms possibly being impossible to integrate on same user interface • No access to school servers • Minimal access to automated home environment.
Expected outcomes	**What outcomes would you like to see?**
	• A fully integrated system with all the healthcare sensors working and communicating with one another. • All devices controlled from 1 or 2 app which is user friendly and intuitive to use for the instructors, students, nurses, and the elderly. • Furthermore, an outcome in which both the client (Vamia) and our supervisors are happy about the end result. And there haven't been any misconceptions about the project goal.
Success metrics	**How will you measure success?**
	• Satisfaction from project coach (Hans) and Anu • Minimization of apps controlling automated home • Successfully connecting as many devices as possible to google home • Fully documenting bill of materials for devices used at Vamia and ABB

2.7. Project goals

➢ Create a bill of materials:

- List all available devices and become familiar with them
- Obtain as much information as possible about these devices through documentation and research
- Create a list with links, description, functioning, type of communication (for future teams)

➢ Integrate devices into the ABB platform:

- Make the Raspberry pi work with the ABB system access point
- Integrate as many non-ABB products as possible into the ABB platform using the Raspberry pi

- Find out if the ABB free@home system is suitable for connecting non-ABB devices

- List ideas for the final interface:
  - Choose an easy-to-use IoT platform
  - Manage as many devices as possible from a single interface
  - Be able to visualise the data collected by smart devices
  - Be able to intervene remotely (turn on the light, the heating…)
- Deliver a prototype:
  - Demonstrate to interested stakeholders
  - Simulate use on tablet, computer…
  - Design the interface
  - Validate or not the project before programming
- Programming the application:
  - Search and choose a software to create our application

2.8. Stakeholders

Vamia: Anu Ylikahri is our contact at Vamia, our client. At the first meeting, Anu was enthusiastic about our arrival. A good cooperation between Novia and Vamia is considered desirable. The highly qualified people in the field of our project are not present in their vocational school. That is why the presence of students from Novia was required.

Nursing students: Speaking from our own experiences as students, new ideas and technologies have been rejected when something goes wrong. With this in mind the idea is to deliver a solution that can set a good impression for the students that have to work with the room. Therefore, the solution should work very smoothly in the first demonstration.

Teachers (Vamia): Teachers know the ABB system. We could learn a lot from them. We must also take into account that this room will be mainly used as a showroom, which means that it will be more demonstration oriented. So, it must be easy to teach with one application, while at the same time making the students want to use these devices.

Health care workers: For them too, caring for the elderly should become easier. By providing them with a technological environment, the interface should be easy to use and intuitive, with the possibility of intervening remotely or being alerted in case of problems.

ABB Company: Oscar is Vamia's contact at ABB. He could provide us with product information. ABB is involved in this project because of the amount of ABB devices in the room.

Novia: We have a mentor (Hans) who supervises us throughout the project. The mentor is our contact when the project group runs out of material during the project. The task of coaching the students has been given to the mentor. It is his job to help us in difficult moments and to enthuse us during the project. This project could be a seed for new cooperation projects in the future.

The elderly: We could test our product with the elderly. We should provide an intuitive and easy to use interface with legible (large enough) writing. We should also link the emergency button to our final system.

Relatives of the elderly: They need to be informed when something goes wrong. The house tracks the behaviour of the elderly and their movements. The family could also be informed about what is happening in the house.

2.9. System Requirements

In this part of the document the requirements for the final solution are defined. These requirements are defined from the customer meeting that have been held with our costumer Anu. The most important defined requirements are as follows:

1. **At least 4 smart devices integrated and working**
   To accomplish this requirement, we need to integrate 4 smart devices into the system. This meaning that 4 different devices can be controlled using only one master that can send commands and control these devices.
2. **Reduce the number of apps that need to be used**
   Reducing the number of apps comes from the wish of anu to have less apps to control everything. So, the main aim here is to have 1 app that controls all of the devices.
3. **A program that integrates other applications in one interface**
   This requirement states that we must create a program that integrates the other applications into one. With this meaning that we need to choose/design an application that is able to handle all the applications into one.
4. **Have the Smart fridge also connected to the system.**
   The smart fridge is from Samsung and therefore Android operated. It can also connect to the smart thing's application offered by Samsung. And the plan is to integrate this application into the main interface. So, the Smart fridge would make a nice addition to the complete picture of a smart home.
5. **Create an application with interface for demonstration**
   For this requirement, the plan is to create an application or user-interface that can be used to control the smart devices in the room. This user-interface is mainly used for showroom application where it can be easily used for demonstrations.
6. **Have the system working on android operating system**
   If there will be an application created, the most promising option will be to create this application for the Android operating system. This is because the Android

operating system is more user-friendly for making new applications. And it has significantly less restrictions than for instance the apple IOS operating system.

7. **The Google assistant working**

   The google home assistant is a voice operated assistant in your house. For this requirement to succeed the google assistant must be able to control the smart devices in the home. This must all be controlled by voice and would be a great achievement in turning the home into a true automated home.

8. **The system working on the ABB interface**

   The ABB interface controls the majority of the smart systems in the house. For example, all the lights and window blinds are controlled by the ABB free@home system. This requirement is fulfilled when 3rd party devices can be controlled from the ABB free@home system.

9. **Have the system working on the apple IOS platform**

   After there is a working application on the Android platform. The next step will be to try and create one for the apple IOS platform. This is so the apple users can also control the system.

10. **Create a notification system for healthcare workers or family**

    Create a way to notify the family or healthcare workers when the person living in the smart environment needs the presence of a physical person.

These requirements can be further categorized in Must have, should be included and Nice to have. When looking at the following table, the importance all the requirements is displayed here.

Table 3: Requirements Sorted into Different Priorities

Must have	Should be included	Nice to have
At least 4 smart devices integrated and working	Have the Smart fridge also connected to the system	The Google assistant working
Reduce the number of apps that need to be used	Create an application with interface for demonstration	The system working on the ABB interface
A program that integrates other applications in one interface	Have the system working on android operating system	Have the system working on apple operating system
		Create a notification system for healthcare workers or family

3. Research

3.1. Internet of Things (IoT)

Because the nature of this project is heavily utilizing the Internet of Thing (IoT); our team devoted some time digging into the what, why's, and how's of the Internet of Things.

3.1.1. What is Internet of Things?

IoT describes an embedded system of a group of things (devices) which are connected and exchanging data mainly over the internet. Hence, the term Internet of things. There are a multitude of fields which have collectively enabled the use of IoT. Such fields include computing, control systems, embedded systems, machine learning, and wireless sensor networks. [9].

3.1.2. Why and How to use Internet of Things?

Attraction to IoT has been continuously increasing because how advantageous it is for civilian, industrial, and commercial systems. This is because different smart devices many times utilize different protocols in their systems. Using an IoT platform for automation purposes significantly decreases the complexity of integrating smart devices which utilized different protocols into one easy-to-use interface [10]. This ease of integration of smart devices has led to the great attraction toward IoT.

The first step in harnessing the benefits of the IoT is to find an appropriate device. Some devices will be further discussed in section 3.2. The IoT device which is chosen acts as the main communicator to all your other devices [11]. What is meant by this is that all the devices which you want to integrate into one user-friendly interface will be communicating to the IoT device. When a device chosen an IoT hub must be selected which allows for one to customize their user-interface to their liking.

The IoT hub which one chooses is highly dependent on their respective applications [10]. IoT hubs are going to be further discussed in section 3.3. This IoT hub is what takes care of all the background software processing which is needed for the external devices to communicate to your IoT device. In addition, the IoT hub is what allows the user to customize the user-interface and convertibility to their liking.

3.1.3. How IoT will be Implemented in this Project

The Vamia smart home currently has multiple smart devices which all utilize different applications. These different applications are a nuisance because the smart home is geared most towards elderly and disabled living. IoT will be used to integrate the usage of all the respective smart devices into one intuitive user-interface. In addition, the user-interface should be able to be used by the family and healthcare workers of living within such an automated home.

3.2. IoT Devices

To start with explaining what IoT devices are, we first need to understand what kind of devices we may use for IoT purposes. So, an IoT device is a device that has supported internet connectivity and is able to communicate with other devices across the internet [11]. It is also able to give users remote access to devices to manage their needs.

Some examples of IoT devices are smart refrigerators or smartwatches and smart fire alarms. You also have smart bicycles, door locks and lights. This can go on endlessly, so you can see there are a load of ordinary devices that can be made smart. Mentioned below are the two very well-known IoT devices used in the Vamiaeasy home.

- Google home speaker:

This one is most definitely an IoT device. This is because the device has the ability to control almost all devices in your house. It can control your thermostat in your home as well as your TV, lights, and media player. Furthermore, this speaker is compatible with a load of IoT hubs what makes it very useful when used on a more professional basis.

- Phillips Hue:

This is a Lighting system developed by Philips for smart homes. These lights are excellent for setting the ambience in every room wirelessly. These lights are also able to synchronise with your movies or music that you are watching or listening to. Next to this it also has the ability to be used as a wake-up light.

Although the two above devices are considered IoT devices, they do not assist our team in accomplishing the task of integrating third party devices into one user-interface. Through research our team found that using a microcontroller or single-board-computer as our main IoT device for the other smart devices to communicate to would be more advantageous [11]. A microcontroller board is more advantageous for generic off-the-shelf hardware/sensors while a single-board-computer is more advantageous for so called “blackbox” hardware which companies already have communcation protocals for [11].

For the reasons mentioned above, the better option for our team was the raspberry pi. This is because many of the devices we have to integrate into one user-interface are already utilizing their own protocals. After finding which respective IoT device our team would like to use as the main communicator between the different smart devices, we began to search for different IoT gateways to implement onto our raspberry pi, which is further discussed in section 3.3.

3.3. IoT Gateways

Upon finding an appropriate IoT device, being the raspberry pi, which would allow us to integrate different smart devices into one user-friendly interface, our team began doing further research into different IoT gateways.

3.3.1. What is an IoT Gateway?

An IoT gateway acts as the main point of communication between the smart and IoT devices. All the data moving through the smart devices and cloud are passing through such an IoT gateway. It does this by acting as a network router which is routing the data between the smart devices and cloud. This type of gateway is generally able to handle both inbound and outbound data transfers. [12].

There are different IoT gateways which have different functionalities. For example, an individual using IoT for industrial purposes will utilize a different gateway than somebody utilizing IoT for home automation purposes. Some of the popular IoT gateways available are Samsung smarthings, Amazon Alexa, Google Home, Apple Homepod, HomeAssistant, and OpenHAB [13]. However, out of these six options, only HomeAssistant and openHAB offer integration of a variety of devices. In other words, Samsung smarthings, Amazon Alexa, Google Home, and Apple Homepod only offer integration of devices which they are compatible with while openHAB (OH) and HomeAssistant (HA) have the flexibility of integrating a variety of non-compatible devices.

3.3.2. IoT Gateway Options: openHAB Vs. HomeAssistant

Of the gateways previously mentioned, it seems that the two best options for integration of non-compatible devices are openHAB and HomeAssistant [13, 14] . Both platforms offer great customization and flexibility for home automation purposes, but their differences arise in the developing of the user-interface and integration of third-party devices. Below there are some key points in the differences of openHAB and HomeAssistant.

- Installing and Maintenance
    - The installation and maintenance of both gateways are quite intuitive. However, HomeAssistant seems to have a more developer friendly environment for both the installation and maintenance [14].
- Pairing and Supporting Devices
    - Both of these gateways offer a wide variety of integration of devices. HomeAssistant and openHAB advertise integration of 1400 and 792 different

devices, respectively. In addition, the integration of the devices is accepted to be a little bit easier in HomeAssistant. [14].

- Creating Automation Scenes and Rules
    - The creation and customization of automation scenes and rules are more flexible in openHAB in comparison to HomeAssistant. The downfall to this is that the developer needs to be a little more advanced in high-level programming. [14].

3.4. Home Assistant & openHAB

Home Assistant and openHAB are open-source home automation software, [15, 16]. The main difference in these software's is the programming in which these software's were developed, where HomeAssistant is developed using Python 3 and openHAB is developed using Java, [14, 15, 16]. Furthermore, it emphasizes on local control and privacy [16]. The home assistant software should run perfectly on a Raspberry Pi.

Figure 8: HomeAssistant logo [16]

The project team also decided to use a home automated opensource smart system. HomeAssistant and openHAB were selected because they are two of the most used home automation gateways. HomeAssistant is particularly easy to use and step up, for this reason our team put a big emphasis on using HomeAssistant in the beginning months of our project. Furthermore, controlling the user interface is also easily understood by those with a bit of tech knowledge. In addition, HomeAssistant supports a load of different brands and smart devices. Therefore, HomeAssistant will be a good choice for installing most of the 3rd party device

Figure 9: openHAB Logo [15]

Following our experimentation with HomeAssistant during the first half of the semester, we began experimenting with openHAB. What we found was that openHAB is a little bit more difficult for the developer, which was known through reference [14] and confirmed through our experimentation.

In order to familiarize our team with the developer environment of HomeAssistant and openHAB we began to dig into the opensource forums available to HomeAssistant and openHAB developers. The two big opensource forums for HomeAssistant are linked in references [17] and [18] and for openHAB are linked in references [19] and [20]. These opensource forums allowed us to know what devices we will be able to integrate into HomeAssistant and openHAB as well as knowing what obstacles we will have to overcome during the integration of these devices.

Figure 10: openHAB Dashboard example [19]

Figure 11: HomeAssistant dashboard example [17]

3.5. The ABB free@home System

In this section, ABB will be discussed. The Vamiaeasy room contains many ABB devices which resulted in our team wanting to gain a better understanding the ABB and their free@home system. This company has been researched because of its importance in the project.

3.5.1. General information

ABB is an international company. More than 100.000 people are employed by ABB worldwide. The company is decentralized and has locations in more than 100 countries. The operational power lies in the different geographical divisions. ABB is a combination of two companies: ASEA (Sweden) and BBC (Switzerland). These two companies have been merged since 1988 [21].

Products

The divisions of ABB are divided into four different sectors:

1. Electrification
2. Process automation
3. Motion
4. Robotics & discrete automation

Strategy

Having in regard to the theory of Treacy and Wiersema, ABB is a good example of product leadership. Innovation is desired by all its customers. In general, customers of companies with the strategy pays the highest amount of money to receive the best products with the newest innovation. Innovation is not only created by ABB, but by its customers as well. It is a cooperation between many parties. For example, customer, ABB and other specialized companies [22].

ABB's Mission

ABB has the following values [23] :

1. Courage: take action if you need to do something.
2. Care: Take care of people, customers and the environment.
3. Curiosity: Try to improve yourself and the technologies.
4. Collaboration: Work together and build on other's strengths.

Purpose

The mission of ABB is summarized in the next three rules:

"We succeed by creating superior value."
"We push the boundaries of technology to drive performance to new levels."
"We energize the transformation of society and industry to achieve a more productive, sustainable future."

Sustainability is also an important unique selling point. ABB offers solutions for energy efficient business processes. The company wants to be a major player in the energy transition. This is ABB's way to help companies improve the productivity and sustainability in the coming years [24].

What is ABB free@home?

ABB free@home is a product from ABB. ABB free@home applies various devices to make living easier. Examples of devices are sensors, lights, thermometer, cameras and tablets. It is easy to install, and it is easy to adjust the panel to your own preferences. For example: all the lights in the kitchen could be switched on with one click on your laptop/smartphone/device. One goal of the project is to integrate those devices in one program.

After connecting the device to the right wifi network, the configuration of the system could easily be modified to ones desiredness.

4. Bill of Materials

The team had an assumption that the majority of the devices within the Vamiaeasy room were smart devices. Here, smart devices are considered as devices which have some form of wireless connectivity (i.e., WiFi, Bluetooth, Z-wave, or ZigBee).

After spending more time in the automated living environment, the team ended up finding that many of the devices which were thought to be smart are actually not at all smart (i.e. they have no wireless connectivity). Upon finding that some of the devices have no wireless connectivity, which in turn makes them non-smart devices, the team decided to make up a bill of materials.

A bill of materials, in essence, is a list which allows one to quickly identify the materials and subproducts which are used for a specific product [25]. However, in our case, we created a bill of materials to identify all the devices within the smart living environment.

Because the majority of the devices within the smart living home are Vamia are ABB devices, the team decided to break the bill of materials into 3 separate categories: ABB_free@home devices, smart devices, and no smart devices. These categories can be found in tables 4, 5, and 6 which can be seen on pages 40,41, and 42, respectively.

Table 4: ABB free@home devices

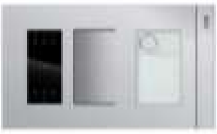
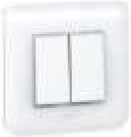
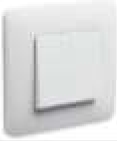
ABB_free@home devices		
Device name	**Pictures**	**Description**
Mini outdoor station		Smart doorbel with integrated camera.
System Access Point 2.0		Enables configuration of system and access for control by mobile devices, both locally and remotely.
A switch		Touch sensor for bus system with two buttons (2 circuits).
A switch		Touch sensor for bus system with one button.
Indoor movement detector		Movement sensor that detects movement.
Thermometer		Heating/cooling operation. Control element with room temperature.
Outdoor motion detector		Outside movement is motioned by this Motion detector. It can observe 3 angles.
User-interface		Touch screen for controlling all the systems in the house.
A curtain motor		The motor controls the opening and closing of the curtains.

Table 5: Smart Devices

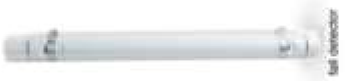
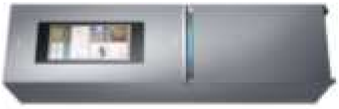
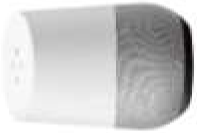
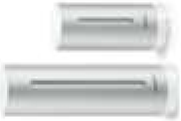
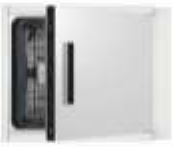
Smart devices		
Device name	**Pictures**	**Description**
eLsa activity sensing		The solution is a wireless system that detects and alerts health care personnel in case of a fall.
Samsung Family Hub 2.0		It is a connected refrigerator with a touch screen connected in Wifi. It is possible to have applications, to see in real time the inside of the fridge thanks to cameras.
TAIKOFON® FeelSound Player		It is a sound-sensing player that lets you both hear and feel sound in an amazing new way. You can listen to online radio or downloaded music, choose a channel on Spotify…
Philips Hue		Control the light in the room and is connected to the ABB_free@home system. Use a Wi-Fi connection.
Medicine box		This is machine that dispenses medicines at the right time for a elderly person. It has to be restocked every 2 weeks by a nurse.
Television		It is a smart TV and uses the application smart things
Google home assistant		It is a smart speaker, moreover it is possible to control other devices by voice (television, lights ...) through the Google Home application.
CareSens Dual		CareSens Dual offers an ideal platform to perform blood glucose and blood ketone tests on a single meter.
Netatmo Weather station		Allows to take the temperature of a room, the rate of humidity… The motor controls the opening and closing of the curtains.

Table 6: No Smart Devices

No smart devices		
Device name	**Pictures**	**Description**
iRobot Roomba 606		It is a robot vacuum cleaner no connected for this model
Alarm clock philips		It is a bright alarm clock that promotes natural awakenings.
Coffee machine		It is a coffee maker with some functions: possibility to set a timer or to choose the time at which the coffee should be made.
Key Finder: an object locator		Helps you find lost items with an alarm sound quickly and easily. The locator also has an LED light to search for objects in the dark. Range up to 30-40 meters in open space.
Maxi personal amplifier		This device allows to amplify the sound for people with hearing difficulties by wearing a headset.
Timer socket		This is a plug with a timer ranging from 15 to 60 minutes. When you click the button, the timer adds 15 minutes. When the time is up, the socket turns off and prevents the power from flowing.
Wireless optical smoke alarm		It is an optical smoke detector that detects the presence of smoke and emits an audible signal to quickly respond.
Daily calendar - morning, day, evening, night		The daily calendar shows the time of day in words. The time of the morning, day, evening and night texts can be customized according to the user's needs.
Dishwasher BEKO		It is a classic dishwasher.
Solon memo minder		The Solon Memo Minder is an innovative memory aid that allows the recording of a 15 second message, which is played whenever someone walks past the unit's motion sensor.

5. Visualization of the Vamiaeasy Showroom

5.1. Home By Me – 3D model

Figure 12: Logo Home By Me

HomeByMe is an online interior design software. It allows you to visualize your project in 3-dimenstion to simulate a visit and to obtain 2-dimensional plans. The functionalities of this software are free thanks to the numerous companies that provide their furniture in a complete and diversified catalogue. This software is developed by Dassault System, a well-known publisher of software specializing in 3D design. [26].

During the modelling phase, the HomeByMe software proved to be very intuitive, the learning is very fast and many forums are available if needed. Thus, if any modifications were to be made in the future, it would be very easy to continue the work done by the EPS group. The access codes to the HomeByMe account of the EPS group for the Vamiaeasy project are available in the documentation within Microsoft Teams.

5.1.1. Realization of the 3D model

Normally, this software is used to create, draw, and propose projects before realizing them. In the case of the EPS group, it is more a question of copying the Vamia showroom as realistically as possible, in order to present it from anywhere. Indeed, having explanations of the devices present in the room is important, but a visualization of the latter allows to have a more immersive and understandable presentation. Moreover, until now, only pictures of the room were available to have an overview of it.

The first step of this 3D model was to model the room based on the photos that the group had taken. It was complicated to go on site to take measurements because the access to this room is cumbersome due to being used by many people. During this first phase, it was necessary to find furniture which corresponds well to those present in the room. Thanks to custom-made furniture, it was easy to size them as in reality. The group was able to put together a first sketch which was then presented to Vamia before they provided a scale plan. The second step was to update the whole model to respect the exact dimensions of the plan.

Finally, after spending many hours modelling this room as accurately as possible, the EPS group was able to provide a complete final model in which we can provide a walkthrough as in reality (Figure 13).

5.1.2. Limitations of this software

Although the choice of furniture is very replicated, it was not possible to model all the connected devices present in the room, such as the medicine box and the key finder. For this reason, we looked for a way to be able to move around the demonstration room while adding precise information on each device. For example, a solution would be to zoom in on a device and then display information about it: smart or non-smart devices or devices already integrated to our IoT platform.

This type of presentation could be a support for the presentation of our project but also for the presentation of the room which may be used by Vamia to present their room. So, the EPS group decided to use the Prezi platform.

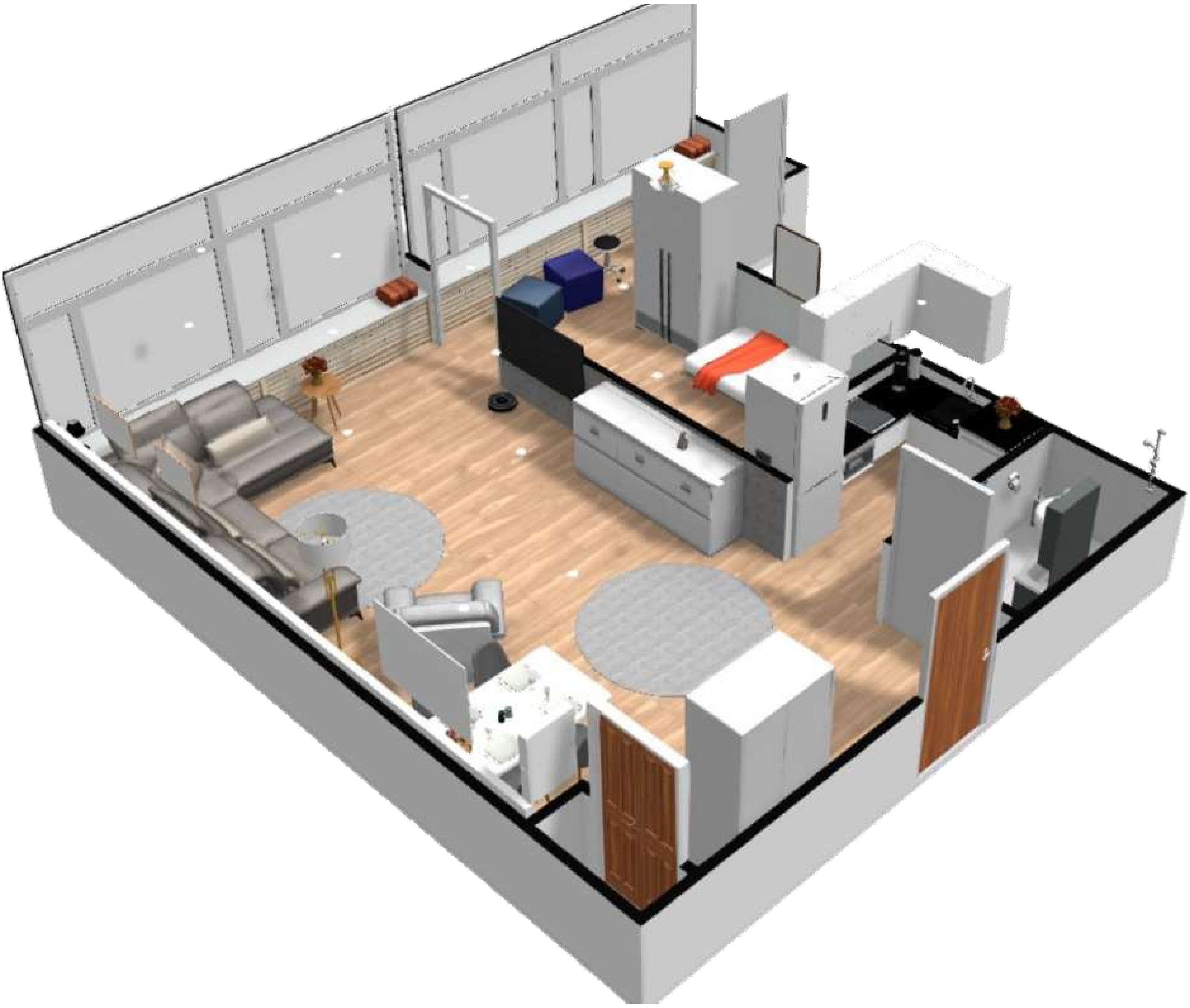


Figure 13: 3D model of the Vamiaeasy showroom created via HomeByMe

5.2. Showcasing the Vamiaeasy Room via Prezi

5.2.1. Description of the Prezi tool

Figure 14: Logo Prezi

Prezi is a presentation tool in the same style as Power Point. However, its uses are different because it is based on a presentation map that you can zoom in or out infinitely. The zoom animations are created automatically and make the presentation much more interactive. But be careful in some cases it can give an opposite impression with a too loaded file. [27].

The use of this tool can be a bit complicated at first. A certain amount of practice is necessary to be able to achieve what you want. But once launched, the different possibilities of use are really interesting. It is possible to duplicate your project in order to vary the information inside according to your audience. The access information to the account with the final Prezi presentation are available in the documentation on Microsoft Teams.

5.2.2. Realization of the final Prezi – Vamiaeasy Showroom

The first step was to combine HomeByMe with Prezi to create an immersive presentation of the showroom (Figure 15). Knowing that Prezi uses the principle of a map to move around, the group chose the 3D model as the home page. On this first page, it is easy to access the different rooms of the showroom and to have an overview of the project. In addition to the bubbles that access the rooms, the red bubbles allow you to have additional information:

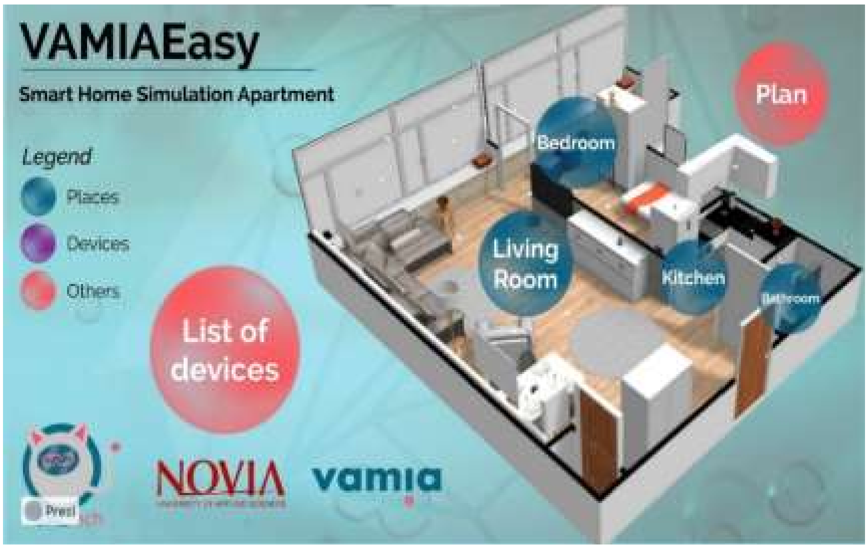
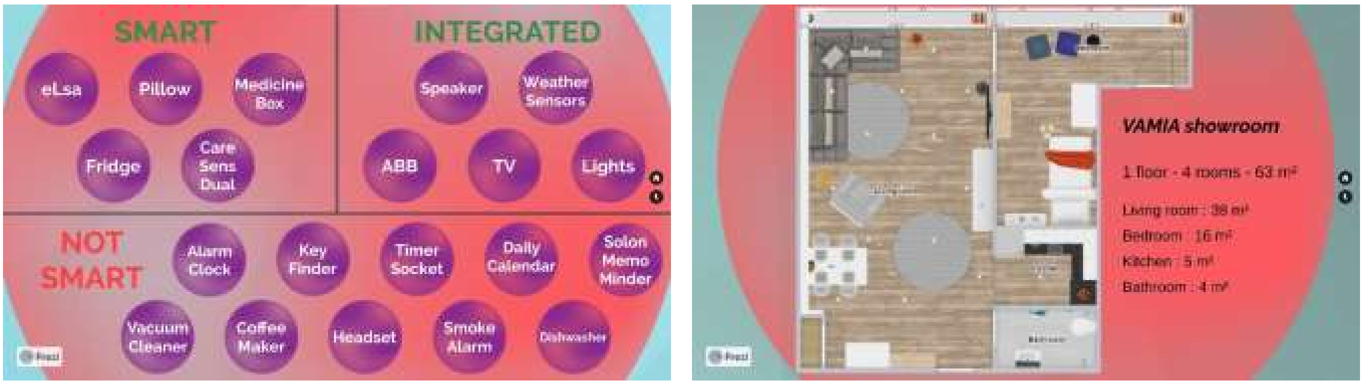


Figure 15: Prezi home page of the Vamiaeasy room

- The "plan" bubble allows you to have a 2D plan of the room with the surface areas in m2
- The "List of devices" bubble allows you to quickly access the devices without searching for them in the rooms



VAMIA showroom

1 floor - 4 rooms - 63 m²

Living room : 38 m²
Bedroom : 16 m²
Kitchen : 5 m²
Bathroom : 4 m²

Figure 16: "List of devices" and "Plan" bubbles

The second step was to integrate all the devices present in the room into the Prezi. Thanks to the bill of materials previously developed by the EPS group, all that was left to do was to dispatch them to the different rooms. (Figure 16)

For each device, several pieces of information had to be displayed:

- Is it a smart device, non-smart device, or already integrated in the IoT platform
- A picture of the respective device
- A description of the device and its uses

To make this presentation more realistic, the zoom on each room corresponds to a real image. Figure 17 shown below is the example when zooming on the "Living Room" bubble.

