FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Context Based Lighting Control

Mariana Almeida Truta

U. PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Master in Informatics and Computing Engineering

Supervisor: Prof. Miguel Pimenta Monteiro

External Supervisor: Eng. Carlos Silva

July 27, 2023

Context Based Lighting Control

Mariana Almeida Truta

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

President: Prof. Jorge Barbosa
External Examiner: Prof. Paulo Baltarejo Sousa
Supervisor: Prof. Miguel Pimenta Monteiro

July 27, 2023

Abstract

Illumination is constant and essential in everyday life and affects a person's performance unnoticeably. It is thus important to control it according to each individual's preferences and context. Lighting Management Systems (LMS), used to control and manage lighting in a building, are continuously evolving to improve the user experience and increase user satisfaction. New generations of LMS prioritise the exploitation of dynamic context-based interactions over manual controls so that users can focus on their tasks rather than the lighting system. Users can therefore benefit from a more natural and intuitive interaction with the system, which is more tailored to their needs and preferences.

Context-based interactions are based on the idea that the system should adapt to the context in which the user is. Such context can be provided by the user or the system itself. This dissertation aims to explore how to change the lights in a room through this type of interaction using the preferences that the user had previously inserted into the system and the context of the environment detected by the system. Firstly, the system will locate the user in an indoor space, such as an office or a residence, to be able to interact with the correct luminaires. To achieve this, it will be used an accurate and precise range/location technology, resistant to obstacle interference, named ultra-wideband (UWB) technology.

Furthermore, the second part of this thesis will focus on developing a dynamic and user-friendly application that allows the user to establish their preferences, constituting their profile, as well as manual control, and make adjustments to the lighting in their current location. This application aims to provide a smoother and effortless interaction with the LMS, by automatically setting and displaying the controls to the user to alter the lights of their location.

The purpose of the application is to ensure that, in addition to greater user satisfaction, there is also a concern for the well-being of users. Human-centric lighting (HCL) is light that is concerned with visual and non-visual (circadian, neuroendocrine and neurobehavioral) responses. Therefore, knowing that not all users have the same needs, besides providing default light scenes designed, it is also essential to give the user the flexibility to change them to make the environment more comfortable and suitable, providing the system with the necessary data to automate the process.

Accordingly, this dissertation will explore and take advantage of the recent technology advances regarding UWB and mobile phone sensors and upgrade and automate the existing light systems.

Keywords: Context based lighting control, Lighting management system, Ultra-wideband, UWB, Human-centric lighting

ACM Classification: Networks → Network services → Location based services, Networks → Network types → Cyber-physical networks → Sensor networks, Human-centered computing → Ubiquitous and mobile computing → Ubiquitous and mobile computing theory, Concepts and paradigms → Ambient intelligence

Resumo

A iluminação é constante e imprescindível na vida quotidiana e afeta o desempenho de uma pessoa de forma impercetível. É, portanto, importante controlá-la de acordo com as preferências e o respetivo contexto de cada indivíduo. Os Sistemas de Gestão de Iluminação (LMS), utilizados para controlar e gerir a iluminação num edifício, estão em contínua evolução para melhorar a experiência do utilizador e aumentar a sua satisfação. As novas gerações de LMS dão prioridade à exploração de interações dinâmicas baseadas no contexto em detrimento dos controlos manuais, para que os utilizadores possam concentrar-se nas suas tarefas e não no sistema de iluminação. Os utilizadores podem, assim, beneficiar de uma interação mais natural e intuitiva com o sistema, sendo este mais adaptado às suas necessidades e preferências.

