Creative Technology
Bachelor Thesis

Privacy-Friendly Room Occupancy Measurement System

Hylke Jellema

Supervisor: Daniel Davison
Critical Observer: Juliet Haarman

October 31, 2023

Creative Technology
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Abstract

The modern data-driven society requires methods to collect accurate occupancy data for informed decision making, and comes with user privacy concerns and challenges. This thesis addresses the challenge of collecting accurate occupancy data while prioritizing user privacy. The study focuses on the Interaction Lab at the University of Twente, where a privacy-friendly occupancy measurement system is desired by the client.

The research revolves around identifying stakeholder requirements, suitable data collection methods, achievable privacy given the requirements, and the prototyping choices. Finally, a system evaluation is performed. Reviewing literature and current implementations, highlights the privacy weaknesses of current technologies and the need for a private-by-design approach.

Requirements were collected through interviews with lab users and the Lab Team. Ideation and specification phases led to the development of a prototype. The prototype utilizes an VL53L5CX Time-of-Flight sensor with an 8x8 pixel array connected to a Raspberry Pi 4, with a developed algorithm. The device observes a 2x2 meter zone at the entrance with a achieved sample rate of 17-20Hz. Data is collected locally and mitigated into day-parts. The limited observation zone, low resolution sensor, and data handling choices, are to ensure maximum user privacy. Additionally, Lab Team users can request the current occupancy for planning.

The requirement evaluation confirmed the system's compliance with the set requirements. The performance evaluation achieved an accuracy of 96% in normal conditions. User and lab team supported the system with a positive reception, with suggestions for increased transparency.

In conclusion, this thesis demonstrates the feasibility of a privacy-friendly occupancy measurement system, which respects user privacy through hardware design while maintaining high accuracy. Future research should explore the role and importance of hardware selection in ensuring privacy.

Acknowledgements

First of all, I want to thank my supervisor Daniel Davison, for this assignment, the elaborate feedback in both mails and meetings, and the extra support during the summer break. I learned many new things outside of my comfort zone due to the devoted supervision. Additionally, I would like to thank Juliet Haarman for her feedback and time. It was good to have another perspective.

Next, I want like to thank my niece, Myrna Boer, for her help with figuring out Solidworks, and design input to get everything in a mountable case.

Last, I want to thank my family and friends for supporting me, as always.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Context

Our society is becoming more and more data-driven in recent decade. Accurate data can support good decision making, and collecting and using data to become a data-driven decision maker is accessible to smaller organisations due to the advancements made. In this context, gathering consistent occupancy data on the population within a room or building can offer significant utility. The collected data can be employed for diverse applications and data-driven decision making, such as: identifying overcrowded areas, evaluating event success, monitoring real-time occupancy to locate appropriate study spaces or determine staff presence, planning future structures, among others. Moreover, the information procured by these systems can contribute to economic savings and sustainability by facilitating effective energy management, by optimizing the thermal design of buildings[10][19]. However, the modern day data-driven society and with that the collection of occupancy data, comes at the cost of user privacy[22]. This project is aimed at researching if the required data can be collected while prioritising privacy, this introduces the problem statement.

1.2 Problem Statement

Current approaches may be excessively invasive, resulting in the potential collection of more data than necessary or the possibility to do so. In numerous instances, privacy is guaranteed by assurance rather than by hardware design. Rapid advances in this field have intensified these concerns[22]. A common implementation involves the deployment of an artificial intelligence model in conjunction with a camera capable of recognizing people. Although these systems typically focus on quantitative data, they remain susceptible to software breaches and misuse, due to the device having sensors that are not needed or too detailed for the primary goal. Also, devices have additional sensors for collecting other data points in the future and often have extra options such as demographic analysis and staff detection. With closed-sourced code the overall inner workings of such devices, and how these sensors are used, can not be fully investigated by a external party. As a result the promises of the manufactures, function as the sole indicator for user safety.

1.3 The client

The Interaction Lab, situated at the University of Twente, serves as a connecting hub for various research fields [1]. It provides a venue for students to engage in academic growth, explore technologies, and access resources. By offering these technologies, instructional tutorials, staff, and fellow students in a single location, the Interaction Lab fosters a sociable environment that motivates the creation of innovative ideas. The main coordinator for the lab space is interested in collecting occupancy data for a new additional lab in the "tower room". To collect data on the occupancy, user privacy must be a priority.

1.4 Goal and Research Questions

The goal of this project is to build a privacy-friendly occupancy measurement system. The system should prioritise the privacy of the user. The main focus of the project will be on the hardware used for collecting occupancy data and how the client requirements can be satisfied with minimal invasive and privacy friendly hardware. This would be hardware that is only able to detect and measure data points needed for an accurate measurement. Also, the deployment should not have any additional unneeded hardware or sensors then needed for the data collection requirements. Additional to the hardware, software considerations will also be made. The following research questions conclude the introduction:

> "To what extent can accurate occupancy data be collected in the Interaction lab when prioritizing user privacy?"
>
> ---
>
> Main Research Question

To help answer this research question some sub research questions have been made:

1. What are the stakeholder requirements for a private occupancy measurement system?
2. What methods could be used to collect occupancy data in the lab?
3. What level of privacy can be achieved with the requirements for data collection?
4. How can these requirements become a working prototype?
5. What is the overall evaluation of the implemented system?

Chapter 2

Background Research

In this chapter a review of relevant literature[1] and state-of-the-art research will be conducted on the current occupancy tracking systems at the University of Twente. Finally, the technologies that have been mentioned throughout the chapter will be compared in a table.

2.1 Literature Review

This review of the literature aims to provide a comprehensive comparison of the available technologies and systems for collecting room occupancy data. The technologies will be discussed based on the privacy-friendliness of the method and possibility of excess data to be collected by future software updates. To mitigate the risk of software modifications or exploits by the adversary, hardware data privacy could be an important part of the design cycle. Taking this approach may be necessary to avoid collecting sensitive data from the core sensor. Additionally, retaining data accuracy to a certain degree will also be taken into account. The review will look at a variety of methods to collect room occupancy data. e.g.: CO2, motion detection, or chair sensors, and conclude on which systems are most suitable to be used in a privacy-friendly and non-invasive manner while retaining real-time data accuracy to an acceptable degree.

2.1.1 Privacy when measuring occupancy

When collecting occupancy data on the whereabouts of people, personal information may be collected. Since this review aims to find a privacy-friendly system, it is essential to investigate what factors contribute to this goal and are important when considering a suitable system. Since all systems have to deal with storing data, this review will not investigate data privacy per method discussed from a software perspective. This section will inform the reader of the reasoning behind the factors that will be used for the systems discussed.

[1] section 2.1 The original review was written for the Academic Writing course, with slight adjustments for continuity and sentence structure.

2.1.1.1 Data handling and encryption

All systems that will be considered collect data in some way or another. Correct data handling and encryption are important methods to guarantee the privacy of users and achieve a system that is less vulnerable to software exploits. Old methods primarily focus on how to keep adversary from accessing the data. However, recent research is more focused on how data can be stored in such a way that less harm can be done after a data breach, as mentioned in[12]. There is sufficient literature to be found on how to treat data handling, storage, and encryption correctly. When investigating encryption, camera data encryption can become a problem. This is because encryption is a very computer intensive process when exposed to a large data stream that comes from video. A possible solution to this problem is partial-encryption, where only the region of interest is encrypted by a light-weight model[11]. Compared with other systems, this is not a problem that arises. Therefore, because all possible systems collect data, this is not a factor that will be further explored per investigated technology.

2.1.1.2 Privacy factors

The last paragraph above covered data privacy and why these factors will not be considered in the discussion. This can also be referred to as software privacy and applies to every system.

This discussion is interested in what technology is private-by-hardware-design. For hardware privacy in the occupancy measurement systems in this review, the following factors are used:

Minimizing the possibility of excess data collection by hardware design Limiting data collection by software is a method often used to mitigate privacy concerns. However, this may not be a future guarantee of data safety. An exaggerated example of this could be the need for occupancy data at a single desk. This could be achieved with a binary chair weight sensor with only the ability to send a yes or no. In contrast, a camera could be used in combination with an AI video detection model. Data collection from these two methods could be programmed in a similar way. However, it could be argued that problems may arise in the future.

Minimizing invasiveness The technology should not be invasive to the users. Using the same example as above, of the camera and chair sensor, one can say that a camera gives the user an privacy invasive feeling. The hardware should reflect the reality of a privacy-friendly technology. An important disclaimer to mention is that this does not mean that the technology should completely be hidden. The user should be informed about the collection of data.

Minimizing possibility of collecting personal identifiable data The technology should not collect data that can identify a person. This concept could be taken to great lengths as there might always be a method to extract data and cross-compare this with other collected data[36].

To summarize, the main takeaway from these factors is that occupancy data collection should not be comparable to any form of surveillance[36]. To further explain and categorize the different aspects of privacy in a system for the project, three levels have been formed. All privacy factors that this review places into level 2 & 3 can be applied to any sensor technology used. As a result of this, to select the scope of this literature review and focus on selecting a suitable sensor, all the factors in this section belong to level 1 as illustrated in Table 2.1.

Privacy Levels		
Level 1	Level 2	Level 3
Hardware Data Collection	Sensor Node Transmission	Data Storage
- Low sensor precision. - Limiting resolution.	- Limiting transmitted data - Adding noise to transmitted data[11] - [20]	- Limiting stored data. - Encrypting stored data. Covered in subsubsection 2.1.1.1

TABLE 2.1: Illustrating 3 levels where privacy measures can be applied. To help support this review

2.1.2 Methods for measuring occupancy

With the increasing need for accurate and real-time data on building and/or room occupancy, various methods have been developed to meet these demands. This section will examine the most commonly used technologies for collecting room occupancy data, including experimental technologies. In addition, the review will discuss privacy and data accuracy for all technologies that are investigated. At the end of this chapter, the reader will have a better understanding of the available technologies and their strengths and weaknesses in collecting room occupancy data.

To find suitable technologies, it is important to mention some methods and technologies that will not be considered. Firstly, no invasive method will be investigated that requires a user to actively carry a specific device, such as RFID tag detection seen in[23]. Second, any method that requires action from the user will not be considered. This does not include devices that users already have, such as phones. To find suitable methods which are widely used, some literature has been consulted that have done statistical research on the most common methods for occupancy detection[40][18]. From here, other sources have been snowballed to collect more information on the specific occupancy measurement systems. Additionally, other potential methods are considered which are more experimental.

2.1.2.1 Camera detection

Using a camera for people counting is a method that is often used for occupancy detection. It is based around the idea of applying certain AI models and blob detection to images from a video[28][17]. One big advantage of this method is the ease of implementation, as cameras are already widely used in many places and also serves other purposes. Additionally, the accuracy of the occupancy data of a system that uses real-time video is high. An experiment conducted in an office room shows an accuracy of up to 97.32% for correct detection[30]. Furthermore, a camera can collect data on the exact location of a person[25]. Consequently, one camera can be used to measure occupancy for multiple parts of a room and/or area. However, tracking exact locations for occupancy measurements creates excess data. Additionally, users could be identifiable through video. As a result, the use of a camera can create an invasive experience for users. Nevertheless, many of these problems can be mitigated by limiting the transmitted data from the camera node, as described in level 2 & 3.

2.1.2.2 ToF camera

Using a ToF (Time-of-Flight) camera for people counting is a method that works well for occupancy measurements. This method is based on the principle of measuring the time it takes for light to travel from a ToF camera to the object and back again. The data can then be processed to a depth map, which can be used to determine the location of people in a room. One of the main advantages of using a ToF camera in comparison to a traditional camera is the 3D output. Consequently, the background of the image can be removed and provides a more accurate understanding of the occupancy data compared to 2D images[29]. Additionally, a ToF camera uses IR (infra red) making it work in variable lit environments[34]. Moreover, TOF cameras have the ability to collect data without capturing identifiable images of people. This makes them a more privacy-friendly option. However, it could still be argued that excess data could be collected. For example, the exact location and time of a user could be tracked. In both[21] and[15] a method is shown where arrays of single-pixel ToF cameras are used to preserve user privacy. An experiment conducted using a TOF camera showed an accuracy of up to 100% for correct detection[39]. In addition, other research found an accuracy of 95%[34].

2.1.2.3 CO2 concentration sensor

CO2 sensors are considered the most effective sensor for occupancy detection among all environmental variables that can be monitored, such as temperature, humidity, and light [13]. The concept is based on the principle that an increased human presence raises CO2 levels in a room[41]. However, the accuracy of a CO2 based measurement system is not high. Namely, human CO2 production varies from person to person depending on many different factors[37]. Similarly, many other location-specific factors can alter the concentration of CO2 and may vary over time[24]. One solution to this is to combine other environmental factors into a learning model. Candanedo et al.[16] developed an statistical learning model using CO2 together with illumination, humidity, and temperature factors. Other studies also use CO2 measurements together with data from wireless communication sensing. For instance, Wang et al.[35] developed a method using WI-FI sensing. When comparing traditional and newer models, the accuracy varies between 55-95%[16].Multiple methods that bring together variables of varying accuracy use some form of statistical learning to create accurate occupancy measurements. Consequently, implementing similar learning models may be necessary for less invasive methods of collecting occupancy data.

2.1.2.4 Chair sensors

A less common experimental method is the use of chair sensors. This method relies on the usage of multiple sensor nodes that communicate with a base station. When using solid chairs, integrating weight sensors can become a challenge. A solution to this problem could be to use temperature sensors to sense the presence of a person[31]. This method of occupancy measurement is very accurate for areas where people tend to sit down. However, if the area has a less predictable ratio between standing and sitting down, The data becomes less accurate.

Although this method is not invasive, the data it collects can become a privacy concern. For example, if the same person uses a chair every day, the data from that particular chair can be used to collect personal information. Furthermore, when temperature sensors are used, data about the health status of a person could be exploited. Although, these concerns can be mitigated by limiting the transmitted data (level 2). For example, by transmitting a binary value when the temperature changes to above 30 degrees[31]. Also, the node identifier could be randomized so it is more complicated to track a specific chair node[27]. These methods could still be vulnerable to possible software exploits.

2.1.3 Technology conclusion

To summarize this section, multiple methods have been discussed. It can be concluded that there is no technology that suits all requirements. An increase in accuracy and real-time data is likely to be paired with privacy concerns. As can be seen in Table 2.2. However, some noteworthy differences should be mentioned. At first, using a ToF camera over a traditional camera can increase accuracy due to its ability to remove the background from an image. Additionally, a ToF camera collects less personal data and can be trimmed down to lower resolutions while retaining accuracy. It can be concluded that, in the scope of occupancy measurements, an ToF camera is more suitable over a traditional camera. Second, a CO2 sensor has issues with accuracy due to many other factors that alter CO2 concentration in a room. However, other data can be used together with statistical models to increase accuracy. Finally, the use of chair sensors might be suitable if the setting allows. In addition, having occupancy data per zone or chair can help with dynamic lighting and other optimizations. Ultimately, when choosing a suitable technology, a suitable balance has to be struck between these strengths and weaknesses for the specific needs of a situation.

2.1.4 Literature Review Conclusion

Selecting a suitable technology for collecting occupancy data is crucial to avoid collecting excess data. Many privacy concerns can be mitigated by software precautions. However, the sensor technology used may still become an issue in the future because of a software update. Additionally, these technologies add to the invasive nature of a method such as traditional camera. This literature review was aimed to compare different methods to support a balanced decision making. To achieve this, multiple methods have been discussed and compared. Additionally, the review showed the importance of choosing a method that accommodates the required data.

Using multiple sources the most common methods were found and supporting sources were snowballed from there. Afterwards, literature was investigated in regards to the privacy of these technologies and as a broader term. From this it can be concluded that there was limited literature on hardware privacy, named level 1 in this review. The implication of these limitations is that less is known about absolute privacy, by hardware design. Future research could, for instance, investigate how hardware can support the main goal of a system and limit the possibility of additional usage when dealing with data collection in certain spaces. Such research could contribute to protecting user privacy in times of growing possibilities in user data collection, AI models, and increasingly capable technology.

2.2 State of the Art

In this section, the state of the art in room occupancy systems that are currently used at the University of Twente are explored. In section 2.1 a literature review has been carried out, discussing various methods and evaluating their accuracy and individual privacy. In contrast, this section will investigate various existing deployments. These will be analyzed and contrasted to underscore their distinctive features, advantages, and potential limitations.

2.2.1 Implementation at University of Twente

A current implementation for occupancy measurement is present at the University of Twente, particularly focusing on "Occupancy Management"[3]. Since late 2020, sensors have been installed in the Ravelijn building. This project aims to tackle the increasing challenge of scheduling and effectively utilizing campus spaces and facilities due to a growing student population. The project aims to achieve faster and better insights into the use of lecture and project rooms. The current scheduling approach is based on registered timetables, not on the actual usage of spaces. Sensor data allows for a better understanding of room utilization and trends over time, as opposed to manual counting, which provides only a momentary impression.

This pilot project aims to evaluate different sensor technologies to achieve the desired results: faster, improved, and more accurate information on the use of educational facilities and meeting rooms. The rapidly evolving sensor market offers improved technology and accuracy, following this pilot, decisions will be made on the installation of people counters and presence sensors in other campus buildings.

2.2.1.1 Types of sensor technologies:

Motion Sensors All motion sensing technology used at the university are based on PIR (passive infrared sensor).

- Capgemini Desk sensor - PIR (general movement), installed under desks
- Capgemini meeting room sensor - PIR (general movement)
- Elsys ERS Eye - PIR (general movement)
- Elsys ERS Sound - PIR (general movement) and sound
- OAC Electronics NightHawk - PIR, temperature, humidity and light

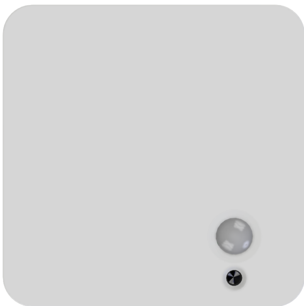

FIGURE 2.1: Picture of the ERS Eye sensor

Discussion In the scope of this project, privacy by design is of significant importance. The listing of all different motion sensors used shows the sensors that are named on the website of the University of Twente[5]. Investigating these sensors from their original product pages, it should be noted that both Elsys ERS sensors have additional sensors in addition to the PIR sensor[5]. Both ERS sensors have the ability to sense temperature, humidity, light, and sound. Additionally, the ERS Eye has a "Panasonic Grid-Eye© infrared sensor" capable locating the location of detected motion.

People Counters All people counters used at the university are either based on overhead stereo vision or ToF sensing in combination with a camera, as shown in Figure 2.2. Stereo camera systems use 2 high resolution cameras.

- Flir Brickstream 3D Gen 2 (BPC-2500M-25W) - Stereo vision with 2 digital cameras.

- Xovis PC3-M - Stereo vision with 2 digital cameras.
- Xovis PC2S - Stereo vision with 2 digital cameras.
- Irysis Vector 4D Analytic - ToF sensor and digital camera.

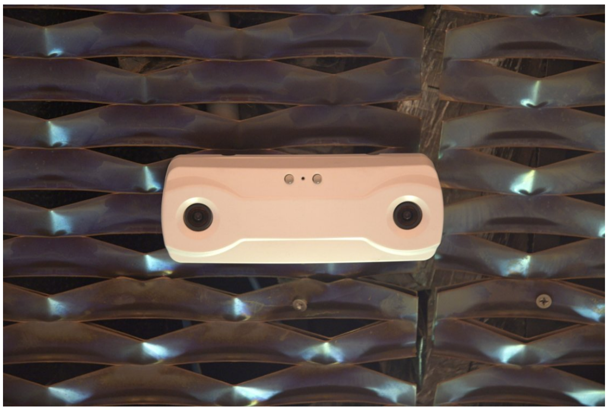

FIGURE 2.2: Picture of a Flir Stereo Vision sensor at the University of Twente

Discussion Further exploring these technologies via there product page raises concerns in the scope of privacy-by-design. First of all, the implementations use digital cameras with a high resolution. The university website states that all devices are configured to convert images to low-resolution images. However, these are software implementations, not hardware limitations. On the product pages further capabilities can be found. One of these capabilities is the improved AI computer on the Xovis PC2S, capable of classifying gender and age[6]. Additionally, the Irisis device digital camera is only used for setup and calibration. However, on the product page the digital camera can be used for staff identification. In the scope of privacy by hardware design, these sensors could raise questions. Furthermore, to improve the privacy the stereo cameras are mounted top-down to ensure the technology is not able to identify faces.