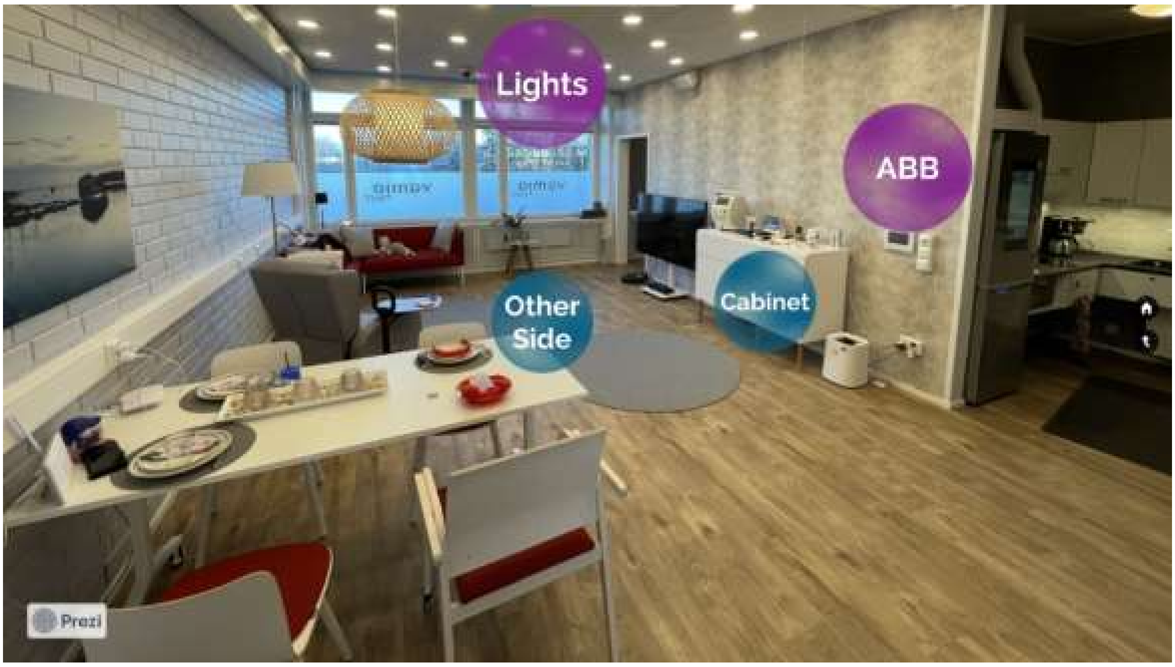
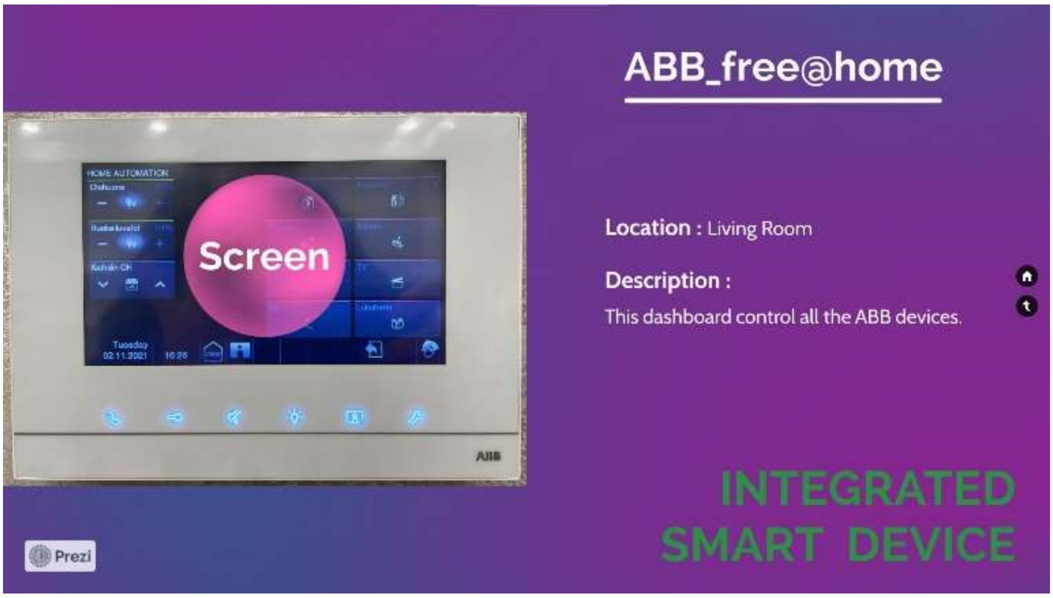


Figure 17: "Living Room" bubble

Following the legend on the home page, if you zoom in on a new blue bubble, you will zoom in on a specific area of the room, while a purple bubble will bring you to the description of the device (Figure 18 &Figure 19).



ABB_free@home

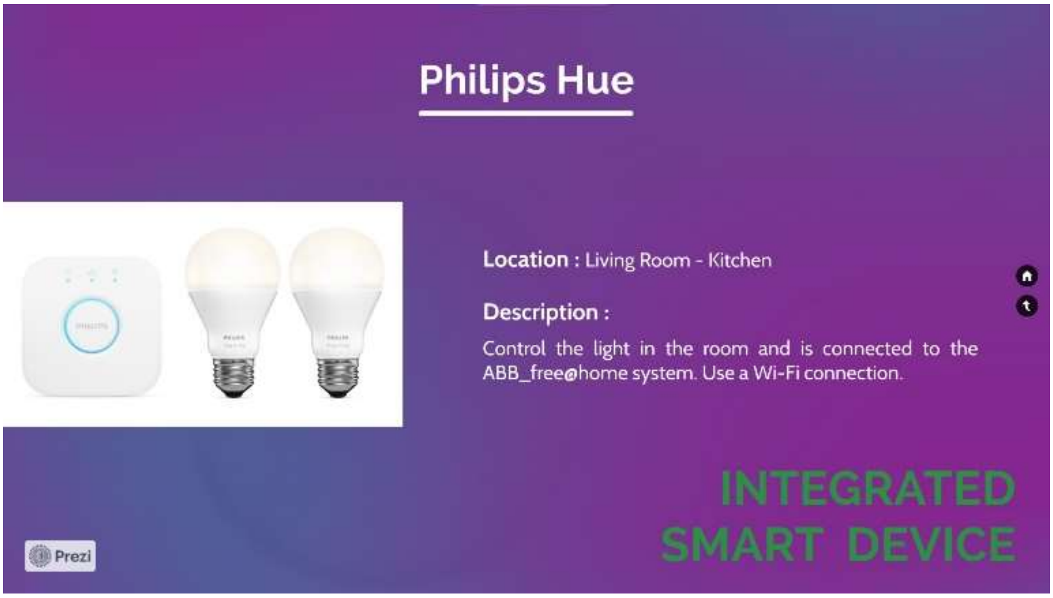
Location : Living Room

Description :

This dashboard control all the ABB devices.

INTEGRATED SMART DEVICE

Figure 18: "ABB" bubble



Philips Hue

Location : Living Room - Kitchen

Description :

Control the light in the room and is connected to the ABB_free@home system. Use a Wi-Fi connection.

INTEGRATED SMART DEVICE

Figure 19: "Lights" bubble

Finally, we come to the end of this Prezi presentation, its big advantage is to be able to vary the information present inside. In the case of the EPS group, it is oriented towards the multitude of devices present in the room and if they are already integrated in our IoT platform. But by modifying its content, it is possible to direct this Prezi more towards teaching for Vamia teachers.

6. Vamiaeasy’s ABB free@home System

6.1. Getting Access to the free@home Configuration Environment

The ABB system can be configurated through a PC’s web browser. In order to do so, the user’s PC must be connected to the same WiFi that the free@home system is connected to. Upon being connected to the same WiFi that the free@home system is connected to, the user will need the free@home system access point’s IP address and the password to the “installer or asennus” account. The coming subcategories will describe how this is done in detail.

In order to find the IP address which is associated with the free@home system, then, as previously mentioned, it is required that the user’s PC is connected to the WiFi. Now, the user can go to their files, then click on “network” to find the router they are connected to. The router should be found under the “Network Infrastructure” tab. Double clicking on their router will give them access to the IP addresses associated with all the devices connect to this router. Below has step-by-step instructions to do this.

1. Connect to the same WiFi which the free@home system is connected to
2. Go into the PC’s files folder and find the “network” tab (Figure 20)
3. Double click on the router which the PC and free@home system is connected to (Figure 20)
    - Following this, you will be prompted to input the admin password for the router. Generally, this will be the same password which is used to connect to WiFi; However, if this is not the case, then the user must obtain the appropriate password to gain access into the router’s system.
    - The user may find respective IP address for all devices connected to this router after gaining access to this system

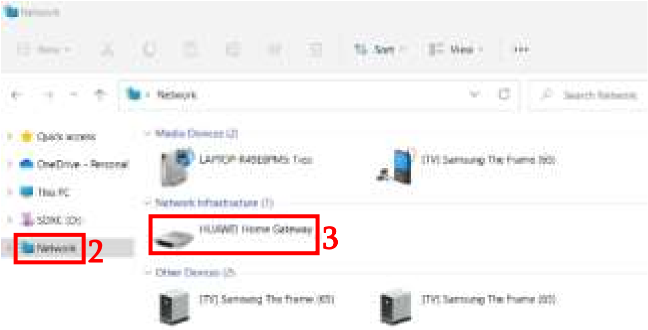
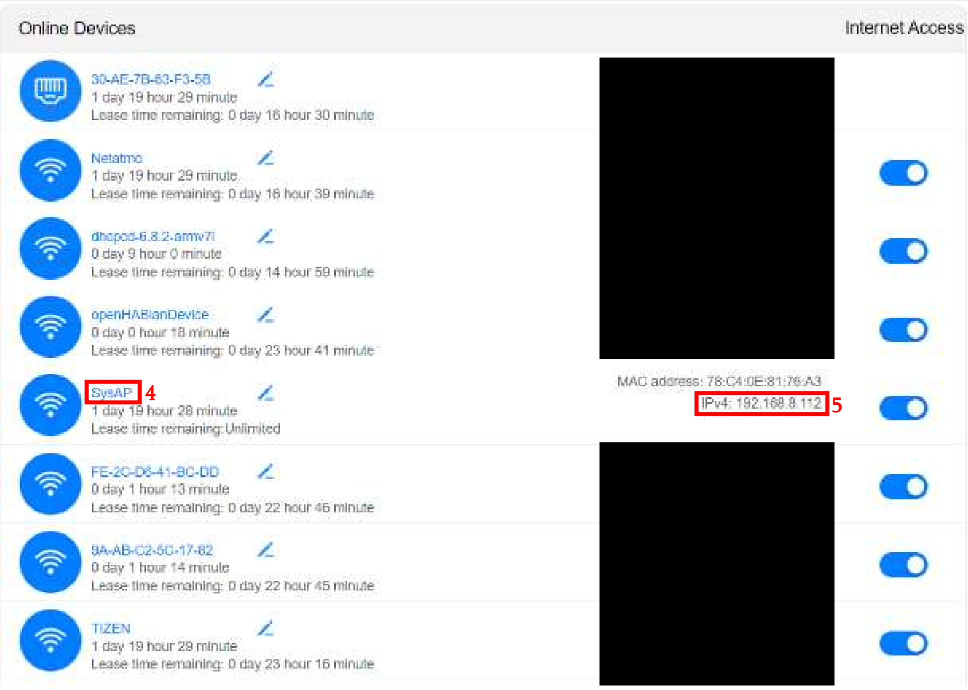


Figure 20: Finding the router which a PC is connected to

4. After finding the list which has the devices that are connected to the router, find the device with the name “SysAP” within this list (Figure 21)
    - Note: SysAP is an acronym for System Access Point
5. Note the SysAP IPv4 address (Figure 21)
6. Type the SysAP IPv4 address into the web browser of the PC being used (Figure 22)
    - The user should now be in the login screen for the system access point configuration environment

This completes getting access to the system access point configuration environment.



Online Devices | Internet Access

30-AE-7B-63-F3-5B
1 day 19 hour 29 minute
Lease time remaining: 0 day 16 hour 30 minute

Netatmo
1 day 19 hour 29 minute
Lease time remaining: 0 day 16 hour 39 minute

dhcpcd-6.8.2-armv7l
0 day 9 hour 0 minute
Lease time remaining: 0 day 14 hour 59 minute

openHABianDevice
0 day 0 hour 18 minute
Lease time remaining: 0 day 23 hour 41 minute

[redacted]

SysAP 4
1 day 19 hour 28 minute
Lease time remaining: Unlimited
MAC address: 78:C4:0E:81:76:A3
IPv4: 192.168.8.112 5

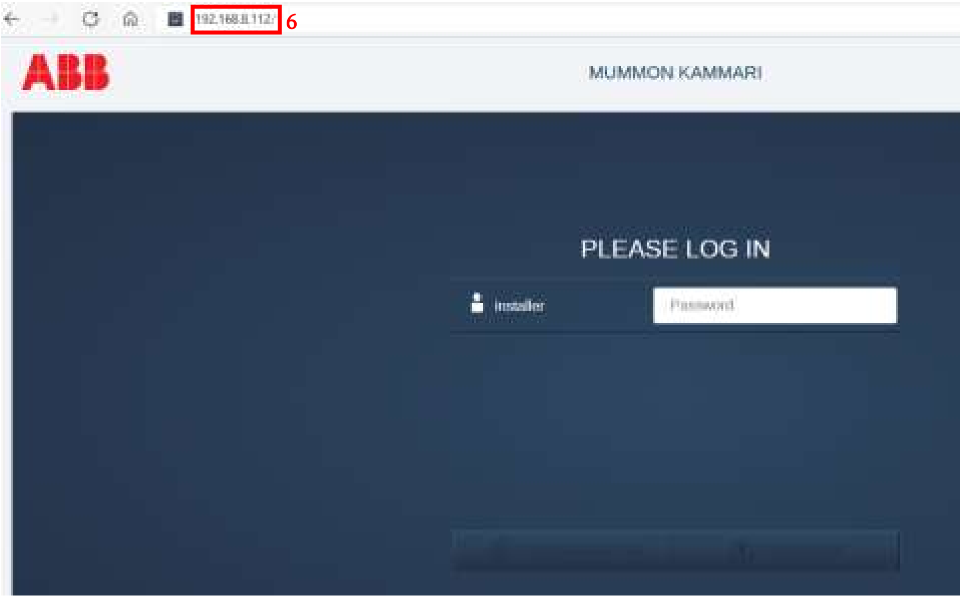
FE-2C-D6-41-8C-DD
0 day 1 hour 13 minute
Lease time remaining: 0 day 22 hour 46 minute

9A-AB-C2-5C-17-82
0 day 1 hour 14 minute
Lease time remaining: 0 day 22 hour 45 minute

TIZEN
1 day 19 hour 29 minute
Lease time remaining: 0 day 23 hour 16 minute

[redacted]

Figure 21: Router device list example



192.168.8.112/ 6

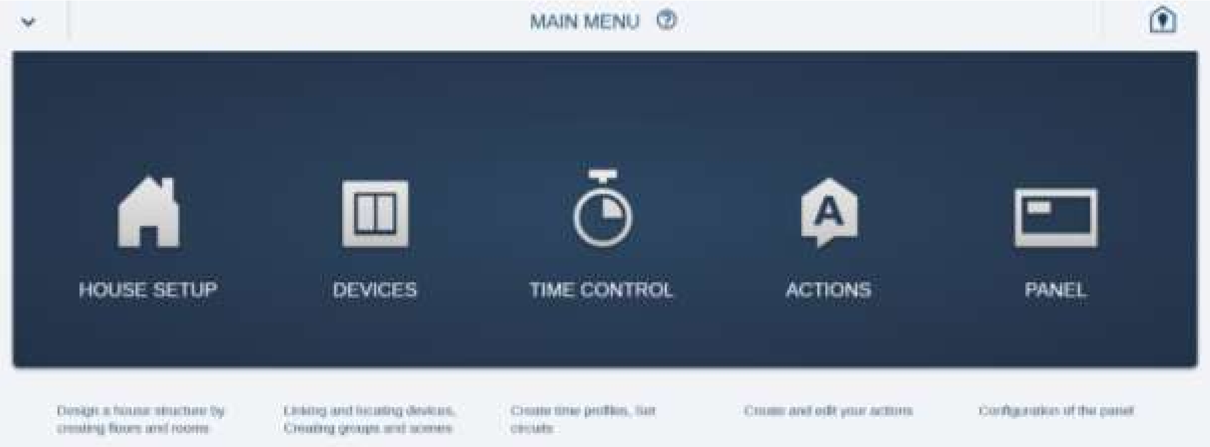
PLEASE LOG IN

installer | Password

Figure 22: Login page for System Access Point

6.2. The free@home Configuration Environment

The main menu depicted in Figure 23 is shown after logging in. The main menu exists of 5 tappable applications:



Design a house structure by creating floors and rooms

Linking and locating devices, Creating groups and scenes

Create time profiles, Set circuits

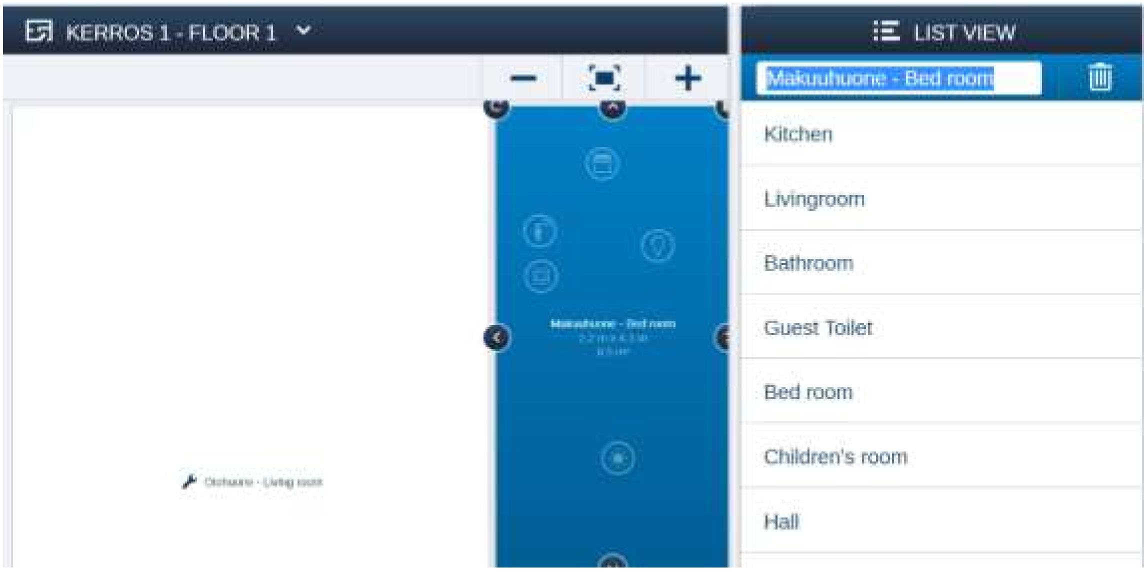
Create and edit your actions

Configuration of the panel

Figure 23: ABB free@home main menu

6.2.1 House setup

The house setup can be used to change the floor plan of the respective home. Firstly, the tile should be tapped. After this, the tile could be dragged to a different location. It is also possible to adjust the size of the tiles. The arrows on the side of the tiles must be used to increase or decrease the size of the tile (Figure 24).



Makuuhuone - Bed room

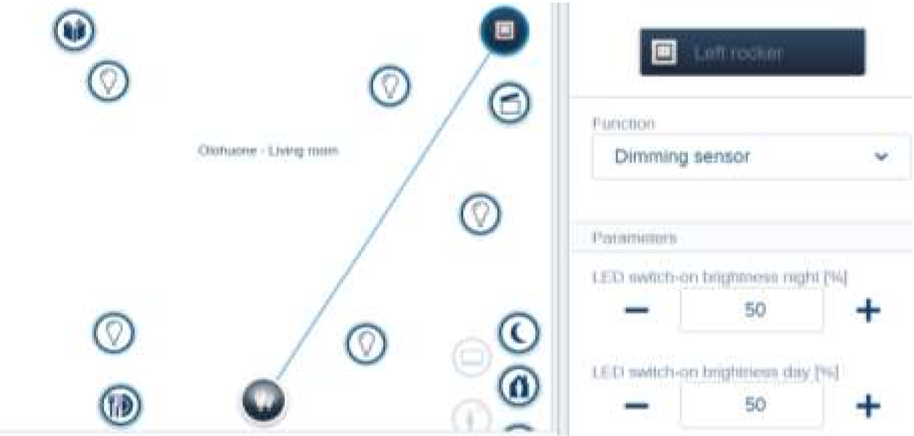
Figure 24: ABB free@home house configuration

The pictograms in the tile could not be adjusted in the housing setup. The name (marked blue) could be changed in the list view. All the names displayed in the white list below the name are just suggestions for names. These could be ignored.

6.2.2. Devices

After tapping the application ‘devices’, the following screen shown in Figure 25 appears. Here you can:

- Add pictograms (devices): New ABB devices could be connected to the network in the ‘devices’ application. However, this is not within the scope of our project. All the devices are already connected to the system so the possibility to add new devices is not used.
- Connect devices: After adding the devices to the right tile, the devices need to get connected. For example, a light and a switch. All the devices are already connected at the moment. However, there might be a situation in the future where it is desirable to change these connections. Firstly, the rocker should be tapped. A choice has to be made between the right rocker and the left rocker. For this example, the choice does not matter. After this, it has to be connected with a lamp by tapping an existing lamp on the map. It is also possible to amplify or attenuate the light by adjusting the brightness (see parameters in Figure 25).



Function: Dimming sensor

LED switch-on brightness night [%]: − 50 +

LED switch-on brightness day [%]: − 50 +

Figure 25: Connecting ABB free@home devices

- Making scenes: In a scene, different lights, blinds and the television could be controlled by just one tap. Examples for scenes are:
    - Good morning: All the lamps turn on and the blinds open.
    - Good night: All the lamps deactivate and the blinds close.
    - Diner: Only the lamps above the dinner table activate. All the order lamps deactivate.
    - Reading: Only the lamps above the sofa activate. All the order lamps deactivate.
- Different scenes had been made before this project started. During this project, one scene has been added. A scene could be added by tapping on the “Scenes/Groups” (see the Figure 26).

- After tapping the “Scenes/Groups” button, the custom scene must be chosen.

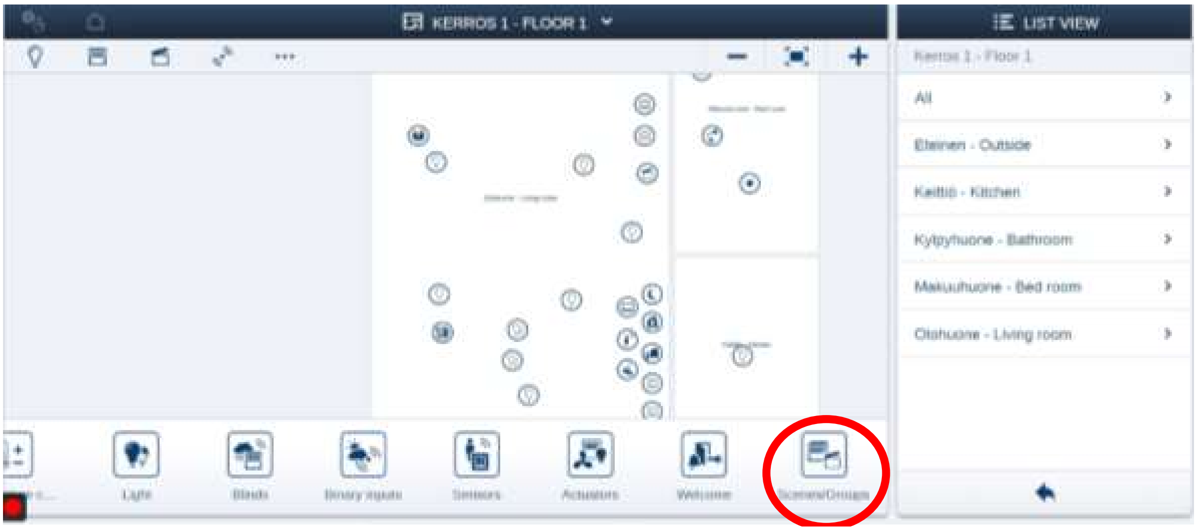


Figure 26: Locates scenes/groups in the ABB free@home systems

6.3. VAMIAeasy’s free@home Configurations

All the devices have been installed by the ABB technician. During the installation, a few scenes has been programmed as well. The installation was completed before the project group started to work on this project. The dashboard within the Vamiaeasy room is shown in Figure 27. Three different colored tiles are visible. They could be recognized from their colour at the top of the tile.

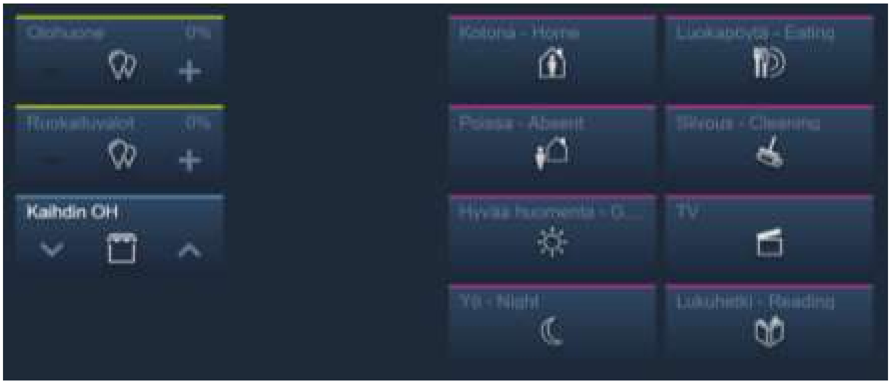


Figure 27: ABB dashboard panel

- Yellow: A group of lights. Different lights have been coupled into a combination of lights. For example all the lights in the “olohuone” (living room) or all the dining lights ruokailuvalot.
- Blue: The blinds. Every window (two in total) has its blind. The blinds can be controlled by using tapping on the “Kaihdin OH” (blind) tile on the dashboard.
- Purple: Scenes. Different devices are coupled into one scene. If this tile is tapped, A complete set of tasks is being activated on different devices.
  - Kotona (Home): The lights turn on and the blinds open. This scene could be used when an elderly person comes home.
  - Poissa (Absent): The lights turn off and the blinds close. This scene could be used when an elderly person leaves the house.
  - Hyvää huomenta (Good morning): The blinds open up and some lights are being activated. The activated lights are the lights between the bedroom and the kitchen/bathroom. An elderly person uses this scene for rising.
  - Yö (Night): The blinds close and all the lights turn off. The user can use this scene when this person would like to sleep.
  - Luokapöytä (Eating): Alle the lights turn off except the lights above the sofa. Blinds stay/go open.
  - Slivous (Cleaning): The blinds open up and all the lights turn on on 100% strength. The cleaner of the apartment (probably not the elderly person itself) could clean.
  - Lukuhetki (Reading): All the lights turn off except the lights above the sofa. So the habitant has a pleasant atmosphere for reading. Blinds stay/go open.

7. Automating Motorized Cabinet

The Vamiaeasy room is equipped with a motorized cabinet which is triggered by using a controller with an up and down button (Figure 28). The configuration of this cabinet from the manufacturer was originally programmed so that the user must the press and hold the up and down buttons to bring the cabinet up and down, respectively. To bring the cabinet completely down takes 28 seconds. Although this may seem like a short time, it does not leave a good impression when demonstrating it to a group. The Vamiaeasy group found a solution to automate this cabinet for wireless connectivity while still allowing the user to control the cabinet as originally designed.