As interações baseadas no contexto têm por base a ideia de que o sistema deve adaptar-se ao contexto em que o utilizador se encontra. Este contexto pode ser fornecido pelo utilizador ou pelo próprio sistema. Esta dissertação visa explorar como alterar as luzes numa sala através deste tipo de interação utilizando as preferências que o utilizador tinha anteriormente inserido no sistema e o contexto do ambiente detetado pelo sistema (por exemplo, a hora do dia e a intensidade e cor da luz na sala). Além disso, antes disso, o sistema localizará o utilizador num espaço interior, como um escritório ou uma residência, para poder mudar a luz das luminárias corretas. Para o conseguir, será utilizada uma tecnologia precisa de alcance/localização, resistente a interferências de obstáculos, denominada tecnologia de banda ultralarga (UWB).

Para além disso, a segunda parte desta tese centrar-se-á no desenvolvimento de uma aplicação dinâmica e de fácil utilização que permita ao utilizador estabelecer as suas preferências, constituindo o seu perfil, bem como o controlo manual e ajustes à iluminação na sua localização atual. Esta aplicação tem como objetivo proporcionar uma interação mais suave e sem esforço com o LMS, ajustando e apresentando automaticamente os controlos ao utilizador para alterar as luzes da sua localização.

O propósito desta aplicação é assegurar que, para além de uma maior satisfação do utilizador, haja também uma preocupação com o bem-estar dos utilizadores. A iluminação centrada no ser humano (HCL) é luz que tem em conta respostas visuais e não visuais (circadianas, neuroendócrinas e neurocomportamentais). Portanto, sabendo que nem todos os utilizadores têm as mesmas necessidades, para além de fornecer cenas de luz predefinidas concebidas com base em estudos de HCL, é também essencial dar ao utilizador a flexibilidade necessária para as alterar de modo a tornar o ambiente mais confortável e adequado, fornecendo ao sistema os dados necessários para automatizar o processo.

Assim, esta dissertação irá explorar e tirar partido dos recentes avanços tecnológicos em relação aos sensores UWB e telemóveis e melhorar e automatizar os sistemas de luz existentes.

Acknowledgments

Firstly, I would like to thank my supervisor, Prof. Miguel Pimenta Monteiro, for accepting this challenge and for the guidance and feedback throughout this project.

I am deeply thankful for the warm welcome and support I received from Tridonic. Their constant assistance, encouragement, and provision of resources made this project possible. I never felt alone, and I am grateful for their help.

I am eternally thankful to my family, especially my brother and parents, for their unconditional love, sacrifices, and support. They have been my pillars of strength throughout my life, and I owe my achievements to their belief in me.

I want to express my sincere appreciation to my friends, who made this experience more memorable and enjoyable. I am grateful for the knowledge and joy they bring into my life. They inspire me, and I hope I can always be there for them as they have been for me.

Lastly, I want to give a special thanks to André. His love, support, and infinite patience have been a constant source of strength and motivation. I am incredibly grateful for his presence in my life and the comfort he has provided throughout this journey.

To all who have contributed, whether big or small, to this project and my degree, I thank you from the bottom of my heart.

Mariana

“Carpe diem.
Seize the day, boys.
Make your lives extraordinary.”

Dead Poets Society, 1989

Contents

List of Figures

List of Tables

Listings

Abbreviations

AoA	Angle of Arrival
API	Application Programming Interface
BaaS	Backend-as-a-Service
BLE	Bluetooth Low Energy
CCD	Central Control Device
CNN	Convolutional Neural Network
GPS	Global Positioning System
HCL	Human-centric Lighting
iOS	iPhone Operating System
IoT	Internet of Things
IPS	Indoor Positioning System
LAN	Local Area Network
LED	Light Emitting Diode
LMS	Lighting Management System
LOS	Line of Sight
MEMS	Micro Electro-Mechanical System
MM	Magnetic Matching
MQTT	Message Queuing Telemetry Transport
NLOS	Non Line of Sight
OoB	Out-of-Band
PDR	Pedestrian Dead Reckoning
PoC	Proof of Concept
REST	Representational State Transfer
RSSI	Received Signal Strength Indicator
SLMS	Smart Lighting Management System
SVM	Support Vector Machine
TDoA	Time Difference of Arrival
ToA	Time of Arrival
ToF	Time of Flight
TWR	Two-Way Ranging
UWB	Ultra-Wideband
VLP	Visual Light Positioning
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

Chapter 1

Introduction

This chapter begins with a clear understanding of the context of the problem. Following that, it presents the motivation for the research and defines the objectives. Finally, the chapter concludes with a brief description of the document's structure, highlighting the analyses of state-of-the-art and the planning of the work to achieve the goals.