2.3 Technology summary

To help support the further exploration and selection of a suitable sensor, all technologies will be added to a visual representation in Table 2.2, which shows the strengths and weaknesses of each implementation. The summary is a combination of the sensors from both the literature review, and the accompanying conclusion in subsection 2.1.3 and the existing technology at the University of Twente, discussed in section 2.2. A conclusion that can be made from the technology summary is that sensors that offer accurate real-time occupancy are mostly also less private. Some technologies, such as, Time of Flight sensors offer many privacy benefits while having the possibility to offer accurate data. These findings can be used for the further exploration of possible implementations.

	Hardware Privacy			Data Accuracy	
	1. Excess data	2. Invasiveness	3. Personal data	Real-Time	Accuracy
Camera detection	○	○	○	●	●
Stereo Camera	○	◐	◐	●	●
ToF camera	◐	◐	◐	●	●
ToF camera (single pixel)	●	◐	●	●	◐
PIR	●	◐	●	●	◐
CO2 sensor	●	●	●	○	○
CO2 sensor (combined)	●	●	◐	◐	◐
Chair sensors (binary)	◐	◐	◐	●	◐
Chair sensors (temperature)	○	◐	○	◐	◐

● = Good; ◐ = Acceptable; ○ = Not desirable;

Table 2.2: Comparison of methods

Chapter 3

Methods and Techniques

With the introduction of a problem and the background research completed, the next step is to work toward a solution. This chapter will explain the structure and method of how this project will be structured, by describing the phases for this project to find a suitable solution for the problem statement.

3.1 Creative Technology Design Process

In this section, the chosen design process will be motivated and explained. The Creative Technology Design Process (CTDP) is a specialized design methodology tailored for the Bachelor of Creative Technology program at the University of Twente. This process incorporates a suitable balance between models, namely the divergence and convergence models, as well as spiral models, as depicted in Figure 3.1 below. Its primary objective is to provide a framework for navigating the often intricate design process, fostering iterative and adaptable ideation and prototyping. The classic CTDP comprises four distinct phases: Ideation, Specification, Realization, and Evaluation. Each of these phases will be elaborated upon, along with their application in this bachelor thesis [26].

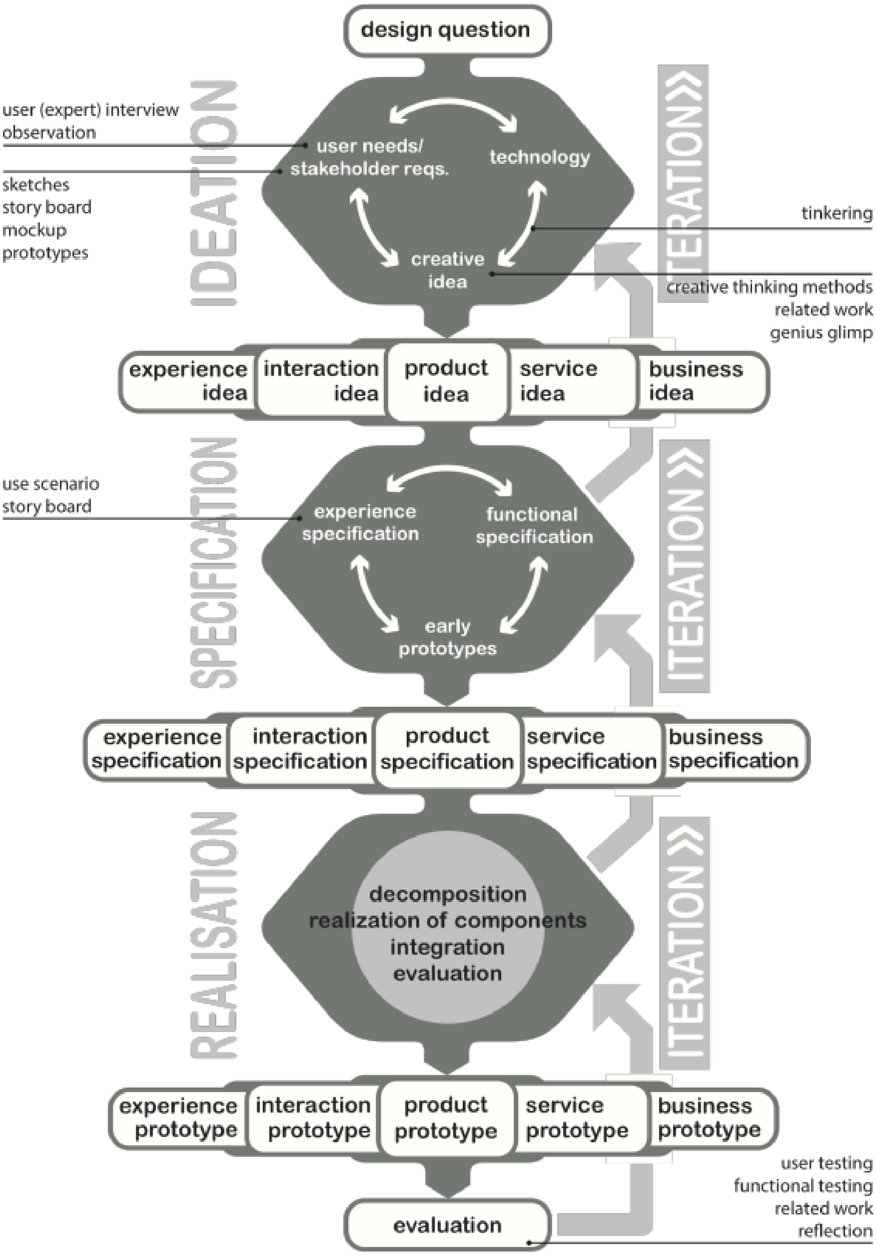

FIGURE 3.1: The Creative Technology Design Process [26]

3.1.1 Ideation phase

In the first phase, a couple of possible concepts will come foreword from the ideation process. The section will start with a stakeholder analysis. This analysis will identify the main stakeholders and investigate what is important to them based on user interviews. The information from stakeholders and background research is then used for two brainstorm sessions. In the first session, possible implementations will be brainstormed and sorted according to requirements. In the second session, possible methods to communicate this data will be explored. From this, some direction and conclusions can be made for the next phases.

3.1.2 Specification phase

In the specification phase the requirements are specified, and a final mix of technology for the prototype and implementation will be outlined. To support this process and the start of the realisation phase, user representations are created based on the stereotypical user the project is trying to address. Additionally, use cases and user stories are created to map the interactions required for the next chapter.

3.1.3 Realisation phase

In the realisation phase, the prototype will be created based on the specified requirements and interactions required. At this point, the outline of the approach is known. However, a part of the selection and choices made during this phase will be discussed in further detail in parallel to the prototype realisation process, such as the exact brand chosen for the sensor. During the realisation chapter multiple methods are used to document the back-end workings of the interactions. Such as, pseudo code and process diagrams.

3.1.4 Evaluation phase

In the final phase an evaluation of the prototype will be conducted. The evaluation will focus on three main parts. First, an assessment of the requirements is conducted to determine the success of the implementation of the requirements identified in the MoSCoW analysis from the specification stage. Second, a performance evaluation will determine both the quantitative accuracy in normal operation, and include a qualitative analysis of more unique situations.

Chapter 4

Ideation

In this chapter, a stakeholder analysis will be conducted. The analysis will be based on the client's requirements, and interviews conducted with both users and a team member of the interaction lab. Then, two brainstorm session will be held for both the technologies that could be used for certain requirements, and the communication methods. A preliminary selection will be conducted to select the most promising combination of technologies. From the outcomes, conclusions can be drawn on the best fitting concepts, and which technologies these will use for the next specification phase.

4.1 Stakeholders

Stakeholder	Role	Contact
Interaction Lab (client)	Decision-Maker	Daniel Davison
Users of the lab	User	Interviews
Lab Team	Staff	Interviews

TABLE 4.1: Stakeholder identification

4.2 Stakeholder Analysis

4.2.1 Interaction Lab (Client)

The client and also the supervisor for this project, Daniel Davison, is the main stakeholder for this project. He is the discision-maker when it comes to implementation and holds the project to certain standards, thus his power is high. Additionally his interest is high, as the project is based around his requirement of building a system that is able to measure occupancy. For these reasons it is important to manage the client closely.

4.2.2 Lab Team

The lab team consists of the people that manage the day-to-day activities in the lab, and form the basis of the social lab space. They are very valuable in collecting information on the needs and wants around the space. As they can use the collected data for planning they also have a high interest. Additionally the lab team is consulted when decisions have to be made. Thus, their power is high and they should also be managed closely.

4.2.3 Users of the lab

The users of the lab consists of students, researchers and other members of the UT community. They use the Interaction Lab as a facility and for personal growth. As the data could potentially be used for real-time occupancy there is some interest. Their power is relatively high as privacy concerns could lead to problems with the implementation. They should be consulted on how to mitigate these concerns.

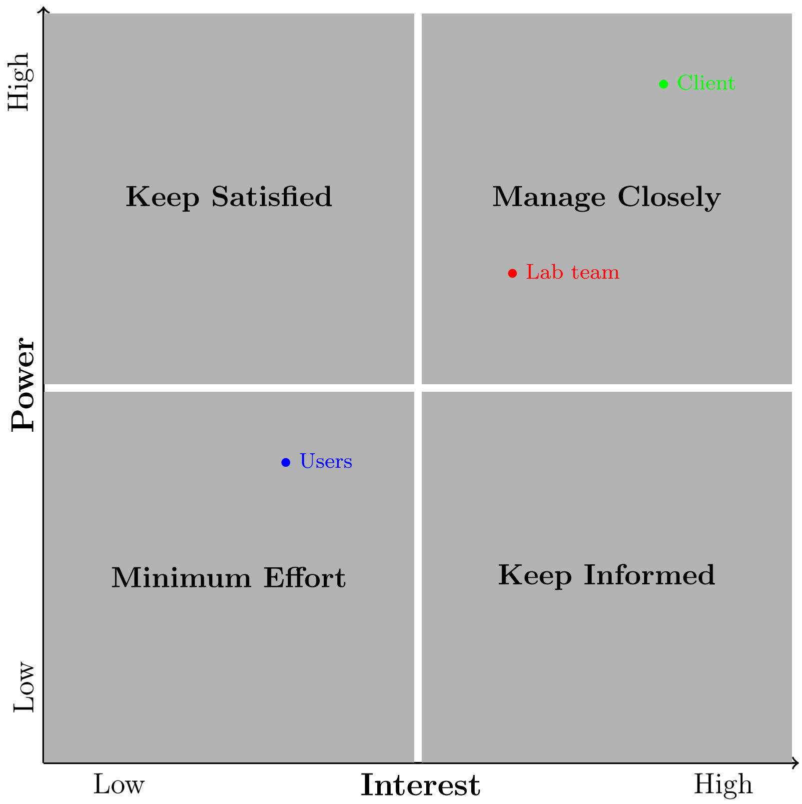

Table 4.2: Power-Interest Matrix

User and Lab Team interviews

4.3 Stakeholder Requirements

For the ideation phase it is important to know the requirements from different stakeholders that will interact with the final product. These requirements are based on the initial client requirements and interviews conducted with the lab team and users of the lab.

4.3.1 Interviews

The User and Lab Team interviews are conducted in the lab space. The client requirements were known as starting blocks for this project. This subsection will explain how the interview was setup to collect the stakeholder requirements. For these interviews the EEMCS ethics procedure was used and completed.

Method The user evaluation process began with the recruitment of potential participants, who were present in the lab environment. These participants were provided with an information brochure, as shown in Appendix B, to clearly communicate the evaluation's objectives and what it entailed, along with their right to opt out at any time. After this, the participants were requested to sign a consent form, as presented in Appendix C, to affirm their willingness to engage in the interview. The user evaluation incorporated both lab team members and general lab users in a single unstructured interview. Topics explored included privacy invasiveness, data privacy, trust in software, and the potential interest in an API, for further development, and other requirements or preferences. The demographic of the lab users are mostly students of both Creative and Interaction Technology, looking for a place to study. During the interview notes will be made on the key conclusions made in the open thematic conversation. Finally, the participant is thanked for participating and offered to check the notes made.

Analysis The results from the unstructured interview are concluded into an overall conclusion, and the key insights used for defining the user and lab team requirements.

Interview The interviews were held with 6 users, of which 2 are members of the Lab Team. In conclusion, the interview data reveals varying perspectives on privacy, data collection, and technology trust among the participants. The interviewees from the lab seem relatively comfortable with technology, but negative remarks were made about using cameras, they value transparency and express the importance of not hiding the sensors. They are not interested in using an API or developing anything. Further key insights are documented in the requirements section below, together with the client requirements.

4.3.2 Client requirements

- Measure occupancy and collect this data.
- Time resolution of at least multiple parts of a day.

- Make the data accessible.
- Respect the privacy of users.

4.3.3 User requirements

- Consider privacy.
- Limit the amount of stored data to ensure privacy.
- Not use a digital camera.
  - This is a concern for users that are handling vulnerable data on their computer screens.
- Measure occupancy for different desks/zones.
  - So users are able to see the availability of an installation they need.
- Have a public request system.
  - Users expressed that certain data could be accessible through a request system with public logging. This ensures privacy through transparency.
- Show the presence of staff members.

4.3.4 Lab Team requirements

- Measure the amount of users walking into the lab.
  - This data could be used to identify the moments when extra staff is needed.
- Notify lab team when many users come to the lab.

4.4 Brainstorm session 1

In the first brainstorm session, implementations are generated based around multiple ways occupancy data may be used by the stakeholders. To keep the session as open as possible only one requirement is used per time. Afterwards, the session will explore if a certain implementation can also serve another requirement.

- Measure occupancy real-time, being the current occupancy in the lab.
  - Client Requirement
- Measure occupancy in parts of the day.
  - Minimum client requirement
- Measure occupancy of zones and desks.
  - User requirement
- Measure in- and output of users.
  - Lab team requirement

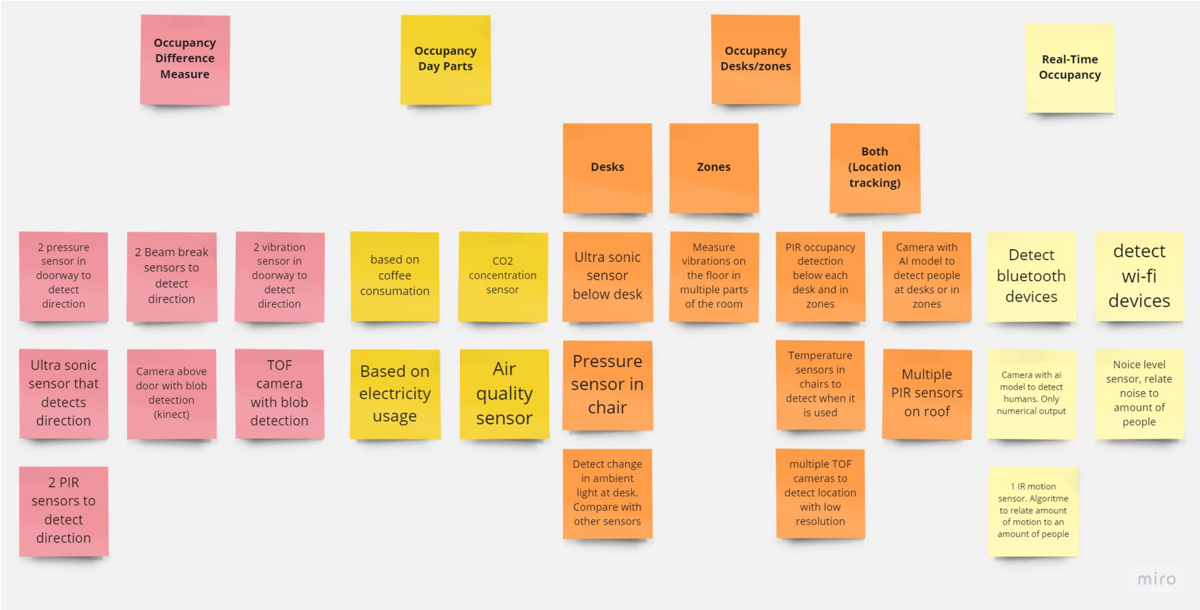

FIGURE 4.1: Post-its from first brainstorm session

4.4.1 Preliminary selection

Before the concepts are generated based on this brainstorm session, some methods can be selected for their potential. First, the methods are sorted based on which requirements can be satisfied with the methods from the brainstorm session. Second, the same methods are reviewed based on user privacy. Finally, the method offering the most privacy for the set requirement group is selected. For example, from the methods that satisfy three requirements, the one is chosen with the least privacy concerns.

4.4.1.1 Methods that satisfy multiple requirements

Most methods that have been generated to achieve a certain requirement would also function to meet other requirements. First of all, the user requirement of measuring the occupancy of desks and zones using multiple passive infrared sensors (PIR) or an AI driven camera can satisfy the client requirement of real-time occupancy reading for the complete room. Using a learning model, most of these methods could predict overall occupancy. Similarly, the lab team requirement that measures the number of people who enter or exit the lab could also be calculated to give an overall count. In Table 4.3 this is mapped and sorted.

	Day Parts	Difference	Desks and/or zones	Real-Time Overall	total
Camera with AI model to detect people at desks or in zones	☑	☑	☑	☑	4
Multiple PIR sensors on roof	☑	☑	☑	☑	4
multiple TOF cameras to detect location with low resolution	☑	☑	☑	☑	4
2 pressure sensor in doorway to detect direction	☑	☑	☐	☑	3
2 Beam break sensors to detect direction	☑	☑	☐	☑	3
2 vibration sensor in doorway to detect direction	☑	☑	☐	☑	3
Ultra sonic sensor that detects direction	☑	☑	☐	☑	3
Camera above door with blob detection (kinect)	☑	☑	☐	☑	3
TOF camera with blob detection	☑	☑	☐	☑	3
Camera with ai model to detect humans. Only numerical output	☑	☑	☐	☑	3
2 PIR sensors to detect direction	☑	☑	☐	☑	3
PIR occupancy detection below each desk and in zones	☑	☐	☑	☐	2
Based on electricity usage	☑	☐	☑	☐	2
Detect bluetooth devices	☑	☐	☐	☑	2
detect wi-fi devices	☑	☐	☐	☑	2
Noice level sensor, relate noise to amount of people	☑	☐	☐	☑	2
1 IR motion sensor. Algoritme to relate amount of motion to an amount of people	☑	☐	☐	☑	2
Measure vibrations on the floor in multiple parts of the room	☑	☐	☑	☐	2
based on coffee consumation	☑	☐	☐	☐	1
CO2 concentration sensor	☑	☐	☐	☐	1
Air quality sensor	☑	☐	☐	☐	1
Temperature sensors in chairs to detect when it is used	☐	☐	☑	☐	1
Ultra sonic sensor below desk	☐	☐	☑	☐	1
Pressure sensor in chair	☐	☐	☑	☐	1
PIR below desks	☐	☐	☑	☐	1
PIR That cover certain area/zone	☐	☐	☑	☐	1
Detect change in ambient light at desk. Compare with other sensors	☐	☐	☑	☐	1

TABLE 4.3: User requirements and methods comparison

4.4.1.2 Privacy

Now that the methods have been sorted based on requirements some methods, that satisfy the same requirements but are more invasive, can be eliminated. To support this process the methods are mapped in Table 4.4 based on the following privacy factors:

- Always-on devices that always capture user information, not just at the door or by looking at certain events.
- Able to recognize detailed personal features, such as cameras.
- Able to recognize other personal features, such as temperature, pressure sensors, or cameras from above.
- Creating possible other concerns such as; a sensor that is installed on desks or in zones, as these could give insights into the behavior of a certain user using the same desk, and technology that uses Wi-Fi and Bluetooth solutions, as they could track device ID.