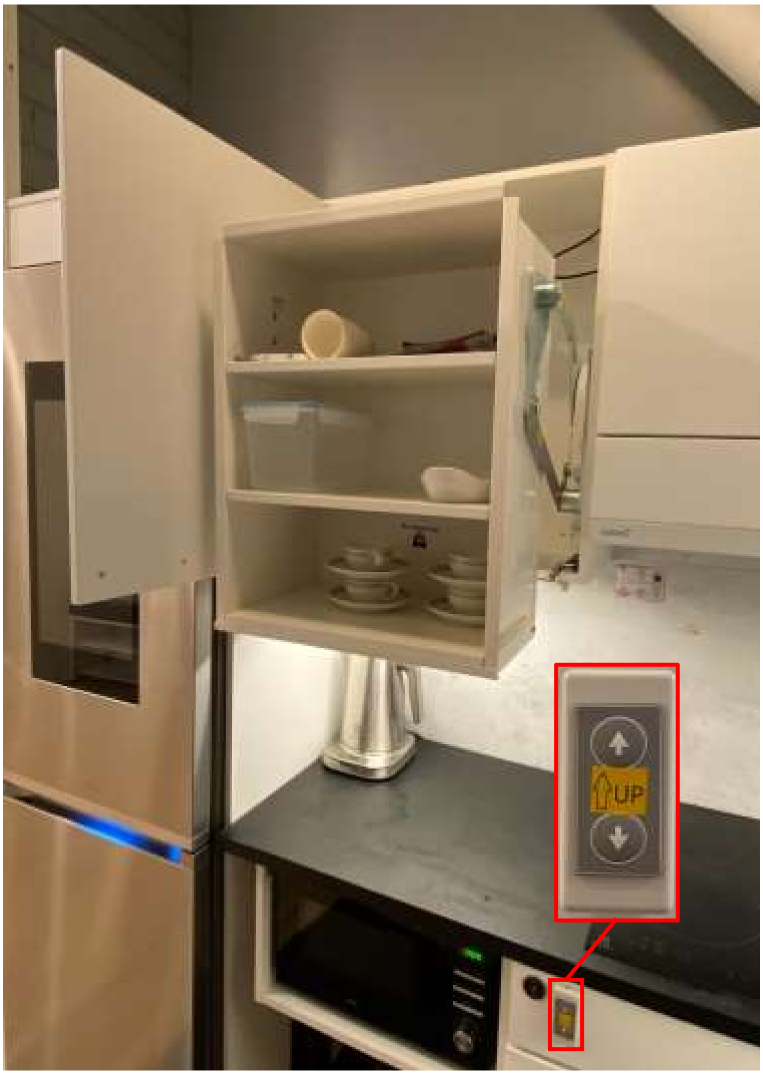


Figure 28: Motorized cabinet with up & down controller

The group utilized an esp32 programmed in the Arduino IDE to control an instant-on relay circuit. This allowed the Vamiaeasy team to create 3 separate scenarios in the automation of this cabinet: (1) using the cabinet controller as originally designed; (2) bringing the cabinet completely down/up in a single-button-click action; (3) bringing the cabinet completely down/up wirelessly via voice command or through the HomeAssistant app.

7.1. ESP32-Triggered Instant-on Relay Circuit

The instant-on relay circuit was the most advantageous circuit construct for the purposes of automating the motorized cabinet. This is because the cabinet controller does not have a true ground (i.e., the common wire for the cabinet connector is not sitting at 0V). In addition, the team provided Vamia a failure diagnostics sheet (Appendix 9). This ensures that Vamia administrators can find the root-cause of the problem if the cabinet were to fail.

7.1.1. First automating cabinet circuit configurations (no success)

The first iteration of automating this cabinet using an ESP32 was done by connecting the controller connector the to ESP32 as shown in Figure 29 on page 56. What this connection does is waits for signals at pin 26 and pin 14 to reach the GND pin. For example, when the up button is pressed, then the signal from pin 26 is able to make its connection to GND and visa-versa for the down button. The difference in our first circuit configuration compared to the final one shown in Figure 29 is that pin 33 and pin 19 were connected to the up and down pins at the cabinet connector, respectively, and the common pin at the cabinet connector was connected to GND.

This configuration was sending a signal from pin33/pin19 to the up/down (red/black) pin when the up/down button was pressed on the cabinet controller. The thought was that this signal system was configured to take in the signals to these respective pins which in turn will then bring the cabinet up and down. However, after running this configuration, it was learned that this is the wrong approach to take and has many faults associated with it.

The second approach was to utilize a transistor switch to complete the circuit between the up/down pins to the common pin on the cabinet connector. The difference in the circuit diagram shown in Figure 29 would be to remove the relays and diodes and connect the base pins of the transistors to pins 33 and 19 with the same valued resistors. Following this connection, we would then connect the up/down pins to the collector pin and the common pin to the emitter of the respective transistors.

This second approach would in theory work; however, there are some faults still associated with this approach. The fault lies in the fact that there would be a voltage drop across the emitter/collector pins of 0.7V and a small increase in current to the emitter pin. This is not an ideal approach because one should ensure the cabinet system is configured to have the same voltage and current as originally designed. This is why the instant-on relay circuit approach is the most ideal and least fault tolerant.

7.1.2. Final automating cabinet circuit configuration (success)

The instant-on relay circuit is the most ideal due to the fact that the circuitry of the motorized cabinet will be untouched. Whereas in the cases mentioned about, the circuitry will have some associated losses and additions of voltage or current with them. The relay shown in Figure 29 is triggered by pins 33 and 19 for the up and down motion of cabinet, respectively. When the up or down button is pressed, then a 0.15mA current will be sent to the base of the transistors which then allows the 5V signal to make it to the GND connection on the esp32. Upon the 5V signal making a complete connection to ground, the relay will be switched and closes the circuit from the up/down pins of the cabinet connector to their common pins. Which, as previously mentioned, is the most advantageous because the cabinet circuitry is left untouched and in essence is configured in its original design. The datasheets for the respective components found in Figure 29 can be found in the reference [28, 29, 30]. A hand-drawn version of this circuit schematic shown in Figure 29 can be found in Appendix 8 on page 94.

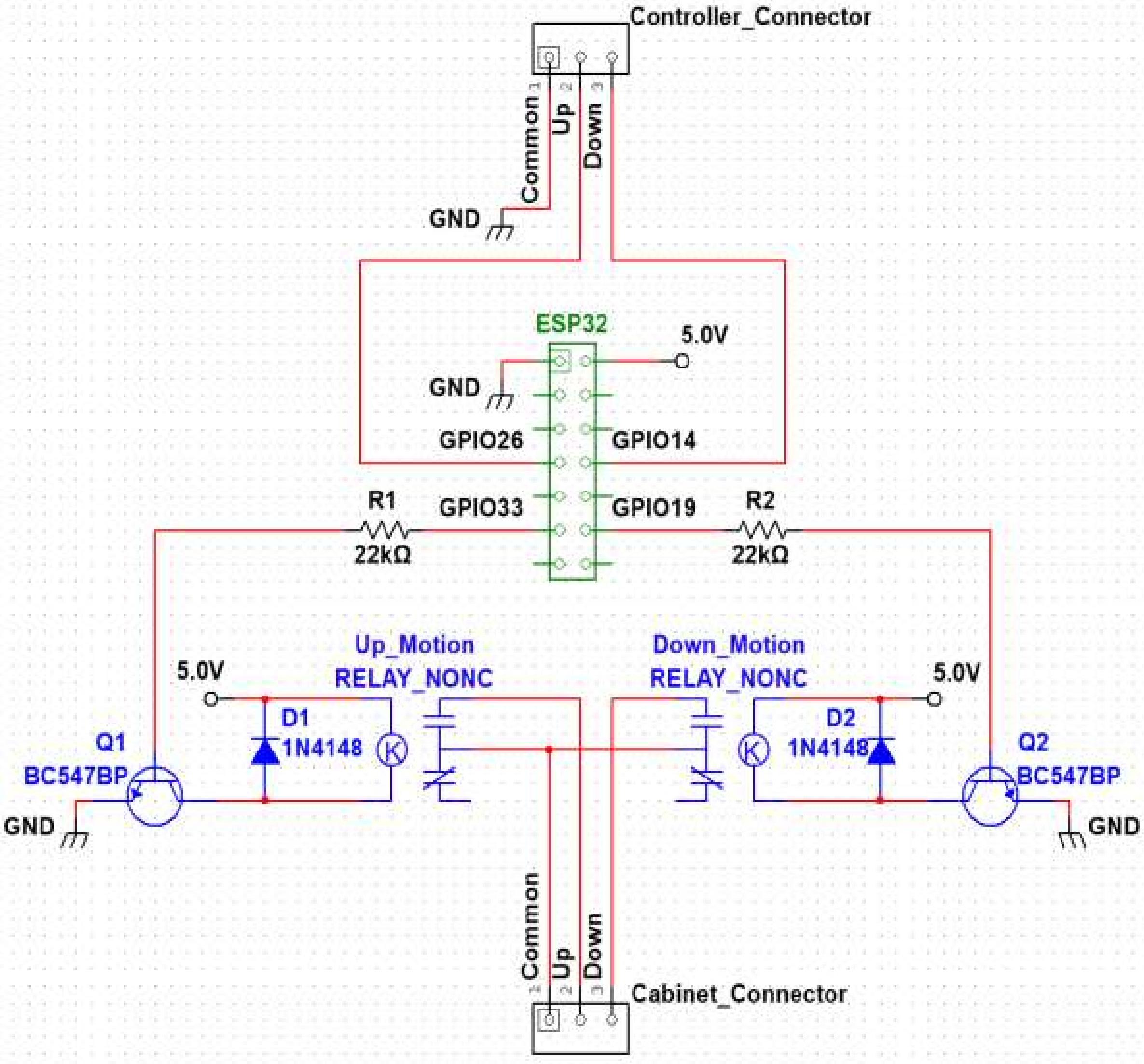


Figure 29: ESP32-Triggered instant-on relay circuit schematic

The circuit diagram shown in Figure 29 was then implemented onto an electrical breadboard, put into a blackbox, and installed in the Vamiaeasy room. The blackbox containing the ESP32-triggered instant-on relay circuit is shown in Figure 30.

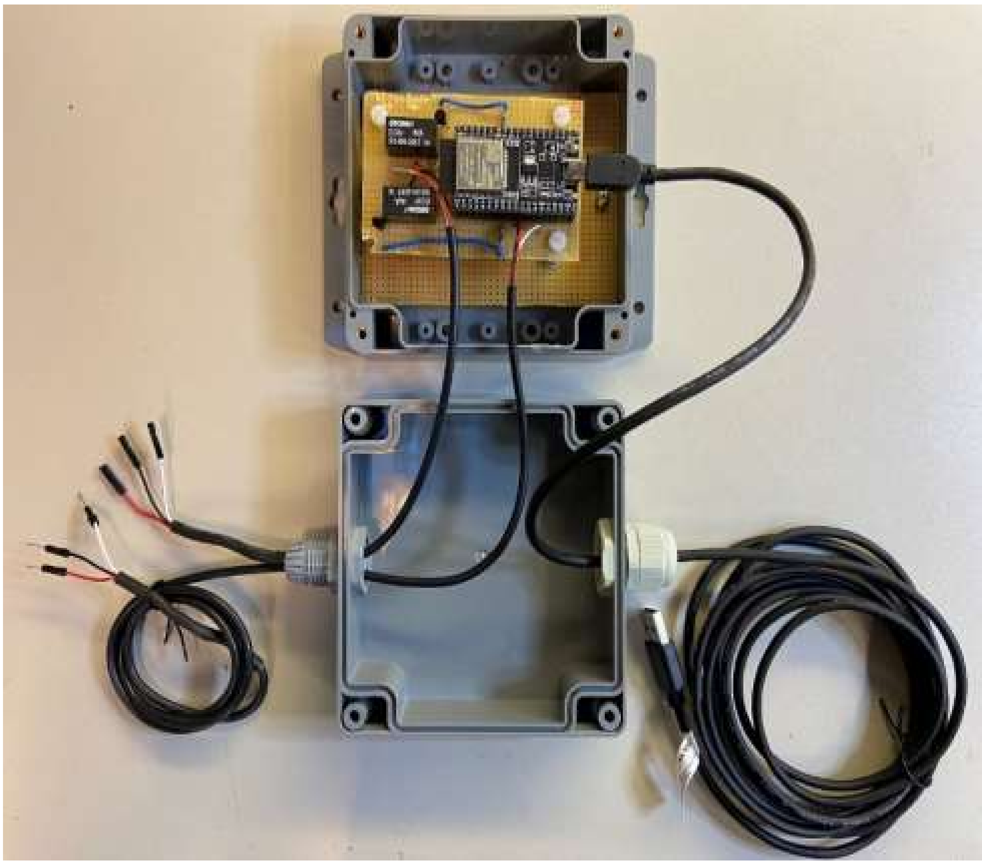


Figure 30: Blackbox circuit for automation of cabinet

7.2. Different Cabinet Automation Scenes

As previously mentioned in section 7.1, there were three separate scenarios which the Vamiaeasy group created in the automation of this cabinet. This section will describe them in further detail.

7.2.1. Preliminary Information

The scenes which were automated using the ESP32 were all coded in the Arduino IDE. A library known as "OneButton" was used to ease the coding configuration to recognize the different press-button actions (single press, long press, and releases). This library and its respective components can be found in reference [31].

In addition to the press-button action requests, there was an http request configured so that the cabinet can be brought up and down through voice command or the HomeAssistant app. An open-source code which had an http web server configured was adjusted and tailored for the cabinet purposes. This open-source code and its description can be found in reference [32].

7.2.2. Utilizing Cabinet Controller as Originally Designed

Being able to use the cabinet as originally designed is important for intuitive reasons. This implementation was done using the attachDuringLongPress and attachLongPressStop functions found in the OneButton library [31]. The way this was implemented is done by detecting a long press which is defined to be a button being held down for more than 1 second [31].

Recall that when pin33/pin19 sends the 0.15mA signal to the base of the transistor, then the transistor allows the 5V connection to reach ground which switches on the respective relay. The attachDuringLongPress function is entered when up/down down buttons are held down for more than 1 second which turns on pin33/pin19 sending a 0.15mA to the base of the respective transistors. Upon the release of the up/down buttons, the attachLongPressStop function is entered which turns off pin33/pin19 stopping the 0.15mA signal to the base of the respective transistor. This configuration allows for the cabinet to be used as originally designed.

7.2.3. Bringing Cabinet Completely Up/Down in a Single-Button-Click Action

When the demonstration of the cabinet was first shown to the Vamiaeasy easy group it was noticed that the press and hold action was not the most impressive. So, the Vamiaeasy group thought that a single-action press of the button would add convenience to the end user and impress our customer (Anu). The implementation of the single-press action utilized the OneButton library to recognize the single-button-press.

The detection of the single-press action was done by utilizing the attachClick function found in the OneButton library [31]. The ESP32 entered the attachClick function upon detecting an up/down single-button-press which then turns on pin33/pin19 for 28 seconds. Toggling pin33/pin19 for 28 seconds was done by simply triggering the pins to turn on, adding a line of code to pause for 28 seconds, then toggling the pins off.

7.2.4. Bringing Cabinet Completely Up/Down Wirelessly

Controlling the cabinet wirelessly was implemented using an http server. The code found in reference [32] configured an http server which was then specifically tailored to control the cabinet for Vamiaeasy's purposes. Instead of detecting a button-press, the code is waiting for the http server to be toggled by a request which looks like the following: <IP address of esp32>/*action.* Where the action is either *up* or *down*. When this action is called, the cabinet will go completely up or down. Following the creation of the http server, a cabinet up and cabinet down block was created in the HomeAssistant environment which called the respective http request. The cabinet was then able to be controlled through the HomeAssistant app and by voice because the HomeAssistant integration into google nest.

7.3. Mitigating Faults and Errors

An ESP32 is only going to do exactly what it was programmed to do. So, programmed errors are bound to occur. In addition, there are other faults which may occur, such as the WiFi going out which may potentially result in the ESP32 not running the button-actions at all. The Vamiaeasy group added mitigation of faults and errors through additional lines of code.

7.3.1. Buttons Pressed Synchronously

The way the ESP32 was initially configured did allow for the simultaneous press of both the up and down button. What occurred when both buttons were pressed at the same time was the cabinet would go completely down and then completely up (or vice versa). However, this is not an ideal fault mitigation; rather, the ideal approach would be that nothing happens when both buttons are pressed at the same time. A simple boolean value was put into the callable button-press detection function to mitigate this action of the up and down buttons being pressed at the same time. The boolean value in pseudocode is as follows: if the up button is pressed and the down button is not pressed, detect if the up button was pressed once or if it is being held down; if the down button is pressed and the up button is not pressed, detect if the down button was pressed once or if it is being held down. This ensures if both buttons are pressed at the same time, the code will not jump into the up and down button-press detection functions, which results in no action being done.

7.3.2. If Down Button is Held Down and then Up Button is Pressed Once

Another fault which may occur is that if the cabinet down button is held down, for example, for 10 seconds, and then the up button is pressed once. What would have originally occurred is the cabinet would go completely up and the signal would continuously be sent for a non-needed 18 seconds (because the cabinet down button was held down for 10 second means that the up action only needs to occur for 10 seconds). To mitigate this problem a value was created which detects how long the down button was held down for. Then, in the attachClick function, it checks to see if the down button was held down for more than one second. If the down button was held down for more than one second, then the single-click up action would only occur for the duration of how long the down button was previously held down for.

7.3.3. If WiFi is Disconnected from the ESP32

This problem is straight-forward to fix. A while loop was put into the code to detect if the ESP32 was disconnected from WiFi. In the case that the WiFi was disconnected, the ESP32 will stay in this while loop until the WiFi is reconnected. This resolved the WiFi connected; however, in our first implementation, the button-press action did not work while reconnecting to WiFi. This problem was mitigated by added the code to detect the button-press actions into the while loop which detected when the WiFi was disconnected.

8. Google Nest

Figure 31: Google Nest Logo [33]

The google nest is a smart speaker that can control compatible smart devices by voice. Utilization of this smart speaker allows users to integrate in-house and third-party services into the google assistant. By integrating these types of services, the users may control them by voice. Some possibilities may be listening to music, receiving news updates, and updating your calendar with tasks. The users may also integrate supported smart devices for home automation purposes. [33]. Some device manufacturers may include but are not limited to Phillips Hue, Samsung, Nest, Chamberlain, Roomba, and the list goes on [34]. These devices may also be used by voice commands [33].

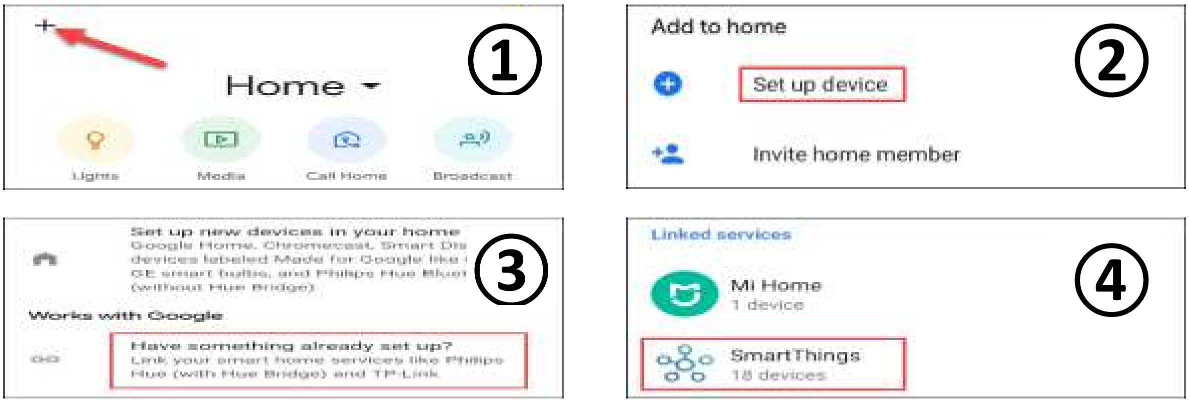
Vamiaeasy had not yet configured the google nest. In fact, it was still sitting in its original box and had never been powered on before the Vamiaeasy team arrived. Configuring this smart speaker did not have the highest priority; however, our group decided to move forward with getting this device connected because the configuration of this smart speaker and the integration of the devices into the speaker is among the easiest tasks to complete.

8.1. Configuring Google Nest

① Plug in your Google Home connected speaker. The device will start searching for your phone.

② Download the Google Home app ( ) with your phone which must be connected to Wi-Fi.
③ Sign in with a Google account and then tap the setup icon in the upper right corner to add the device.

④ After that you can add other connected devices like SmartThings (Samsung devices), Philips Hue bulbs…
Below is an example to add Samsung devices using the application



Set up new devices in your home
Google Home, Chromecast, Smart Dis
devices labeled Made for Google like (
GE smart bulbs, and Philips Hue Bluet
(without Hue Bridge)

Works with Google

Have something already set up?
Link your smart home services like Philips Hue (with Hue Bridge) and TP-Link

Mi Home
1 device

SmartThings
18 devices

Figure 32: Getting Started with Google Home Assistant

8.2. Connecting ABB free@home to the Google Nest

The team has dedicated a small section for configuring the ABB free@home system to the google nest because it is not as intuitive as the other google nest compatible devices. Although it is not as intuitive, ABB has made a presentation which walks through the necessary steps to get the free@home devices connected to the google assistant [35]. This step-by-step process assisted us in getting ABB connected to the google home assistant, but there were some other miscellaneous steps not mentioned in reference [35] which will be discussed in detail below.

1. Create an account at mybuildings.abb.com
2. Log in to the free@home user interface
    - You may access the free@home system by either entering the IP address of the system access point into your browser (see section 5.1.) or on the mobile application
    - Note: your device must be connected to the same WiFi as the ABB system
3. Go through the following selections "Settings" > "MyBuilding Portal" > "Connection" and enter the mybuildings username and password account information
    - Note: After you click "connect" the user will see a prompt which states "The fee for the use of the remote control is 2.48 EUR per month." This can be disregarded for the integration of free@home into Google nest
4. Check the box next to "Activate the free@home API"
5. Open the Google Assistant app on your smartphone or tablet
    - Open your profile page
    - Press "Home" > "Add" > "Link a smart device" > find and select "free at home"
6. You will be forwarded to the myBusch-Jaeger portal. Type in your MyBuildings Portal username and password.
    - If the connection was successful, then you will be returned to the google assistant app.

9. HomeAssistant (HA)

9.1. Home assistant installation guide

As previously mentioned, HomeAssistant (HA) is an IoT hub that has the ability to integrate all kinds of different devices into one user-interface. This is key for completing our team's main objective. Since HomeAssistant has a large pool of companies and brands it can work with, this makes HA a very attractive option for using in the smart home sector.

For the installation of HA there has been made use of certain guides in information from the internet. This mainly is the information on the HA official website. But also, some other guides and links have been used for this. The guide that has been made for installing is partially instructions of the official guide and partially instructions that we found somewhere else. [36, 37].

HomeAssistant is a free opensource home automation operating system. It is a central control system with an emphasis on privacy and local control. The next part will be a description of how to install HA on a raspberry pi.

9.1.1. Step-by-Step Instructions for Installing HomeAssistant

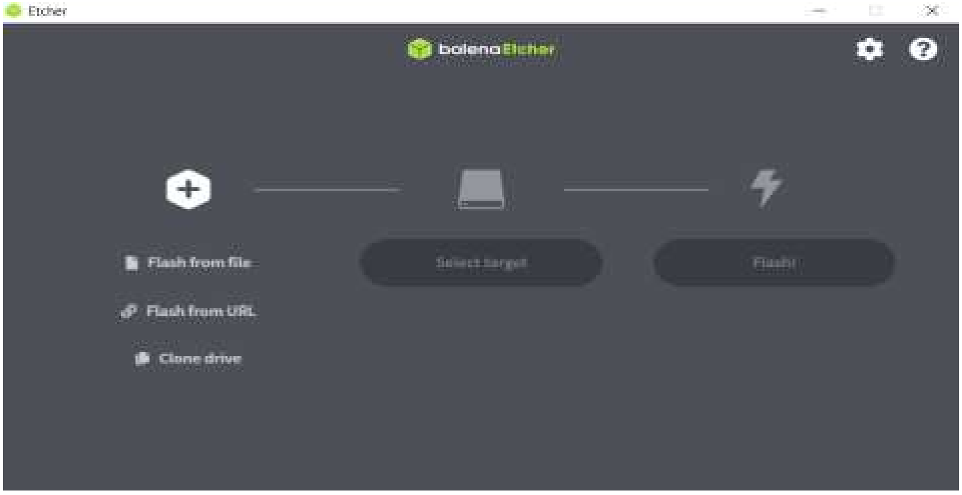


Figure 33: Balena etcher home page

1. The first step is making a boot SD card for Home Assistant. This can be done using Balena Etcher. And it can be downloaded for free from the following link: https://www.balena.io/etcher/
2. After installing etcher, you are presented with the following interface, in this interface you choose the Flash from URL option. You can get the URL from the link below. https://www.home-assistant.io/installation/raspberrypi
3. Paste in the URL and after that select that as a target.

4. The next step is to use the select your SD card that you plugged in as a target to flash. After you have done that click flash.

Wi-Fi setup

1. Plug the SD card back into your PC and open the Hannos Boot folder.
2. In this create a new folder named “CONFIG”.
3. Go into “CONFIG” and create a new folder named “network”.
4. Go into “network” and create a new text file named “my-network”.
5. Open the TXT file and past in there the following lines of code:

```
[connection]
id=my-network
uuid=72111c67-4a5d-4d5c-925e-f8ee26efb3c3
type=802-11-wireless

[802-11-wireless]
mode=infrastructure
ssid=MY_SSID
# Uncomment below if your SSID is not broadcasted
#hidden=true

[802-11-wireless-security]
auth-alg=open
key-mgmt=wpa-psk
psk=MY_WLAN_SECRET_KEY

[ipv4]
method=auto

[ipv6]
addr-gen-mode=stable-privacy
method=auto
```

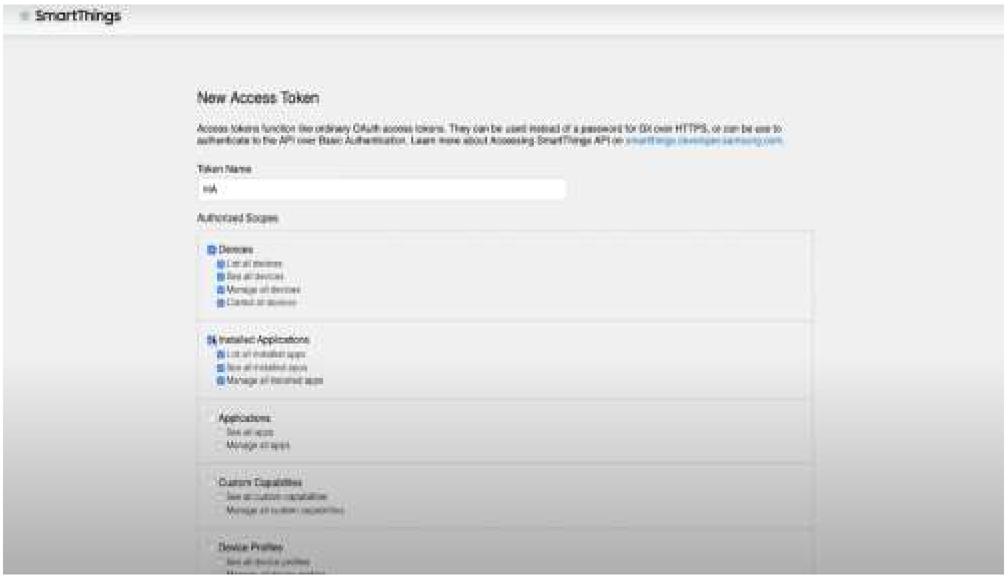
6. Next step is to fill in your ssid (Wi-Fi name) where it says MY_SSID and the Wi-Fi password in the MY_WLAN_SECRET_KEY.
7. After this save the document and you can take the SD card out of your pc and plug it into your Raspberry pi.
8. Then plug in the power cable into your raspberry pi.
9. Then go the following web address in your browser: homeassistant.local:8123
10. After you have done this, you should see a screen saying that you need to wait for about 20 minutes.
11. After that you should see this screen and it means you have successfully created a home assessment file

9.2. Integration of SmartThings into HomeAssistant

SmartThings is the platform used by Samsung to connect their Samsung devices. Besides this they also offer their own smart hub called the SmartThings hub. In the Vamiaeasy room there are two devices that make use of this SmartThings application. These are the Samsung smart tv and the Samsung Family Hub (fridge). To integrate these devices, the SmartThings application which contains all the data needs to have access to HA to share this data.

In order to accomplish this, there are a few steps required. The first step is to get a personal access token. Besides this you will also need access to the HA cloud service (NabuCasa). This cloud service is a paid subscription of €5, - per month. Taking this subscription grants besides the integration of SmartThings also the ability to gain remote access to your HA. Then there is also the ability for your google home to be linked with HA and you will be able to control you HA trough the google speaker. [38]

To start with you need to get your personal access token. Go to the following link to get it: https://account.smartthings.com/tokens
Sign in with your Samsung account and click generate new token. Then write down the name of this token for example HA as I called it in my example down below and check all the boxen for generating the access token (Figure 34).