1.1 Context

Lighting has been used since ancient times when humankind relied on fire as their primary light source. Throughout history, various forms of lighting have been used, such as candles, oil lamps, and electric lamps [23]. The electric lamps marked the first time that lighting could be controlled by a switch. However, the characteristics of the light remained fixed, providing little flexibility in terms of lighting control. This was one of the first steps towards the development of Lighting Management Systems (LMS).

Ultra-wideband (UWB) technology is a cutting-edge wireless communication method that utilises a broad range of frequencies to transmit and receive signals with high precision and low power consumption. Although UWB has been employed in fields like medicine for many years, it has only recently been integrated into mobile phones, making it more accessible to the general public. The latest smartphones with UWB chips allow for more accurate and precise location detection compared to traditional wireless technologies such as Wi-Fi and Bluetooth [30]. The technology's secure nature and accurate localisation capabilities have led to its exploration in various applications, including the automotive industry. For example, using this technology, vehicles are able to automatically unlock, activate lights, and even apply personalised settings when the owner approaches.

This technology allows seamless and context-based interactions, ushering in a new era of automation and user-centric experiences. It enables more precise indoor navigation, which can be used to provide more accurate and personalised services. Furthermore, UWB can revolutionise the Internet of Things (IoT) by providing seamless and secure communication between devices, creating interconnected ecosystems that respond intelligently to user presence and actions.

By integrating UWB technology into LMS, a paradigm shift in user interaction and lighting control can be achieved, enabling more personalised and context-aware lighting experiences. Users can enjoy lighting environments that adapt to their needs and preferences, enhancing comfort and convenience. The precise localisation capabilities of UWB technology open up new possibilities for optimising lighting conditions based on user presence and movement within a space. Ultimately, the integration of UWB technology in LMS paves the way for transformative lighting experiences that respond to individual preferences and create a more immersive and dynamic lighting environment.

Tridonic [1], headquartered in Austria, is a technology company of the Zumtobel Group [2] and a leading provider of lighting solutions. Committed to enhancing the user experience and promoting well-being, Tridonic has proposed the development of an application that enables seamless interaction with the lighting system based on user context, such as location and preferences. With this application and Tridonic lighting solutions, users can have optimal lighting conditions for their needs and preferences without any extra effort.

1.2 Motivation and Goals

In today's fast-paced world, there is an increasing desire for automation and systems that effortlessly adapt to people's needs. The demand for a lifestyle where mundane and repetitive tasks are taken care of, allowing people to focus on more important things, is evident. Whether it is smart homes that intuitively adjust temperature and lighting based on user presence or smart cars that automatically unlock and adjust settings based on the driver's preferences, the desire for automation is clear. As these innovations continue to evolve, the dream of a world where technology seamlessly integrates into people's lives, anticipating and fulfilling their desires, becomes a compelling reality.

Lighting management systems are no exception to this trend. The ability to control lighting conditions based on user contexts, such as location and preferences, is a significant step towards creating a more personalised and immersive lighting experience. Precisely locating the user within a room is a crucial feature to enable this type of interaction. Tridonic initially explored the concept using Bluetooth; however, the results were not satisfactory. Although widely used, Bluetooth is not accurate enough to provide the user's precise location. Therefore, choosing an adequate technology that could provide the desired results was necessary.

This project is motivated by the growing demand for automation, personalisation, and efficiency in lighting systems. Traditional lighting systems often lack flexibility, require manual operation and do not adapt to the dynamic needs of users, resulting in suboptimal lighting experiences. By addressing these limitations, the project aims to develop an intuitive and user-friendly mobile application that provides a smooth and personalised lighting experience.