	Continues sensing	Reconize people feature in detail	Reconize other people features	Other concerns	total
Camera with AI model to detect people at desks or in zones	☑	☑	☑	☐	3
Camera with ai model to detect humans. Only numerical output	☑	☑	☑	☐	3
PIR occupancy detection below each desk and in zones	☑	☐	☐	☑	2
Detect bluetooth devices	☑	☐	☐	☑	2
detect wi-fi devices	☑	☐	☐	☑	2
Noice level sensor, relate noise to amount of people	☑	☐	☐	☑	2
multiple TOF cameras to detect location with low resolution	☑	☐	☑	☐	2
Temperature sensors in chairs to detect when it is used	☑	☐	☑	☐	2
Measure vibrations on the floor in multiple parts of the room	☑	☐	☑	☐	2
Ultra sonic sensor below desk	☑	☐	☐	☑	2
Pressure sensor in chair	☑	☐	☑	☐	2
PIR below desks	☑	☐	☐	☑	2
PIR That cover certain area/zone	☑	☐	☐	☑	2
Detect change in ambient light at desk. Compare with other sensors	☑	☐	☐	☑	2
Multiple PIR sensors on roof	☑	☐	☐	☐	1
Camera above door with blob detection (kinect)	☐	☐	☑	☐	1
TOF camera with blob detection	☐	☐	☑	☐	1
Based on electricity usage	☑	☐	☐	☐	1
1 IR motion sensor. Algoritme to relate amount of motion to an amount of people	☑	☐	☐	☐	1
2 pressure sensor in doorway to detect direction	☐	☐	☐	☐	0
2 Beam break sensors to detect direction	☐	☐	☐	☐	0
2 vibration sensor in doorway to detect direction	☐	☐	☐	☐	0
Ultra sonic sensor that detects direction	☐	☐	☐	☐	0
2 PIR sensors to detect direction	☐	☐	☐	☐	0
based on coffee consumation	☐	☐	☐	☐	0
CO2 concentration sensor	☐	☐	☐	☐	0
Air quality sensor	☐	☐	☐	☐	0

TABLE 4.4: Privacy and methods comparison

4.4.1.3 Conclusion

For the final selection of the methods to be used, the most private method is chosen from every requirement group.

- **Multiple PIR or TOF sensors throughout the lab:** From the group that satisfies all 4 requirements, using cameras can be eliminated due to the privacy concerns. PIR being more privacy friendly, however TOF sensors can be limited in resolution, achieving the same with less sensors.
- **TOF above entrance with blob detection:** Measuring the difference using a technology at the entrance and exit of the lab is a very private way of occupancy measurement, as the collection of data is limited to the small observed zone.
- **1 PIR motion sensor in lab to detect amount of movement:** A very privacy friendly way of detecting occupancy. However, this method is less accurate for measurements due to it requiring motion.

4.5 Brainstorm session 2

In this session methods to communicate the collected data will be explored. The goal is to brainstorm suitable methods to communicate the data to a user. This process is completed by looking at the main requirement per stakeholder, and the possible ways this requirement can be met, as seen in Figure 4.2.

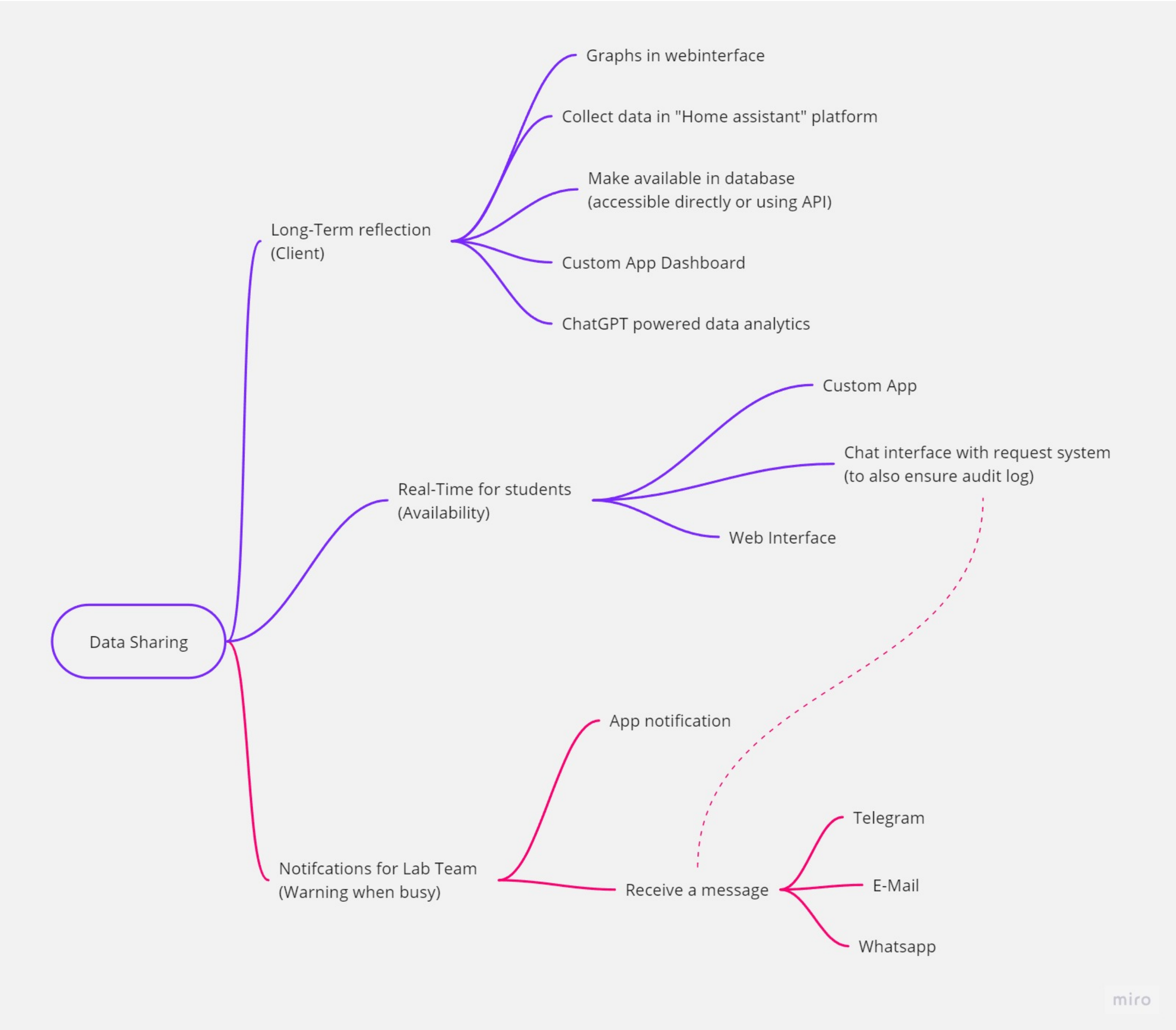

FIGURE 4.2: Mind Map from second brainstorm session

4.5.1 Preliminary selection

When communicating the data collected from a selected technology, authentication plays an important role. The collected data should not be accessed by users who are not authorized. During the user interviews it was mentioned that users do not prefer to have another web interface or platform that requires an account. Additionally, an administrator must keep track of who can access the occupancy data. In addition to the extra required actions, users also expressed the importance of displaying who requested what data in a log. This log could be, for example, accessible by all authenticated users. When considering all these factors, the usage of a chat app becomes interesting due to offering a solution to these preferable requirements. First, most people already have a chat app installed on their smartphone when using a widely-installed application. Therefore, most users would not require a new account or an app to be installed. Second, when displaying the data only in a group chat, an admin could add and remove the users that are allowed to access the data with ease. Moreover, having the data only in a special group chat also allows for auditing by other users. For example, it would be quickly noticed when a person keeps requesting data that is not required by that person at that time. Finally, a chat method can also deliver notifications quickly. For these reasons, this method is preferred. The final selection will be completed in the next chapter.

4.6 Conclusion and concepts

From the ideation some technology approaches and usages have been ideated and discussed. The final prototype will be a chosen mix between a communication method and a sensor technology. Additionally, the further implementation and other technologies will be added to this combination in the next two chapters. One finding was the promising usage of a chat app to communicate data. Another technology with high potential was the usage of a Time of Flight sensor, as they offer more accuracy then cameras when placed top-down, while increasing privacy. From this two concepts are chosen. However, the final implementation could deviate. Both concepts use a Time of Flight sensor. The first concept only observes a zone post-door, for a more private system, and then uses a chat app for data reporting. The second prototype, uses multiple sensors throughout the lab, a more complex web interface is used to communicate the additional data for the different zones.

4.6.1 First concept - Chat Based occupancy system with a Time of Flight sensor observing a zone at the entrance of the lab

The first concept, involves a Chat-Based Occupancy System with a Time of Flight sensor placed at the entrance of a lab. The system operates by monitoring the number of individuals entering and exiting the lab space using the Time of Flight sensor and a blob detection algorithm. Authorized users can interact with the system via a chat interface, making inquiries about current occupancy. The occupancy data is saved locally to ensure privacy.

4.6.2 Second concept - Time of Flight sensors above zones with web interface that shows occupancy per desk and installation

The second concept, introduces Time of Flight sensors placed above designated zones within the lab. These sensors continuously monitor the occupancy of specific areas or desks. The collected data is then made accessible through a web interface, allowing users to view real-time occupancy for different zones and installations within the lab. This concept provides a web-interface that enables users to view when lab resources can be used, optimizing workspace utilization and allowing efficient resource management.

Chapter 5

Specification

In the specification chapter, the user, lab-team and client requirements are reviewed and added to a MoSCoW analysis. Additionally, these requirements are outlined in user representations and user interactions/stories, to get a better understanding of the stakeholders this project is addressing. After further specification of the systems workings, a conclusion is drawn and the MoSCoW analysis is expanded for the final prototype.

5.1 Requirements - MoSCoW Analysis

In the ideation phase and background research, some ideas and concepts have emerged. To further build on these ideas in this phase, the requirements are organized into a MoSCoW analysis.

Category	Requirement	Explanation
Must	Measure occupancy for parts of the day.	The main project requirement from the client. The lab team can also use this data for their planning.
Must	Share this data in a dashboard	The main project requirement from the client. The lab team can also use this data for their planning.
Must	Respect the privacy of users	A project requirement from the client and a shared concern among users.
Should	Limit the amount of stored data	A concern among users.
Should	Not use a digital camera	A concern from some users. Even if nothing is recorded, the lab should feel as a safe social space.
Should	Collect in/output of users	The lab team expressed that looking at the in and out stream of people could be useful. Adionally, this allows for occupancy calculations.
Could	Measure occupancy for different desks/zones.	Users expressed that knowing if the desk or zone with a technology they are using is occupied could be useful.
Could	Have a request system	Users expressed that when per-zone data is collected, this data should only be visible after a user requests this data. Logging this publicly is a solution.
Could	Show the presence of staff members	Users expressed that knowing staff presence can help them decide when to come to the lab
Wont	Host the data trough an API	Users expressed that they are not very interested in developing tools with the data at this time.

Table 5.1: Product Requirements MoSCoW Analysis

5.2 User representations and Stories

To help with further specifying and understanding the requirements, user representations will be outlined that reflect the typical user, based on the user interviews. In addition, user stories will be added to illustrate the interactions with the design.

5.2.1 User representation Michaël

Michaël represents users from the lab team and is based on initial requirements and interviews. Michaël is interested in the real-time data generated by the occupancy system so he can better plan his visits to the lab to support students. His frustration is not knowing the amount of students in the tower room, requiring him to sometimes 'check up'. If he would have access to the current occupancy, he would be able to plan the best moment to take the stairs to the tower room, or come to the Interaction lab at all from another place on campus. This is concluded in

Figure 5.1.

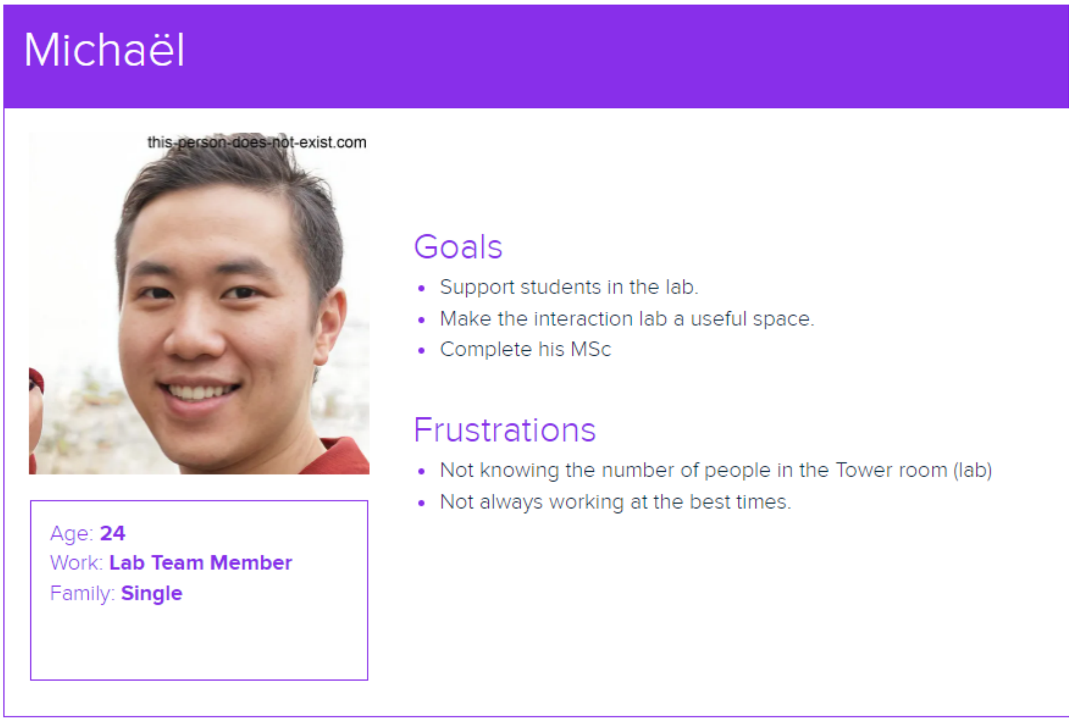

FIGURE 5.1: User representation Michaël, created with Xtensio, picture generated using AI

5.2.2 User representation Pamela

Pamela represents the lab coordinator and is based on initial client requirements. Pamela is interested in long-term data generated by the occupancy system so that she can better plan events and lab team plannings. Additionally, she can share this data with projects at the university that could use these insights. A frustration this project is trying to solve for her, is not knowing the amount of visitors came to the lab compared to another moment when a new installation is placed. When Pamela plans an one day event, that she attends, there is some perspective on the success of this event. However, when the attraction placed or event is multiple days, and she does not attend the lab full-time, this data help her measure the success. Along the line this long-term data can lead to making data-driven decisions. This is concluded in Figure 5.2.

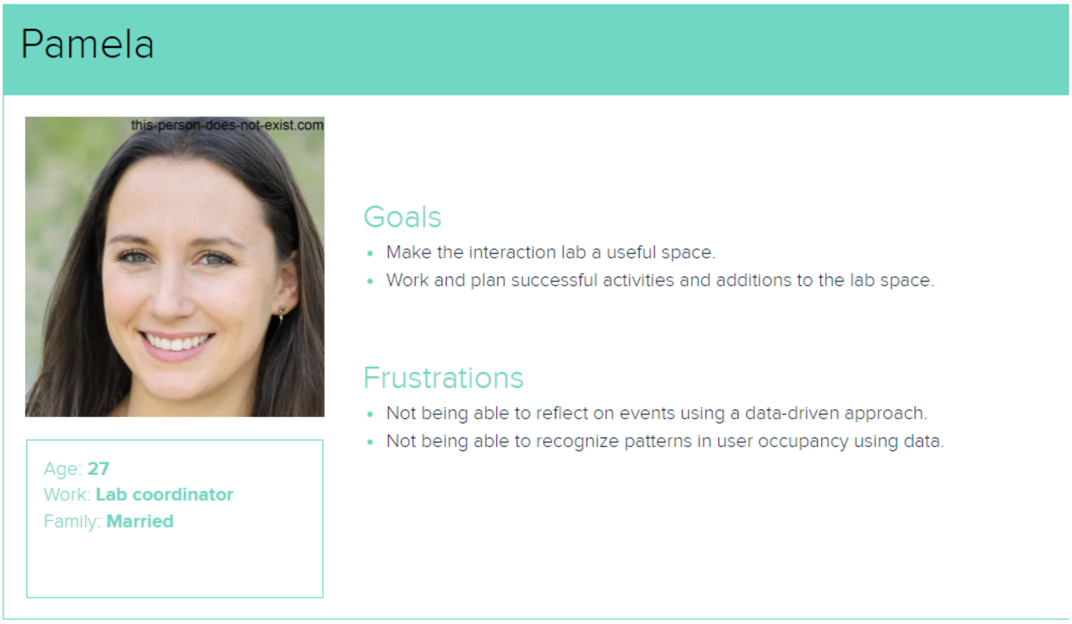

FIGURE 5.2: User representation Pamela, created with Xtensio, picture generated using AI

5.2.3 User representation Tom

Tom represents users from the lab and is based on initial requirements and interviews. Tom comes to the lab to study and sometimes work on projects using the installations available. Tom cares about his privacy and is not comfortable with a hidden system. At the same time, he does not want the implementation to bother him with any required interactions. as he has enough obligations already and the lab is a space for him to focus on his studies. This is concluded in Figure 5.3

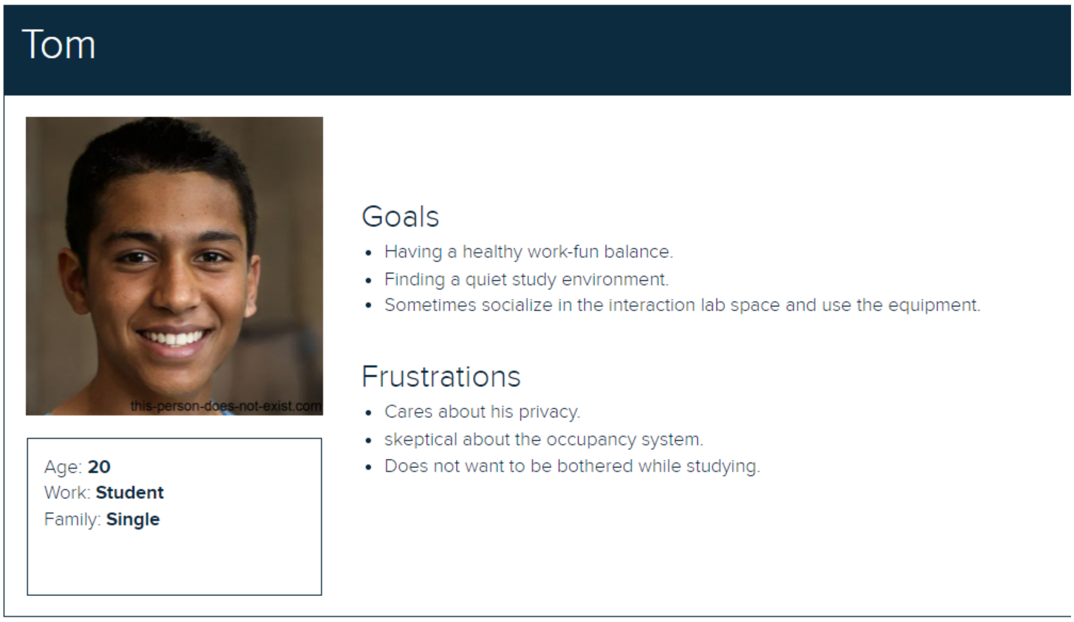

FIGURE 5.3: User representation Tom, created with Xtensio, picture generated using AI

5.2.4 User stories

User stories show the interactions and goals for the user groups. Each entry defines a user representation, an interaction, and the goal that is achieved by that interaction. The user stories are formatted according the the outline by the Agile Alliance [4]

- As a Lab Team member I want to know the current amount of occupancy so that I can decide if I need to go to the lab.
- As a lab user I want the occupancy system to not require an action from my side so that I can study without distraction.
- As a lab visitor I want the occupancy system to not collect more personal data then needed so that I do not have to worry.
- As a researcher I want to be able to access long term data so that I can use it for my project.
- As a lab coordinator I want to use the long term data so that I can use it for planning and reflection.

5.3 Sensor selection

The primary focus on privacy differentiates the development of this implementation from other systems. In the background research it was discussed why privacy at a hardware level is crucial to build private-by-design occupancy measurement systems. This requirement and concern has also been shared by users in the ideation phase. Because of this, a low-resolution time-of-flight camera sensor was chosen above the door zone. This is further explained.

5.3.1 Door zone sensing

By only sensing the zone by a door, users are only observed for a brief moment when entering or exiting the lab. This method makes for a less invasive system that, by design, cannot collect any user data beyond this area. Additionally, this method offers very accurate readings of the total in and out movements, this can offer additional insights.

5.3.2 ToF instead of other sensors

Some sensors are less capable of capturing user sensitive data, such as a method using 2 PIR (typical motion sensing) sensors or using any other 2 point system including lasers, vibrations and sound. However, these methods often have trouble with people entering and leaving simultaneously or detecting larger groups as individuals, as the raw data becomes harder to interpret. Using higher resolution cameras could potentially deliver better data. However this raises many privacy concerns. Additionally, users have expressed the invasive nature of using cameras. Using a Time-of-Flight camera allows for using lower resolutions and distinguish between ground and person/object without the need of detecting colors, which in turn could be used to identify users. Additionally, a Time-of-Flight camera is able to show the highest point in the area of a pixel, making even lower resolutions possible.