SmartThings

New Access Token

Access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be use to authenticate to the API over Basic Authentication. Learn more about Accessing SmartThings API on smartthings.developer.samsung.com

Token Name

HA

Authorized Scopes

☑ Devices
- ☑ List all devices
- ☑ See all devices
- ☑ Manage all devices
- ☑ Control all devices

☑ Installed Applications
- ☑ List all installed apps
- ☑ See all installed apps
- ☑ Manage all installed apps

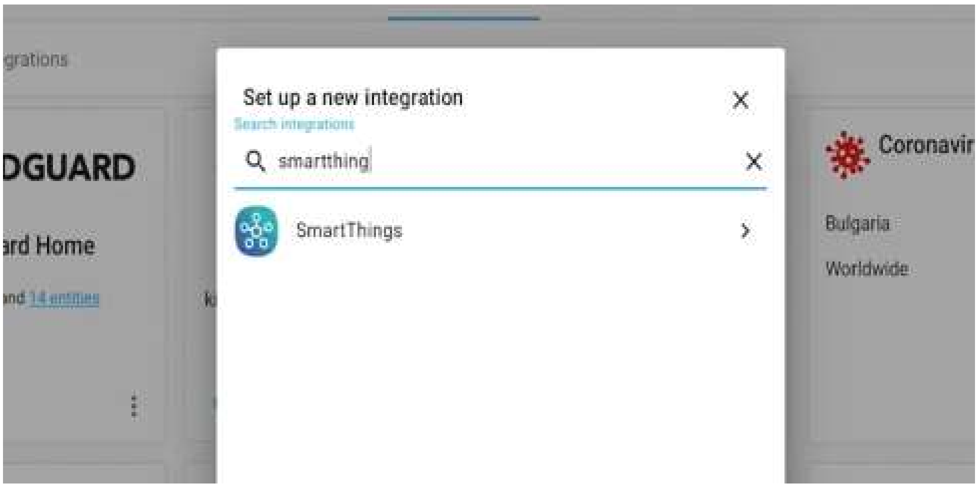
□ Applications
- □ See all apps
- □ Manage all apps

□ Custom Capabilities
- □ See all custom capabilities
- □ Manage all custom capabilities

□ Device Profiles
- □ See all device profiles

Figure 34: SmartThings Token Generator

When you have pressed generate token you will be presented with a big string of letters and number. Copy this string and save it somewhere you don't lose it. Now after this go back into Home Assistant and go to integrations and search for SmartThings (Figure 35 on page 65).



Set up a new integration

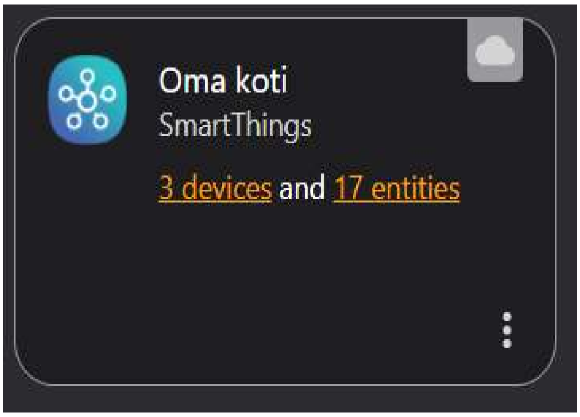
Search integrations

smartthing

SmartThings

Figure 35: SmartThings icon in HomeAssistant integrations

Now smarterthings will ask you for your personal access token. Enter this token and press submit. After this it will ask you to authorize HomeAssistant and press open website. Press the done button in the following screen and after that press authorize in the right bottom corner. Now all the devices from smarterthings should be visible in HomeAssistant (Figure 36). For our instance this means that we will have access to the current temperature of the fridge and see the power usage of the fridge and if it is closed or open.



Oma koti

SmartThings

3 devices and 17 entities

Figure 36: Confirmation of successful SmartThings integration

9.3. Integration of Google Nest into HomeAssistant

The google nest is also one of the devices that is integrated into the room. It is a very important device to integrate into the HomeAssistant. This is because it is essential for when you want to control the room via voice command and having this ability creates a big wow factor for when using the room for demonstration purpose. Furthermore, having the ability to control certain equipment can make the life of an elderly also easier. It is a big plus when an elderly or disabled person doesn't have to get up from their chair to turn on the light.

The integration of the google nest can be accomplished by using the build in auto discovery function. This way it saves a load of time, and the process is easy. To start with you go into you HA menu and go the configuration menu in the left bottom (Figure 38).

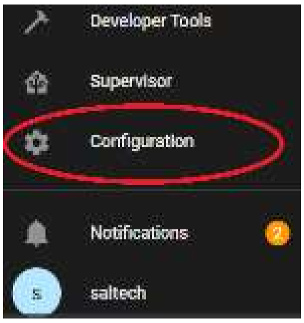


Figure 38: Locating configuration tab in HomeAssistant

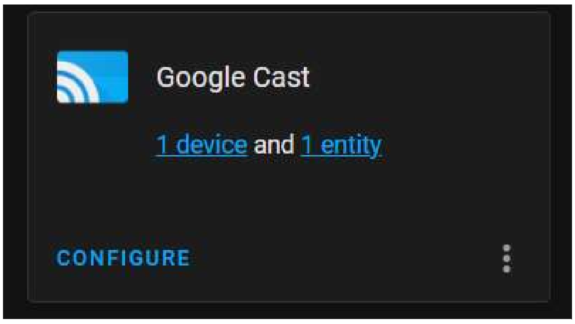


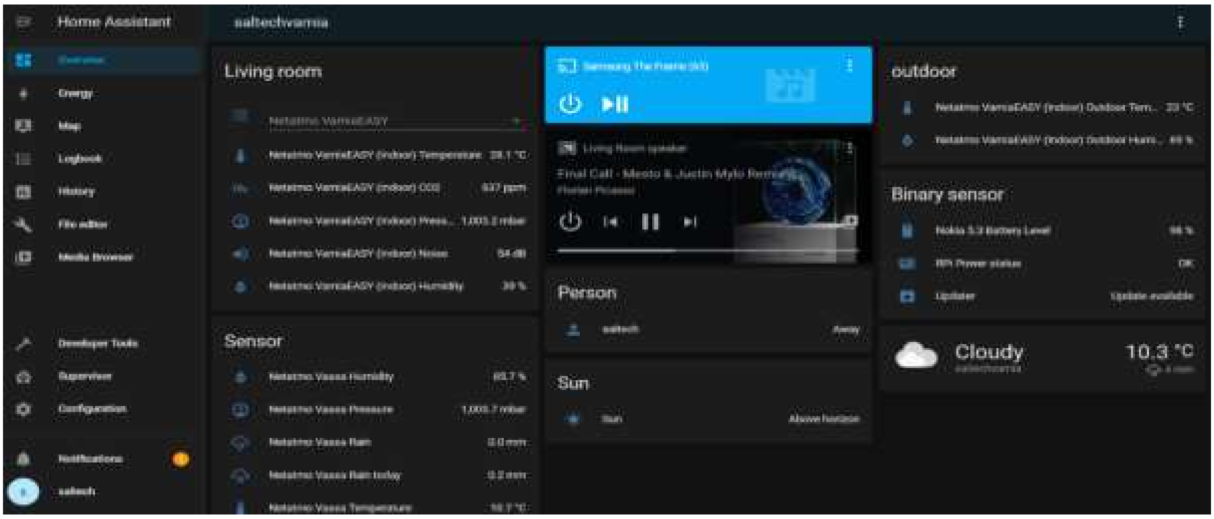
Figure 37: Configuring Google cast into HomeAssistant

After that go to integrations. And there the google nest will be visible as a device that you can configure (Figure 37). Once you press configure, it will automatically integrate your google home as a castable device and you will be able to control it form the home assistant main dashboard.

9.4. Integration of Netatmo into HomeAssistant

The Netatmo weather station in the Vamiaeasy room consist of two devices. The first device is an outdoor weather station that can be setup somewhere outside and the second device is an indoor weather station. Both of these devices can measure several in-room stats. For example, they can measure the room temperature and the room humidity.

In order to connect the Netatmo weather station the user must go to the integrations tab in Home Assistant. Then press the add integrations button and search for Netatmo. Press add and follow the instructions on screen. After this is done you should have access to the Netatmo weather information in the main dashboard. And it should look like the image shown in Figure 39. [39]



Living room

Netatmo VarnaEASY

Netatmo VarnaEASY (indoor) Temperature 28.1 °C

Netatmo VarnaEASY (indoor) CO2 637 ppm

Netatmo VarnaEASY (indoor) Press... 1,005.2 mbar

Netatmo VarnaEASY (indoor) Noise 54 dB

Netatmo VarnaEASY (indoor) Humidity 39 %

Sensor

Netatmo Vaasa Humidity 85.7 %

Netatmo Vaasa Pressure 1,005.7 mbar

Netatmo Vaasa Rain 0.0 mm

Netatmo Vaasa Rain today 0.2 mm

Netatmo Vaasa Temperature 10.7 °C

Person

saltech Away

Sun

Sun Above horizon

outdoor

Netatmo VarnaEASY (indoor) Outdoor Tem... 23 °C

Netatmo VarnaEASY (indoor) Outdoor Humi... 49 %

Binary sensor

Nokia 5.3 Battery Level 98 %

RPi Power status OK

Updater Update available

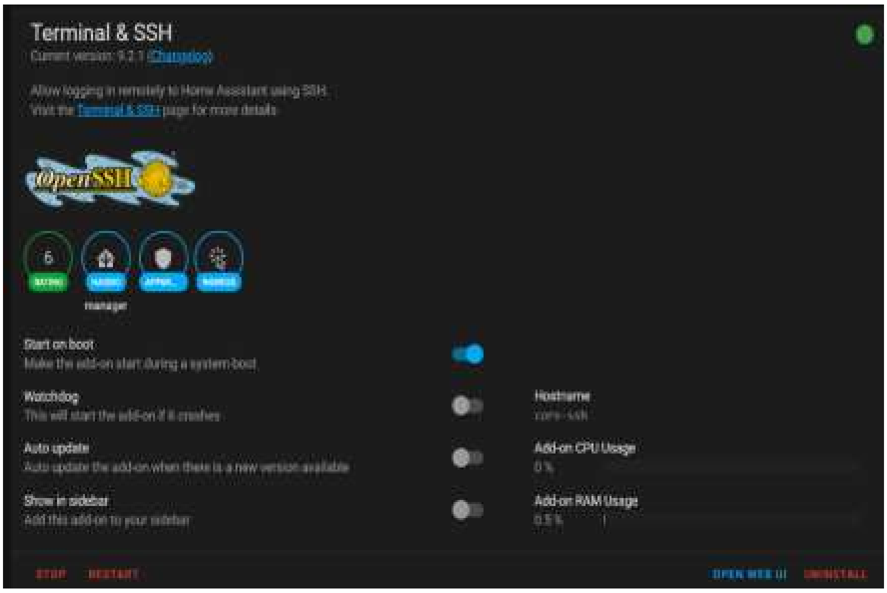
Cloudy 10.3 °C

Figure 39: HomeAssistant dashboard with the Netatmo data

9.5. Integration of free@home into HomeAssistant

To get the free@home integration working in HomeAssistant, its components first had to be installed into a directory. The installation directory needs to be in the same directory as the configuration.yaml file. But first to gain access to this directory, we need access to HA’s files.

We do this with the Terminal & SSH addon in HomeAssistant (Figure 40). To get this addon to show up in the addon store we had to enabled advanced mode in the profile settings [40].



Terminal & SSH
Current version: 9.2.1 (Changelog)

Allow logging in remotely to Home Assistant using SSH.
Visit the Terminal & SSH page for more details

Start on boot
Make the add-on start during a system boot

Watchdog
This will start the add-on if it crashes

Auto update
Auto update the add-on when there is a new version available

Show in sidebar
Add this add-on to your sidebar

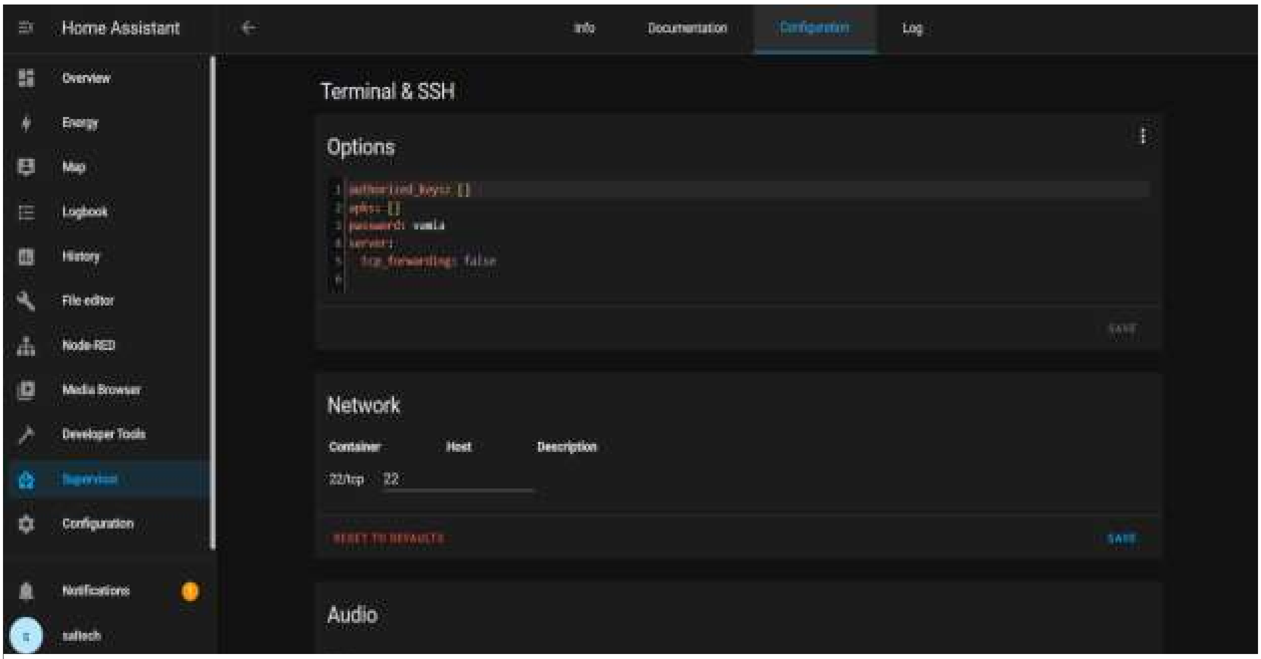
Hostname
core-ssh

Add-on CPU Usage
0 %

Add-on RAM Usage
0.5 %

Figure 40: Terminal & SSH addon for free@home integration into HomeAssistant

After installing the addon go into configuration and add a password in there. Then in network fill in 22. This is port 22 which is used for SSH communication protocol. (Figure 41).



Terminal & SSH

Options

```
authorized_keys: []
apks: []
password: vamia
server:
  tcp_forwarding: false
```

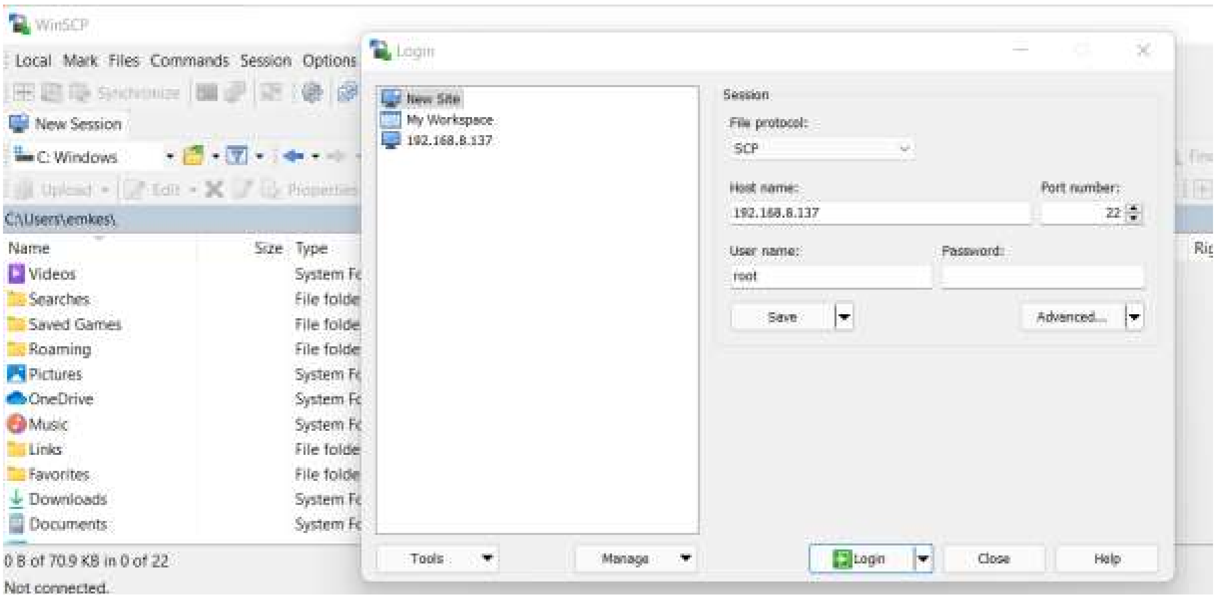
Network

Container   Host   Description
22/tcp   22

Audio

Figure 41: Configuration credentials for free@home integration into HomeAssistant

Then you have to install the program WinSCP on your computer. After this you have to press new site and then the screen shown in Figure 42 pops up. There you have to fill in the ip address of HomeAssistant and in the port number section you fill in port 22. The username is root and the password you have to use is the password you entered in the configuration file of the Terminal & SSH addon.



Session

File protocol: SCP

Host name: 192.168.8.137

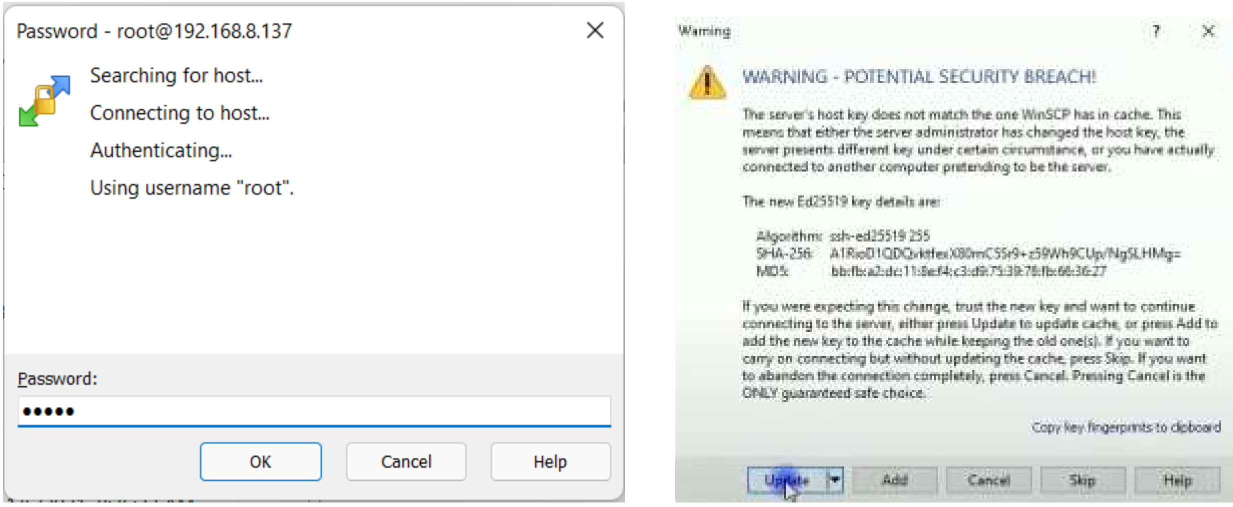
Port number: 22

User name: root

Password:

Figure 42: Winscp file transfer screenshot.

Next what will happen is that it will prompt you asking your password (Figure 43). Then you fill in your password again and press ok.



Password - root@192.168.8.137

Searching for host...
Connecting to host...
Authenticating...
Using username "root".

Password: •••••

Warning

WARNING - POTENTIAL SECURITY BREACH!

The server's host key does not match the one WinSCP has in cache. This means that either the server administrator has changed the host key, the server presents different key under certain circumstance, or you have actually connected to another computer pretending to be the server.

The new Ed25519 key details are:

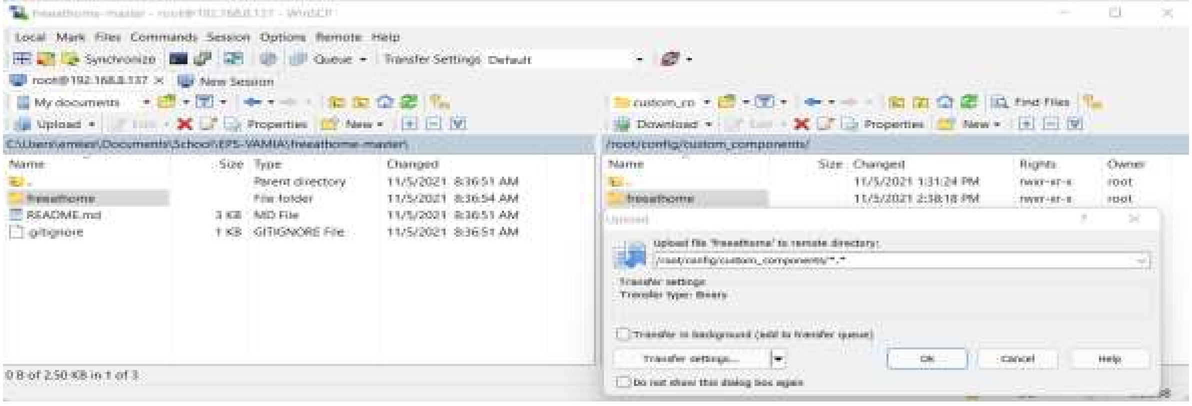
```
Algorithm: ssh-ed25519 255
SHA-256:   A1RioD1QDQvktfexX80mC55r9+z59Wh9CUp/NgSLHMg=
MD5:       bb:fb:a2:dc:11:8e:f4:c3:d9:75:39:78:fb:66:36:27
```

If you were expecting this change, trust the new key and want to continue connecting to the server, either press Update to update cache, or press Add to add the new key to the cache while keeping the old one(s). If you want to carry on connecting but without updating the cache, press Skip. If you want to abandon the connection completely, press Cancel. Pressing Cancel is the ONLY guaranteed safe choice.

Figure 43: winSCP password and connection

When you see the image shown in Figure 43 you may click the update button. This will enable you to get access to the files located on the raspberry pi.

After this you have access to the files of your local Home assistant instance. Then you go to your config folder and create a new folder named: “custom_components” (Figure 44). When you have done this you can move the freeathome folder that you can download on github [41].



C:\Users\emiel\Documents\School\EPS-VAMIA\freeathome-master\

Name	Size	Type	Changed
..		Parent directory	11/5/2021 8:36:51 AM
freeathome		File folder	11/5/2021 8:36:54 AM
README.md	3 KB	MD File	11/5/2021 8:36:51 AM
.gitignore	1 KB	GITIGNORE File	11/5/2021 8:36:51 AM

0 B of 2.50 KB in 1 of 3

/root/config/custom_components/

Name	Size	Changed	Rights	Owner
..		11/5/2021 1:31:24 PM	rwxr-xr-x	root
freeathome		11/5/2021 2:38:18 PM	rwxr-xr-x	root

Upload file 'freeathome' to remote directory:
/root/config/custom_components/*.*
Transfer settings
Transfer type: Binary
☐ Transfer in background (add to transfer queue)
Transfer settings... OK Cancel Help
☐ Do not show this dialog box again

Figure 44: Exhanging filed to raspberry pi through WinSCP

After this a restart of HA is required, this is so the system will be able to properly install the new placed custom components. After the reboot you will be able to go to the integrations page on HA and on the page it there you can search for the freeathome binding. It should now be visible in there and you can click on it then. After that you will be prompted to input your free@home username and password which is found within the following directory in the free@home system: setting > activate free@home API (Figure 46).

In this pop up shown in Figure 46 you fill in the the proper information. So for the host the input is the ip address of the System accespoint of ABB. The username is the username of the ABB system and the password is the abb system password.

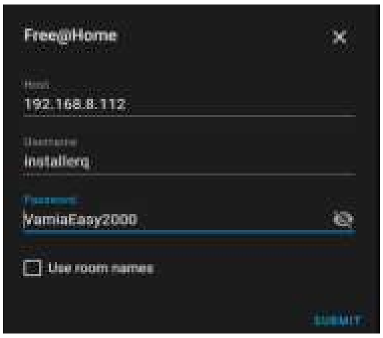

Free@Home
Host
192.168.8.112
Username
installerq
Password
VamiaEasy2000
☐ Use room names
SUBMIT

Figure 46: Inputting free@home information for HomeAssistant integration

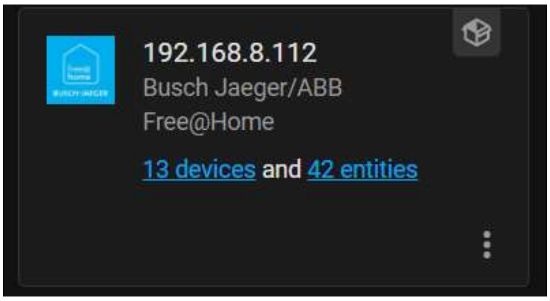

192.168.8.112
Busch Jaeger/ABB
Free@Home
13 devices and 42 entities

Figure 45: Confirmation of free@home HomeAssistant integration

Press the submit button and after that the system should be integrated into HA. You can check it by going to the current integrations and there it should say Busch Jeager/ABB (Figure 45). Here you can find all the 13 devices that are connected which contain 42 entities. These devices are lights, sensors, window blinds and the buttons.

9.6. Home Assistant Community Store (HACS) integration

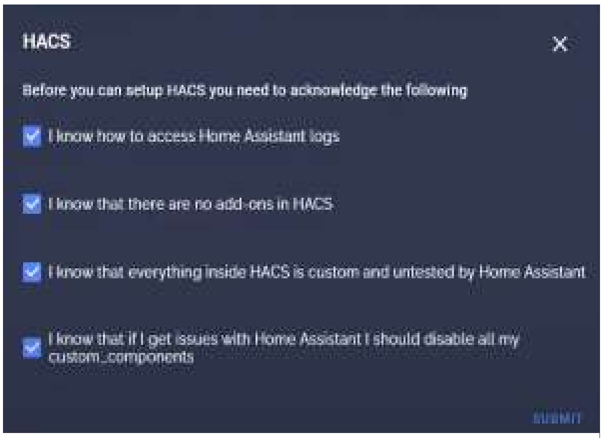
The HA base main interface is an interface that has all the devices put on one main page. This means that it holds a load of information on them. Since the HA will be used by a variety of other people, for which such interface might not be as intuitive. There is need for making a good-looking dashboard that is easy to understand and doesn't boast as much information as the current one. To fix this problem there are different tools that we can we use and one of them is called the HACS (Home assistant community store).

This community store holds varies by community members created overhauls for the main dashboard or better called Lovelace. To gain access to this we first need to enable to the HACS. We do this by going to our terminal to talk to HomeAssistant. Then we type and run the following command in there.

```
wget -q -O - https://install.hacs.xyz | bash –
```

After this command has been executed HA needs to be restarted. When HA is rebooted, you should be able to find HA in the integrations tab when you search for a new integration. After this you should check all the boxes in the prompt shown in Figure 48.

Then the prompt seen in Figure 47 will be shown. For this there was a github account created for the Vamiaeasy project team to link this with HA. Besides creating an account you also need to use your authorization key for HACS. With this key you give HACS to use your github account to download repositories from there.



HACS

Before you can setup HACS you need to acknowledge the following

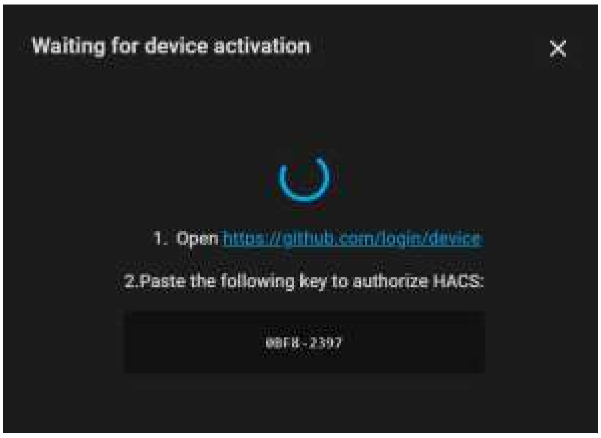
☑ I know how to access Home Assistant logs

☑ I know that there are no add-ons in HACS

☑ I know that everything inside HACS is custom and untested by Home Assistant

☑ I know that if I get issues with Home Assistant I should disable all my custom_components

Figure 48: HACS integration prompt



Waiting for device activation

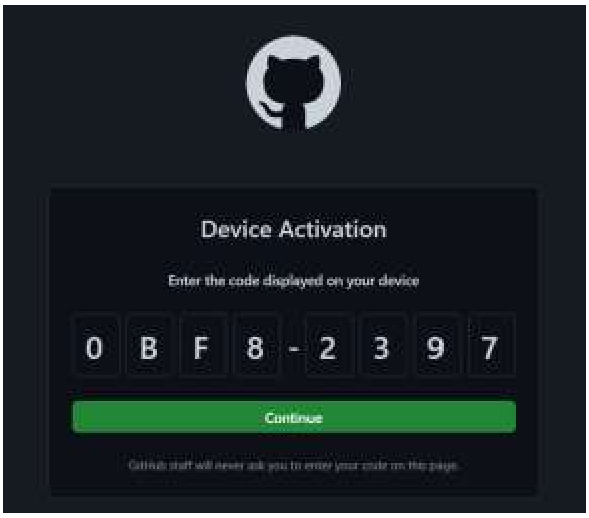
1. Open https://github.com/login/device

2.Paste the following key to authorize HACS:

0BF8-2397

Figure 47: HACS device activation

When you click on the link you will be redirected to the website where you are asked to fill in the code displayed on your device (Figure 49). Fill in the code and press continue to link the accounts. Then in the following menu shown in Figure 50, you must authorise the account that you linked. This way you finalize the link.