[1] https://www.tridonic.com/
[2] https://www.zumtobel.com/

The core of the mobile application lies in its ability to accurately detect the user's location within a room and adjust the lighting conditions accordingly. To achieve this, the project leverages the integration of UWB technology into mobile phones. With the recent release of inter-device communication code for UWB on iOS and Android platforms, this technology has become more accessible, offering new possibilities for various applications. By utilising UWB, the system can overcome the limitations of previous technologies and provide precise user location data accurate to a few centimetres. This level of accuracy is crucial in delivering the desired user experience.

Furthermore, individuals have unique preferences regarding lighting conditions. While some prefer brighter environments, others may prefer a dimmer ambience. The ability to adjust lighting conditions based on that preferences is another essential feature of the application. Users can set their preferred lighting conditions, and the system will automatically adjust accordingly. This feature proves particularly valuable in shared spaces where multiple individuals with varying preferences coexist. By allowing users to define their preferences, the system ensures a more personalised and enjoyable lighting experience for everyone involved.

1.3 Document Structure

In this document, there are five more chapters. Chapter 2 lays out the fundamental concepts necessary for understanding the topic, as well as an overview of current industry advancements and related technologies. Chapter 3 focuses on the specific problem and presents the proposed solution, including system overview and requirements. In Chapter 4, all solution components are described in detail. Chapter 5 presents the results of the tests conducted to validate the system. Finally, Chapter 6 concludes the document and discusses future work.

Chapter 2

State of the Art

Before introducing the solution, it is essential to understand the relevant concepts and technologies thoroughly. This chapter explores the current state of the art in lighting management systems and indoor positioning using a mobile phone, giving a comprehensive overview of these subjects.

2.1 Lighting Management System

The importance of lighting in everyday life cannot be overstated, as it profoundly affects an individual's mood, comfort and productivity. With advances in lighting technology, the introduction of lighting management systems has become increasingly common as a solution to optimising the lighting experience.

The current section provides an overview of lighting controls, starting with their definition and purpose. Next, smart lighting management systems, including their features and benefits, are analysed. Finally, the section concludes by discussing the impact of light on people and the concept of human-centric lighting, which aims to optimise lighting based on human needs and behaviour.

2.1.1 Lighting Controls

Lighting controls are devices and systems used to regulate the operation and performance of lighting systems. They provide a means of adjusting lighting levels and turning lights on and off as needed.

In the following sections, the most common manual and automatic types of lighting controls are described, presenting the pros and cons of each one.

2.1.1.1 Switch

Switches (Figure 2.1) are a basic manual lighting control form that allows the user to turn lights on and off manually [26].

They are easy to install and use, making them a cost-effective solution for basic lighting control needs. Nevertheless, switches have limited functionality, providing only basic on/off control, which can result in inefficient lighting and higher energy consumption. Additionally, traditional switches do not allow for any adjustments to lighting levels or provide any means of creating custom lighting scenes, which makes them less flexible and versatile than other forms of lighting control. However, it is worth noting that modern switches already incorporate dimming control by long pressing the switch, for example.

Figure 2.1: Example of a switch

2.1.1.2 Dimmer

Dimming (Figure 2.2) is manual lighting control that allows for adjustable lighting levels [26]. It provides a means of reducing energy consumption and creating a desired ambience by controlling light intensity. Using total light levels is often unnecessary since, for example, different tasks may require different intensities. Dimmers allow for these variations, eliminating the need to move to a room with better lighting to continue a task.

However, dimmers can be more complex to install and use than simple switches and may not be compatible with all light sources.

Figure 2.2: Example of a dimmer

2.1.1.3 Scene Setter

Scene setters (Figure 2.3) are a type of lighting control that enables users to create and store multiple lighting scenes for different situations and times of the day [26]. These scenes can be activated with just one button press, providing a convenient and flexible way to manage lighting. They are ideal for homes and offices, providing a simple and intuitive means of controlling lighting and creating the desired ambience.