5.4 Data saving and accessing

The data collected by the system must be saved and shared. For this section, privacy is again a leading requirement. The data will only be saved locally. At the end of each day the data will be removed and saved to day parts. Real-time data is communicated through a chat app.

5.4.1 Minimal data saved

From the client requirement, occupancy data must be saved in day parts or higher resolution. Another requirement from the lab team is to make the in and output of users accessible. To achieve both these requirements the visitors walking into and out of the lab will be saved into short-term storage. The sensors depth data will not be saved. The algorithm will change the sensor data to a binary map and from this the only output a person entering or leaving the room, and register this into the database. At the end of the day the data will be saved into a long-term database and aggregated into 3 day-parts. When this process is completed with short-term storage, used for the current occupancy, is removed.

5.4.2 Accessing data

Using a chat app, data can be requested from the system. The system will give a rough estimate of the current occupancy, and is able to notify the lab team when a large group enters. Then long-term data can be accessed by an administrator by accessing the device. This way any data safety concerns are mitigated, and only rough data can be viewed.

Advantages using a chat app:

- Data is shared on a request-base in a group chat. - This allows for easy auditing by other lab team members.
- Allows easy permission controls. - Everyone in the group can access.
- Easier to access then opening and authenticating on a web app.
- No additional app needed.
- Future adaptations to which data can be requested are simpler to implement or change.

5.5 Final Specification

5.5.1 Use cases

Using a use case diagram, all possible interactions with the system are visualized. On the left side all possible actors are shown, on the right right any interaction with external systems is shown.

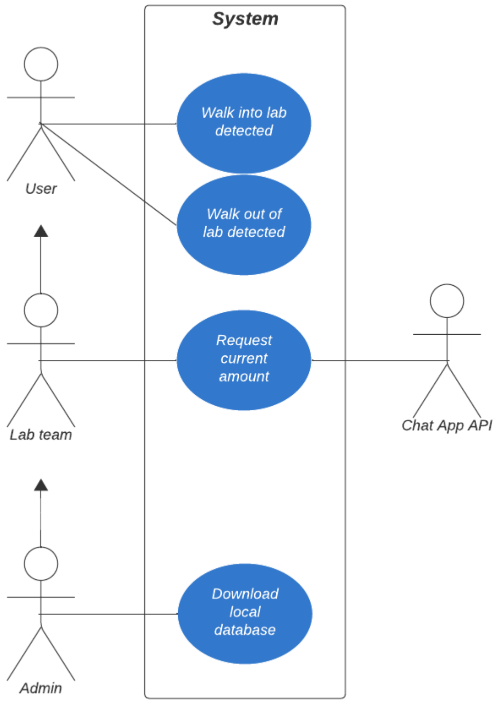

FIGURE 5.4: Use-case diagram

5.5.2 MoSCoW analysis specification

After additional specification, the MoSCoW analysis from the ideation phase is revisited and the scope for the realisation phase is defined.

Category	Requirement	Specification Status
Must	Measure occupancy for parts of the day.	Will measure 3 day parts
Must	Share this data in a dashboard	Will share data through the chat app. Will make data accessible through local database.
Must	Respect the privacy of users	Will not save any more information than is needed. Real time occupancy only shared in averages
Should	Limit the amount of stored data	Will save accumulated amounts in day parts.
Should	Not use a digital camera	Will use low resolution ToF sensor
Should	Collect in/output of users	Will measure change in occupancy
Could	Measure occupancy for different desks/zones.	Will not be realized.
Could	Have a request system	Will implement a request system
Could	Show the presence of staff members	Will not be realized

Table 5.2: Product Requirements MoSCoW Analysis

Chapter 6

Realisation

During the realization phase, the prototype is constructed according to the specified requirements and interactions. At this stage, the general approach chosen is further elaborated upon. Such as, the selection of brands, hardware and software, while simultaneously working on building the prototype. Various techniques are used to document the inner workings of the interactions, including pseudo code and process diagrams. Finally, some challenges are discussed.

6.1 Location and placement

To start this chapter it is important to know how the sensor will be placed since this can be used for the selection of a suitable sensor. In the specification, it was discussed why door zone sensing offers more privacy than other room-wide solutions. However, measuring the zone inside of a door can make the implementation harder, as the sensor has to be implemented into the door frame. Additionally, the algorithm would have to take into account the opening and closing of the door in the sensing field. To mitigate both of these issues, only the zone after a door will function as the measurement zone. In this zone there will be no objects, which offers a clean sensing area without the possibility of multi-path interference[38]. In subsubsection 2.1.2.2 of the background research ToF sensing is explained. When the IR light takes a different route back to the sensor this interference can start playing a role, as illustrated in Figure 6.1. The sensor will be placed top-down above an empty zone, attached to the ceiling, so the light reflects directly back to the sensor. Additionally, the sensor could be placed in different places when these requirements are met.

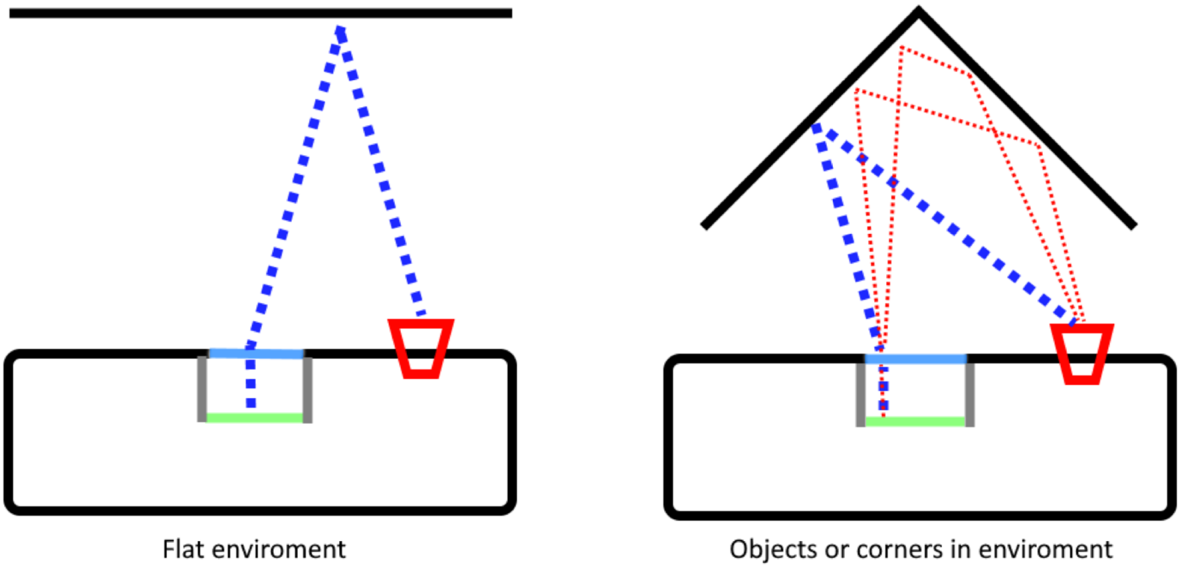

FIGURE 6.1: Illustration of ToF multipath interference. Wrong path shown by red line.

6.2 Platform selection

The Raspberry Pi 4 (RPi4) was selected for this project due to its versatility, affordability and performance. It has a fast processor and enough RAM to run a relatively light algorithm on the real-time data coming from the sensor and some of the other tasks. Additionally, WiFi is available to remotely access the interface, making live tweaking and debugging less labor intensive. The Raspberry Pi also has a good interface for ToF sensors using I2C communication, because of this many sensors can be directly attached. Also, it can be powered using standard 5v USB, while being widely available and open source. As a result, it is a perfect fit for a project in this category.[7] A 4GB RAM model was chosen due to it being available with the limited stock due to the chip shortage.

6.3 Sensor selection

For the sensor selection, low-resolution ToF cameras that can communicate using the I2C bus connection are considered. A market leader for accurate ToF sensors is STMicroelectronics, whos sensors are ported to the RPi4 by various brands. For this project, Pimoroni was chosen because they are able to ship quickly, and offer an affordable price.

To pick the correct sensor a couple of requirements are of importance. First, the range of the sensor must be suitable for the situation. When attached to the ceiling at a certain height, this height becomes a rough minimum as the detection zone is placed above the floor. Second, the sensor should have a suitable field of view. Finally, the minimum refresh rate should be calculated to get enough track data per visitor.

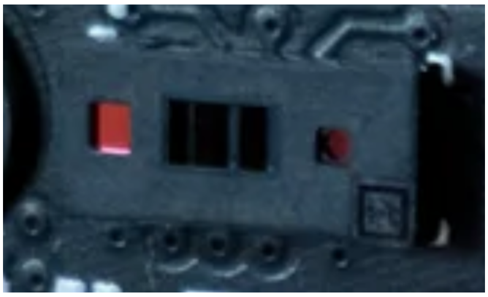

FIGURE 6.2: The VL53L5CX Time of Flight sensor.

The VL53L5CX, shown in Figure 6.2, by STMicroelectronics is the most current gen multi zone ToF sensor, which is also ported to RPi4 and widely available. The sensor supports a range of up to 4 meters, making it more then suitable to measure from the lab ceiling at 2.4m and additionally, making it suitable for less typical higher ceilings. The sensor supports a field of view of 65° diagonal [9]. This would result in a detection area of 45° in length, resulting in a 2^2 meter area at a height of 2.4 meters.

Additionally, the refresh/sample rate should be checked as sufficient data is required to make a blob track of a passing visitor. To control if this requirement is met, we will use a high human walking speed of 5.6km/h. This would result in a passing time of 1.33 seconds. The selected ToF sensor advertises a sample rate of 60hz. However, using the full 8*8 pixel array that we need for this project the sample rate is reduced to 15hz[9]. Resulting in 20 samples for a fast paced visitor. As a result, more samples are available then the 8 pixels in a row. Therefore, this is more than enough for the algorithm to keep track of multiple different blobs in the sensing zone.

6.4 Case design and mounting

To house both the RPi4 and the sensor, a case should be selected or designed. A couple of requirements are of importance. First, the case should allow for light cooling, which is sufficient for the low CPU usage required for the algorithm. A sufficient requirement is that the case should have openings for air to pass through. Consequently, this allows for cooling blocks to be added if needed. Second, the case should have an opening for the sensor to be placed in a top-down position. Third, all other ports on the RPi4 should be accessible for future development and integration's[1]. Finally a mounting system should allow for non-permanent ceiling mounting.

Starting with the case selections, a case that satisfies all these requirements and is able to house the specific sensor used is not available. In order to achieve a suitable design, a custom case was designed by tweaking an existing CAD RPi4 case design and 3D printing the result. The case in Figure 6.3 shows the selected design.

[1]This could include the audio jack for sound feedback; The HDMI port for displaying real-time data, offering data transparency; The LAN port to offer a more permanent integration with the network.

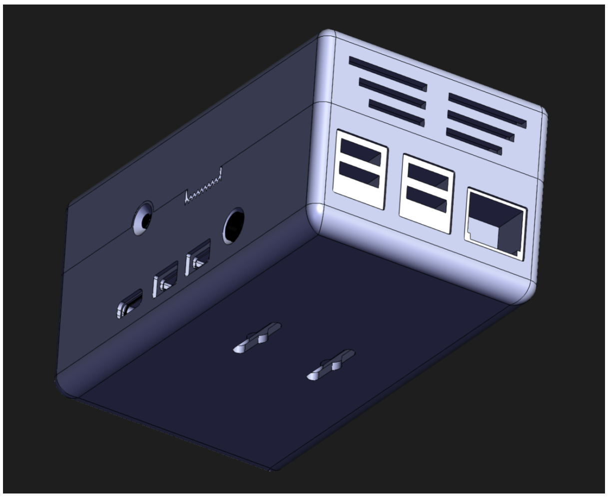

FIGURE 6.3: Original RPi4 case design by Mohamed Gamal on GrabCAD.com

The case offers cooling vents on both sides, access to all ports, mounting screws and sufficient room for the sensor mounting and cables.

We start editing the case by removing the existing ports that were present for an other HAT module. For the sensor mounting on the top part of the case, an outline of the sensor is created. Another sensor breakout board from Pimoroni was found on GrabCAD.com and the dimensions matched the measurements of the board we are using. The measurements of the sensor module on the breakout board were not published in the data sheet. As a result, an image of the breakout board was projected on the found board as illustrated in Figure 6.4 to arrive at the correct outline. Subsequently this outline can be projected onto the top part of the case to create the cutouts as illustrated in Figure 6.5, where the sensor can be seen behind the top-lid.

6.4.1 Temporary mounting

Even though mounting holes exist on the case design for more permanent mounting. For the development part of this project, being able to test at different locations and quickly have access to the device for debugging, an additional mounting method was sought. An new part going all around the case as illustrated in Figure 6.6 solves this problem. It can be taped to one of the ceiling strands, and allows the prototype to be easily attached and removed. Additionally, a tight fit allows the prototype to function without the required screws to keep the parts together. As a result, the device can be easily accessed during the development.

6.5 Software selection

To develop the system, software and libraries have to be chosen that comply with the needs. Python has been selected as programming language due to its many libraries and strengths when analyzing data. Additionally the advantages of Python being an easily learnable syntax and an

FIGURE 6.4: VL53L5CX Sensor Breakout Board by Pimoroni projected on the breakout board of the BME-680

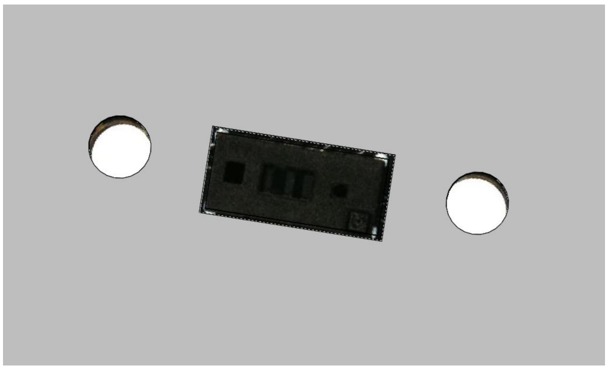

FIGURE 6.5: Cutouts in top part of case using outline of Pimoroni breakout board

extensive support for a wide range of libraries, facilitates the potential for future projects to build upon this system.

For the communication of the occupancy a chat app has to be chosen. For this interaction Telegram was selected. Even though Telegram is the second most downloaded chat app[2], Whatsapp whose usage is more popular, only allows development by large and registered companies. In order to use Telegram, a library has to be selected to support the communication with the Telegram API. Thus, the Telebot library was selected.

The Telebot library facilitates the communication with the Telegram API. The library is both streamlined and uncomplicated, allowing for less code and quick changes to the response handling. Additionally, the Threading library will be used to run the infinite polling of the Telegram API, parallel to the blob algorithmic loop. Keeping these processes separate supports stability of the overall system and ensures persistent monitoring and response capabilities for Telegram.

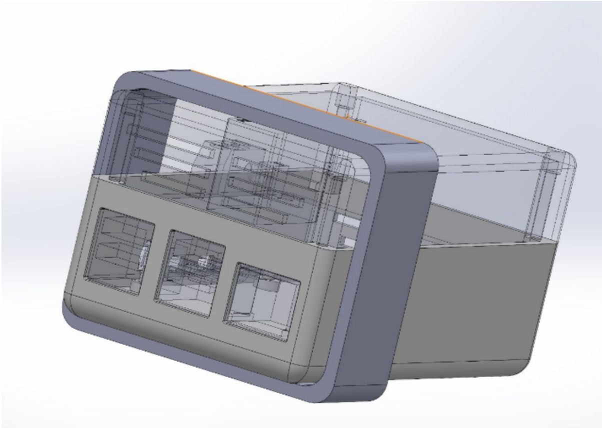

FIGURE 6.6: Case in Solidworks with mounting

For managing the local database, TinyDB has been selected for lightweight data storage within the system.

6.6 Software design

In this section the design of the software and detection algorithm will be documented. First, the interactions are documented in a rough sketch used to show the outline of the final system. These diagrams show the multiple systems and their workings inside of the installation, and the user interaction, which is always located in a top node. In a later section, the algorithm that is used for the detection of blobs is discussed using a choice diagram and pseudo-code.

6.6.1 Occupancy measurement, process outline

In Figure 6.7 the outline of the blob detection process is shown. When a user walks into the sensor zone located directly after the door of the room a new blob is detected and tracking starts, when the blob leaves the sensing zone, a conclusion is made based on the entry and exit position. This conclusion can confirm if a person entered or left the room. When the difference between this positions is not significant enough the blob is deleted and the visitor is not counted. When a visitor is confirmed, a new entry is made into the database.

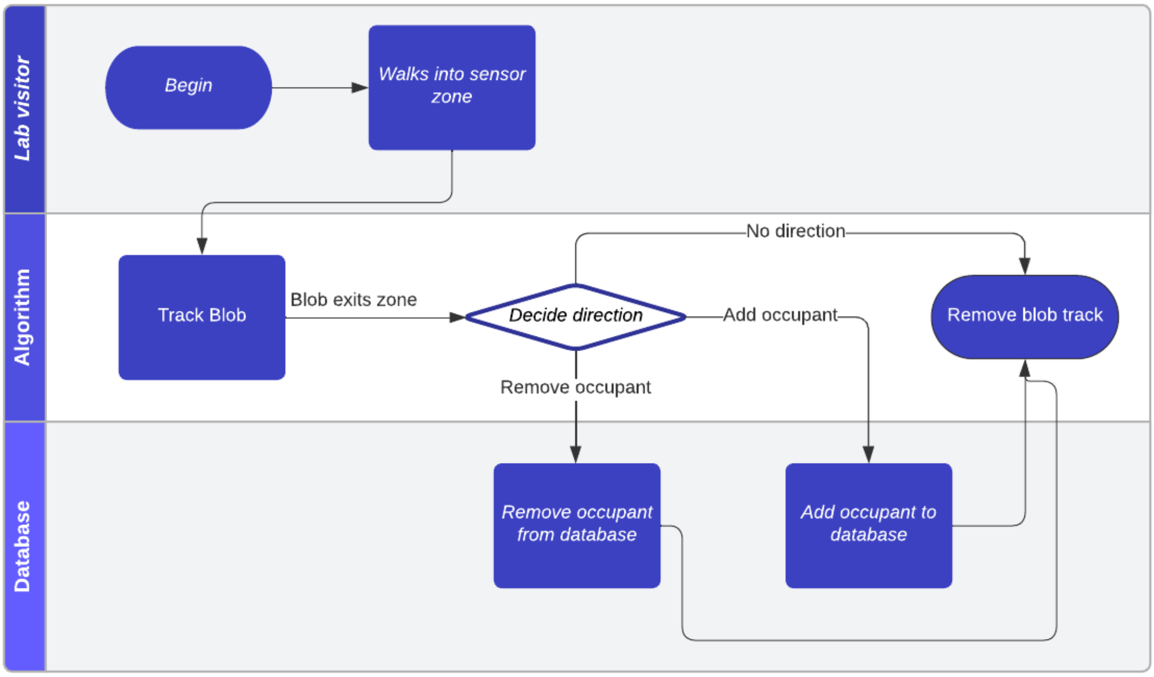

FIGURE 6.7: A process diagram outlining the process of a user entering the sensor zone

6.6.2 Requesting occupancy amount, process outline

In Figure 6.8, the process outline of the request interaction is shown. When a lab team or administrator user requests the amount of occupants inside the Telegram app, the API implementation calls a method. Next, the database handler class starts to count all the entries in the database. When completed, the amount is returned inside of a range to not share exact amounts.