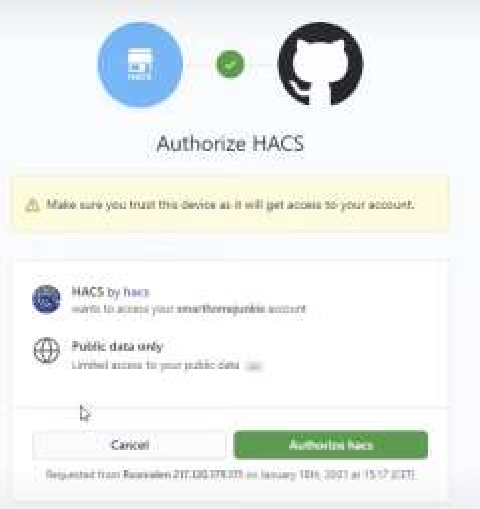


Device Activation

Enter the code displayed on your device

0 B F 8 - 2 3 9 7

GitHub staff will never ask you to enter your code on this page.



Authorize HACS

⚠ Make sure you trust this device as it will get access to your account.

HACS by hacs
wants to access your smarthomepunkie account

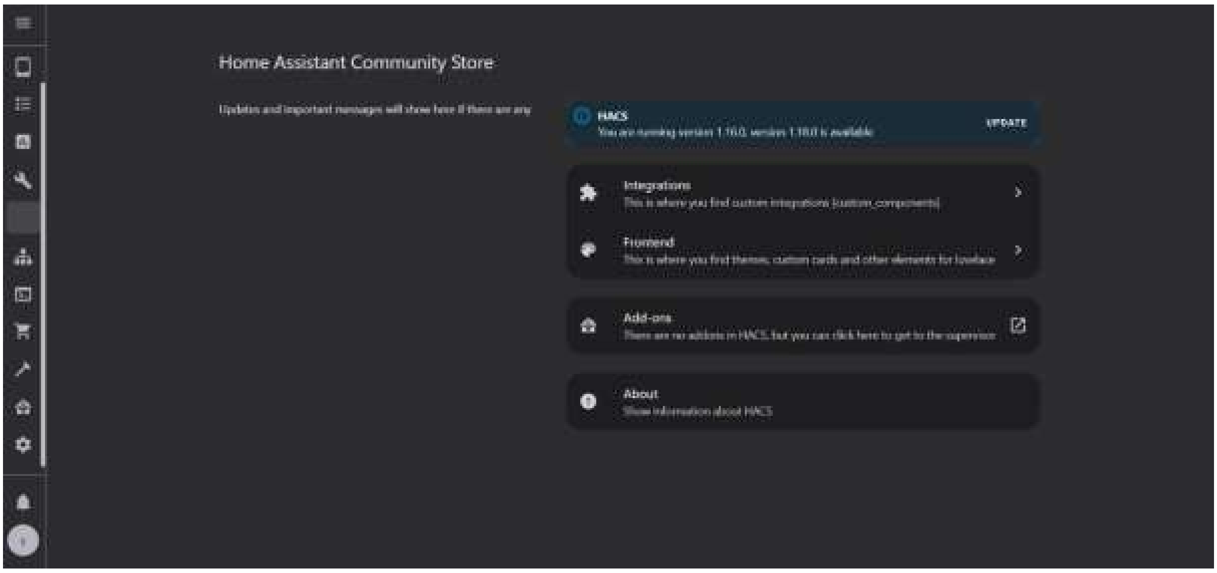
Public data only
Limited access to your public data

Requested from Raaiwalen 217.120.179.175 on January 18th, 2021 at 15:17 (CET)

Figure 49: Entering device activiation code for HACS integration

Figure 50: Authorizing HACS

After this you have access to the HACS if you go into the left panel (Figure 51). You will be fronted with the following page in HACS. Now you can go into the frontend section and from there download custom panels for the Lovelace.



Home Assistant Community Store

Updates and important messages will show here if there are any

HACS
You are running version 1.16.0, version 1.18.0 is available

Integrations
This is where you find custom integrations (custom_components)

Frontend
This is where you find themes, custom cards and other elements for lovelace

Add-ons
There are no addons in HACS, but you can click here to get to the supervisor

About
Show information about HACS

Figure 51: Confirmation of properly integrated HACS

Besides this you also have the ability to choose a theme for your Home Assistant. The Theme that we went for is an IOS dark theme. This is an apple inspired theme to give it a sleek and well looking touch.

As you can see in the image shown in Figure 52, we made a few big changes to the dashboard. The entire dashboard is overhauled and made more intuitive and easier to understand.

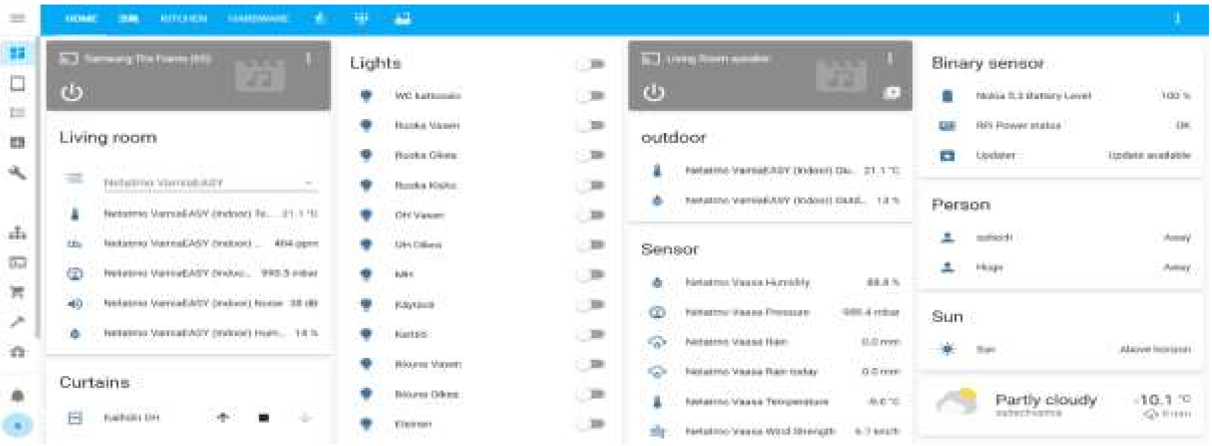


Figure 52: The first iteration of the HomeAssistant dashboard

The image shown in Figure 53 represents the dashboard the elderly will be seeing when they look on their iPad. There has been some thought into the dashboard and what features would be nice to have for an elderly. First of all, we have all the scenes that are there to control the interior lighting and blinds in the house. These are shown on the left side of the screen. In the middle it shows the current temperature of the room and the temperature for the rest of the day. And the last part are the controls for the cabinet and the add a shopping list.

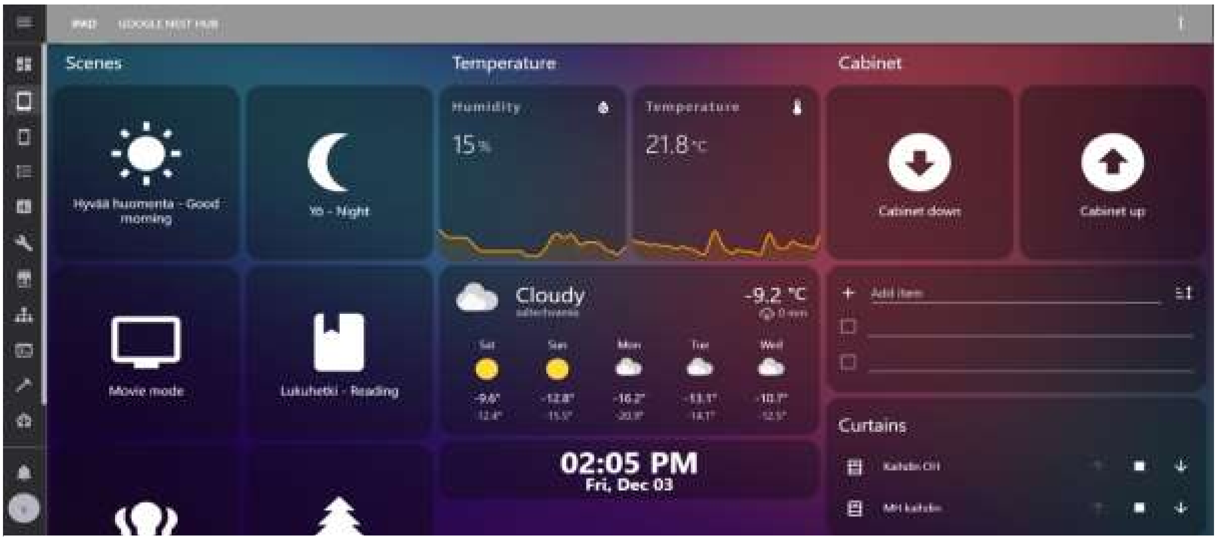


Figure 53: The HomeAssistant dashboard for the iPad

Then there has also been created a dashboard for the google nest hub that is sitting in the bedroom shown in Figure 54. The dashboard has a very intuitive and easy interface. This is because it also is serving as an alarm clock and a home control interface.

This dashboard can be shown on the google home assistant trough the means of casting. The protocol used is CATT(cast all the things). The only problem with this that we encounter at the moment is that it stops casting after a few minutes and renewing the cast is difficult. So there has been an automation created for renewing this service, but it is not functioning as intended. Therefore, it is something we leave for the next group to continue on and finish.

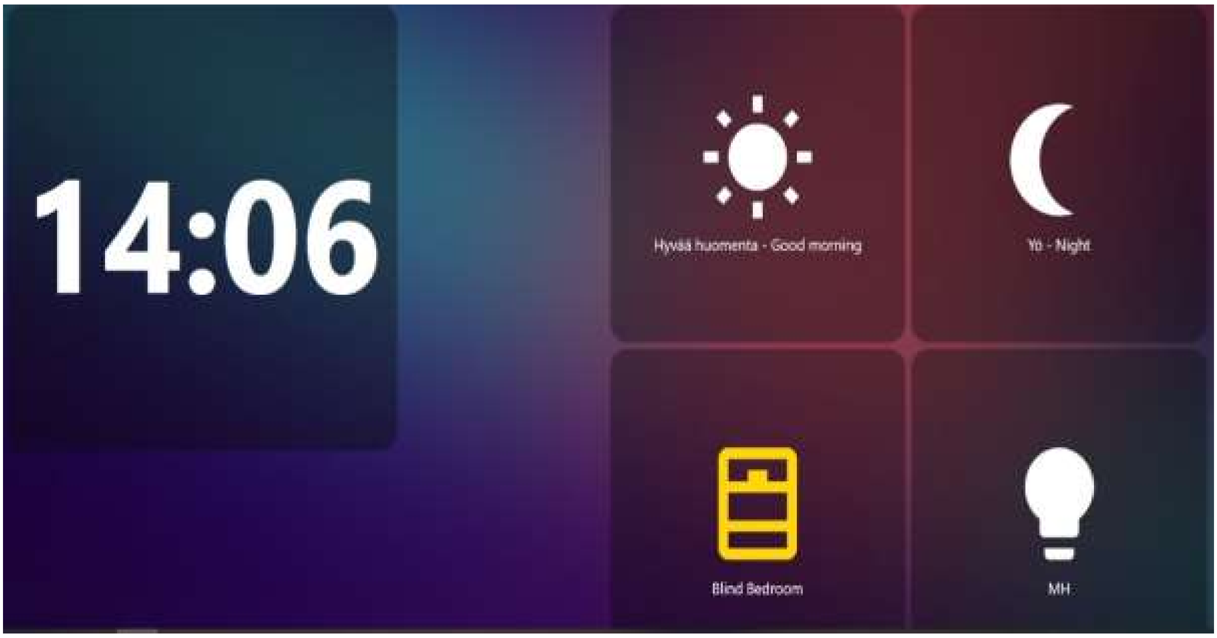


Figure 54: The HomeAssistant dashboard for the google nest hub

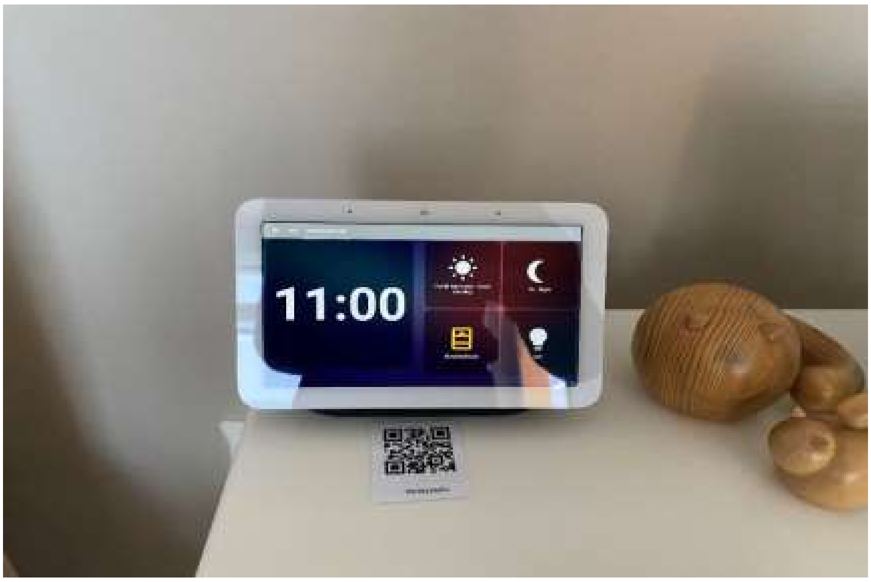


Figure 55: image of how the display looks in the room

9.7 Concluding HomeAssistant

Then to finalise this section there needs to be a little conclusion on why HomeAssistant was chosen instead of openHAB. To start off, the use of openHAB was more advantageous because openHAB offered the usage of ABB. This was also addressed by ABB itself that they had the ability to connect with this service, but after finding a way to get the ABB system into HomeAssistant the decision was made to work with HomeAssistant from that point on. So, after the breakthrough with ABB and looking at the other features HA has to offer, the team decided to continue with HA.

Some other great features from HomeAssistant is that it is more widely known and used. Their user community is bigger than openHAB and that offers more integrations and customization. Next to this point is the integration process in HomeAssistant is much easier for the developer. Another great advantage for HomeAssistant is that there is more information online about it. This makes it easier for somebody who has no experience in the smart home technology to use and learn about this platform. This especially applied for the project team this semester who had to start from scratch learning about home automation software.

To conclude HomeAssistant, the team delivered a fully functioning application that has the ability to control all the devices. The HomeAssistant platform links ABB and the 3rd party devices all together in one interface. This interface is easy to use for an elderly person and meets the requirements of the client. Besides this, the HomeAssistant cloud enables the ability to use voice commands which can almost control every device in the room. This also was one of the wishes of the client that has been fulfilled. Therefore, we can say that the usage of HomeAssistant was a good choice in regards to fulfilling the wishes of the client.

10. OpenHAB

10.1. Installation & Configuration of openHAB

Although HomeAssistant is being used by the group, we decided to find the pros and cons of the openHAB and HomeAssistant for ourselves. The number of devices which can be integrated into HomeAssistant is more than that of openHAB. However, the free@home system is easier to integrate into openHAB versus HomeAssistant, which is advantageous because the majority of the home is already being controlled through the free@home system.

The installation of openHAB has some similarities to HomeAssistant but they are not identically the same. In the similar case to HomeAssistants installation, there are many open-source which one may utilize to find the right openHAB configuration for their purposes [19, 20, 41, 42]. For our specific purposes we decided to stick with reference [43] because of their detailed steps for integrating third-party devices into, not only openHAB, but the ABB system as well.

Reference [43] is the open-source forum which has a subforum, referenced as [42], describing the installation of openHAB onto the raspberry and integrating the free@home system into openHAB. This reference does lack some essential steps for the installation and configuration of openHAB which is discussed further below.

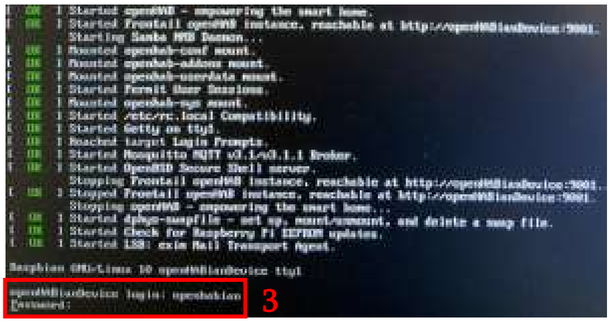
10.1.1. openHAB Installation

1. Download the following raspberry pi image file located in reference [42]
    - Preconfigured OpenHAB3 free@home Grafana influxDB NodeRed Image
2. Open the same balenaEtcher which was used for the HomeAssistant installation
3. Select the downloaded image file
4. Select the drive which routed to your SD card
5. Click flash

The installation of openHAB onto the raspberry pi should now be complete.

10.1.2. openHAB Configuration

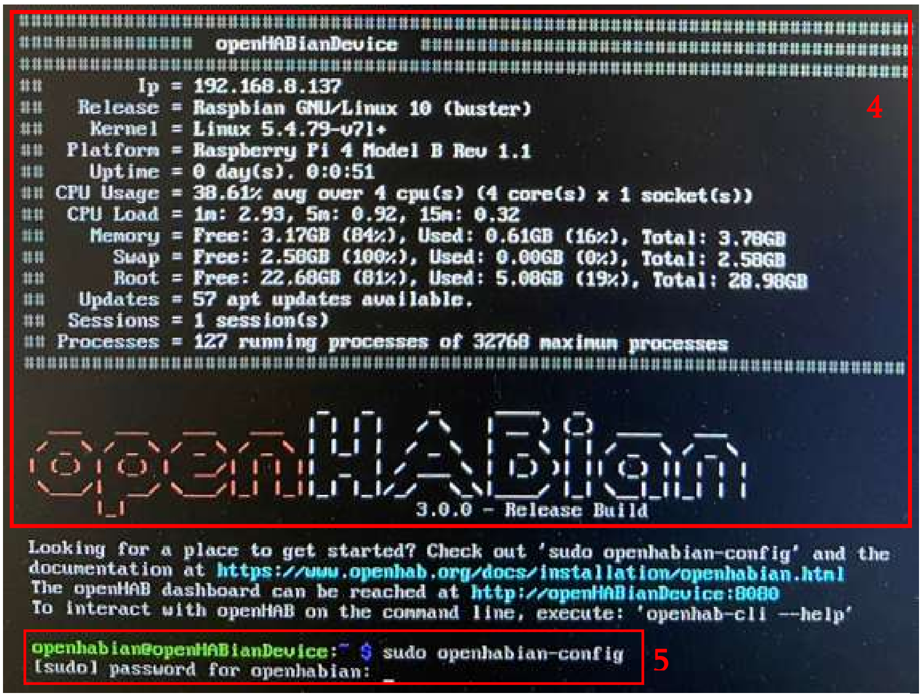
1. Before apply power, connect the raspberry pi to a monitor, keyboard, and an ethernet cable with internet connection.
2. Supply power to the raspberry pi
3. A prompt for the username and password will arise (Figure 56)
    - Username: openhabian
    - Password: openhabian
    - Note: Linux does not show the password as it is being written



```
[  OK  ] Started openHAB - empowering the smart home.
[  OK  ] Started Frontail openHAB instance, reachable at http://openHABianDevice:9001.
         Starting Samba NMB Daemon...
[  OK  ] Mounted openhab-conf mount.
[  OK  ] Mounted openhab-addons mount.
[  OK  ] Mounted openhab-userdata mount.
[  OK  ] Started Permit User Sessions.
[  OK  ] Mounted openhab-sys mount.
[  OK  ] Started /etc/rc.local Compatibility.
[  OK  ] Started Getty on tty1.
[  OK  ] Reached target Login Prompts.
[  OK  ] Started Mosquitto MQTT v3.1/v3.1.1 Broker.
[  OK  ] Started OpenBSD Secure Shell server.
         Stopping Frontail openHAB instance, reachable at http://openHABianDevice:9001...
[  OK  ] Stopped Frontail openHAB instance, reachable at http://openHABianDevice:9001.
         Stopping openHAB - empowering the smart home...
[  OK  ] Started dphys-swapfile - set up, mount/unmount, and delete a swap file.
[  OK  ] Started Check for Raspberry Pi EEPROM updates.
[  OK  ] Started LSB: exim Mail Transport Agent.

Raspbian GNU/Linux 10 openHABianDevice tty1

openHABianDevice login: openhabian
Password:
```

Figure 56: Logging into Raspberry Pi

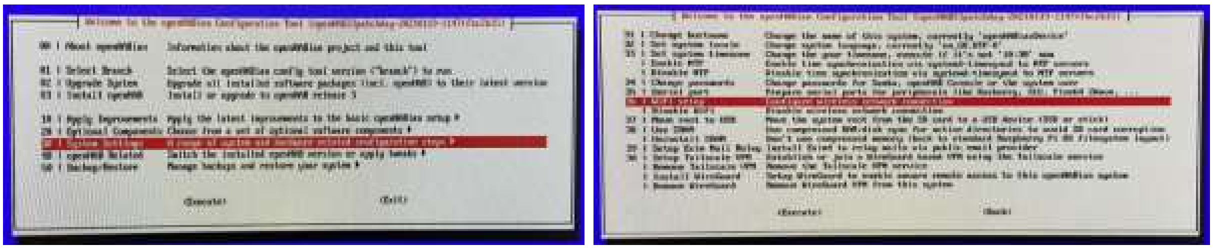
4. The devices IP address and openHABian welcome screen will arise (Figure 57)
    - Note: This IP address shown may not be the exact IP address the user will need to use in the coming steps, so disregard it for the time being.



```
#################################################################################
###############  openHABianDevice  ##############################################
#################################################################################
##        Ip = 192.168.8.137
##   Release = Raspbian GNU/Linux 10 (buster)
##    Kernel = Linux 5.4.79-v7l+
##  Platform = Raspberry Pi 4 Model B Rev 1.1
##    Uptime = 0 day(s). 0:0:51
## CPU Usage = 38.61% avg over 4 cpu(s) (4 core(s) x 1 socket(s))
##  CPU Load = 1m: 2.93, 5m: 0.92, 15m: 0.32
##    Memory = Free: 3.17GB (84%), Used: 0.61GB (16%), Total: 3.78GB
##      Swap = Free: 2.58GB (100%), Used: 0.00GB (0%), Total: 2.58GB
##      Root = Free: 22.68GB (81%), Used: 5.08GB (19%), Total: 28.98GB
##   Updates = 57 apt updates available.
##  Sessions = 1 session(s)
## Processes = 127 running processes of 32768 maximum processes
#################################################################################

openHABian
                          3.0.0 - Release Build

Looking for a place to get started? Check out 'sudo openhabian-config' and the
documentation at https://www.openhab.org/docs/installation/openhabian.html
The openHAB dashboard can be reached at http://openHABianDevice:8080
To interact with openHAB on the command line, execute: 'openhab-cli --help'

openhabian@openHABianDevice:~ $ sudo openhabian-config
[sudo] password for openhabian: _
```

Figure 57: openHAB Welcome Screen

5. In the "openhabian@openhab:~$" line, type "sudo openhabian-config" (Figure 57)
    - You will be prompted to input a password, it is the same password previously used, being "openhabian"



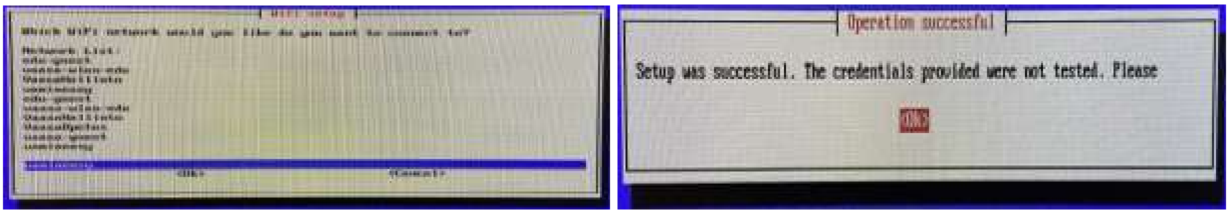
```
Welcome to the openHABian Configuration Tool [illegible]

00 | About openHABian      Information about the openHABian project and this tool

01 | Select Branch         Select the openHABian config tool version ("branch") to run
02 | Upgrade System        Upgrade all installed software packages (incl. openHAB) to their latest version
03 | Install openHAB       Install or upgrade to openHAB release 3

10 | Apply Improvements    Apply the latest improvements to the basic openHABian setup ►
20 | Optional Components   Choose from a set of optional software components ►
30 | System Settings       A range of system and hardware related configuration steps ►
40 | openHAB Related       Switch the installed openHAB version or apply tweaks ►
50 | Backup/Restore        Manage backups and restore your system ►

            <Execute>                         <Exit>
```

```
Welcome to the openHABian Configuration Tool [illegible]

31 | Change hostname         Change the name of this system, currently 'openHABianDevice'
32 | Set system locale       Change system language, currently 'en_US.UTF-8'
33 | Set system timezone     Change the your timezone, execute if it's not '[illegible]' now
   | Enable NTP              Enable time synchronization via systemd-timesyncd to NTP servers
   | Disable NTP             Disable time synchronization via systemd-timesyncd to NTP servers
34 | Change passwords        Change passwords for Samba, openHAB Console or the system user
35 | Serial port             Prepare serial ports for peripherals like Razberry, SCC, Pine64 ZWave, ...
36 | WiFi setup              Configure wireless network connection
   | Disable WiFi            Disable wireless network connection
37 | Move root to USB        Move the system root from the SD card to a USB device (SSD or stick)
38 | Use ZRAM                Use compressed RAM/disk sync for active directories to avoid SD card corruption
   | Uninstall ZRAM          Don't use compressed memory (back to standard Raspberry Pi OS filesystem layout)
39 | Setup Exim Mail Relay   Install Exim4 to relay mails via public email provider
3A | Setup Tailscale VPN     Establish or join a WireGuard based VPN using the Tailscale service
   | Remove Tailscale VPN    Remove the Tailscale VPN service
   | Install WireGuard       Setup WireGuard to enable secure remote access to this openHABian system
   | Remove WireGuard        Remove WireGuard VPN from this system

            <Execute>                         <Back>
```

Figure 58: Getting to openHAB WiFi set up

- At this point the "openhabian configuration tool" should be present (Figure 58)

6. Using the arrows keys on the keyboard, go down to "system setting" > "WiFi setup" (Figure 58)
   - At this point you should have a list of available WiFi networks
7. You must physically type in the WiFi SSID (WiFi name) which you would like to connect to. (Figure 59)
   - A prompt with "operation successful" will show. If this prompt is not present, then it is likely that there is a typo in the WiFi SSID or password (Figure 59)
8. After pressing "OK" in the operation successful page, the user will be forwarded to the "openhabian configuration tool" page. Press tab on the keyboard and exit the "openhabian configuration tool" page.



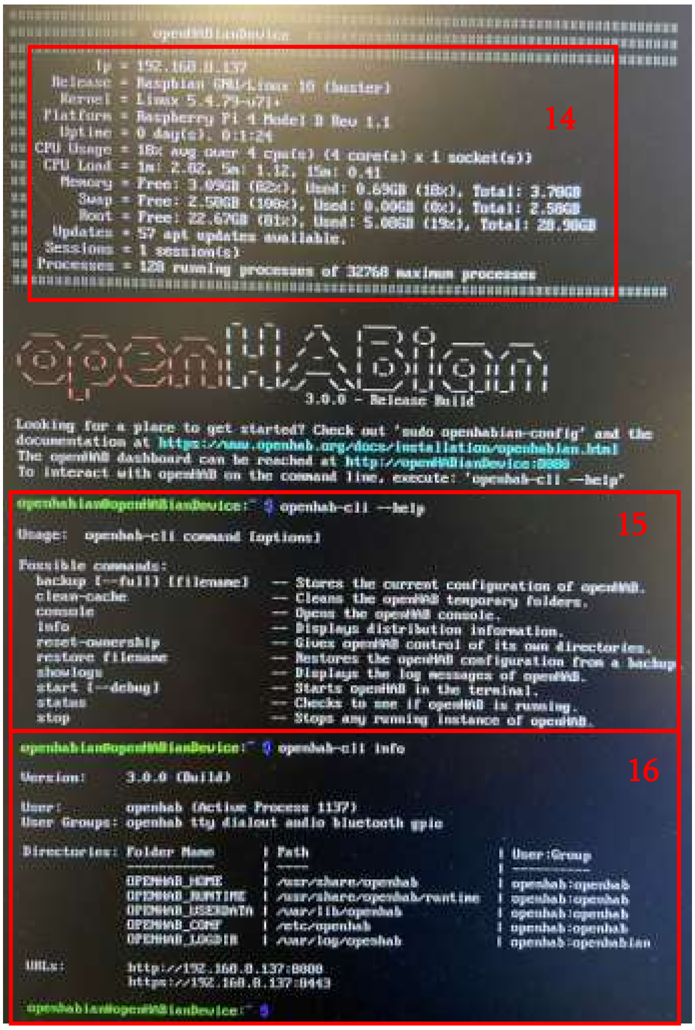
```
WiFi setup
Which WiFi network would you like do you want to connect to?

Network List:
[illegible]

            <Ok>                         <Cancel>
```

```
Operation successful

Setup was successful. The credentials provided were not tested. Please

                    <Ok>
```

Figure 59: WiFi set up

9. Remove the ethernet cable from the raspberry pi
10. In the "openhabian@openhab:~$" line, type "sudo reboot"
11. Power off the raspberry pi for a full reboot
12. Ensure that the ethernet cable is unplugged and power on the raspberry pi
13. Input the same username and password shown in step 3 within section 7.1.2.
14. The openHAB welcoming screen will show with the correct IP address associated with the device (Figure 60)
    - Write this IP address down as you will need it to gain access to the developer environment
15. In the "openhabian@openhab:~$" line, type "openhab-cli --help" (Figure 60)
    - This will show the user a list of useful commands
16. In the "openhabian@openhab:~$" line, type "openhab-cli info" (Figure 60)

- Take note of the IP address and ports listed.



```
                 openHABianDevice
        Ip = 192.168.0.137
   Release = Raspbian GNU/Linux 10 (buster)
    Kernel = Linux 5.4.79-v7l+
  Platform = Raspberry Pi 4 Model B Rev 1.1
    Uptime = 0 day(s). 0:1:24
 CPU Usage = 18% avg over 4 cpu(s) (4 core(s) x 1 socket(s))
  CPU Load = 1m: 2.02, 5m: 1.12, 15m: 0.41
    Memory = Free: 3.09GB (82%), Used: 0.69GB (18%), Total: 3.78GB
      Swap = Free: 2.50GB (100%), Used: 0.00GB (0%), Total: 2.50GB
      Root = Free: 22.67GB (81%), Used: 5.08GB (19%), Total: 28.98GB
   Updates = 57 apt updates available.
  Sessions = 1 session(s)
 Processes = 128 running processes of 32768 maximum processes

                              3.0.0 - Release Build

Looking for a place to get started? Check out 'sudo openhabian-config' and the
documentation at https://www.openhab.org/docs/installation/openhabian.html
The openHAB dashboard can be reached at http://openHABianDevice:8080
To interact with openHAB on the command line, execute: 'openhab-cli --help'
openhabian@openHABianDevice:~ $ openhab-cli --help

Usage:  openhab-cli command [options]

Possible commands:
  backup [--full] [filename]   -- Stores the current configuration of openHAB.
  clean-cache                  -- Cleans the openHAB temporary folders.
  console                      -- Opens the openHAB console.
  info                         -- Displays distribution information.
  reset-ownership              -- Gives openHAB control of its own directories.
  restore filename             -- Restores the openHAB configuration from a backup.
  showlogs                     -- Displays the log messages of openHAB.
  start [--debug]              -- Starts openHAB in the terminal.
  status                       -- Checks to see if openHAB is running.
  stop                         -- Stops any running instance of openHAB.
openhabian@openHABianDevice:~ $ openhab-cli info

Version:     3.0.0 (Build)

User:        openhab (Active Process 1137)
User Groups: openhab tty dialout audio bluetooth gpio

Directories: Folder Name      | Path                            | User:Group
             -----------      | ----                            | ----------
             OPENHAB_HOME     | /usr/share/openhab              | openhab:openhab
             OPENHAB_RUNTIME  | /usr/share/openhab/runtime      | openhab:openhab
             OPENHAB_USERDATA | /var/lib/openhab                | openhab:openhab
             OPENHAB_CONF     | /etc/openhab                    | openhab:openhab
             OPENHAB_LOGDIR   | /var/log/openhab                | openhab:openhabian

URLs:        http://192.168.0.137:8080
             https://192.168.0.137:8443

openhabian@openHABianDevice:~ $
```

Figure 60: openHAB homepage as well as help and info

The raspberry pi has now configured openHAB, and a user may begin developing automation scenarios in the openHAB environment.