Despite that, scene setters can be more complex and expensive than other lighting controls, making them unsuitable for basic lighting control needs. Advanced wiring and installation may also be required, which increases costs and makes installation more challenging. Besides, scene setters usually lack advanced features and functionalities, making them less flexible and versatile than other lighting control forms.

Figure 2.3: Example of a scene setter

2.1.1.4 Date & Time Scheduling

Date & Time Scheduling (2.4) is a type of lighting control that enables users to establish automatic on/off light schedules and trigger lighting scenes according to specific times, dates or events [26].

This type of control can be used for energy savings, convenience, and security. By setting automatic schedules, users can decrease energy consumption and costs by ensuring lights are turned off when not in use. Date & Time Scheduling also offers an easy way to automatically adjust lighting levels for different times of day, eliminating the need to switch lights manually. This feature not only enhances convenience but also contributes to energy efficiency. Moreover, the system enhances security by enabling automated control of lights in buildings, even when the user is absent. For instance, it can automatically turn lights on and off, creating the sense that the building is occupied and discouraging possible intruders.

The limitations of Date & Time Scheduling include the need for frequent adjustments if schedules change frequently. In addition, no lights will be on if unplanned users are in the space, and no manual switch is available [12]. Also, the complex installation and setup may limit accessibility for some users.

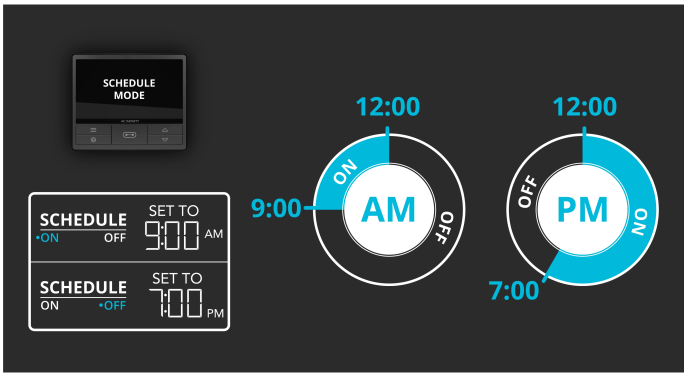

Figure 2.4: Example of a time scheduling configuration

2.1.1.5 Group Control

Group control (Figure 2.5) refers to the ability to control multiple luminaires as a single unit rather than controlling each luminaire individually. This type of lighting control offers greater efficiency and ease of use and is ideal for large rooms and spaces with multiple luminaires in a single area [26]. This saves time and makes lighting control more convenient, especially in larger spaces.

Nonetheless, group control also has some limitations. One is the potential for reduced flexibility, as users may not be able to make individual adjustments to each light in a group. Furthermore, group control may be more complex and require advanced wiring and installation, making it less accessible and more expensive for some users.

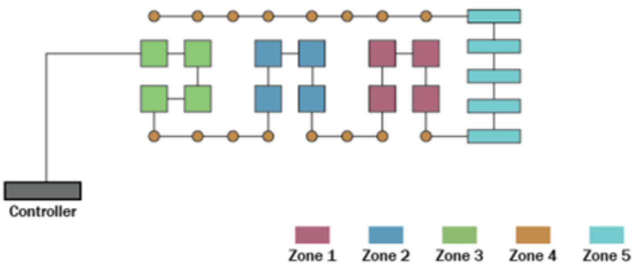

Figure 2.5: Example of a group control lighting

2.1.1.6 Occupancy Adaptation

Occupancy adaptation (Figure 2.6) consists of a lighting control system that automatically adjusts light levels based on the presence of people in a space. This type of control uses sensors to detect occupancy and adjust lighting accordingly [26].