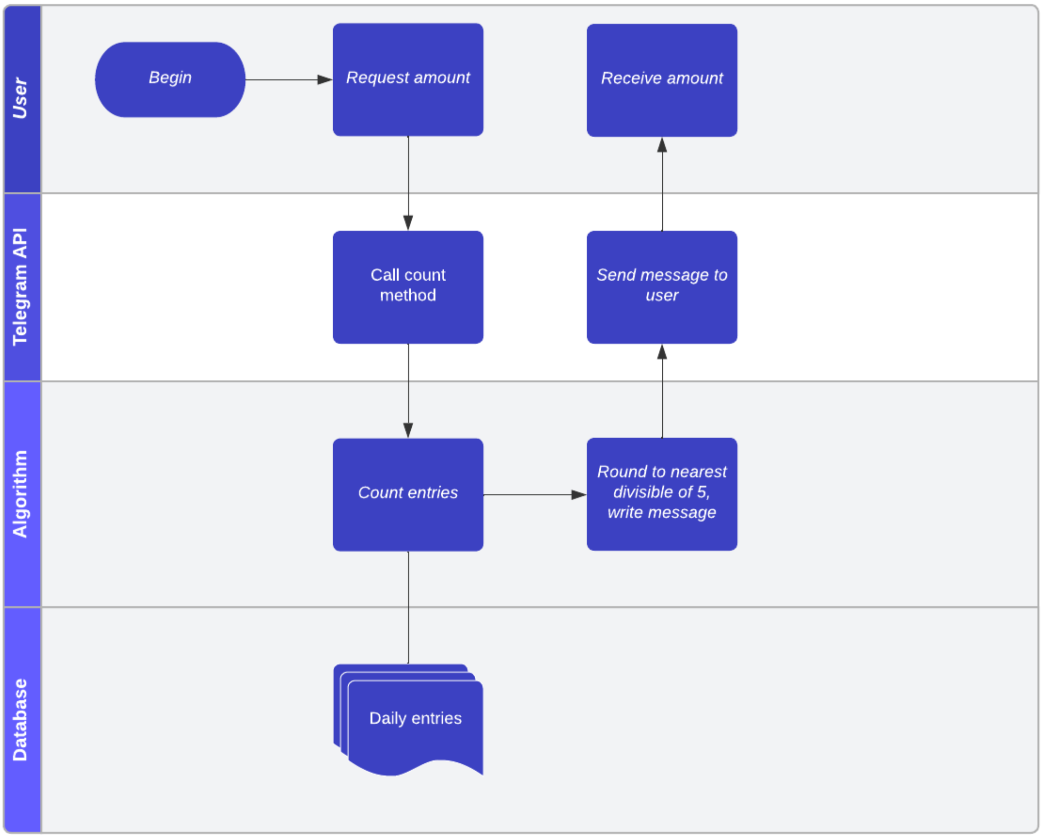

FIGURE 6.8: A process diagram outlining the process of a user requesting the real-time amount

6.6.3 Daily minimizing saved data, process outline

In Figure 6.9 the process outline of the daily data minimisation is shown. The process is automatically activated at night. First, it counts all the entries in the database and allocates them to day parts. Next, the day parts are saved to the long-term database. When completed, the local daily database is cleared.

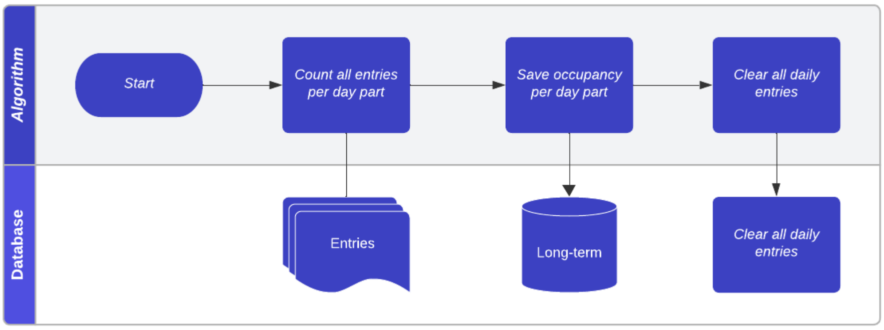

FIGURE 6.9: A process diagram outlining the process of the daily data minimization

6.7 Final Code

Using the software design and the system process diagrams, the code is realised. First, the blob algorithm shown in the pseudo-code and appendix is explained. The blob algorithm is used to detect the direction and keep track of multiple visitors. Then an UML diagram is used to show the other classes and additional explanations. Finally, in the next section, some of the challenges faced are discussed.

6.7.1 Blob algorithm explanation and pseudo-code

When a visitor enters the lab a blob track is created, based on the contour of the binary map from the sensors depth data. This starting position is stored inside of an instance of the blob class, shown in section A.5. When the visitor leaves the detection zone, the final location is send to the blob instance and the blob instance makes an entry to the database, and is then removed. To decide if a visitor is walking into or out of the lab, the first and last x position of the blob is compared. If this value covers more then half of the detection zone, a direction is chosen. Else, a stationary conclusion is drawn and no data is stored. The code made for the blob detection is explained further using pseudo code. To support the development of the algorithm, sensor data was mapped to add visual feedback. This code is shown in the GUI version of the algorithm in section A.4. Additionally, When multiple blobs are in the sensor zone a new ID is given. This is shown in Figure 6.10. This illustration also shows the binary data, where the white pixels are the middle of a detected blob with sufficient size.

Due to the size of contours not always being sufficient, some changes had to be made to make the algorithm more stable and therefore, accurate. One change was making the ToF sensor detect the closest peak in the light response, this is further explained in section 6.9.

Another change was how the contours are handled. Due to the size of a contour not being large enough in some mid frames, during a track, the contours are also appointed to a blob track. The last blob tracks position, remains known if the algorithm does not identify a contour. When the contour reappears, the blob track can continue by updating its position with the position off the corresponding contour. If the contour does not reappear, the visitor has probably left the zone and the blob is processed and removed as intended and explained above.
The following pseudo-code explains the blob algorithm in section A.3.

```
-Create an empty dictionary to keep track of blobs and their information.
(blob_mapping)
-Initialize a counter for blob IDs (blob_id_counter) to 1

-Repeat indefinitely:
    -if blob_id_counter exceeds 10, set it back to 1

    -if the ToF sensor has new data available:
        -Get the 8*8 ranging data from the ToF sensor API
        -Convert the data into a 2D array (arr)
        -Remove pixels that return an error reading

        -Create a binary mask based on the array values (binary_mask)
        -Convert the binary mask to a binary array (binary_arr)

        -Find contours in the binary array using cv2 (contours)

        -For each blob in blob_mapping:
            -If a blob hasn't been seen for more than 1.5 seconds,
            remove it
            -Otherwise, update the blob's position and timestamp

        -For each contour in the list of contours:
            -If the contour area is greater than 0:
                -Calculate the center of the contour

                -Check if the contour matches an existing blob:
                -For each blob in blob_mapping:
                    -Calculate the distance between the center of the
                    contour and the previous position of the blob
                    -If the distance is less than 3, consider it the
                    same blob and assign it as a matched_blob_id
```

```
    -If no matching blob is found:
        -Create a new blob with an ID (blob_id_counter)
        and update the counter
        -Add the new blob to blob_mapping

    -If a matching blob is found, update its position
    and timestamp

    -Increment blob_number by 1 and assign it the ID
    of the matched blob or the new blob
-Continue the loop
```

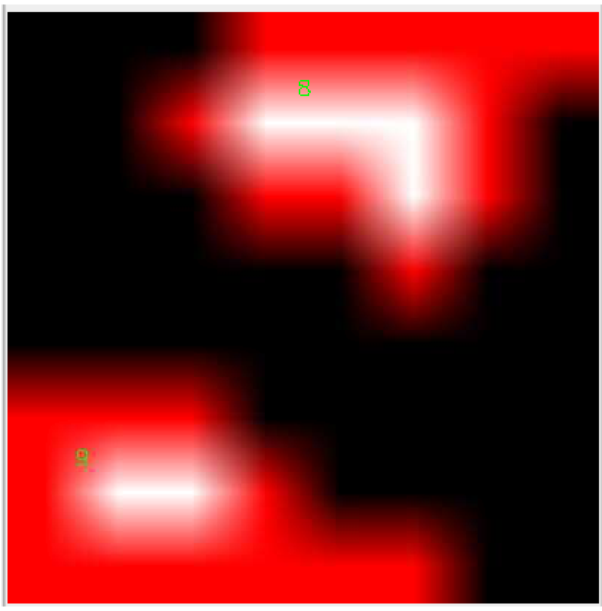

FIGURE 6.10: Binary map from sensor data.

6.7.2 Classes

Additional to the blob algorithm, code is written for the other classes shown in Figure 6.11. The final implemented code for all classes can be found in Appendix A. The Telegram handler class takes care of the communication with the Telegram API. The infinite polling process required for Telegram to respond promptly, is handled in a separate thread to keep these processes separate. The class can access the Database handler class in order to count the occupancy and communicate this back. Finally the blob class is responsible for comparing the starting and ending position of a visitor and uses the Database Handler to make an entry based on the result.

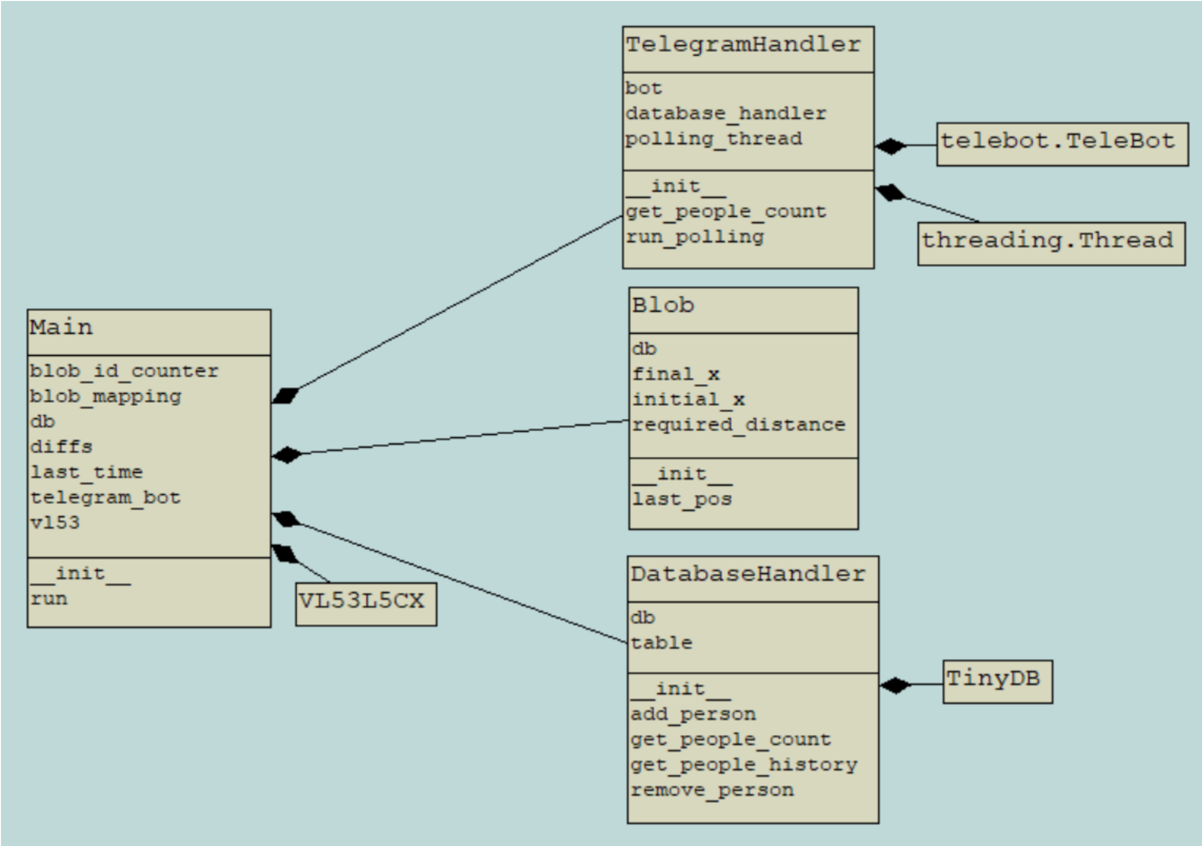

FIGURE 6.11: UML diagram of classes

6.7.3 Telegram implementation

The implementation of the Telegram API is as followed. A user of the Lab Team can send a 'request' command, which triggers a counting method in the database. The returned value is rounded to the nearest 5 and send back to the user, as shown in Figure 6.12. The code used for handling these requests and responses is shown in the appendix section A.2.

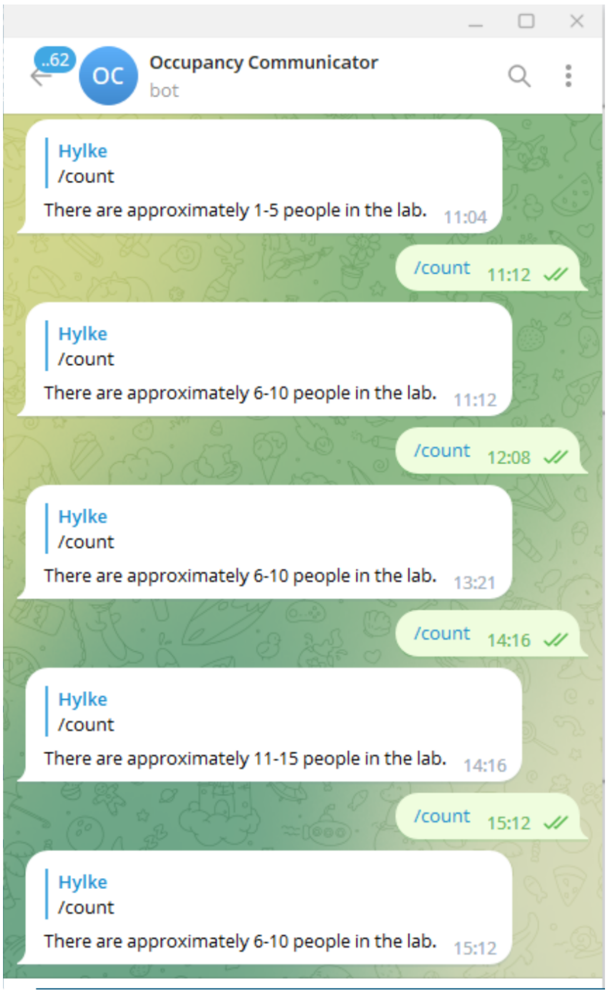

FIGURE 6.12: Telegram bot communication

6.8 Realisation of hardware

The case for the RPi4 was 3D printed together with the mounting system in Figure 6.13. When completed, the RPi4 was added. The ToF sensor came without any pins so a male to male header was soldered to the breakout board, allowing the sensor to be connected to the pins 1,3,5,7,9 on the RPi4. Where 1 and 9 are 3.3v and ground for the power supply. Additionally, 3 and 5 form the I2C bus where 3 (SDA) is for data and 5 (SCL) for the clock. Pin 7, is connected for the possibility to use an interrupt. However, for this implementation the code checks when the data is available once every loop. Finally m2 screws are used to attach the sensor to the lid in Figure 6.14 and closed prototype in Figure 6.15. Using the printed mount attached to the ceiling using easily removable electrical tape in Figure 6.16, the whole system is ready to be tested.

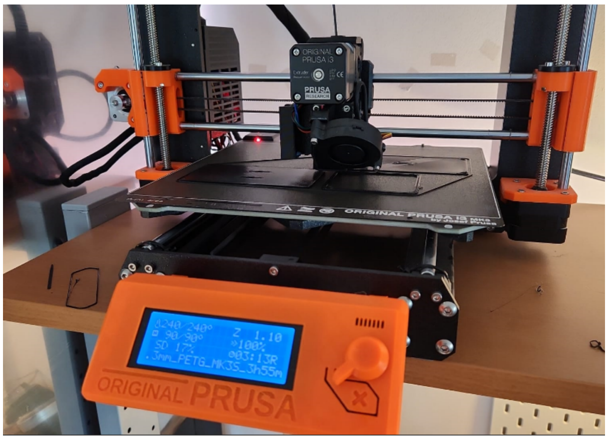

FIGURE 6.13: Printing the case on a Prusa 3D printer

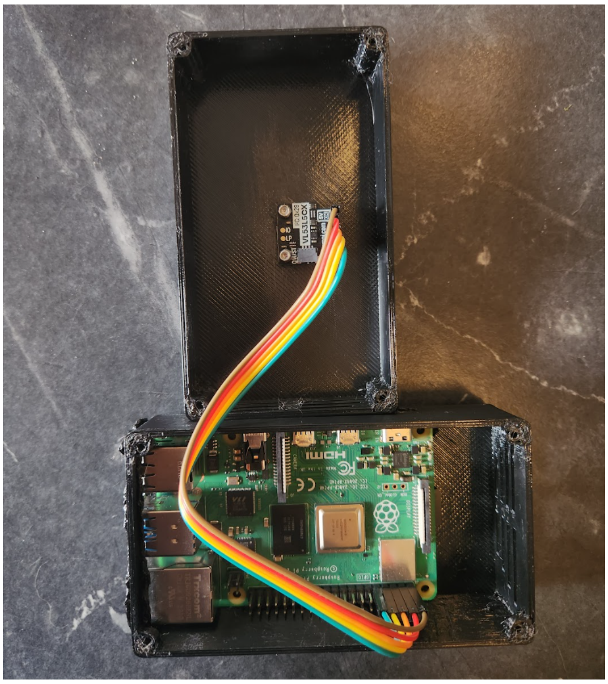

FIGURE 6.14: The complete system apart

FIGURE 6.15: The complete system

FIGURE 6.16: Case with temporary mounting attached to ceiling

6.9 Realisation considerations and challenges

During the realisation some challenges were met, they will be discussed in this section.

During testing it was discovered that visitors, in some cases only 1 or 2 pixels big, are not detected as expected. As the blob algorithm requires a minimal size to function, to mitigate errors of rogue pixels caused by reflections, a solution had to be found. During the initial research a quick calculation was done that a visitor will always be stretched over a couple of pixels, so the problem wasn't the low resolution, field of view or height. Looking further into the issue it was discovered that this was caused by the higher amplitude of reflection of IR light coming from the floor, instead of the side of a visitor, as shown in Figure 6.18. For example, an arm swinging through one pixel might reflect some light, but not enough to be seen as a target. The amplitude signal response is shown in Figure 6.17. In the driver guide [8], a setting was found that could tweak this behaviour and the 'Target Order' was changed from the default 'strongest' to the 'closest' peak in the graph. This solution worked. However, a new problem of false positives was introduced where pixels sometimes reported back an object at 1-5cm distance for 1 frame. This was resolved by always setting pixels that reported back a value lower then 5 to a 0 in the binary array. As a result, clean reliable data goes into the detection algorithm.

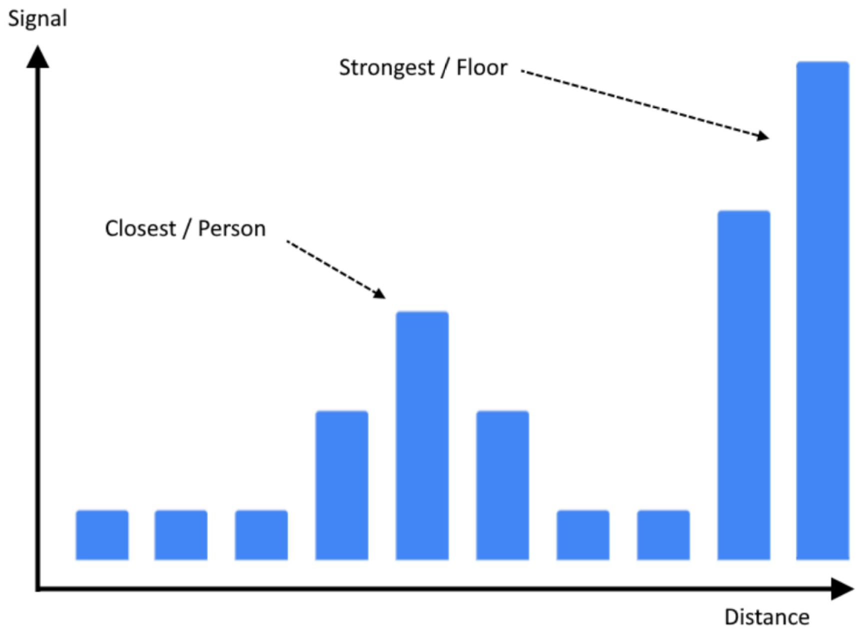

FIGURE 6.17: Illustration of ToF returning signal amplitude.