Now that the raspberry pi has configured openHAB, the user must gain access to the developing environment. To gain access to the openHAB developing environment, the user must be connected to the same WiFi as the raspberry pi and type the following in their web browser: <ip address of the raspberry pi>:8080

10.2. Moving Forward with openHAB

Although the free@home system was easier to integrate into openHAB versus homeassistant; the Vamia team decided not to move forward with the device integrations into openHAB. This is due to the fact that homeassistant offers a significant increase in device integrations and we were able to integrate the free@home into homeassistant as well.

Before the successful integration of the ABB free@home into homeassistant, the team researched whether it would be possible to integrate the free@home system into openHAB then integrate openHAB into homeassistant. This does seem relatively convoluted, but we were determined in getting the ABB free@home system into homeassistant.

10.2.1. Integrating OpenHAB into home assistant.

The team has done some research into integrating openHAB into homeassistant. The findings are that it is possible to integrate one into the other; however, it is rather a long and intricate process. This is because both systems are ‘master’ control systems and cannot be controlled by one another [44]. In order to integrate openHAB into homeassistant one must use an MQTT broke. With MQTT you can run commands between OH and HA which in turn would allow you to run the free@home system through homeassistant.

11. Project Management Evaluation

During this project, changes and decisions could be made because of new insights. Examples for these new insights are unforeseen circumstances or an earlier miscommunication with the client. These choices are justified in this chapter. The attention is focused on the next project management topics:

- Work Breakdown Structure
- Scheduling
- Budgeting

11.1. WBS

For the duration of this EPS program, the project group noticed that the priorities were different than our expectations. The group expected that it was also possible to design an application based on the wishes of the client. During the project, the group discovered this was lower on the priority list. This conclusion is based on conversations with our client and conversations with the supervisor from Novia. The original WBS could be found in Appendix 6 on page 94.

After this discovery, the decision was made by all the members of the project group to spend this time connecting all the devices in one application. That is why the interviews have been postponed.

One task has been added to the WBS. It is the task of the 3D design of the Vamiaeasy room. The project group brainstormed about the project during one of the teambuilding sessions. During the covid pandemic could it be undesirable to receive guests in Vamia. It could also be prohibited by covid restrictions. These restrictions are exposed by local governments. The project group and Vamia do not influence these circumstances. An online 3D design has been created to counter this risk. A virtual tour could be given by the host on distance through using the 3D room. In addition to the 3D room, a presentation has been made. This interactive presentation contains all the devices with a description. The visitor can visit the Vamiaeasy room virtually without a guide and will not miss important information. The client approved this idea.

11.2. Risk assessment

In the beginning of the project, 24 risks were identified by the project group. These risks have caused some difficulties during the last months. These difficulties will be discussed in this section. Firstly, the three main risks will be discussed. These risks were identified in chapter 2.4. After that, the other risks which occurred will be discussed.

Three main risks:

1. **The room is always occupied by Vamia students or teachers:** After a conversation with the client, the decision was made to reserve the room far in advance. The project group was able to reserve the room at least three times a week. So this risk was mitigated very well.
2. **The password for free@home is not available:** The password for free@home was found a week after the midterm presentation. This was not a problem for our project.
3. **Covid-19 restrictions**: The group had not had any problems till the middle of November appeared. Covid-19 has been brought to Vaasa after a visit to an international party boat by international students. None of us were on the boat, but one person from the European Project Semester got infected. The infected person was a flatmate of Hugo. So Hugo had to stay in quarantine for ten days. Fortunately, it was at the end of the project. Documentation work had to be done and it was also possible to do at home. The covid-19 restrictions have not caused big problems.

Other risks

4. **Some devices cannot be integrated:** During the project, the discovery was made that some devices could not be integrated into the ABB free@home or Home Assistant. All these devices were not smart products at all. It was impossible to connect them to the internet. The whole team decided to create a list for the client with smart devices that can replace the current non-smart devices. This list was given to the client so she could make a decision easily.
5. **Lack of information about the devices**: In the beginning of the project, There was a desire to create an overview of all the devices that are present in the Vamiaeasy room. During the research, the project group discovered that some given information was false. For example: the documentation about the vacuum cleaner was wrong. The vacuum cleaner was different than the documented one.

11.3. Scheduling

The midterm report mentioned that the project is on schedule. The project members succeeded to finish the introduction documents, the project management documents and the system requirements on time. All these phases were passed well. Although the outcomes of these phases were not documented well. For example, references were not available in the midterm report. Other small corrections had been made before the project group started to work on the other tasks.

Secondly, another adjustment has been made. After a discussion, the group decided to combine two different tasks. It is about the following tasks: “choosing operating pi system” and “design data flow”. These tasks had similarities during the execution. It was hard to distinguish the two tasks in the project management tools. It was not clear on which task the worked hours should be registered. This is the reason why these tasks are combined under “choosing operating pi system”.

So the project group had to reschedule the Gantt chart. At the same time, the discovery was made that the priorities of the client were different than expected. The time for the interview has been transferred to the time for improvements. The second adjustment in tasks has been made to the schedule. The 3D design has been added to the schedule and replaces the “design data flow” task.

In the end, all the tasks (except the interviews) has been completed within time.

11.4. Budgeting

A budget has been presented in the midterm report (chapter 2.5). The research team made some predictions at the beginning of the project about the amount of time and money that was needed for this project. This chapter discusses the outcome of this prediction.

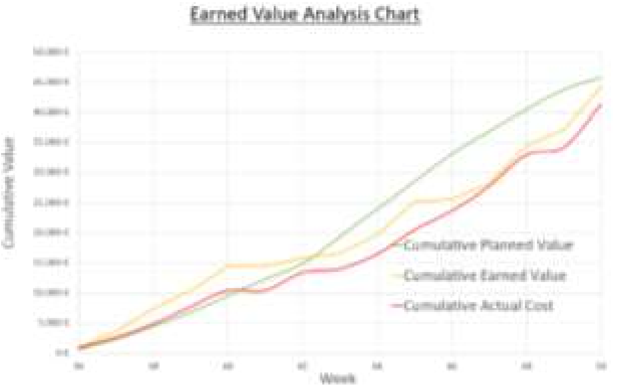


Figure 61: Earned value analysis chart

Three lines are being shown in the graph shown in Figure 61. The green line is the cumulative planned value line. This green line displays the project team’s prediction of the project. The prediction was that most of the tasks could not be finished in the first weeks. In addition, more tasks must have been finished in the last weeks of the project.

The yellow line shows the cumulative earned value. This line shows what tasks have effectively been finished. The line shows that not all the tasks have been completed at the moment. This is true because the elderly have never been interviewed by the research team. The reason for this is explained in chapter 11.1.

Finally, there is a red line. This line shows the cumulative actual cost. This line shows the amount of money that has been spent on the project. This graph visualizes the efficiency of the project. The yellow line is higher than the red line. This means that the earned value is higher than the actual costs. It is an efficient project when a project is completed with fewer costs than the scheduled costs. It could be concluded that this project has been executed efficiently.

12. Tracking Hours

Throughout the EPS semester, the group filled out an Excel spreadsheet of hours completed for the project. The principle is simple, there is a sheet with the total amount of hours spent by each member (Project Total_student), tasks performed (TaskList), and a sheet for each member of the group. The sheets can be seen in Figure 62.

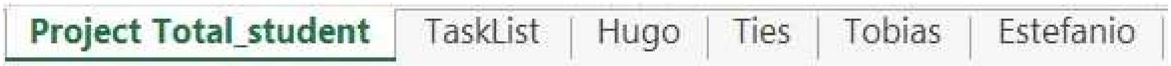


Figure 62: Excel sheets

In order to have an accurate result, the group filled in the spreadsheet each day, specifying the tasks. This allowed for a weekly review of the completed tasks.

Finally, the first summary sheet allows us to know, at the end of the project, how much time the group has produced for the project. Then when we convert it in hourly time, to have an idea of the budget that it represents. The graph shown in Figure 63 depicts the time spet by each group member for the duration of this project.

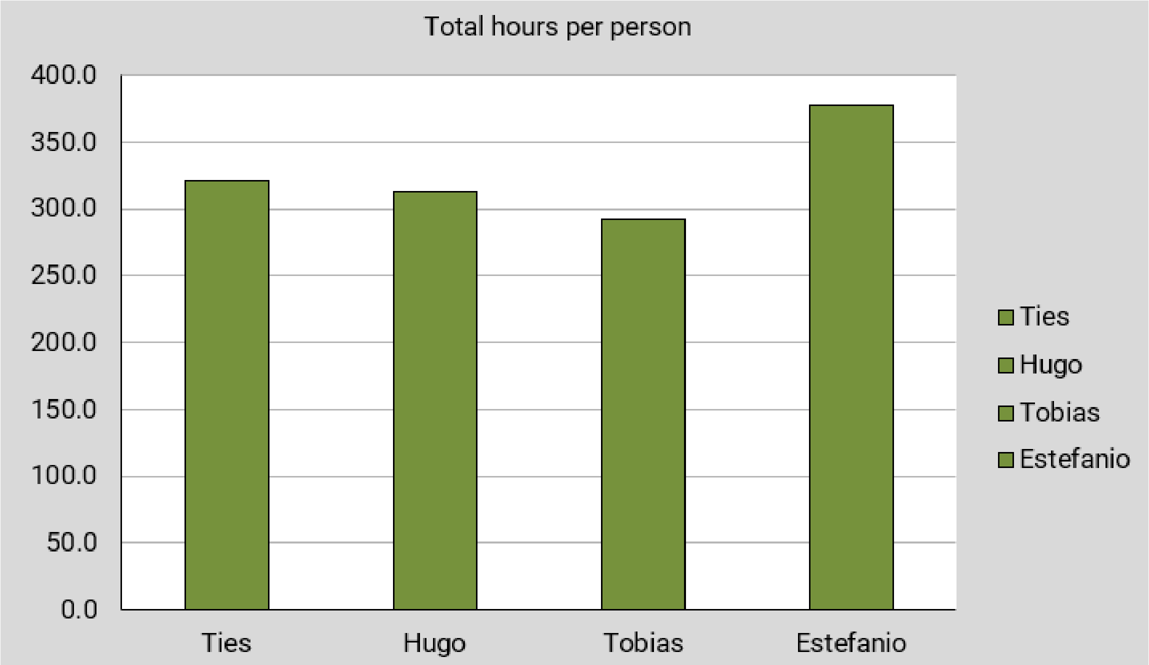


Figure 63: Summary of total hours worked per person

Conclusion

The main objective of this project was to develop an IoT platform to facilitate the use of smart devices in an automated living environment all within one intuitive user-interface. In addition to successfully creating this user-interface, this EPS team encountered the opportunity to further innovate the Vamiaeasy automated room.

The start of the project was the most difficult because we first had to agree on the expectations of the customer (Vamia) and obtain the passwords to all the applications and systems present in the showroom. With limited access to the room, organization was a crucial element of the project. This team did not waste any time defining the objectives and executing on these objective to meet the ultimate end-goal.

All the group members were able to find their path in this project while learning from the others. The efficiency of this group and being able to deliver an end-product to the customer was done by conducting this project using the agile methodology. Here is a list of the main deliverables to the customer:

- Integrated many devices into one user-interface: the ABB free@home system, Samung TV, Samsung Refrigerator, Netatmo weather station, Google home smart speaker, Google Nest Hub, and motorized kitchen cabinet.
- The motorized kitchen cabinet was innovated and made easier to use. It can be controlled wirelessly through HomeAssistant and by voice command. In addition, instead of having to press and hold a button to bring the cabinet up or down, a button can simply be pressed once to bring the cabinet up or down.
- A 3-dimensional visualization of the demo room was constructed using HomeByMe and integrated into Prezi. This 3-dimensional demonstration may allow Vamia to give walk-throughs of their room off-campus or virtually.
- The group developed a bill of material of all the devices located in the Vamiaeasy automated living environment

Finally, the expectations of Vamia have been mostly fulfilled, the HomeAssistant application allows one to control the majority of the connected devices without having to use a different platform for each one. The interface has been reworked to make it more intuitive and easy to use. In addition, voice control is fully functional through the Google Home, for example "Ok Google, I'm home." and all the lights turn on and the blinds open. The group has tried to meet the expectations of Vamia as best as possible, besides this they also tried to impress them with voice commands or improved operation of the kitchen cabinet (also integrated with HomeAssistant).

Recommendations for Prospective Group

Making Non-Smart Roomba Vacuum into a Smart Vacuum

The Roomba vacuum was initially thought to be a smart vacuum. However, it was learned that this model (iRobot Roomba 606) has no wireless connectivity which in turn means this device is not a smart device. This non-smart Roomba vacuum can be turned into a smart device using an ESP32 in a similar way the cabinet was turned into a smart device with wireless connectivity.

The Roomba 606 vacuum has 3 buttons which can be seen in Figure 64. The code which was used for the cabinet can be tailored to toggle each of the 3 buttons on the Roomba vacuum. In addition, using the http request would allow an individual to control this vacuum wirelessly through home assistant or by voice command.

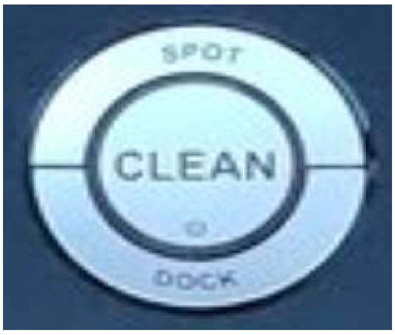


Figure 64: iRobot Roomba 606 Buttons

Another idea for this vacuum would be to implement a virtual device in the ABB free@home system which would allow the programmer to make a push-button on the free@home tablet to control this vacuum as well. Refer to reference [45] on how to complete this task. This would be quite impressive because the programmer could, for example, execute the “clean” mode for the Roomba when the “clean” scene is pressed on the free@home system.

Adding Cabinet Up/Down into the free@home System

Adding the cabinet up/down http requests into the free@home system could be done using the same method for integrating the vacuum into the free@home system. Similarly, you will want to create a virtual device in the free@home system and connect the virtual device to an http request. Once again, the information found in reference [45] will be of great help in completing this task. These cabinet up/down request can be integrated into different scenes once you have successfully connected the virtual device to their respective http requests. For example, in the good morning scene, the cabinet can be brought completely down (because people normally have their coffee or eat breakfast when they wake up).

Utilize Existing Devices for Data Visualization

There are existing ABB motion sensors which are used to turn lights on and off. These sensors may also be used as data visualization tools to see the activity in the Vamiaeasy living environment. The idea here is to make a plot which shows when there was motion and when there isn’t motion. A case can be to check if the patient has gotten up out of bed at an appropriate time or if they went into the living and have not moved for 5 hours. Then, an alarm can trigger for a specific event, for example, if the patient has not moved for 10 hours.

The motion sensors have already been integrated into homeassistant and the data visualization can be done using Grafana (a data visualization tool). Grafana would need to be integrated into the existing homeassistant application. There are many other devices which can be visualized for the healthcare workers using Grafana as well: when the lights were turned on and off in specific rooms; temperature data from the Netatmo device; and refrigerator data (being the temperature of the fridge and how many times the door has been open or closed).

Contacting Maricare Health Sensors

Our group tried getting the Maricare fall-detector healthcare sensors integrated into the homeassistant application. Moreover, complications arose when integrating this device into the homeassistant application. The conclusion is that this device is made specifically for healthcare purposes, so the manufacturer (being Maricare) has strict encryptions which did not allows us to have access to their systems data. What is known is that their detectors utilize zigbee communication protocol.

What the next prospective Vamiaeasy group could potentially do is connect a zigbee dongle to the raspberry pi and see if they can pick up any information from the Maricare hub. If no information is found through the zigbee communication, then the next step would be to contact Maricare. When contacting Maricare, the group can explain they are students working on a project to combine healthcare information into one data visualization environment. Then, ask Maricare if there is a solution to get their fall-detection information so the Vamiaeasy team can implement it into the Grafana data visualization environment.

Fixing the Casting of Homeassistant into the Google Nest Hub

As of now our team has been successful in casting the homeassistant user-interface onto the google nest hub which is located near the bed in the Vamiaeasy living environment. However, there is a small bug associated with this casting; if somebody swipes left or right on the google nest hub, then the homeassistant cast gets removed. In order to recast the homeassisant onto the google nest hub, the user must restart the google nest hub and then recast through the homeassistant dashboard.

This is quite an inconvenience for the instructors and students at Vamia because it may not be intuitive for them to do. Through some research it has been found that this is a common problem for people in the homeassistant community. However, there are programmers who are actively working on fixing this bug. So perhaps when the prospective Vamiaeasy team begins to work on this project, the programmers may have found a solution to mitigate this problem.

Samsung Smart Fridge Live Images Integrated into Homeassistant

The smartthings application was integrated into the homeassistant application, but our team was unable to get the live images from the camera casted into homeassistant. What the team was able to get from the smarterthings integretion was control of the TV, temperature information from the fridge, and whether the fridge door has been open or closed. Whether this is possible or not was not investigated. The prospective group can potentially dig deeper to see if casting the live images to homeassistant is possible or not.

Z-Wave and Zigbee Integrations

The group has not connected Z-wave and Zigbee dongles to the raspberry pi to see what other devices may potentially be of use for the Vamiaeasy automated home. The prospective group could potentially see what other devices may be able to be connected through these communication protocols or if there are other devices which may be integrated through these communications. This would be on the lower end of the priority list, but it is worth giving it a try.

References

[1] European project semester, “About EPS,” [Online]. Available: http://www.europeanprojectsemester.eu/concept.

[2] United Nations, “World population ageing,” Department of economic and social affairs, New York, 2017.

[3] F. Swain, “The technologies that could transform ageing,” 5 november 2020. [Online]. Available: https://www.bbc.com/future/article/20201104-the-technologies-that-could-transform-ageing.

[4] Belbin, “Why Use Belbin Team Roles?,” [Online]. Available: https://www.belbin.com/about/why-use-belbin.

[5] Wikipedia foundation, “Finnish Sauna,” 21 augustus 2021. [Online]. Available: https://en.wikipedia.org/wiki/Finnish_sauna.

[6] M. Ehrs and R. Nylund, “EPS coursebook,” Novia, Vaasa, 2021.

[7] “Responsibility Assignment Matrix,” Wikipedia, Wikimedia Foundation, 11 October 2021. [Online]. Available: https://en.wikipedia.org/wiki/Responsibility_assignment_matrix.

[8] Nuclino, “design-brief,” [Online]. Available: https://www.nuclino.com/articles/design-brief.

[9] Matt Burgess, “What is Internet of things? Wired explains,” 16 februari 2018. [Online]. Available: https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot.

[10] Oracle, “What is IoT?,” [Online]. Available: https://www.oracle.com/internet-of-things/what-is-iot/.

[11] IBM, “Choosing the best hardware for your next IoT project,” [Online]. Available: https://developer.ibm.com/articles/iot-lp101-best-hardware-devices-iot-project/ .

[12] B. Posey and T. Lavery, “IoT gateway,” April 2021. [Online]. Available: https://internetofthingsagenda.techtarget.com/definition/IoT-gateway.

[13] S. Cope, “Smart home hubs guide,” 26 december 2020. [Online]. Available: https://stevessmarthomeguide.com/smart-home-hubs-guide/.

[14] A. Brice, “Your smart home platform: Home assistant vs Open Hab,” Januari 2020. [Online]. Available: https://smarthome.university/your-smart-home-platform-home-assistant-vs-openhab/.

[15] “OpenHAB Wikipedia,” Wikimedia Foundation, 11 September 2021. [Online]. Available: https://en.wikipedia.org/wiki/OpenHAB.

[16] Wikimedia Foundation, “Home Assistant,” 18 October 2021. [Online]. Available: https://en.wikipedia.org/wiki/Home_Assistant .

[17] “Home Assistant,” GitHub, [Online]. Available: https://github.com/home-assistant.

[18] “Home Assistant Community,” Home Assistant Community, [Online]. Available: https://community.home-assistant.io/ .

[19] “openHAB,” openHABian, [Online]. Available: https://community.openhab.org/.

[20] “openHAB gitHUB,” github, [Online]. Available: https://github.com/openhab.

[21] ABB, “about abb,” [Online]. Available: https://global.abb/group/en/about.

[22] ABB, “Our strategy,” [Online]. Available: https://global.abb/group/en/about/our-strategy.

[23] ABB, “Why ABB?,” [Online]. Available: https://careers.abb/global/en/why-abb.

[24] ABB, “Our purpose,” ABB, [Online]. Available: https://global.abb/group/en/about/our-purpose. [Accessed 9 november 2021].

[25] A. Jenkins, “NETSUIT,” ORACLE, 24 February 2021. [Online]. Available: https://www.netsuite.com/portal/resource/articles/erp/bill-of-materials-bom.shtml.

[26] “HomeByMe,” [Online]. Available: https://home.by.me/en/.

[27] “Prezi Presentations,” [Online]. Available: https://prezi.com/.

[28] onsemi, “BC547B - NPN Bipolar Transistor,” August 2021. [Online]. Available: https://www.onsemi.com/pdf/datasheet/bc546-d.pdf.

[29] RAPID, “5V DPDT SUBMINIATURE RELAY,” [Online]. Available: https://static.rapidonline.com/pdf/60-4754e.pdf.

[30] V. Semiconductors, “1N4148 - Small Signal Fast Switching Diodes,” 06 July 2017. [Online]. Available: https://www.vishay.com/docs/81857/1n4148.pdf.

[31] mathertel, “github - OneButton,” 27 October 2021. [Online]. Available: https://github.com/mathertel/OneButton.

[32] kunshmaurya, “Arduino Project Hub - WiFi ESP32 LED Control Project,” 1 May 2020. [Online]. Available: https://create.arduino.cc/projecthub/kunshmaurya/wifi-esp32-led-control-project-42f36c.

[33] “Google Nest (Smart Speakers),” Wikipedia, Wikimedia Foundation, 2 November 2021. [Online]. Available: https://en.wikipedia.org/wiki/Google_Nest_(smart_speakers). .

[34] J. e. a. Johnson, “Get the Most out of These Smart Devices and Services with Google Assistant,” Android Central, 7 October 2021. [Online]. Available: https://www.androidcentral.com/best-google-home-compatible-devices.

[35] “Google Assistant Installation Instruction for ABB-Free@Home,” ABB, 9 July 2020. [Online]. Available: https://library.e.abb.com/public/c9e5067412c948e3ab39c5d92bcf836c/free-at-home_Installation-Google_MBA_EN_V1_2CKA0014B5016_ABB.pdf?x-sign=B5b+BpAFakbBOcLuqFty4kR1zxHUbD54m4kyltGE1f8b+IsEzMm8mBGymU98bpsE.

[36] “Home Assistant,” [Online]. Available: https://www.home-assistant.io/installation/raspberrypi. [Accessed 16 september 2021].

[37] L. T. Channel, “Home Assistant WiFi Setup - YouTube,” [Online]. Available: https://www.youtube.com/watch?v=j35ytqa_Pfc. [Accessed 16 September 2021].

[38] “Home Assistant,” [Online]. Available: https://www.home-assistant.io/integrations/smartthings/. [Accessed 12 november 2021].

[39] “Home Assistant,” [Online]. Available: https://www.home-assistant.io/integrations/netatmo/. [Accessed 2 December 2021].

[40] “Home Assistant Community,” 13 september 2021. [Online]. Available: https://community.home-assistant.io/t/home-assistant-community-add-on-ssh-web-terminal/338200. [Accessed 5 november 2021].

[41] jheling, “GitHub,” [Online]. Available: https://github.com/jheling/freeathome. [Accessed 5 november 2021].

[42] Ati, “free@home integrated into IoT Hub (OpenHAB2 and OpenHAB3, Grafana, Node-RED and influxDB database) for IoT use cases,” Busch-Jaeger Community, 05 November 2020. [Online].

[43] “ABB free@home how-tos,” Busch-Jaeger Community, [Online]. Available: https://community.busch-jaeger.de/neueste-howtos.

[44] Fxcoolby. OpenHAB Community, “Bridging OpenHab and Home Assistant,” 18 januari 2020. [Online]. Available: https://community.openhab.org/t/bridging-openhab-and-home-assistant/90742 .

[45] ABB, “free@home Third Party Integrations,” December 2020. [Online]. Available: https://library.e.abb.com/public/b1bfd91c3c8446c5bcc14d0a78f8d6f0/Webinar-free@home-REST-API-December2020_PR_EN_V1-0_9AKK107991A6482.pdf.

[46] “Samsung Smartthings - Bindings,” 2 november 2021. [Online]. Available: https://www.openhab.org/addons/bindings/smartthings/.

[47] “What is an IP Address – Definition and Explanation,” Kaspersky, [Online]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address.