The main advantage of occupancy adaptation is increased energy efficiency, especially in spaces that are not frequently occupied. By automatically turning lights off when a room is unoccupied and turning them back on when someone enters, occupancy adaptation helps to reduce energy consumption and costs. Studies have shown that this type of control can result in energy savings ranging from 17% to 60% [13].

Nevertheless, occupancy adaptation also has some disadvantages. It requires increased upfront expenses resulting from additional hardware (presence sensors) and more complex installation and setup, making it less accessible for some users. Occupancy adaptation may also require frequent adjustments to ensure optimal performance.

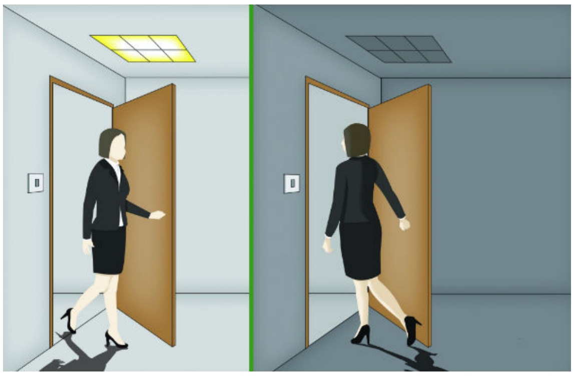

Figure 2.6: Example of an occupancy adaptation lighting

2.1.1.7 Daylight Harvesting

Daylight harvesting (Figure 2.7) is a lighting control method that leverages natural daylight to minimise the use of artificial light. It leads to energy savings of around 40% [26] and a more bright and comfortable indoor environment. It adjusts the intensity of artificial lights based on the daylight level in a room, making gradual changes to avoid discomfort for users.

Daylight harvesting, though effective, has drawbacks such as specialised equipment requirements and higher initial investment. Proper placement of lighting sensors is crucial, as movement and obstacles can affect readings. Pandharipande and Newsham [26] and Chew *at al.* [10] have compiled studies with suggested placement strategies for lighting sensors. The system must also be calibrated and maintained for optimal performance, adding to cost and complexity.

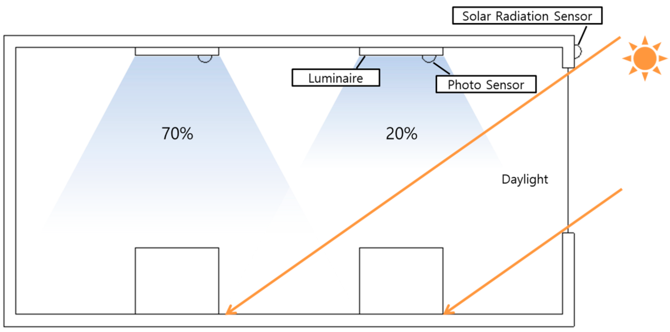

Figure 2.7: Example of a daylight harvesting lighting

2.1.1.8 Personal Control

Personal control (Figure 2.8) is a lighting control option that allows users to customise and control the lighting in their environment through light intensity and colour adjustments [26].

General lighting may not always meet the diverse needs of tasks and individuals. Personal control provides greater customisation and flexibility, allowing users to tailor the lighting environment to their specific needs and preferences for improved comfort and experience.

Regarding limitations, personal control may require advanced technology and setup, making it less accessible or more expensive compared to other lighting control options. It also may require more technical knowledge to use effectively.