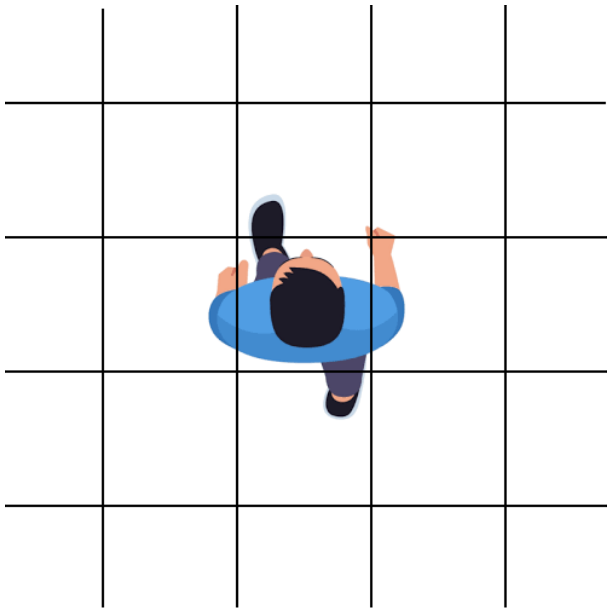

Figure 6.18: Person walking below sensors grid. Illustration is cropped, actual grid size is 8x8

Another issue was with an unstable sample rate. The sensor was set to being used at 15hz as originally found in the datasheet during research for the realisation. The higher sample rate was needed to reliably match blobs from frame to frame. As discussed in section 6.3. If a visitors blob moves to far from the original location in the frame before, the amount allowed should be raised. This could result in matched IDs from other blobs to be allocated to another person in the same frame. By adding a sample rate counter to the code, the rate was confirmed anywhere between 3-7hz, which is very unreliable and unacceptable for the project. In order to debug, many approaches had been taken.

We started with all visualizers, used for the programming of the blob algorithm and debugging, to be removed from the code. However, even only loading the data from the sensor gave the same results. The RPi4 I2C bus was clocked at 1MHz, the maximum allowed for the ToF sensor, this was reduced to investigate if the bus data was saturated, only giving results when lowering further then 400KHz. This put the attention on the sensor itself. Even the demo video on the project page from Pimoroni, with example code using the 15hz sample rate, was not running at this sample rate. After some additional research into the original API in C, behind the Python wrapper, it was discovered that other not needed data was also being shared. After forking the Python library, the back-end C firmware was adjusted, shown in Figure 6.19. As a result, data could be collected at 17-20hz and the findings were also shared with the Pimoroni team.

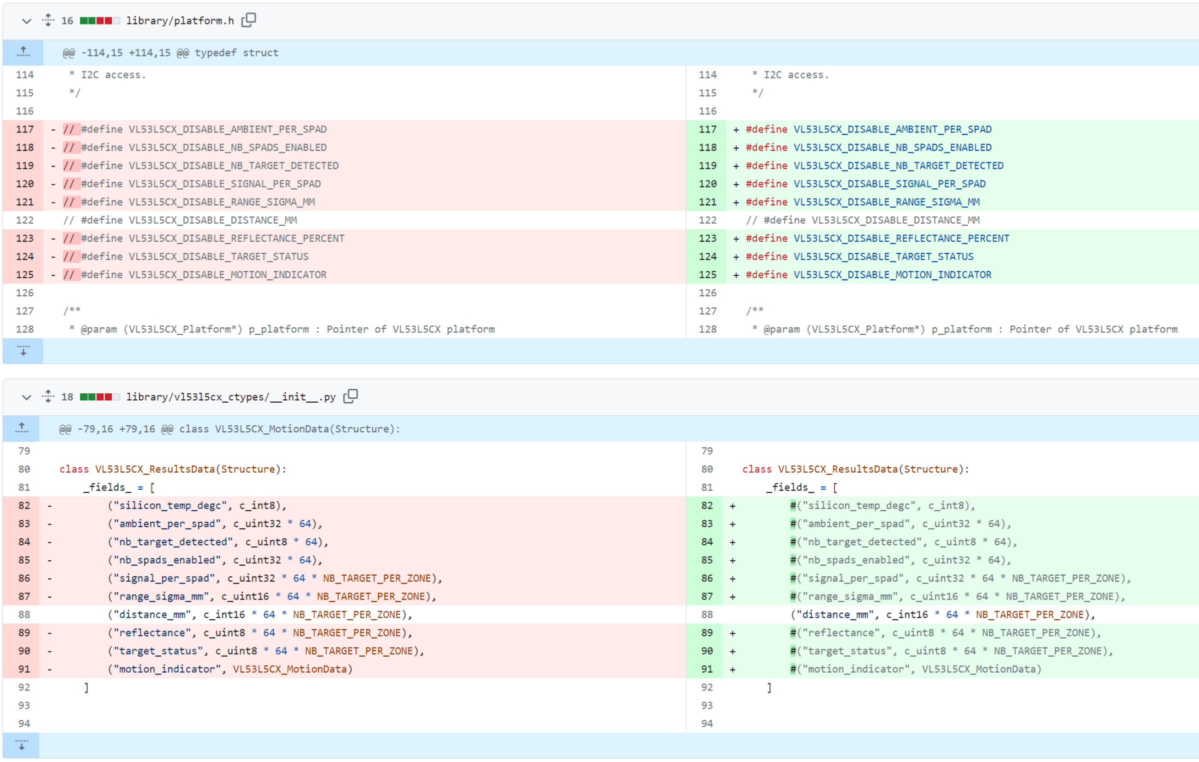

FIGURE 6.19: Forked adjusted firmware inside Python wrapper

Chapter 7

Evaluation

The final system described in chapter 6 will be evaluated. First, a requirement evaluation will be conducted. Second, a performance evaluation will be done, comparing the sensors observation with a ground truth. Additionally, the performance is also evaluated by putting the system in unique situations. Third, a user evaluation is done to complete the evaluation. The user evaluation is both interested in the lab teams experience with the request system and how users of the lab experience the presence of the system.

7.1 Requirement evaluation

The requirement evaluation will be performed by evaluating the requirements listed in the MoSCoW analysis (Table 5.2) concluded the specification chapter. The following requirements are evaluated in Table 7.1 and further discussed in this section.

Category	Requirement	Evaluation
Must	Measure occupancy for parts of the day.	Measures occupancy data all day.
Must	Share this data in a dashboard	Current occupancy can be requested using a chat app. Long-term data is available.
Must	Respect the privacy of users	Multiple user privacy requirements have been met.
Should	Limit the amount of stored data	The occupancy data does not contain any personal data and is only saved in day parts long-term.
Should	Not use a digital camera	Does not use a camera.
Could	Measure occupancy for different desks/zones.	Only measures in a zone at entrance/exit to prioritise privacy.
Could	Have a request system	Has a Request system using chat app.
Could	Show the presence of staff members	No, does not differentiate using personal features to respect privacy.
Wont	Host the data trough an API	No API, but local database could be used for further development of applications.

Table 7.1: MoSCoW Analysis Evaluation

7.1.1 Must - Measure occupancy for parts of the day.

The first requirement was given by the client and served as a minimal data resolution needed for the project to be useful. The final product is implemented measuring all traffic entering and exiting the lab room. This data is then used so the current occupancy can be shared with the lab team. At the end of the day the data is saved into the required day parts and all additional data removed. As a result, the minimal requirement is satisfied with the long-term data, while prioritizing user privacy by removing all excess occupancy data.

7.1.2 Must - Share this data in a dashboard. Could - Have a request system

The second requirement was given by the client so that the collected data can be viewed. Additionally, the lab team could use the data on the dashboard to look at the current occupancy. Some of the drawbacks of making a typical dashboard are the hard to implement authentication and the requirement for the user to open a website of some sort, these reasons have also been discussed in subsection 4.5.1. In the final product the long-term JSON data can be downloaded from the local storage. Additionally, the current occupancy is made available using the request based chat app.

7.1.3 Must - Respect the privacy of users. Should - Limit the amount of stored data. Should - Not use a digital camera.

These requirements were motivated both by the original client requirement and by the first user interviews. The user privacy was a priority during the product development and has been implemented in different considerations. First of all on the hardware side, the sensor is only able to see a 8*8 pixel data array. This is sufficient for multiple blob detection while not collecting more information then needed. Additionally, The sensor is not able to see an image, nor color. Even thought the sensor does see depth, the resolution is not high enough to clearly identify a person and the depth data is not recorded. Furthermore, the placement of the sensor allows the system to only record the movement of a visitor during leaving and entering the room. As a result, a visitor is not observed when in the room. Additionally, the software considerations are that no personal data is saved to a database entry, other than the direction travelled which data is only saved locally. Also, after the day ends, the data is removed and saved to a long-term database for the occupancy during day parts.

7.1.4 Concluding remarks

In conclusion, the final system described in Chapter 6 satisfied all the "Must" and "Should" requirements, addressing the key functionality set out in the MoSCoW analysis, which is based on the three different stakeholders. The system has successfully met the "Must" requirements, which included measuring occupancy for parts of the day, sharing occupancy data, and the multiple elements of respecting the privacy of users. The "Could" requirements for a request system was also implemented, and became a primary interaction, as it is able to share data in a simple and transparent way. The "Could" requirement for measuring occupancy in multiple zones and the detection of when staff is present, were not implemented to prioritise privacy. Overall, the requirement evaluation indicates that the system successfully addresses the most important requirements from the stakeholders offering a privacy friendly solution for collecting occupancy data.

7.2 Performance Evaluation

To ensure the reliability and effectiveness of the system, a performance evaluation will be conducted. First the plan of the performance evaluation will be outlined. Then, the setup and approach is documented. Finally, the results for the performance test will be shared and discussed.

7.2.1 Setup for Performance Evaluation

To conduct the performance evaluation, the final prototype had to be installed in the lab. An important consideration for the evaluation is the right for a user to not participate in the tests.

For this reason, an opt-out lane is created and the sensor can not be installed directly after the door as intended. As a result, the zone is placed more into the room as can be seen in Figure 7.2. The two walls flanking the zone are intended to guide visitors through it, emulating the behaviour of a visitor as if the sensor were positioned post-door. To help users understand what is being tested and the possibility to opt-out, a sign (Appendix D) is shown on both sides, as shown in Figure 7.1.

FIGURE 7.1: Sign at door for performance test.

FIGURE 7.2: Testing zone for performance test.

7.2.2 Plan for Performance Evaluation

To test the system on performance the following tests are conducted.

Normal Operation Test During this test, the system is implemented and observed during normal operation. The systems observations are then compared with a ground truth which will be observed by a human observer. Depending on the number of users entering and exiting the lab, and the consistency of the accuracy, the time needed to evaluate will be decided. To compare the product observations to a ground truth the local database is compared with the observations of the researcher. To collect this data, a Python script (section A.7) is written to note the timestamps in a database, so the two can be compared.

Unique Situations Test During this test, the system is put through some unique situations to evaluate if the system is able to withstand more complicated situations. The following situations, indicated by the client as typical for the lab space, will be observed and discussed.

1. Multiple people entering at the same time (in frame simultaneously).
2. People entering and leaving at the same time.
3. People stationary

4. People entering while other person is stationary.

5. People entering with boxes and chairs.

7.2.3 Performance evaluation

During the performance evaluation days, the lab did not have enough visitors to collect the required useful data. Due to the time constraints, it was decided that the normal operation test will be performed by walking through the sensor field continuously with a group of students. In a total time of 5 minutes the product was tested by 3 people with a total of 50 samples. The participants were asked to walk in random directions, angles and at random times. Thus, the product can not accidentally be evaluated in ideal circumstances. However, the unique situations are purposely mitigated and tested separately.

After completing the first evaluations the data was recorded and plotted in Figure 7.3. The blue line shows the data collected by the products local database, every visitor that entered or left is added to the total sum and a point is created. The orange line shows the ground truth. After a couple of samples the system misses a visitor that entered the room. As a result, the total occupancy is off by 1. Further in the samples another error is made with a visitor leaving the room, bringing the total back to the ground truth. At the end of the test the ground truth and occupancy are concluded equal. To conclude, some remarks can be made about this experiment. First, in addition to this test, no false positives were ever observed. Second, the accuracy of a single detection is around 96%, and there does not seem to be a preference for direction. Third, when a visitor is not detected, the total occupancy deviates from the actual occupancy. As the errors do not seem to prefer a direction, the average will remain accurate. However, the error margin of the total sum grows throughout the day.

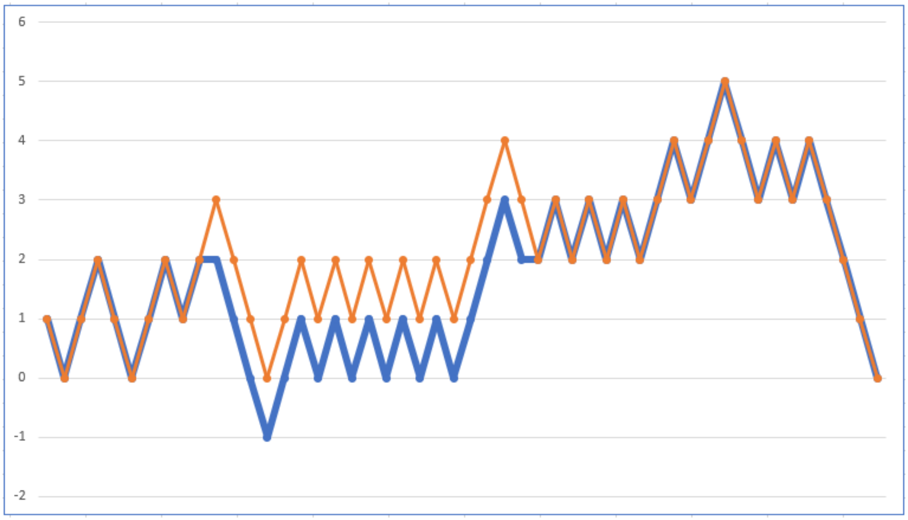

FIGURE 7.3: Performance evaluation. Blue line is the result, orange the ground truth.

For testing the unique situations a similar approach is used where the possible situations are simulated by participants. However, the collected data will not be evaluated on a quantitative basis, but analysed by reviewing the sensor data and making remarks about what causes certain behaviour. These remarks can be used for the discussion to advise on future development and are shown in Table 7.2.

Situation	Requirement
Multiple people entering at the same time	When multiple people enter the frame in the same direction the system detects these. However, while testing is was noticed that when the participants come to close to each other, they are detected as 1 blob, and only 1 person is registered. Nonetheless, the door frame is not big enough to allow for parallel entering, and visitors tend to not walk as close to a person compared to the evaluation.
People entering and leaving at the same time.	For this test again, the results are accurate when the blobs do not collide. When they do, the system may register a stationary blob, as the total distance needed to register is not achieved.
People stationary	When a participant walking into the observed zone and then remains stationary, the system will keep track of the created blob for as long as the participant stays. When the participant leaves the zone the same way they entered, the track will be removed as intended.
People entering while other person is stationary.	For this test the same conclusions can be drawn, when the blobs are not close to each other the system can differentiate the blobs and performs as expected.
People entering with boxes and chairs.	This test gave perfect results. As the system is already very accurate in normal operations, it is hard to make a remark about the influence of a person holding chairs and boxes. However, it could be expected that more area to track, makes for more accurate results.

TABLE 7.2: Unique situation evaluation

7.2.4 Concluding remarks

In the performance evaluation, the system's functionality and reliability were assessed. Two primary tests were conducted: the Normal Operation Test and the Unique Situations Test. In the Normal Operation Test, the system demonstrated an high average accuracy of around 96%. It effectively tracked user movements. However, did miss some detections, leading to deviations in the total calculated occupancy. As the total occupancy is calculated based only on change, the error margin increases over time. However, as roughly demonstrated by the normal operation test, the errors do not prefer direction and have the potential to average out during the day. Additionally, after the day is completed the short-term data is removed and total occupancy starts at zero. As a result, the collected data is a sufficient representation of the total occupancy and is a positive sign for the successful implementation of the product. The Unique Situations Test examined scenarios like simultaneous entries, entries and exits at the same time, stationary individuals, and entries with objects. While the system generally performed well, it struggled when multiple individuals came too close together, resulting in single detections. In reality many of these situations would not become an issue, but it does leave room for improvement.

In summary, the system effectively tracks visitor traffic in a lab environment, with some minor limitations.

7.3 User evaluation

This section provides an overview of the user evaluation process.

Method First, potential participants are invited to take part in the user evaluation. The participants are recruited in the lab space and are known to be users of the lab, consisting mostly of Creative and Interaction Technology students. The potential participants are then provided with an information brochure, shown in Appendix E, containing details about the evaluation's objectives, what the evaluation entails, and the right to opt out at anytime. This step is to ensure participants have a clear understanding of what is expected. Subsequently, participants are requested to sign a consent form, shown in Appendix F, as an affirmation of their willingness to engage in the evaluation and their acknowledgment of the specifics. The interviews will be setup as semi-structured, where set questions are asked, with the option to explore further on the theme. Users who are not lab team members, are not able to directly interact with the prototype. As a result, the user evaluation is divided into a lab team part, and a user part. Due to this the specifics will be documented separate for both the lab team and lab users. For both interviews the key findings and conclusions of the open conversation are noted down by the researcher per asked question. Finally, the participant is thanked for the participation and given the opportunity to review the notes taken.

Lab Team approach The lab team participants will be asked to use the chat bot to request the current occupancy. When the participant is satisfied and understands the workings, a conversation is held using a couple of questions. The primary focus is to find out how the lab team experiences the interaction. Additionally, the participant is asked to comment on the method used to communicate this data. Finally, another question is asked on how the system is used, to find out if the lab team can already see themselves using the system. The following questions are asked.

- Can you comment on the interaction with the chatbot?
- Can you comment on how the data is communicated from the system?
- Can you comment on how the system will be used?

Lab Users approach The user evaluation is primarily focused on how the user experiences the presence of the occupancy system. During this conversation some questions are used to steer the focus. First, participants are asked to comment on the presence of this system. The goal of this question is to start a conversation on the privacy aspects. Also, a question is asked about possible improvements regarding, but not limited to, privacy concerns. In the final question users are asked how the collection of occupancy data impacts them. This question is asked to gain insights in how the user weights the collection of occupancy data against being observed by the device. Due to there not being a direct benefit of this system for users this is an important aspect. The following questions will be asked.

- Can you comment on the presence of the room occupancy system?
- How could the implementation be improved?
- The client can now collect occupancy data, how would that impact you?

Analysis In contrast to the unstructured interviews from the ideation phase, the semi-structured evaluation interview will give more specified answers per theme related to the question. The overall findings, key insights and possible recommendations will be summarized per question.

7.3.1 Lab-Team evaluation results

In the interviews with the lab team, there were two separate participants, and several take-aways emerge regarding their interaction with the chatbot and their thoughts on the system's implementation and data communication:

Interaction with Chatbot: Both users expressed that the chat app was easy and clear to interact with. User 1 suggests using WhatsApp for the chatbot interaction as it is more popular, while User 2 is satisfied with Telegram. User 2 appreciates the request-based interaction as it is very simple and quick. They both express interest in receiving notifications when large amount of people visit the lab.

Data Communication: User 2 comments on the usage of Telegram bringing work life into personal life. However, due to the system only working with requests, it is not a huge problem. The user advises on implementing into Microsoft Teams, as this is used for communication by the lab team. Additionally, User 2 suggests a dashboard for more widespread use. User 1 finds the data communication good. However, would like to also know when the last 2 people leave the lab, as the chat app will only show "0 to 5 people".

How the system will be used: User 1, expresses interest in sharing the chat app with other users to know the current occupancy but raises concerns about students accessing this information. Both lab team members express the usefulness of the system, especially when downstairs and not knowing the current status of the lab room upstairs. Additionally, user 2 expresses that using the data to look back at events is good, but asks why a simple head count wouldn't be sufficient.

7.3.1.1 Concluding remarks

The advice of switching to Whatsapp was considered and discussed in the realisation chapter. Additional to the mandatory business registration, the limitations of the API also form an issue. To make the system open and accessible for future developments, the Telegram API was chosen to offer more possibilities. User 1, advised on communicating the actual occupancy through Telegram. However, earlier in the development, this was discussed with the ethics commission and chosen to only communicate a range to further respect user privacy. To conclude, both users from the lab team were overall positive regarding the chatbot interaction and the usefulness of the occupancy tracking system. Also, the lab team evaluation gave some insights and some points to research with future development.

7.3.2 User evaluation results

In the interviews with users, there were three separate participants. Participants are allowed to ask questions and discuss how the system collects data and overall working. Several takeaways emerge after a conversation about the implementation:

Can you comment on the presence of the room occupancy system? Participants generally had positive perceptions of the room occupancy system's presence. User 1 expressed concerns about privacy as a student but saw its potential value for lab team members. They appreciated the system's non-intrusive nature once its operation was clarified and after reading the information brochure located at the door (Appendix D). User 3 desired a way to view the occupancy data for convenience of knowing when the lab room was busy, and had no privacy concerns, while appreciating the non-camera approach.

How could the implementation be improved? In terms of improving the system's implementation, User 1 thought it was "an amazing method" after discussing using a ToF sensor, but recommended additional transparency, allowing users in the room to see the data being collected. The user mentioned making the raw sensor data visual on a screen or show what the sensor can see on a sign. User 2 suggested the inclusion of a dashboard for all users. User 3 expressed a need for a screen downstairs to view the occupancy data. Additionally, showing if the room is currently being used for a class by showing this outside before entering could be useful.