Table of Appendices

Appendix 1: Gantt Chart 94

Appendix 2: Risk assessment – List and Matrix..94

Appendix 3: Planned Value 94

Appendix 4: Actual Cost..94

Appendix 5: Earned Value..94

Appendix 6: Work Breakdown Structure..94

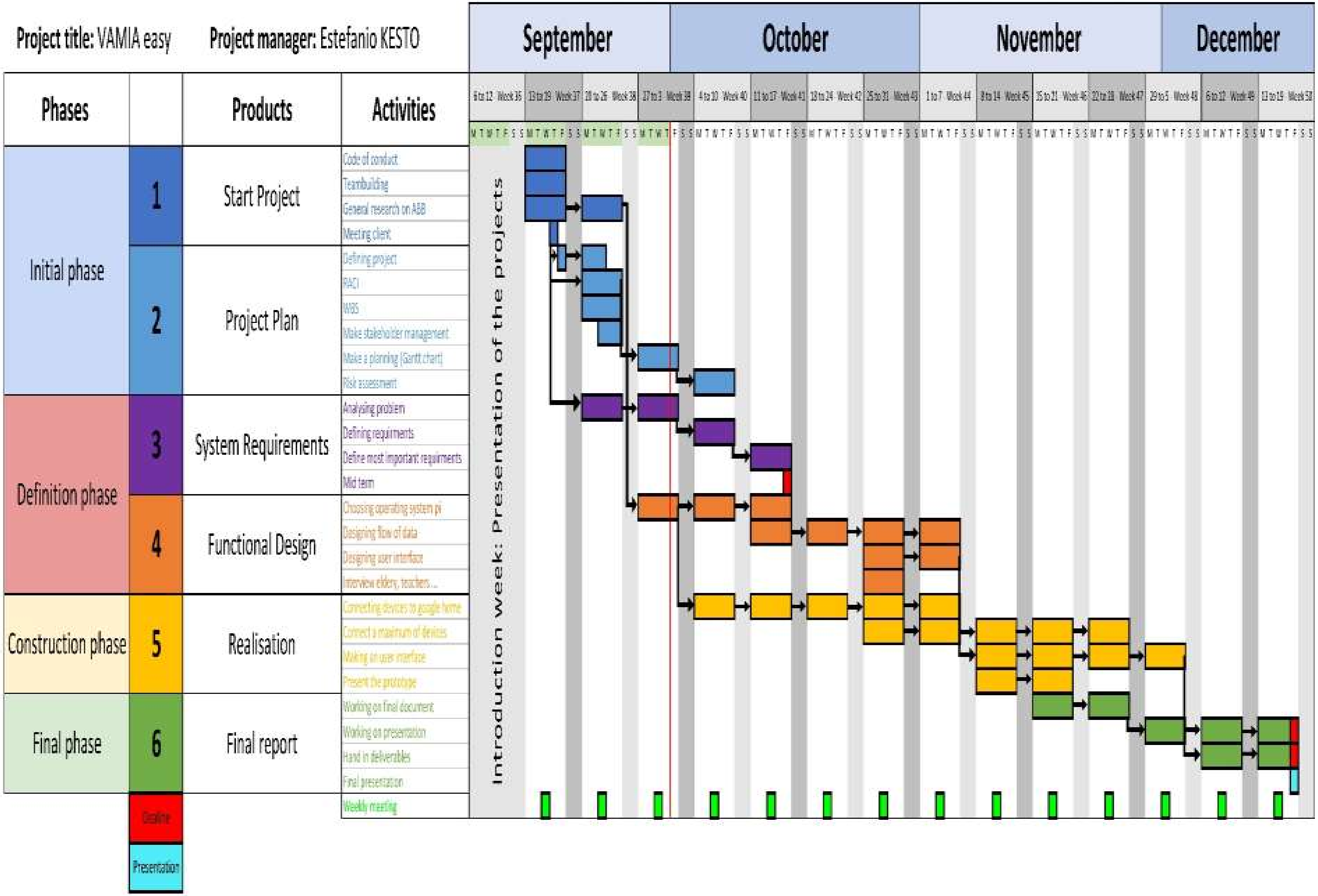
Appendix 7: RACI Tables 94

Appendix 8: Hand drawn circuit schematic for automation of motorized 94

Appendix 9: Failure diagnostics for motorized cabinet..94

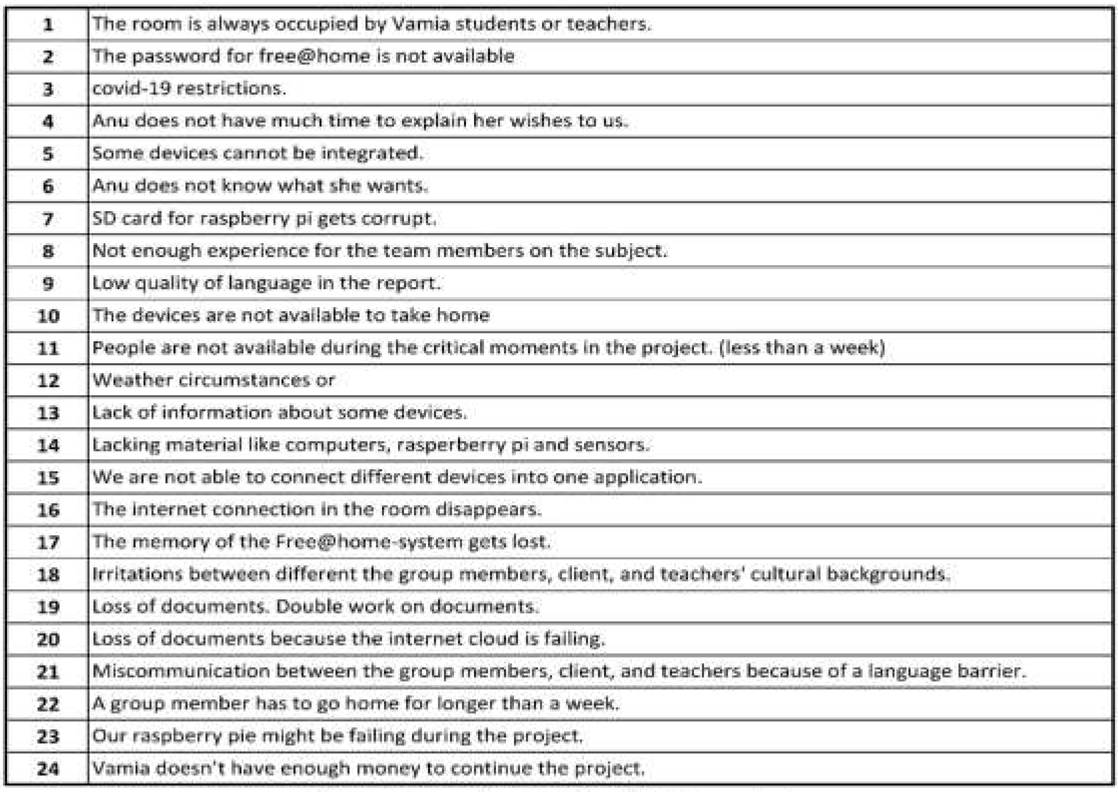
Appendix 10: The groups code of conduct 94

Appendix 1: Gantt Chart



Appendix 2: Risk assessment – List and Matrix

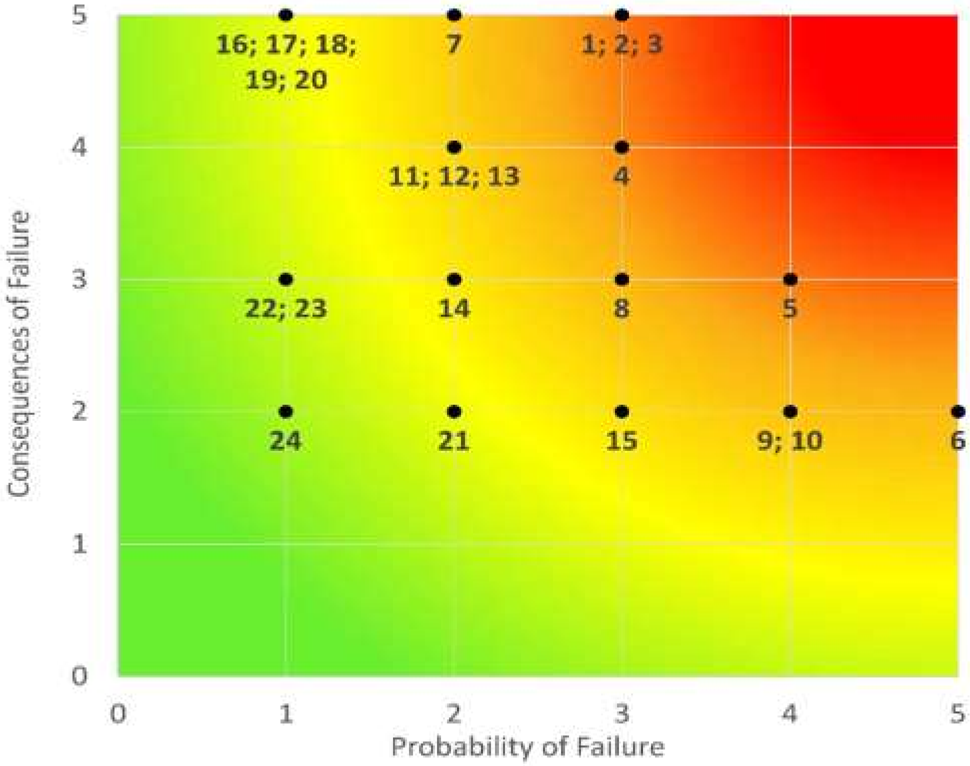
List of project risks



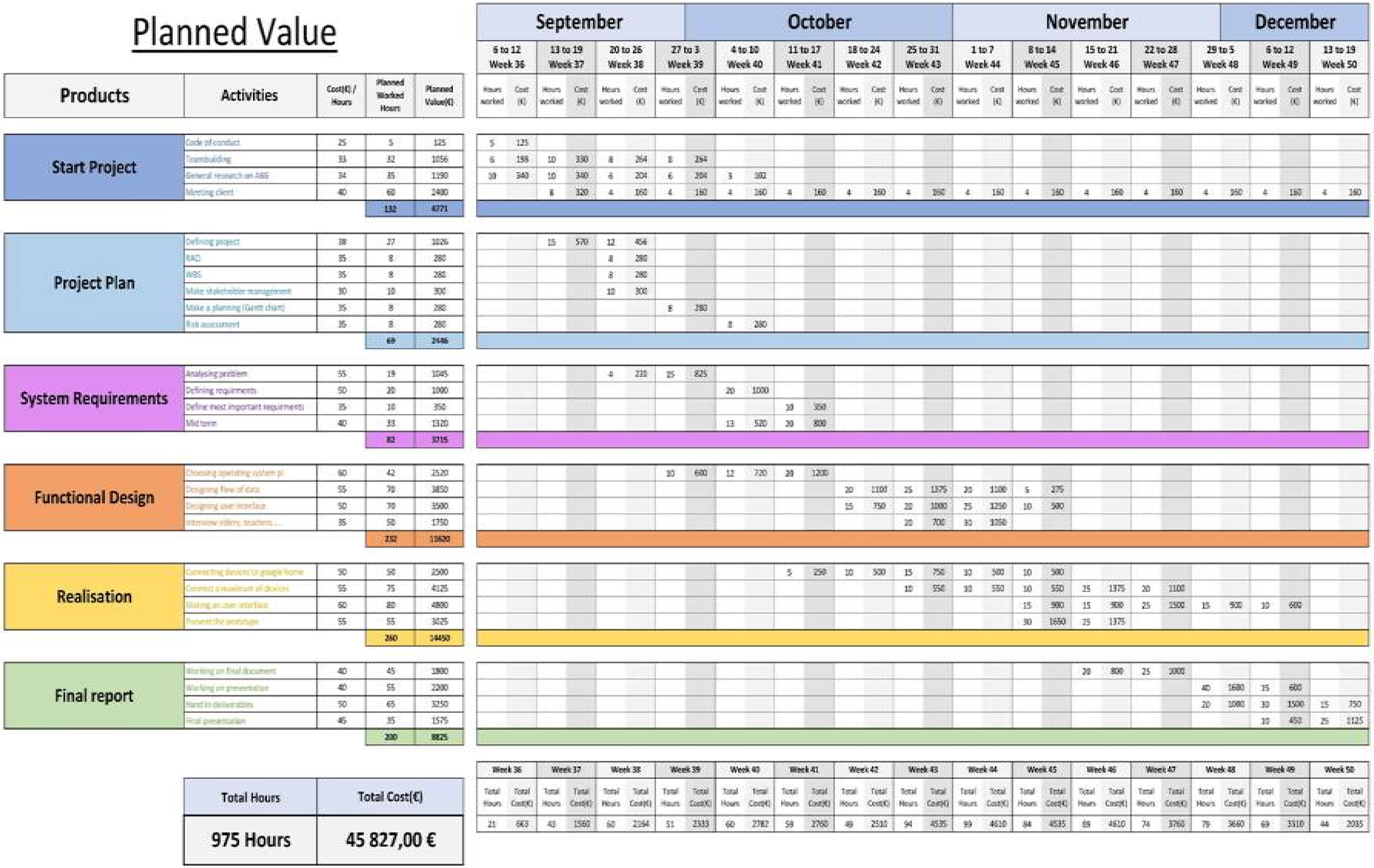
1	The room is always occupied by Vamia students or teachers.
2	The password for free@home is not available
3	covid-19 restrictions.
4	Anu does not have much time to explain her wishes to us.
5	Some devices cannot be integrated.
6	Anu does not know what she wants.
7	SD card for raspberry pi gets corrupt.
8	Not enough experience for the team members on the subject.
9	Low quality of language in the report.
10	The devices are not available to take home
11	People are not available during the critical moments in the project. (less than a week)
12	Weather circumstances or
13	Lack of information about some devices.
14	Lacking material like computers, rasperberry pi and sensors.
15	We are not able to connect different devices into one application.
16	The internet connection in the room disappears.
17	The memory of the Free@home-system gets lost.
18	Irritations between different the group members, client, and teachers' cultural backgrounds.
19	Loss of documents. Double work on documents.
20	Loss of documents because the internet cloud is failing.
21	Miscommunication between the group members, client, and teachers because of a language barrier.
22	A group member has to go home for longer than a week.
23	Our raspberry pie might be failing during the project.
24	Vamia doesn't have enough money to continue the project.

All the risks are categorized from hight to low priority.

Risk matrix



Appendix 3: Planned Value



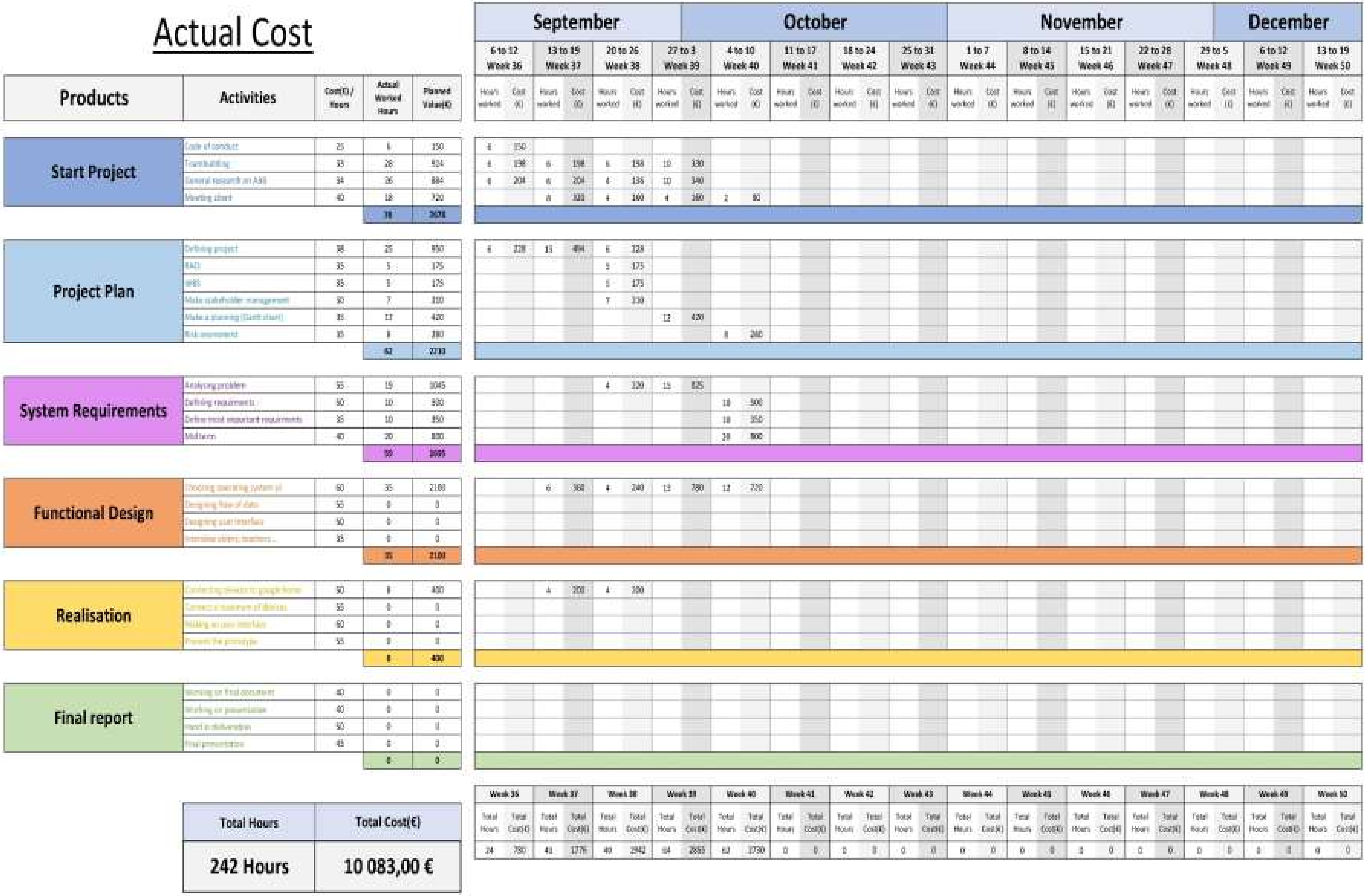
Planned Value

Products	Activities	Cost(€) / Hours	Planned Worked Hours	Planned Value(€)	September								October								November								December					
					6 to 12 Week 36		13 to 19 Week 37		20 to 26 Week 38		27 to 3 Week 39		4 to 10 Week 40		11 to 17 Week 41		18 to 24 Week 42		25 to 31 Week 43		1 to 7 Week 44		8 to 14 Week 45		15 to 21 Week 46		22 to 28 Week 47		29 to 5 Week 48		6 to 12 Week 49		13 to 19 Week 50	
					Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)
Start Project	Code of conduct	25	5	125	5	125																												
	Teambuilding	33	32	1056	6	198	10	330	8	264	8	264																						
	General research on ABS	34	35	1190	10	340	10	340	6	204	6	204	3	102																				
	Meeting client	40	60	2400			8	320	4	160	4	160	4	160	4	160	4	160	4	160	4	160	4	160	4	160	4	160	4	160	4	160	4	160
			132	4771																														
Project Plan	Defining project	38	27	1026			15	570	12	456																								
	RACI	35	8	280					8	280																								
	WBS	35	8	280					8	280																								
	Make stakeholder management	30	10	300					10	300																								
	Make a planning (Gantt chart)	35	8	280							8	280																						
	Risk assessment	35	8	280									8	280																				
			69	2446																														
System Requirements	Analysing problem	55	19	1045					4	220	15	825																						
	Defining requirements	50	20	1000									20	1000																				
	Define most important requirements	35	10	350											10	350																		
	Mid term	40	33	1320									13	520	20	800																		
			82	3715																														
Functional Design	Choosing operating system pi	60	42	2520							10	600	12	720	20	1200																		
	Designing flow of data	55	70	3850													20	1100	25	1375	20	1100	5	275										
	Designing user interface	50	70	3500													15	750	20	1000	25	1250	10	500										
	Interview eldery, teachers ...	35	50	1750															20	700	30	1050												
			232	11620																														
Realisation	Connecting devices to google home	50	50	2500											5	250	10	500	15	750	10	500	10	500										
	Connect a maximum of devices	55	75	4125															10	550	10	550	10	550	15	1375	20	1100						
	Making an user interface	60	80	4800																			15	900	15	900	25	1500	15	900	10	600		
	Present the prototype	55	55	3025																			30	1650	25	1375								
			260	14450																														
Final report	Working on final document	40	45	1800																					20	800	25	1000						
	Working on presentation	40	55	2200																									40	1600	15	600		
	Hand in deliverables	50	65	3250																									20	1000	30	1500	15	750
	Final presentation	45	35	1575																											10	450	25	1125
			200	8825																														

Week 36		Week 37		Week 38		Week 39		Week 40		Week 41		Week 42		Week 43		Week 44		Week 45		Week 46		Week 47		Week 48		Week 49		Week 50	
Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)
21	663	43	1560	60	2164	51	2333	60	2782	59	2760	49	2510	94	4535	99	4610	84	4535	89	4610	74	3760	79	3660	69	3310	44	2035

Total Hours	Total Cost(€)
975 Hours	45 827,00 €

Appendix 4: Actual Cost



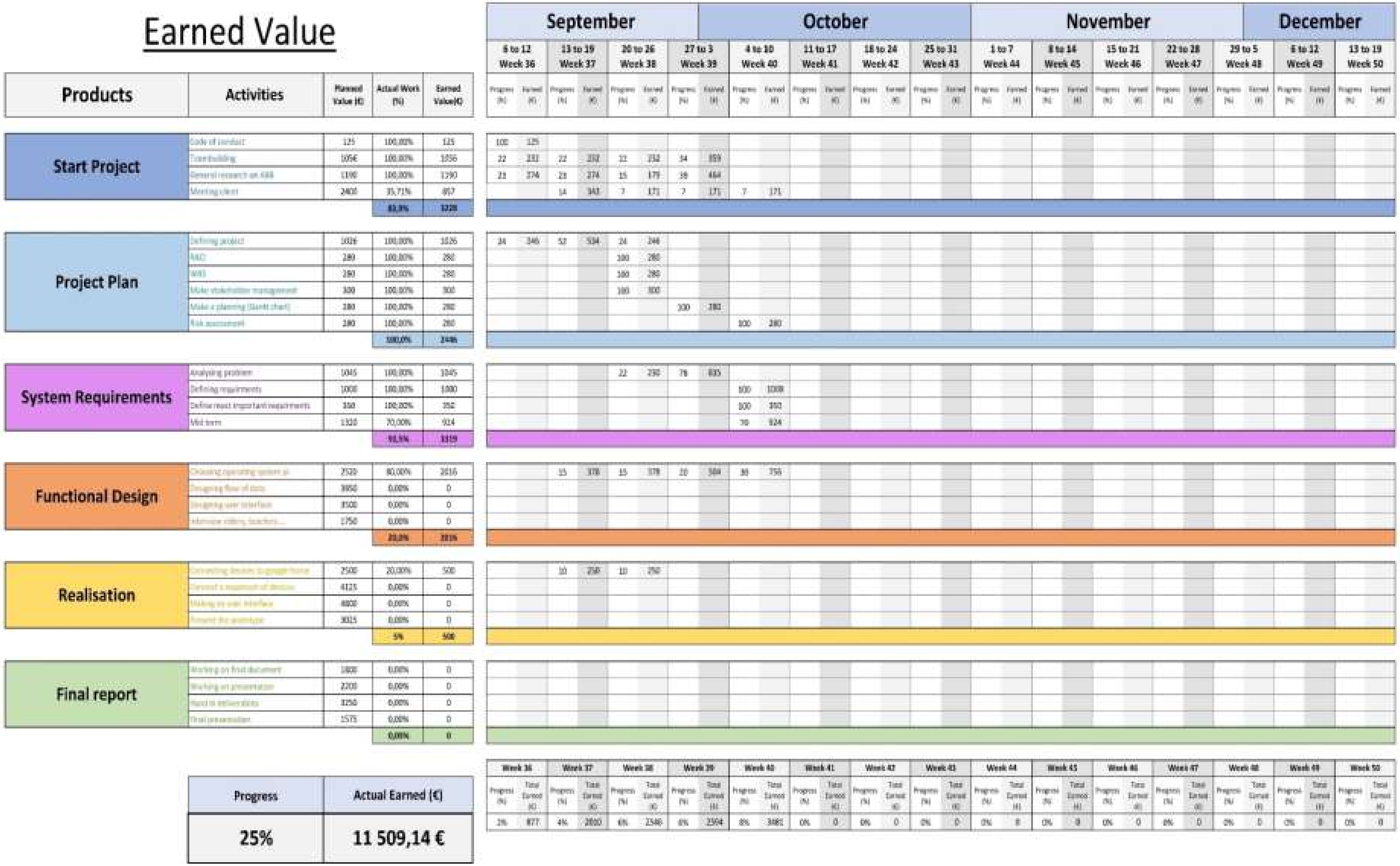
Actual Cost

					September								October								November								December					
					6 to 12 Week 36		13 to 19 Week 37		20 to 26 Week 38		27 to 3 Week 39		4 to 10 Week 40		11 to 17 Week 41		18 to 24 Week 42		25 to 31 Week 43		1 to 7 Week 44		8 to 14 Week 45		15 to 21 Week 46		22 to 28 Week 47		29 to 5 Week 48		6 to 12 Week 49		13 to 19 Week 50	
Products	Activities	Cost(€)/ Hours	Actual Worked Hours	Planned Value(€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)	Hours worked	Cost (€)
Start Project	Code of conduct	25	6	150	6	150																												
	Teambuilding	33	28	924	6	198	6	198	6	198	10	330																						
	General research on ASI	34	26	884	6	204	6	204	4	136	10	340																						
	Meeting chart	40	18	720			8	320	4	160	4	160	2	80																				
			78	2678																														
Project Plan	Defining project	38	25	950	6	228	13	494	6	228																								
	RAD	35	5	175					5	175																								
	WBS	35	5	175					5	175																								
	Make stakeholder management	30	7	210					7	210																								
	Make a planning (Gantt chart)	35	12	420							12	420																						
	Risk assessment	35	8	280									8	280																				
			62	2210																														
System Requirements	Analysing problem	55	19	1045					4	220	15	825																						
	Defining requirements	50	10	500									10	500																				
	Define most important requirements	35	10	350									10	350																				
	Mid term	40	20	800									20	800																				
			59	2695																														
Functional Design	Choosing operating system of	60	35	2100			6	360	4	240	13	780	12	720																				
	Designing flow of data	55	0	0																														
	Designing user interface	50	0	0																														
	Interview elderly, teachers ...	35	0	0																														
			35	2100																														
Realisation	Connecting devices to google home	50	8	400			4	200	4	200																								
	Connect a maximum of devices	55	0	0																														
	Making an user interface	60	0	0																														
	Present the prototype	55	0	0																														
			8	400																														
Final report	Working on final document	40	0	0																														
	Working on presentation	40	0	0																														
	Hand in deliverables	50	0	0																														
	Final presentation	45	0	0																														
			0	0																														

Week 36		Week 37		Week 38		Week 39		Week 40		Week 41		Week 42		Week 43		Week 44		Week 45		Week 46		Week 47		Week 48		Week 49		Week 50	
Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)	Total Hours	Total Cost(€)
24	780	43	1776	49	1942	64	2855	62	2730	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Total Hours	Total Cost(€)
242 Hours	10 083,00 €

Appendix 5: Earned Value



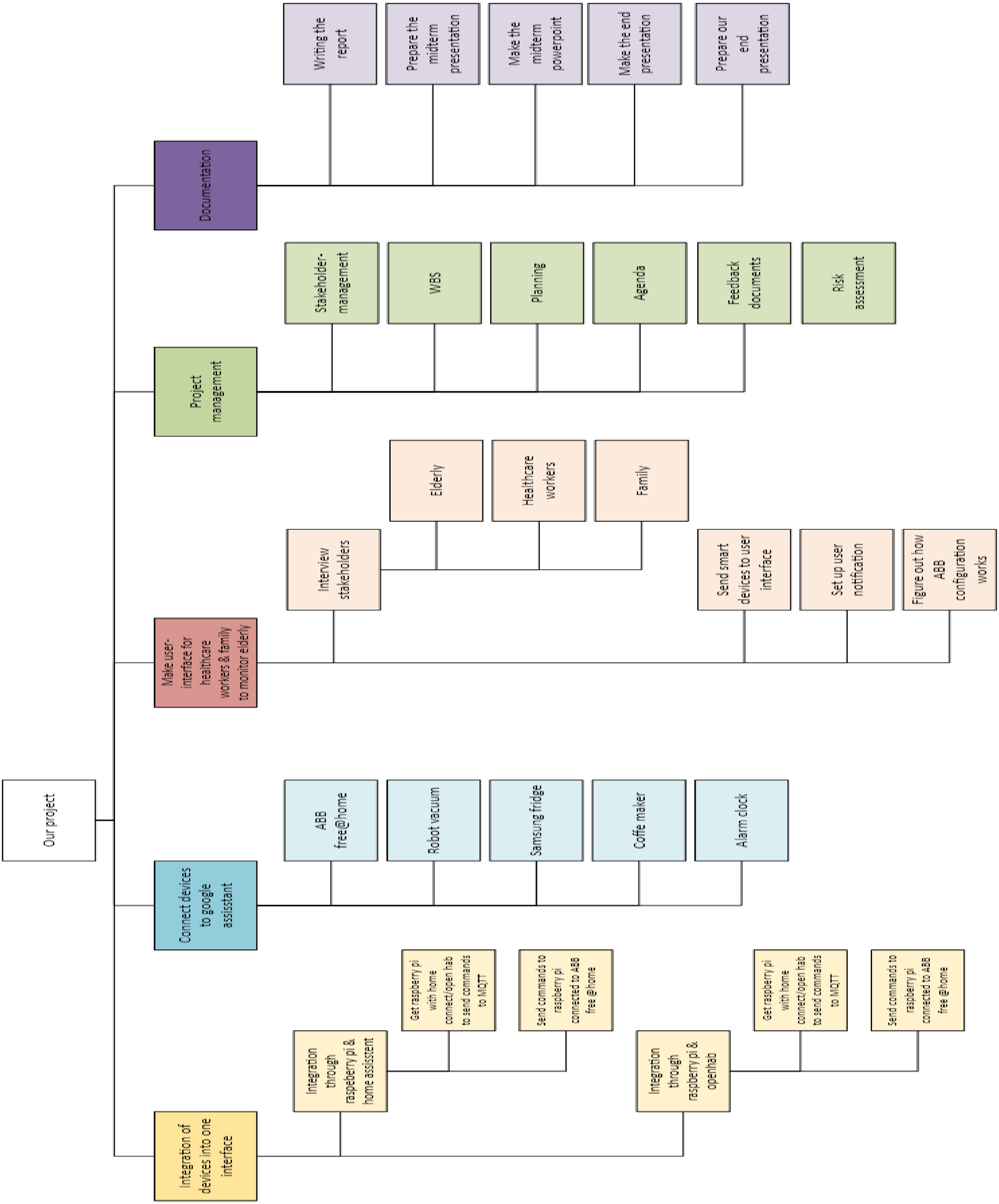
Earned Value

Products	Activities	Planned Value (€)	Actual Work (%)	Earned Value(€)	September							October									November								December				
					6 to 12 Week 36		13 to 19 Week 37		20 to 26 Week 38		27 to 3 Week 39		4 to 10 Week 40		11 to 17 Week 41		18 to 24 Week 42		25 to 31 Week 43		1 to 7 Week 44		8 to 14 Week 45		15 to 21 Week 46		22 to 28 Week 47		29 to 5 Week 48		6 to 12 Week 49		13 to 19 Week 50	
					Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)	Progress (%)	Earned (€)
Start Project	Code of conduct	125	100,00%	125	100	125																												
	Teambuilding	1056	100,00%	1056	22	232	22	232	22	232	34	359																						
	General research on [illegible]	1190	100,00%	1190	23	274	23	274	15	179	39	464																						
	Meeting client	2400	35,71%	857			14	343	7	171	7	171	7	171																				
			83,9%	3228																														
Project Plan	Defining project	1026	100,00%	1026	24	246	52	534	24	246																								
	RACI	280	100,00%	280					100	280																								
	WBS	280	100,00%	280					100	280																								
	Make stakeholder management	300	100,00%	300					100	300																								
	Make a planning (Gantt chart)	280	100,00%	280							100	280																						
	Risk assessment	280	100,00%	280									100	280																				
			100,0%	2446																														
System Requirements	Analysing problem	1045	100,00%	1045					22	230	78	815																						
	Defining requirements	1000	100,00%	1000									100	1000																				
	Define most important requirements	350	100,00%	350									100	350																				
	Mid term	1320	70,00%	924									70	924																				
			92,5%	3319																														
Functional Design	Choosing operating system [illegible]	2520	80,00%	2016			15	378	15	378	20	504	30	756																				
	Designing flow of data	3950	0,00%	0																														
	Designing user interface	3500	0,00%	0																														
	Interview elderly, teachers...	1750	0,00%	0																														
			20,0%	2016																														
Realisation	Connecting devices to [illegible]	2500	20,00%	500			10	250	10	250																								
	Control a maximum of devices	4125	0,00%	0																														
	Making an user interface	4800	0,00%	0																														
	Present the prototype	3025	0,00%	0																														
			5%	500																														
Final report	Working on final document	1800	0,00%	0																														
	Working on presentation	2200	0,00%	0																														
	Hand in deliverables	3250	0,00%	0																														
	Final presentation	1575	0,00%	0																														
			0,00%	0																														