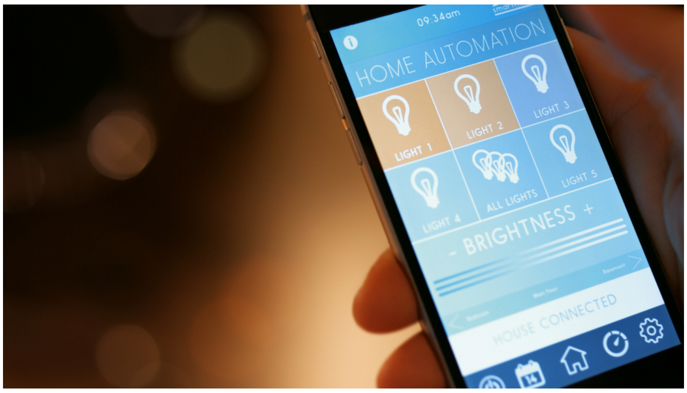

Figure 2.8: Example of a personal control lighting

2.1.1.9 Combined Controls

Combining lighting controls can be a highly effective way to address various issues in lighting systems and enhance the overall lighting experience. It is possible to optimise energy efficiency, improve indoor lighting quality, and provide greater personalisation and convenience by utilising multiple control strategies [26].

For example, combining occupancy adaptation with daylight harvesting can ensure that lighting levels are always optimal and energy efficient. Occupancy sensors can automatically turn off the lights when a room is unoccupied. When someone enters, daylight harvesting can adjust the intensity of artificial lighting based on the available natural light, resulting in a more energy-efficient system that reduces energy consumption and costs.

Similarly, combining personal control with time scheduling can provide users with ultimate control over their lighting environment. Personal control allows users to customise their lighting, while time scheduling can automatically adjust lighting levels based on pre-programmed schedules. This ensures that the lighting is always set to the desired levels without requiring manual adjustments.

To gain a comprehensive understanding of the energy savings achieved through the combination of different lighting controls, it is recommended to refer to [10], which provides a succinct overview of various studies conducted in this area.

Combining lighting controls makes it possible to create a more flexible, efficient, and user-friendly lighting system. Not only do combined lighting controls provide greater control and customisation, but they also help to address the limitations of individual control strategies, resulting in a more comprehensive and effective lighting solution.

2.1.2 Smart Lighting Management System

Smart lighting management systems (SLMS) are advanced technology transforming how lighting is controlled. With a combination of convenience, energy efficiency, and customisation options, they are becoming increasingly popular [28].

SLMS essentially comprises two key components: smart bulbs and a central hub to control the lights. Firstly, smart bulbs are LED lights remotely controlled via a mobile app or wireless connection. They come with various features such as time scheduling, dimming and colour changing, allowing users to adapt the lighting to their needs and requirements. Furthermore, this technology relies on a central hub connecting smart bulbs and other smart home devices. The hub acts as a single point of control and communication and allows users to manage their home or office lighting from a single app or voice assistant [32].

SLMS present numerous benefits that make them popular for modern homes and offices. Some of the most notable advantages include the following [32]:

- **Energy Savings:** With the ability to schedule lights and turn them on and off automatically based on occupancy and natural light, SLMSs can significantly reduce energy waste and lower electricity bills;
- **Convenience:** With remote control via smartphone apps or voice assistants, users can easily manage the lighting of their homes and offices with greater convenience and ease;
- **Customization:** The ability to dim lights, create scenes and change colours enables users to create a personalised lighting environment that suits their moods and preferences;
- **Improved Safety:** By turning lights on automatically through motion sensors, SLMSs can enhance the security of homes and offices by illuminating the space as soon as someone enters;
- **Increased Comfort:** By adjusting lighting levels based on the time of day and available daylight, SLMSs can create a more comfortable and welcoming atmosphere;
- **Enhanced Home Automation:** Integrating SLMSs with other smart home devices can simplify control and enhance overall home automation, making it easier to manage multiple devices.

SLMSs also have several limitations that are worth considering:

- **Heavy Reliance on User Input:** The reliance on manual input can hinder the system's ability to adjust lighting dynamically according to real-time environmental factors;
- **Inaccurate or Nonexistent Location Detection:** The lack of integration with indoor positioning technology can limit the system's ability to precisely determine the user's location within a room and adjust lighting accordingly, thus compromising the overall lighting experience;

- **Complex Interfaces:** Some smart lighting systems may have user interfaces that are complex and unintuitive, presenting a challenge for those without advanced technological skills;

- **High Initial Investment:** The cost of purchasing and installing such a system can also be prohibitively high.