The client can now collect occupancy data, how would that impact you? The personal impact of the client's ability to collect occupancy data varied among participants. User 1 recognized the potential usefulness of the data, particularly for lab team members, and appreciated the provided information on the information sign. User 2 questioned whether the system was actually needed as staff has an idea of when people use the lab.

7.3.2.1 Concluding remarks

Some points were made during the user interviews about offering additional transparency. Suggestions were made to show the occupancy on a screen. These screens could be located downstairs to show the current status of the lab room, or at the door to give an indication about a potential class, or in the lab room itself in order to give data transparency. Also, the view of the sensor could be shown on a screen or sign to further demonstrate the limited invasiveness of the device. Giving users access to the chat app may become an authentication challenge. However, offering this information on a public screen could be a useful addition to the lab space. To conclude, all users expressed a positive attitude towards the presence of the system. Also, they all understood the clients need for this device and data collection, with one user expressing some doubt about the actual need for such data. These interviews, in addition to those with the lab team, have also given extra insights into possible future developments and are both a positive sign for a successfully implementation.

Chapter 8

Discussion

In addition to the evaluation of the system and the insights gained from the user evaluation, this chapter will foster a broader perspective by reflecting on the project as a whole. Some themes throughout the project will be discussed by sharing the findings, limitations and making possible recommendations for future projects.

8.0.1 Privacy

From the moment the data is collected to when it is shared or saved, privacy is the priority. The device only observes a small zone at the entrance, which is sufficient for collecting occupancy data as the performance evaluation has confirmed. The goal of a private design is also served by using a low-res sensor which prevents the collection of excess data, only saving in day parts and by processing all data locally. Some of these choices are reflected on in this section.

Privacy levels In the literature review, the findings on privacy were split in 3 levels. Level 1, which indicated the amount of raw data collected by the sensor, would function as the primary goal for this project and for finding a suitable sensor. Level 2 covered the privacy in transmission from a sensor node to a the controller. Finally, level 3 covered the privacy of data storage. The discussion on privacy will use these three levels.

Level 1 - Sensor selection Literature revealed the weaknesses and strengths of many different technologies that can be used for occupancy tracking. In combination with state-of-the-art research into sensors, currently used on the campus of the University of Twente, the different technologies were mapped in a table based on the sensor accuracy and hardware privacy factors. From the table it could be concluded that technologies that have the ability to offer accurate data, are less private by nature when looking at the hardware capabilities.

The use of a time-of-flight sensor with an 8*8 pixel array, was not the most privacy friendly hardware that could have been selected. This is due to the sensor being able to detect how tall an user is to a quite accurate degree. As a result the length of a stay with the height of a person could be identified, including the person's behaviour over a longer periods. In more busy environments using this variable for tracking would become less of an invasion of privacy, as many visitors with the same height will be identified.[1] Using two lasers or other duo systems, as discussed in the realisation, would have offered a more private hardware design. However, using a two variable system would not handle multiple visitors correctly, as both sensors would activate concurrently, it becomes difficult to distinguish order. When using a two motion sensor setup (PIR), this problem becomes less of a problem because direction can be detected based on the increase in motion, but overall accuracy remains low and challenging[32]. It can be assumed that enterprise brands, as seen in the State of the Art research, use cameras for this exact reason, as they offer very high accuracy[30]. This project tried to solve the issue at hand, by taking a modern approach without the usage of high resolution imaging, nor the possibility of the usage thereof. The blob detection algorithm functioned well and is able to handle these situations with concurrent visitors, achieving a high accuracy compared to more invasive methods, while retaining user privacy.

Level 2 - Data handling When the raw data is captured by the ToF sensor, it is processed internally by a micro-controller. To lower both the bandwidth and retain privacy, the firmware was adjusted to only send the required depth data. Additional data, including reflectance of the subject and motion indicaters are not transmitted on the I2C bus. Moreover, the RPi4 code directly converts the depth data into a binary map. A recommendation is to further adjust the sensor firmware to only transmit a binary map, so the RPi4, which is connected to the internet, would never be able to access the more detailed depth data, further protecting user data. Additionally, to prevent transmission of sensitive data when transmitting data through telegram, the current occupancy is not shared in an exact amount. Level 2 data handling also becomes very important with multiple nodes, this will be discussed in subsection 8.0.4.

Level 3 - Data storage One limitation is the short-term occupancy data, which is not saved in a less detailed approach, such as with the long-term day parts. This was done to support the counting for the chat app, and for the code that minimizes the data at the end of the day. Although these data can only be accessed by accessing the device directly, it does not excuse that the data is available. Due to the time frame of this project, finding a suitable solution was not completed. A solution would be to aggregate changes, and save these totals without taking note of a timestamp in the database. Such an implementation should not alter any of the current workings.

8.0.2 Blob detection algorithm and accuracy

During the performance evaluation, some unintended behaviour was found in more unique situations when multiple blobs overlapped. The result of the algorithm calculating the middle of a contour, issues arise when two blobs form one component. Having a better algorithm for

[1] Depth data (which could indentify height) is not saved nor is it used by the algorithm. The 2d depth array is converted to binary. This paragraph is only about hardware capabilities.

detecting visitors and their walking direction would allow the system to also perform with a higher accuracy in more complex situations, and as a result, be more suitable for places with a high number of visitors. One recommendation for future adaptations would be to make the algorithm remember that two blobs merged, and force it to keep the newly formed component as two visitors. Another solution would be to remove an outer bound from the blob and use a point closer to the center for contour detection, as used in many existing methods [33]. The accuracy achieved with the current algorithm and sensor combination holds up well against more invasive solutions. In the found literature, a system that uses a camera in an office room achieved a accuracy of 97.32% in [30]. Additionally, another paper achieved an accuracy of 95% using a high-res ToF camera and AI model[35]. In comparison, the final prototype achieved a similar accuracy of 96% in the evaluation, while using a minimal amount of data nor AI model, without the limitations of a two point system.

8.0.3 Evaluation Participants

For the performance evaluation, participants were limited. This is due to the tower room lab space being relatively new and not widely used. As a result, during the performance evaluation the required data could not be collected in the time frame set. Therefore, the evaluation had to be done with a small group of people simulating typical behaviour. A discussion could be held if this would influence the performance results and if testing in a normal situation would have led to different results. First, the group was representative as typical aged students from Creative and Interaction technology. Second, the binary map used by the algorithm does not rely on differences in length. Finally, during the realisation similar stable tracking was observed in a different setting. It can be concluded that the performance test accuracy is representative of the actual accuracy.

8.0.4 Rooms with multiple doors

The interaction lab where this system will be implemented only has one door, allowing all measurements to be done by the integrated system. If the system would be implemented in a room with multiple doors, an additional prototype could be used with the same blob detection algorithm. To facilitate easy expansion, an external database is used which can be interacted with from a separate class. When setting up a second device the blob detection algorithm could signal the main device using USB serial, or an encrypted network protocol. If a network protocol is used, privacy factors should be reconsidered with the taken approach. For example, an encrypted data connection could still be monitored by an adversary that would be able to detect the difference between a person entering and leaving the lab based on packet size, or know that there is movement due to this packet being send. One solution would be to encrypt in such a way that the size of the packet does not relate to the content, a common practice in sensor node networks[14]. The problem of detecting the visitors timestamp by only observing when a packet is send, can be mitigated by either: sending random packages to hide the actual data in a flood of other data or, the data could be aggregated on the node until it is required by the main node.

8.0.5 Evaluation recommendation

In the evaluation some lab users expressed that they would appreciate having access to the occupancy data through screens. This again would require carefull consideration of the privacy factors. A recommendation that could be made for data safety, is to use the RPi4 to render an image for a screen that is directly attached, not requiring sharing data over an internet connection. Additionally, to not share more data then needed, it can be mentioned that users are mostly interested in how busy the lab is. Displaying a little clock without any numbers could be a way to communicate only the neccecary information, as seen in Figure 8.1.

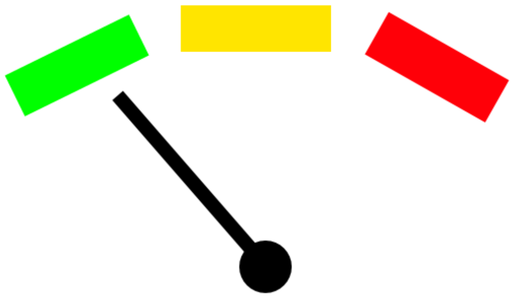

FIGURE 8.1: Current occupancy dashboard

The lab team also mentioned the idea of receiving notifications when a large group enters. This would help them to quickly send a team member to support new visitors that may need something. Due to time constraints this element was not implemented.

Chapter 9

Conclusion

In this chapter, the conclusion is made for this bachelor thesis aimed at building a privacy-friendly occupancy measurement system for the Interaction Lab at the University of Twente. The main research question will be answered. Additionally, a recommendation is made for future research.

> "To what extent can accurate occupancy data be collected in the Interaction lab when prioritizing user privacy"
>
> Main Research Question

The literature research showed that technologies capable of precise data collection compromise privacy due to their hardware capabilities. Interviews with lab users and the lab team confirmed a shared commitment to safeguarding privacy. After an ideation and specification phase, the requirements for the prototype were used to build the final system.

The prototype developed, consists of a Time-of-Flight sensor connected to a Raspberry Pi 4 concealed in a custom case. The device is mounted with a top-down view observing a 2*2 meter zone at a sample rate of 20Hz. The sensor uses a low resolution 8*8 pixel array to communicate depth data to the RPi4. The small observation zone and low resolution both contribute to respecting user privacy by hardware design. To further protect user data, long-term data is aggregated into day-parts. Additionally, the current occupancy is shared with the Lab Team through the Telegram chat app.

The evaluation phase tested and evaluated the final prototype. Three evaluations provide a comprehensive overview of the system. The Requirement Evaluation confirms that the system effectively fulfills the key stakeholders' "Must" and "Should" requirements, offering a privacy-friendly solution for collecting occupancy data. The Performance Evaluation highlights the system's robustness in normal operation, with an average accuracy of around 96%, with some missed detection occurring in more unique situations. During the evaluation, these missed detections balanced out over time when calculating the total occupancy, with the calculated occupancy equal to the actual occupancy. Lastly, the Lab Team and User Evaluations reveal positive attitudes toward the system, with suggestions for increased transparency, and other valuable insights for potential future improvements.

In summary, this bachelor's thesis project demonstrated the potential for a privacy-friendly occupancy measurement system using less invasive hardware while maintaining a high degree of accuracy.

9.1 Future research

In the state-of-the-art research it was found that some of these devices, currently implemented at the University of Twente, have additional sensors such as microphones, which are not used for any tracking. Furthermore, stereo camera implementations use software to convert high-resolution images to low resolutions, with the brand promoting the possibility of demographics tracking which is not needed on campus. This is seen in many places, as using high resolution cameras and AI models occupancy tracking is accurate and easy to implement. Questions can be asked if purely software can mitigate all user privacy concerns. The existing body of literature on hardware privacy, classified as level 1 in this thesis, is quite limited. However, this project demonstrates how privacy-friendly hardware can be selected to meet these minimal requirements, generating user support, while maintaining a high degree of accuracy. Future research should look into the question how hardware can help a system satisfy its main requirements and insure privacy. This kind of research will be important in protecting user privacy in the quick developing world of big-data and AI, by limiting excess data to be collected. Another recommendation is to use this research to create a criteria for sensors used in occupancy tracking systems. This criteria can be used to ensure a degree of user privacy. Where this criteria will be used as a framework to position the occupancy tracking method or technology into a certain group. This group, would make organisations and businesses more conscious when selecting a system for their specific purpose and requirements.

Bibliography

[1] Interaction Lab | The Interaction Lab | Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS). URL: https://www.utwente.nl/en/eemcs/facilities/interaction-lab/.

[2] Most Popular Communication Apps in Netherlands. URL: https://www.similarweb.com/apps/top/google/store-rank/nl/communication/top-free/.

[3] Occupancy management. URL: https://www.utwente.nl/en/occupancymanagement/.

[4] User Stories Template. *agilealliance*. URL: https://www.agilealliance.org/glossary/user-story-template/.

[5] What sensors can be found in Ravelijn? | Sensors in Ravelijn. URL: https://www.utwente.nl/en/occupancymanagement/sensors/.

[6] Xovis PC2SE sensor. URL: https://www.xovis.com/technology/sensor/pc2se-sensor.

[7] raspberry-pi-4-datasheet. (Release 1), June 2019. URL: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf.

[8] A guide to using the VL53L5CX multizone Time-of-Flight ranging sensor with wide field of view ultra lite driver (ULD). (Rev 4), August 2023. URL: https://www.st.com/en/embedded-software/stsw-img023.html#documentation.

[9] VL53L5CX Datasheet, March 2023. URL: https://www.st.com/resource/en/datasheet/vl53l5cx.pdf.

[10] Shahira Assem Abdel-Razek, Hanaa Salem Marie, Ali Alshehri, and Omar M. Elzeki. Energy Efficiency through the Implementation of an AI Model to Predict Room Occupancy Based on Thermal Comfort Parameters. *Sustainability*, 14(13), 2022. URL: https://www.mdpi.com/2071-1050/14/13/7734, doi:10.3390/su14137734.

[11] Jawad Ahmad, Hadi Larijani, Rohinton Emmanuel, Mike Mannion, Abbas Javed, and Ali Ahmadinia. An Intelligent Real-Time Occupancy Monitoring System with Enhanced Encryption and Privacy. In *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, pages 524–529, 2018. doi:10.1109/ICCI-CC.2018.8482047.

[12] Rijad Alisic, Marco Molinari, Philip E. Paré, and Henrik Sandberg. Ensuring Privacy of Occupancy Changes in Smart Buildings. In *2020 IEEE Conference on Control Technology and Applications (CCTA)*, pages 871–876, 2020. doi:10.1109/CCTA41146.2020.9206317.

[13] Irvan B. Arief-Ang, Margaret Hamilton, and Flora D. Salim. A Scalable Room Occupancy Prediction with Transferable Time Series Decomposition of CO2 Sensor Data. *ACM Trans. Sen. Netw.*, 14(3–4), November 2018. Place: New York, NY, USA Publisher: Association for Computing Machinery. doi:10.1145/3217214.

[14] Jacques M. Bahi, Christophe Guyeux, and Abdallah Makhoul. Efficient and Robust Secure Aggregation of Encrypted Data in Sensor Networks. In *2010 Fourth International Conference on Sensor Technologies and Applications*, pages 472–477, 2010. doi:10.1109/SENSORCOMM.2010.76.

[15] Indrani Bhattacharya and Richard J. Radke. Arrays of single pixel time-of-flight sensors for privacy preserving tracking and coarse pose estimation. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016. doi:10.1109/WACV.2016.7477671.

[16] Luis M. Candanedo and Véronique Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings*, 112:28–39, 2016. URL: https://www.sciencedirect.com/science/article/pii/S0378778815304357, doi:https://doi.org/10.1016/j.enbuild.2015.11.071.

[17] Jianzhao Cao, Liangliang Sun, Manfred Gilbert Odoom, Fangjun Luan, and Xiaoyu Song. Counting people by using a single camera without calibration. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 2048–2051, 2016. doi:10.1109/CCDC.2016.7531321.

[18] Zhenghua Chen, Chaoyang Jiang, and Lihua Xie. Building occupancy estimation and detection: A review. *Energy and Buildings*, 169:260–270, 2018. URL: https://www.sciencedirect.com/science/article/pii/S0378778818301506, doi:https://doi.org/10.1016/j.enbuild.2018.03.084.

[19] Varick L. Erickson, Stefan Achleitner, and Alberto E. Cerpa. POEM: Power-Efficient Occupancy-Based Energy Management System. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pages 203–216, New York, NY, USA, 2013. Association for Computing Machinery. event-place: Philadelphia, Pennsylvania, USA. doi:10.1145/2461381.2461407.

[20] Marco Gruteser, Graham Schelle, Ashish Jain, Richard Han, and Dirk Grunwald. Privacy-Aware Location Sensor Networks. In *HotOS'03: 9th Workshop on Hot Topics in Operating Systems*, pages 163–168, Lihue (Kauai), Hawaii, USA, January

2003. URL: https://www.researchgate.net/publication/221150775_Privacy-Aware_Location_Sensor_Networks.

[21] Li Jia and Richard J. Radke. Using Time-of-Flight Measurements for Privacy-Preserving Tracking in a Smart Room. *IEEE Transactions on Industrial Informatics*, 10(1):689–696, 2014. doi:10.1109/TII.2013.2251892.

[22] Ellis Kessler, Moeti Masiane, and Awad Abdelhalim. Privacy Concerns Regarding Occupant Tracking in Smart Buildings. *CoRR*, abs/2010.07028, 2020. arXiv: 2010.07028. URL: https://arxiv.org/abs/2010.07028.

[23] Nan Li, Gulben Calis, and Burcin Becerik-Gerber. Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations. *Automation in Construction*, 24:89–99, July 2012. URL: https://www.sciencedirect.com/science/article/pii/S0926580512000283, doi:10.1016/j.autcon.2012.02.013.

[24] Yi-Chen Li, Wen-Chang Tseng, Nan-Hung Hsieh, and Szu-Chieh Chen. Assessing the seasonality of occupancy number-associated CO2 level in a Taiwan hospital. *Environmental Science and Pollution Research*, 26(16):16422–16432, June 2019. doi:10.1007/s11356-019-05084-3.

[25] Daniele Liciotti, Marina Paolanti, Emanuele Frontoni, and Primo Zingaretti. People Detection and Tracking from an RGB-D Camera in Top-View Configuration: Review of Challenges and Applications. In Sebastiano Battiato, Giovanni Maria Farinella, Marco Leo, and Giovanni Gallo, editors, *New Trends in Image Analysis and Processing – ICIAP 2017*, pages 207–218, Cham, 2017. Springer International Publishing.

[26] Angelika Mader and Wouter Eggink. A DESIGN PROCESS FOR CREATIVE TECHNOLOGY. September 2014.

[27] Francesca Meneghello, Matteo Calore, Daniel Zucchetto, Michele Polese, and Andrea Zanella. IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices. *IEEE Internet of Things Journal*, 6(5):8182–8201, 2019. doi:10.1109/JIOT.2019.2935189.

[28] Sergiu Mezei and Adrian Sergiu Darabant. A computer vision approach to object tracking and counting. *Studia Universitatis Babes-Bolyai, Informatica*, 55(1), 2010.

[29] Rudolf Tanner, Martin Studer, Adriano Zanoli, and Andreas Hartmann. People Detection and Tracking with TOF Sensor. In *2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, pages 356–361, 2008. doi:10.1109/AVSS.2008.18.

[30] Paige Wenbin Tien, Shuangyu Wei, and John Calautit. A Computer Vision-Based Occupancy and Equipment Usage Detection Approach for Reducing Building Energy De-

mand. *Energies*, 14(1), 2021. URL: https://www.mdpi.com/1996-1073/14/1/156, doi:10.3390/en14010156.

[31] Danielle N. Wagner, Aayush Mathur, and Brandon E. Boor. Spatial seated occupancy detection in offices with a chair-based temperature sensor array. *Building and Environment*, 187:107360, 2021. URL: https://www.sciencedirect.com/science/article/pii/S0360132320307290, doi:https://doi.org/10.1016/j.buildenv.2020.107360.

[32] F. Wahl, M. Milenkovic, and O. Amft. A Distributed PIR-based Approach for Estimating People Count in Office Environments. In *2012 IEEE 15th International Conference on Computational Science and Engineering*, pages 640–647, 2012. doi:10.1109/ICCSE.2012.92.

[33] Gang Wang, Carlos Lopez-Molina, and Bernard De Baets. Automated blob detection using iterative Laplacian of Gaussian filtering and unilateral second-order Gaussian kernels. *Digital Signal Processing*, 96:102592, 2020. URL: https://www.sciencedirect.com/science/article/pii/S1051200419301460, doi:https://doi.org/10.1016/j.dsp.2019.102592.

[34] Haikuan Wang, Haoyang Luo, Wenju Zhou, and Dong Xie. A Novel Approach of Human Tracking and Counting Using Overhead ToF Camera. In Minrui Fei, Kang Li, Zhile Yang, Qun Niu, and Xin Li, editors, *Recent Featured Applications of Artificial Intelligence Methods. LSMS 2020 and ICSEE 2020 Workshops*, pages 416–429, Singapore, 2020. Springer Singapore.