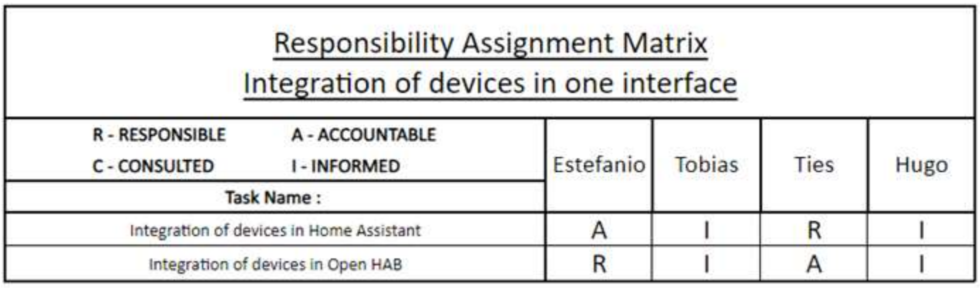
Progress	Actual Earned (€)
25%	11 509,14 €

Week 36		Week 37		Week 38		Week 39		Week 40		Week 41		Week 42		Week 43		Week 44		Week 45		Week 46		Week 47		Week 48		Week 49		Week 50	
Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)	Progress (%)	Total Earned (€)
2%	877	4%	2010	6%	2546	6%	2594	8%	3481	0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%	0	0%	0

Appendix 6: Work Breakdown Structure

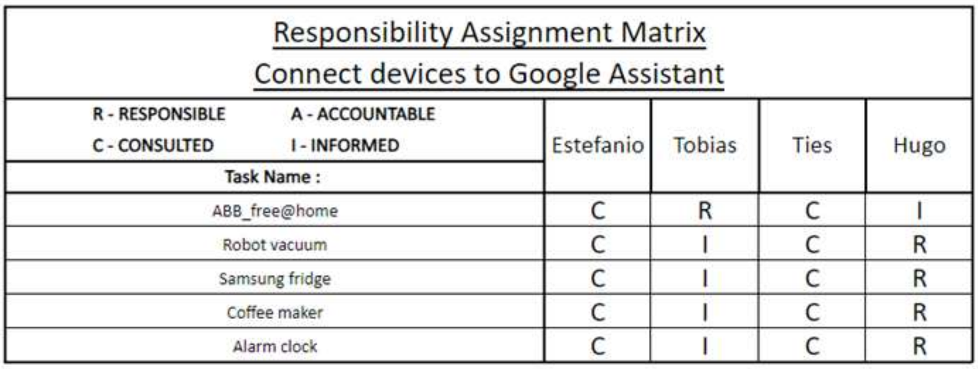

Our project
Documentation
Writing the report
Prepare the midterm presentation
Make the midterm powerpoint
Make the end presentation
Prepare our end presentation
Project management
Stakeholder-management
WBS
Planning
Agenda
Feedback documents
Risk assessment
Make user-interface for healthcare workers & family to monitor elderly
Interview stakeholders
Elderly
Healthcare workers
Family
Send smart devices to user interface
Set up user notification
Figure out how ABB configuration works
Connect devices to google assistant
ABB free@home
Robot vacuum
Samsung fridge
Coffe maker
Alarm clock
Integration of devices into one interface
Integration through raspberry pi & home assisstent
Get raspberry pi with home connect/open hab to send commands to MQTT
Send commands to raspberry pi connected to ABB free@home
Integration through raspberry pi & openhab
Get raspberry pi with home connect/open hab to send commands to MQTT
Send commands to raspberry pi connected to ABB free@home

Appendix 7: RACI Tables



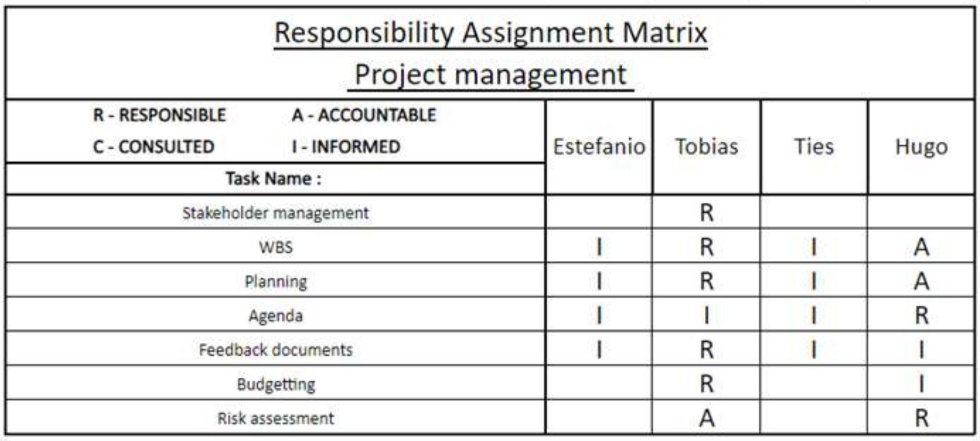
Responsibility Assignment Matrix
Integration of devices in one interface

R - RESPONSIBLE A - ACCOUNTABLE C - CONSULTED I - INFORMED	Estefanio	Tobias	Ties	Hugo
Task Name :				
Integration of devices in Home Assistant	A	I	R	I
Integration of devices in Open HAB	R	I	A	I



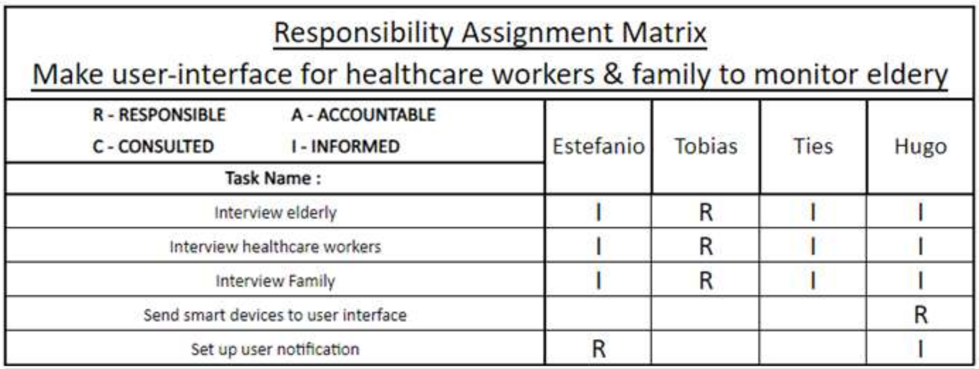
Responsibility Assignment Matrix
Connect devices to Google Assistant

R - RESPONSIBLE A - ACCOUNTABLE C - CONSULTED I - INFORMED	Estefanio	Tobias	Ties	Hugo
Task Name :				
ABB_free@home	C	R	C	I
Robot vacuum	C	I	C	R
Samsung fridge	C	I	C	R
Coffee maker	C	I	C	R
Alarm clock	C	I	C	R



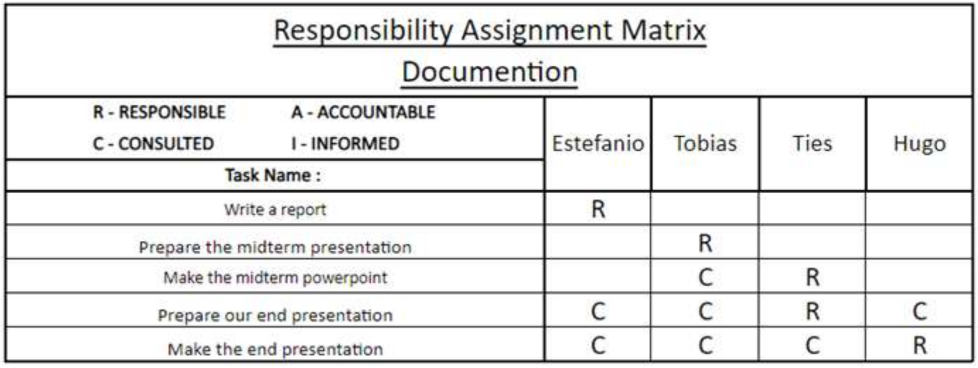
Responsibility Assignment Matrix
Project management

R - RESPONSIBLE A - ACCOUNTABLE C - CONSULTED I - INFORMED	Estefanio	Tobias	Ties	Hugo
Task Name :				
Stakeholder management		R		
WBS	I	R	I	A
Planning	I	R	I	A
Agenda	I	I	I	R
Feedback documents	I	R	I	I
Budgetting		R		I
Risk assessment		A		R



Responsibility Assignment Matrix
Make user-interface for healthcare workers & family to monitor eldery

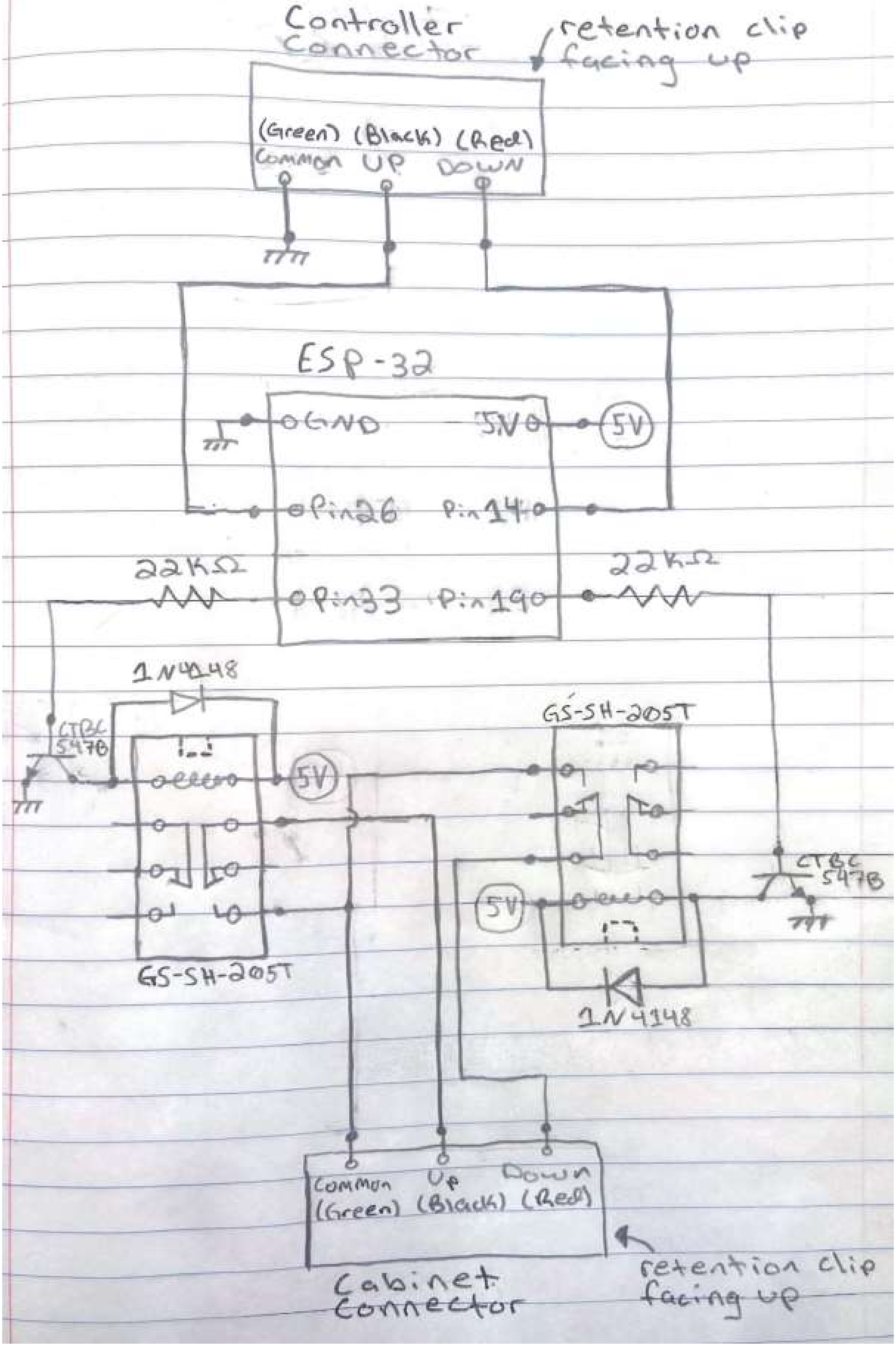
R - RESPONSIBLE A - ACCOUNTABLE C - CONSULTED I - INFORMED	Estefanio	Tobias	Ties	Hugo
Task Name :				
Interview elderly	I	R	I	I
Interview healthcare workers	I	R	I	I
Interview Family	I	R	I	I
Send smart devices to user interface				R
Set up user notification	R			I



Responsibility Assignment Matrix
Documention

R - RESPONSIBLE A - ACCOUNTABLE C - CONSULTED I - INFORMED	Estefanio	Tobias	Ties	Hugo
Task Name :				
Write a report	R			
Prepare the midterm presentation		R		
Make the midterm powerpoint		C	R	
Prepare our end presentation	C	C	R	C
Make the end presentation	C	C	C	R

Appendix 8: Hand drawn circuit schematic for automation of motorized



Appendix 9: Failure diagnostics for motorized cabinet

A. Check controller connector

1. Carefully remove the cabinet controller from the counter (figure 65a)
2. Unscrew the backplate to expose the connector for cabinet controller (figure 65b).
3. Check that the red, black, and white wires are making a good connections with the cabinet connector (figure 65c).
    - With retention clip in the orientation shown in figure 1c: Red wire on right; black wire in middle; and white wire on left.
4. Test up and down buttons

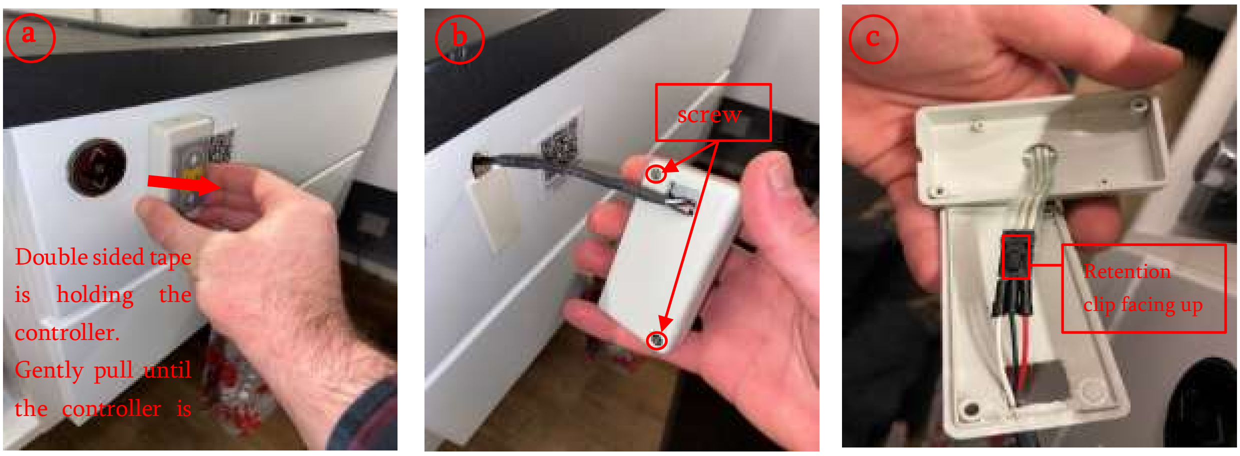

a
Double sided tape is holding the controller. Gently pull until the controller is
b
screw
c
Retention clip facing up

Figure 65: Removing cabinet controller and backplate to cabinet controller to expose the

B. Check cabinet connector

1. Remove the drawer from the drawer slide/hinges (figure 66a)
2. Locate the cabinet connector and gain access to the connection (figure 66b)
3. Ensure good connection is made between terminals from blackbox to the cabinet connector (figure 66c)
    - Red wire to red wire; black wire to black wire; white wire to green wire.
4. Test up and down button

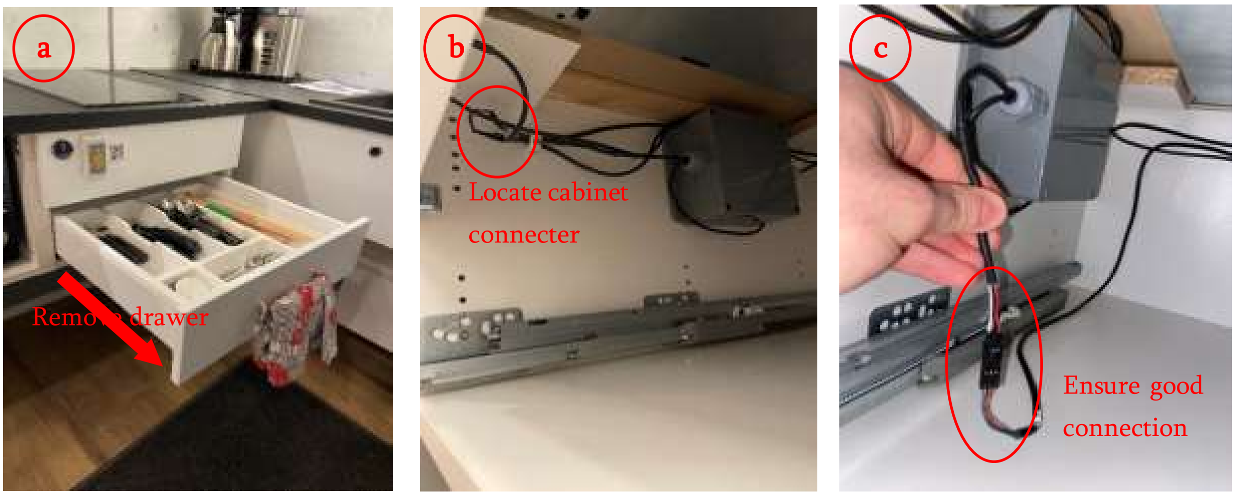


Figure 66: Gaining access to the cabinet connector

C. **Plug controller connector back into the cabinet connector**

1. You may simply plug the controller connector back into the cabinet connector. This will result in the cabinet functioning as it was originally designed.
   - Note: if the cabinet is still not working, then the problem is with the motorized cabinet itself and not the blackbox circuitry. If this is the case, move forward with removing the blackbox and contact the manufacturer of the cabinet.

D. **Check blackbox circuitry**

1. Move the wires which are on top of the blackbox (figure 67a)
2. Life the blackbox up and pull out (figure 67b)
3. Unscrew both caps for the ports on blackbox (figure 67c)
4. Pull out the stand-alone cord which is inserted in the beige port (figure 67c)
   - In other words, one port has two wires sticking out of it and another port has a single wire sticking out of it. You want to pull out the wire in the port which has a single wire sticking out of it.
   - Note: The cord within the beige port is a micro-USB
5. Unscrew and gently remove the top plate from the blackbox (figure 67c)

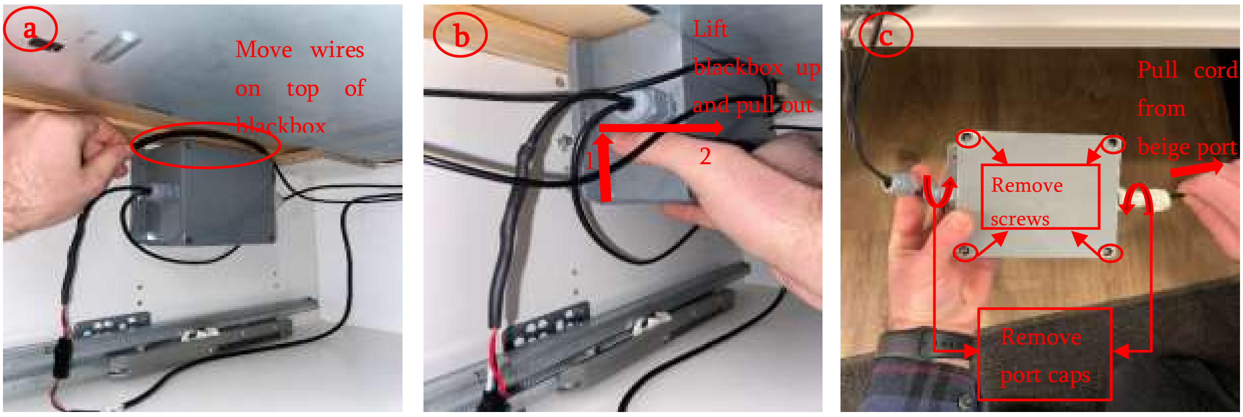

a
Move wires on top of blackbox
b
Lift blackbox up and pull out
1
2
c
Pull cord from beige port
Remove screws
Remove port caps

Figure 67: Gaining access to blackbox circuitry

   - At this point you should have access to the electrical breadboard (figure 4a)

6. Remove the electrical beadboard from the blackbox (figure 68b)
7. Use a digital multimeter to run diagnostics on the esp32 and electrical connections

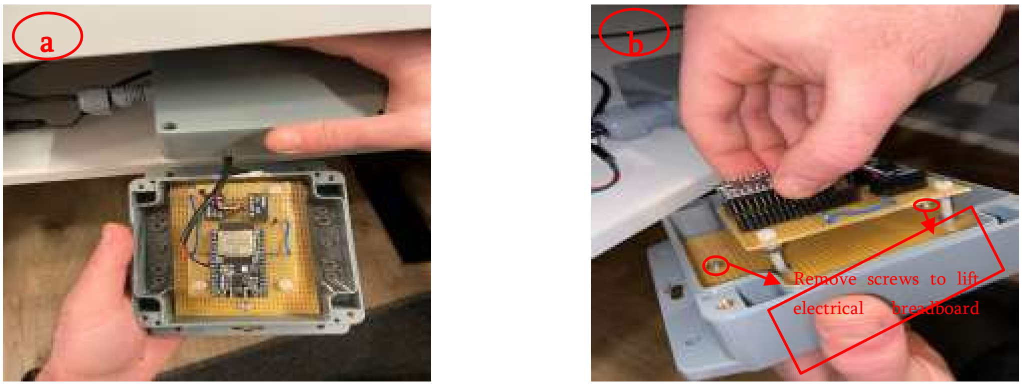


Figure 68: Removing electrical breadboard from blackbox

E. Replace ESP32

1. Obtain an ESP32 devkit
2. Download the Arduino programming environment
3. In the Arduino IDE, locate the “tools” tab and select the “ESP32 Dev Module” as your board
4. Plug in the ESP32 via micro-USB
5. Flash the new board
    - Note: The source code used in the automation of this cabinet was sent to Peter Ahlroos and Anu Ylikahri

F. Rebuild electrical breadboard

1. Obtain all the components shown in the list below
    - ESP32 Devkit
    - 22kΩ Resistors: 2
    - 1N4148 Diodes: 2
    - BC547 NPN Transistors: 2
    - GS-SH-205T 5V Relay: 2
2. Build the circuit shown in figure 5

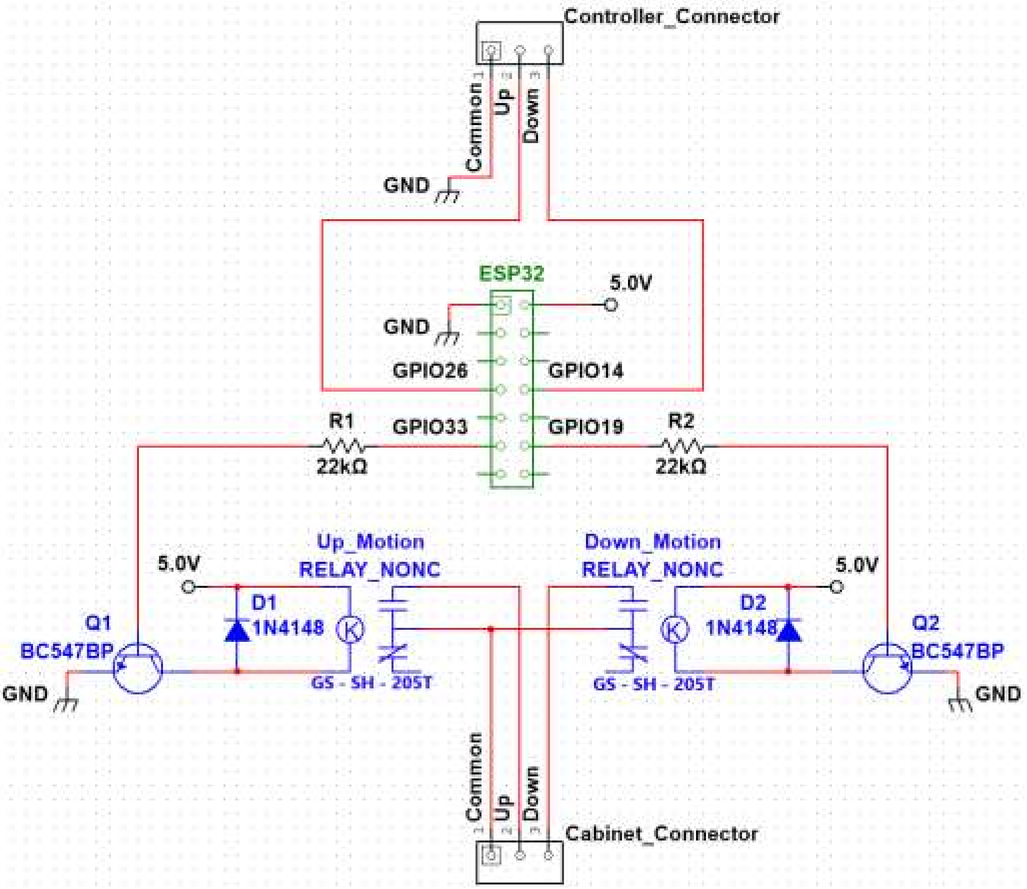


Figure 69: ESP32-triggered instant-on relay circuit diagram for automation of motorized

Please feel free to contact Estefanio Kesto (Estefanio.Kesto123@gmail.com) if any assistance or advice is needed.

Appendix 10: The groups code of conduct

Introduction

This code of conduct facilitates the SALtech group way of engaging in collaborative works. It is created not because bad behaviour is anticipated; rather, because we believe each group member should uphold our values, beliefs, and obligations. Each group member shall adhere to the code discussed hereafter to an exceptional level, which will ensure a clear path is shaved to our cultural beliefs.

Communication

The language of communication will be in English. The main form of communication will be in-person and when off-campus, WhatsApp or Microsoft teams will be used. Documentation and files will be stored in the Vamiaeasy Microsoft teams group.

Core Values

- Communication: Being able to talk among ourselves and discuss problems freely
- Consistency: Being able to accomplish work in a consistent manner, on time and well done
- Collaboration: Being able to work effectively as a team and communicate well
- Balance: distributing the teamwork load
- Consideration: Being considerate of others times and life
- Passion: Being passionate and caring about the project

Group Norms

- Each person will carry their own weight
- Each person will get their assigned work done on time
- If arrangements need to be made, let the group know as soon as possible
- Ask for help of you need it
- Trust your group mates to get their parts done
- Do the best work you possible can

Unacceptable Behaviour

- Discrimination against race, gender identities, physical abilities/appearances, nationality, age, religion, and beliefs
- Unwelcoming tone of voice
- Any form of aggression

Meeting Norms

Weekly meetings will be conducted and will be opened by the chairman. The secretary will distribute an agenda 24 hours before each meeting which will include the following topics of discussion.

- Brief overview of the previous week's tasks
  - Were all the respective tasks complete?
  - Were there any issues within the group or on the tasks?
- Timesheets of each member (hours works on individual task)
- Discuss the work which the plan and tasks that needs to be done for the upcoming.
  - How will they be distributed among the group members?
- Discuss any problems, conflicts, and issues among the group and individual members.
  - Is a member no carrying their weight or completing the tasks given to them?
- Determine the time and date for the next meeting

Team Roles

There are two roles associated with the EPS project. One being a chairman and the other being a secretary. Between the four group members, there will be two members holding the above positions for the first half of the Autumn 2021 semester and the remaining two will take over on the second half of the Autumn 2021 semester. Below describes the positions held for the respective members and time frames.

- 06.09.2021 to 15.10.2021
  - Chairman: Estefanio Kesto
  - Secretary: Hugo Frauciel
- 16.10.2021 to 17.12.2021
  - Chairman: Ties Kolthof
  - Secretary: Tobias van Varik

Reporting a problem

Shall any problems arise between group members, then we should try and address it directly with the respective team member(s) causing said problems. If the problem is unable to be resolved within the group, then it should be escalated to our group coach (Hans Lindén).