[35] Wei Wang, Jiayu Chen, and Tianzhen Hong. Occupancy prediction through machine learning and data fusion of environmental sensing and Wi-Fi sensing in buildings. *Automation in Construction*, 94:233–243, 2018. URL: https://www.sciencedirect.com/science/article/pii/S0926580518302656, doi:https://doi.org/10.1016/j.autcon.2018.07.007.

[36] Xiao Wang and Patrick Tague. Non-Invasive User Tracking via Passive Sensing: Privacy Risks of Time-Series Occupancy Measurement. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, AISec '14, pages 113–124, New York, NY, USA, 2014. Association for Computing Machinery. event-place: Scottsdale, Arizona, USA. doi:10.1145/2666652.2666655.

[37] Kevin Weekly, Nikolaos Bekiaris-Liberis, Ming Jin, and Alexandre M. Bayen. Modeling and Estimation of the Humans' Effect on the CO2 Dynamics Inside a Conference Room. *IEEE Transactions on Control Systems Technology*, 23(5):1770–1781, 2015. doi:10.1109/TCST.2014.2384002.

[38] Refael Whyte. Multipath Interference in Indirect Time-of-Flight Depth Sensors. *Medium*, March 2021. URL: https://medium.com/chronoptics-time-of-flight/multipath-interference-in-indirect-time-of-flight-depth-sensors-7f59c5fcd122.

[39] Hongjie Xiang and Wenbiao Zhou. Real-time people detection based on top-view TOF camera. In Zhigeng Pan and Xinhong Hei, editors, *Twelfth International Conference on Graphics and Image Processing (ICGIP 2020)*, volume 11720, page 1172003. SPIE, 2021. Backup Publisher: International Society for Optics and Photonics. doi:10.1117/12.2589352.

[40] Liang Zhao, Yuxin Li, Ruobing Liang, and Peng Wang. A State of Art Review on Methodologies of Occupancy Estimating in Buildings from 2011 to 2021. *Electronics*, 11(19), 2022. URL: https://www.mdpi.com/2079-9292/11/19/3173, doi:10.3390/electronics11193173.

[41] M. S. Zuraimi, A. Pantazaras, K. A. Chaturvedi, J. J. Yang, K. W. Tham, and S. E. Lee. Predicting occupancy counts using physical and statistical Co2-based modeling methodologies. *Building and Environment*, 123:517–528, 2017. URL: https://www.sciencedirect.com/science/article/pii/S0360132317303268, doi:https://doi.org/10.1016/j.buildenv.2017.07.027.

Appendices

Appendix A

Python code

A.1 Database Handler Class

```
from tinydb import TinyDB, Query
import time

class DatabaseHandler:
    def __init__(self): #Makes database and table
        self.db = TinyDB('today')
        self.table = self.db.table('count')

    def add_person(self): #Method adds person to database
        timestamp = int(time.time())
        self.table.insert({'timestamp': timestamp, 'action': 'person entered'})

    def remove_person(self): #Method removes person from database
        timestamp = int(time.time())
        self.table.insert({'timestamp': timestamp, 'action': 'person left'})

    def get_people_count(self): #Returns count of total occupancy
        add_count = len(self.table.search(Query().action == 'person entered'))
        remove_count = len(self.table.search(Query().action == 'person left'))
        return add_count - remove_count
```

A.2 Telegram Handler Class

```
import telebot
import threading

class TelegramHandler:
    def __init__(self, database_handler):
        self.database_handler = database_handler
        token = "6326184737:AAFYuE1A0IsHhy4kPgC5uwpLp-22bIzQQXg"
```

```
        self.bot = telebot.TeleBot(token)

        @self.bot.message_handler(commands=['count'])
        def get_people_count(message): #Method for returning count
            count = self.database_handler.get_people_count()
            estimate = self.estimate_count(count)
            self.bot.reply_to(message, f"There are approximately {estimate}
                                        people in the lab.")

        self.polling_thread = threading.Thread(target=self.run_polling)
        self.polling_thread.daemon = True #stop thread when main program stops
        self.polling_thread.start() #Infinite polling methods is added to
                                     separate thread


    def run_polling(self):
        self.bot.infinity_polling()

    def estimate_count(self, count):
        lower_bound = ((count - 1) // 5) * 5 + 1
        upper_bound = lower_bound + 4
        return f"{lower_bound}-{upper_bound}"
```

A.3 Main Blob Detection Class (Without GUI)

```
from vl53l5cx.vl53l5cx import VL53L5CX
import time
import datetime as dt
import numpy as np
from matplotlib import cm
import cv2
import os
from blob import Blob
from databasehandler import DatabaseHandler
from telegrambot import TelegramHandler

if(True):
        vl53 = VL53L5CX()
        vl53.init()
else:
        vl53 = VL53L5CX()

alive = vl53.is_alive()
if not alive:
    raise IOError("VL53L5CX Device is not alive")
print("Done!")
```

```
vl53.set_resolution(64)

# This is a visual demo, so prefer speed over accuracy
vl53.set_ranging_frequency_hz(15)
#vl53.set_integration_time_ms(5)

vl53.set_target_order(1) #closest
vl53.set_ranging_mode(1) #Continueus
vl53.set_sharpener_percent(20)
vl53.start_ranging()

blob_mapping = {}  # Dictionary to store blob ID and its properties
blob_id_counter = 1

db = DatabaseHandler()
telegram_bot = TelegramHandler(db)

while True:
    if blob_id_counter>10:
            blob_id_counter=1
    if vl53.check_data_ready():
        data = vl53.get_ranging_data()
        arr = np.flipud(np.array(data.distance_mm).reshape((8, 8))).astype('
                                                        float64')
        arr[arr < 5] = 2000

        #print(arr)
        binary_mask = arr < 1300

        binary_arr = binary_mask.astype(np.uint8)

        contours, _ = cv2.findContours(binary_arr, cv2.RETR_EXTERNAL, cv2.
                                                        CHAIN_APPROX_SIMPLE)

        # Update the existing blobs with their new centers and timestamps
        for blob_id, (prev_x, prev_y, last_seen, blobObj) in list(blob_mapping.
                                                        items()):
            if time.time() - last_seen > 1.5:  # If the blob hasn't been seen
                                                            for more than 1.5 seconds,
                                                            remove it
                blobObj.last_pos(prev_x)
                del blob_mapping[blob_id]

        for contour in contours:
            if cv2.contourArea(contour) > 0:
                M = cv2.moments(contour)
                center_x = int(M["m10"] / M["m00"])
                center_y = int(M["m01"] / M["m00"])
```

```
            matched_blob_id = None
            for blob_id, (prev_x, prev_y, _, _) in blob_mapping.items():
                distance = np.sqrt((center_x - prev_x)**2 + (center_y -
                                                        prev_y)**2)
                if distance < 3:  # Threshold for considering it's the same
                                                         blob
                    matched_blob_id = blob_id
                    break

            if matched_blob_id is None:
                new_blob_id = blob_id_counter
                blob_id_counter += 1
                blob_mapping[new_blob_id] = (center_x, center_y, time.time
                                                        (), Blob(center_x,
                                                        db))
            else:
                blob_mapping[matched_blob_id] = (center_x, center_y, time.
                                                        time(),
                                                        blob_mapping[
                                                        matched_blob_id][-1
                                                        ])

            blob_number = str(matched_blob_id + 1 if matched_blob_id is not
                                                      None else new_blob_id
                                                     + 1)

        cv2.waitKey(1)
```

A.4 Main Blob Detection Class (With GUI)

```
from vl53l5cx.vl53l5cx import VL53L5CX
import time
import datetime as dt
import numpy as np
from matplotlib import cm
import cv2
import os
from blob import Blob
from databasehandler import DatabaseHandler
from telegrambot import TelegramHandler


last_time = dt.datetime.today().timestamp()
diffs = []


if(True):
```

```
        vl53 = VL53L5CX()
        vl53.init()
else:
        vl53 = VL53L5CX()

alive = vl53.is_alive()
if not alive:
    raise IOError("VL53L5CX Device is not alive")
print("Done!")

vl53.set_resolution(64)


# This is a visual demo, so prefer speed over accuracy
vl53.set_ranging_frequency_hz(15)
#vl53.set_integration_time_ms(5)

vl53.set_target_order(1) #closest
vl53.set_ranging_mode(1) #Continuous
vl53.set_sharpener_percent(20)
vl53.start_ranging()

blob_mapping = {}  # Dictionary to store blob ID and its properties
blob_id_counter = 1

db = DatabaseHandler()
telegram_bot = TelegramHandler(db)

while True:
    if blob_id_counter>10:
            blob_id_counter=1
    if vl53.check_data_ready():
        data = vl53.get_ranging_data()
        arr = np.flipud(np.array(data.distance_mm).reshape((8, 8))).astype('
                                        float64')
        arr[arr < 5] = 2000

        #print(arr)
        binary_mask = arr < 1300

        binary_arr = binary_mask.astype(np.uint8)

        contours, _ = cv2.findContours(binary_arr, cv2.RETR_EXTERNAL, cv2.
                                        CHAIN_APPROX_SIMPLE)

        # Update the existing blobs with their new centers and timestamps
        for blob_id, (prev_x, prev_y, last_seen, blobObj) in list(blob_mapping.
                                        items()):
```

```
    if time.time() - last_seen > 1.5:  # If the blob hasn't been seen
                                          for more than 1.5 seconds,
                                          remove it
        blobObj.last_pos(prev_x)
        del blob_mapping[blob_id]

blob_marked_array = np.zeros((8, 8), dtype=np.uint8)
cv2.drawContours(blob_marked_array, contours, -1, 255, thickness=cv2.
                                        FILLED)

blob_marked_array = cv2.cvtColor(blob_marked_array, cv2.COLOR_GRAY2BGR)
cv2.drawContours(blob_marked_array, contours, -1, (0, 0, 255),
                                        thickness=1)

resized_blob_marked_array = cv2.resize(blob_marked_array, (400, 400))

for contour in contours:
    if cv2.contourArea(contour) > 0:
        M = cv2.moments(contour)
        center_x = int(M["m10"] / M["m00"])
        center_y = int(M["m01"] / M["m00"])

        matched_blob_id = None
        for blob_id, (prev_x, prev_y, _, _) in blob_mapping.items():
            distance = np.sqrt((center_x - prev_x)**2 + (center_y -
                                                    prev_y)**2)
            if distance < 3:  # Threshold for considering it's the same
                                                     blob
                matched_blob_id = blob_id
                break

        if matched_blob_id is None:
            new_blob_id = blob_id_counter
            blob_id_counter += 1
            blob_mapping[new_blob_id] = (center_x, center_y, time.time
                                                    (), Blob(center_x,
                                                    db))
        else:
            blob_mapping[matched_blob_id] = (center_x, center_y, time.
                                                    time(),
                                                    blob_mapping[
                                                    matched_blob_id][-1
                                                    ])

        blob_number = str(matched_blob_id + 1 if matched_blob_id is not
                                                  None else new_blob_id
                                                 + 1)
        text_size = cv2.getTextSize(blob_number, cv2.
                                                 FONT_HERSHEY_SIMPLEX, 0
```

```
                                                        .5, 1)[0]
            text_position = (int((center_x / 8) * 400 - text_size[0] // 2),
                                 int((center_y / 8) * 400 + text_size[1] // 2))
            cv2.putText(resized_blob_marked_array, blob_number,
                                                        text_position, cv2.
                                                        FONT_HERSHEY_SIMPLEX, 0
                                                        .5, (0, 255, 0), 1)


        cv2.imshow('Binary Image', binary_arr*255)
        cv2.imshow("Blob Marked Array", resized_blob_marked_array)
        cv2.waitKey(1)
        new_time = dt.datetime.today().timestamp()
        diffs.append(new_time - last_time)
        last_time = new_time

        # Clip the list
        if len(diffs) > 10:
                diffs = diffs[-10:]

        #print(len(diffs) / sum(diffs)) #to count sample rate
```

A.5 Blob Class

```
class Blob:
    def __init__(self, initial_x, db):
        self.db = db
        self.initial_x = initial_x
        self.final_x = None
        self.required_distance = 4

    def last_pos(self, x):
        self.final_x = x

        if (self.final_x - self.initial_x) < -self.required_distance:
            self.db.add_person()
            print("added")
        elif (self.final_x - self.initial_x) > self.required_distance:
            self.db.remove_person()
            print("removed")
        else:
            print("stationary")

        print(self.db.get_people_count())
```

A.6 Day Parts Data Minimization code

```
def get_people_count(self, action, start_time, end_time):
    count = len(self.table.search((Query().action == action) &
                                 (Query().timestamp >= start_time) &
                                 (Query().timestamp <= end_time)))
    return count

def save_totals_and_clear_data(self):
    now = datetime.datetime.now()
    morning_start = now.replace(hour=0, minute=0, second=0, microsecond=0)
    morning_end = now.replace(hour=11, minute=59, second=59, microsecond=
                                        999999)
    afternoon_start = now.replace(hour=12, minute=0, second=0, microsecond=
                                        0)
    afternoon_end = now.replace(hour=14, minute=59, second=59, microsecond=
                                        999999)
    late_afternoon_start = now.replace(hour=15, minute=0, second=0,
                                        microsecond=0)
    late_afternoon_end = now.replace(hour=17, minute=59, second=59,
                                        microsecond=999999)

    morning_entered_count = self.get_people_count('person entered',
                                        morning_start, morning_end)
    morning_left_count = self.get_people_count('person left', morning_start
                                        , morning_end)

    afternoon_entered_count = self.get_people_count('person entered',
                                        afternoon_start, afternoon_end)
    afternoon_left_count = self.get_people_count('person left',
                                        afternoon_start, afternoon_end)

    late_afternoon_entered_count = self.get_people_count('person entered',
                                        late_afternoon_start,
                                        late_afternoon_end)
    late_afternoon_left_count = self.get_people_count('person left',
                                        late_afternoon_start,
                                        late_afternoon_end)

    day_parts_db = TinyDB('day_parts')
    day_parts_table = day_parts_db.table('day_parts_totals')
    day_parts_table.insert({morning_start: {'entered':
                                        morning_entered_count, 'left':
                                        morning_left_count},
                           afternoon_start: {'entered':
                                                  afternoon_entered_coun
                                                  , 'left'
                                                  :
                                                  afternoon_left_count
```

```
                                           },
                    late_afternoon_start: {'entered':
                                           late_afternoon_entered
                                           , 'left'
                                           :
                                           late_afternoon_left_co
                                           }})

        # Clear all data from the 'count' table for the next day.
        self.table.truncate()
```

A.7 Ground truth code

```
from datetime import datetime
from tinydb import TinyDB

db = TinyDB('entry_log.json')

def log_action(action):
    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    db.insert({'timestamp': timestamp, 'action': action})

while True:
    key = input()
    if key == 'w':
        log_action("Leaving")
    elif key == 's':
        log_action("Entering")
    else:
        print("Invalid input")
```

Appendix B

Ideation - Information Brochure

Information Brochure

31-03-2023 // Hylke Jellema // Privacy Friendly Room Occupancy System

The purpose of my research is to find a privacy-friendly way to collect occupancy data for the interaction lab. During the interview, you will be asked questions regarding your privacy concerns, data safety, and need for any possible access to real-time data. Additionally, some questions may be asked regarding your current working environment. The interview will take a maximum of 20 minutes.

If during the interview, for any reason, you would like to withdraw, there is no reason to clarify the reasons. The collected data will also be removed from the study. If the interview is taking too long, please also bring this to my attention. To conclude, you can leave at any time without reason.

There is no risk in participating. If there are any worries or discomforts, you can leave at any time and have the collected data removed. The research project has been reviewed by the ethics commission, and its standards are adhered to.

Regarding the collected data:

- Important keywords and quotes are written down in personal notes.
- Any quotes that can identify you, such as [e.g. certain habits or opinions unique to you], will be collected but not be shared beyond personal notes. Any information unique to you that is used in the report will be made unidentifiable [e.g. “On Wednesday” -> “On a certain day of the week”]
- The personal notes will be removed in less than 2 years.
- You may revise personal quotes directly after the interview, correct answers or have them removed.
- These notes are strictly confidential and will only be used to identify patterns which will be used in the published report.

Contact details:

Researcher: Hylke Jellema, h.j.jellema@student.utwente.nl
Supervisor: Daniel Davison, d.p.davison@utwente.nl

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl

Appendix C

Ideation - Consent Form

Consent Form for *Creative Technology*

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	**Yes**	**No**
Taking part in the study		
I have read and understood the study information dated 31/03/2023, or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	○	○
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	○	○
I understand that taking part in the study involves that my answer may be written down and used for research and development of an occupancy data collection systems.	○	○
Use of the information in the study		
I understand that the information I provide will be used in a report for a graduation project and may be kept for <2 years.	○	○
I understand that personal information collected about me that can identify me, such as [e.g. certain habits or opinions unique to you], will be collected but not be shared beyond personal notes. Any information unique to me that is used in the report will be made unidentifiable [e.g. “On Wednesday” -> “On a certain day of the week”]	○	○
I agree that my information can be quoted (in a way explained in the last question) in research outputs	○	○

Signatures

____________________________	_____________________	_________
Name of participant	Signature	Date

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

Hylke Jellema______________	_____________________	_________
Researcher name	Signature	Date

Study contact details for further information: Hylke Jellema, h.j.jellema@student.utwente.nl

Contact Information for Questions about Your Rights as a Research Participant

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl

UNIVERSITY OF TWENTE.

Appendix D

Evaluation - System Test Information

Note: Also used in a "keep right" version.

OCCUPANCY MEASUREMENT TEST

Using a low resolution camera a system is measuring occupancy in the Interaction Lab. The sensor is situated top-down at the entrance of the Lab. An observer may also note your presence to test the accuracy of the system. No personal identifiable information is collected or recorded.

If there are any concerns or discomforts you can **OPT-OUT by keeping left** from this paper. Please also leave the Lab using this lane. This way you are not participating in any automated data collection or observations.

Contact details:

Researcher: Hylke Jellema, h.j.jellema@student.utwente.nl
Supervisor: Daniel Davison, d.p.davison@utwente.nl

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl

12-06-2023
Hylke Jellema

Privacy Friendly Room Occupancy System

Appendix E

Evaluation - Information Brochure

Information Brochure

03-09-2023 // Hylke Jellema // Privacy Friendly Room Occupancy System

The purpose of my research is to evaluate a device to collect occupancy data for the interaction lab in a privacy friendly manner. During the interview, you will be asked questions regarding possible privacy concerns, and how the device could be improved. Additionally, if you are part of the lab team, you will be asked to interact with a chatbot and share your thoughts. The interview will take a maximum of 15 minutes.

If during the interview, for any reason, you would like to withdraw, there is no reason to clarify the reasons. The collected data will also be removed from the study. If the interview is taking too long, please also bring this to my attention. To conclude, you can leave at any time without reason.

There is no risk in participating. If there are any worries or discomforts, you can leave at any time and have the collected data removed. The research project has been reviewed by the ethics commission, and its standards are adhered to.

Regarding the collected data:

- Important keywords and quotes are written down in personal notes.
- Any quotes that can identify you, such as [e.g. certain habits or opinions unique to you], will be collected but not be shared beyond personal notes. Any information unique to you that is used in the report will be made unidentifiable [e.g. “On Wednesday” -> “On a certain day of the week”]
- The personal notes will be removed in less than 2 years.
- You may revise personal quotes directly after the interview, correct answers, or have them removed.
- These notes are strictly confidential and will only be used to identify patterns that will be used in the published report.

Contact details:

Researcher: Hylke Jellema, h.j.jellema@student.utwente.nl
Supervisor: Daniel Davison, d.p.davison@utwente.nl

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl

Appendix F

Evaluation - Consent Form

Consent Form for *Creative Technology*

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	**Yes**	**No**
Taking part in the study		
I have read and understood the study information dated 03/09/2023, or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	○	○
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	○	○
I understand that taking part in the study involves that my answer may be written down and used for research and development of the occupancy data collection system.	○	○
Use of the information in the study		
I understand that the information I provide will be used in a report for a graduation project and may be kept for <2 years.	○	○
I understand that personal information collected about me that can identify me, such as [e.g. certain habits or opinions unique to you], will be collected but not be shared beyond personal notes. Any information unique to me that is used in the report will be made unidentifiable [e.g. “On Wednesday” -> “On a certain day of the week”]	○	○
I agree that my information can be quoted (in a way explained in the last question) in research outputs	○	○

I agree with, and understand, the answers I have given in this form:

____

(Write Yes or No)

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

Hylke Jellema______________	____________________	_________
Researcher name	Signature	Date

Study contact details for further information: Hylke Jellema, h.j.jellema@student.utwente.nl

Contact Information for Questions about Your Rights as a Research Participant

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl

UNIVERSITY OF TWENTE.