2.1.3 Human-Centric Lighting

Human-centric lighting (HCL) refers to lighting design that considers the effects of light on human health and well-being. It prioritises factors such as visual comfort, sleep quality, mood, and alertness and aims to create lighting environments that support these factors. This approach recognises that light can have both visual and non-visual (circadian, neuroendocrine and neurobehavioral) effects on humans and seeks to optimise these effects in designing lighting systems [5]. Figure 2.9 illustrates the key effects of light on the human body, as presented by Zumtobel.

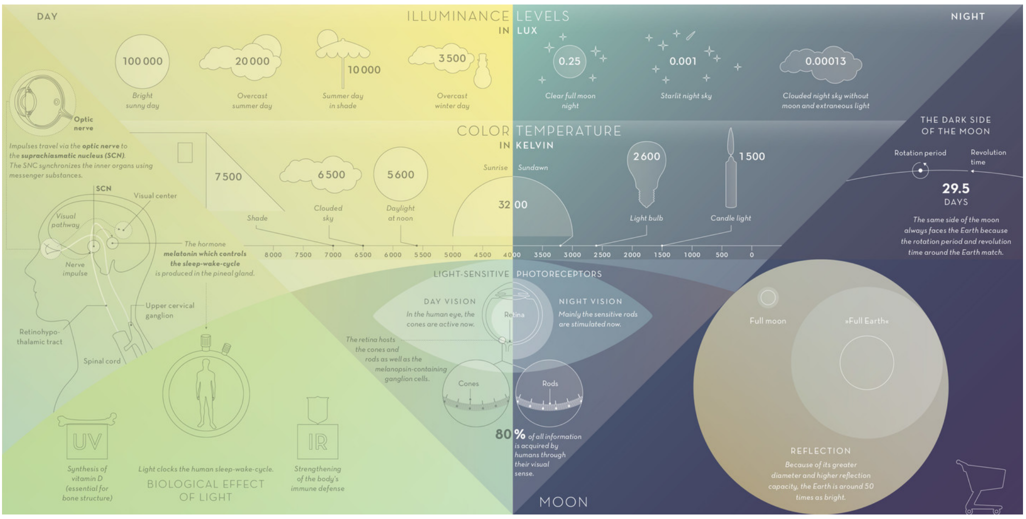

Figure 2.9: Impact of light [37]

Circadian rhythm (Figure 2.10) is the internal biological process that regulates the sleep-wake cycle, hormone secretion, and other vital bodily functions based on the 24-hour day and night cycle. It is important to regulate the light/dark cycle to maintain optimal circadian function. This can be achieved by prioritising natural daylight and properly regulating artificial light to minimise disruptions to the circadian rhythm [16].

Figure 2.10: Light and Circadian Rhythm

Studies show it can improve alertness, cognitive performance, and overall health and well-being [5, 11, 17]. Therefore, it is essential to prioritise HCL in lighting design for optimal outcomes.

2.1.4 LITECOM

LITECOM [38], developed by Zumtobel, is an innovative lighting management system designed to control the entire lighting environment. It introduces a new approach to lighting control by integrating three fundamental elements: intuitive web operation, straightforward installation techniques, and customisable functionality through plug-ins.

LITECOM enables the automation of up to 250 devices using a single LITECOM Central Control Device (CCD). This functionality makes it ideal for smaller buildings or individual floors, offering efficient control and customisation options.

For larger-scale applications, LITECOM infinity takes the capabilities further by allowing the automation of up to 2,500 devices. This expanded capacity is achieved through the integration of up to 15 LITECOM CCDs, providing flexibility to adapt the system to diverse requirements. For instance, an office spread over multiple floors can be easily operated as a unified entity through the LITECOM web application. By combining multiple LITECOM CCDs within an Infinity system, the lighting control can be extended effortlessly across various zones. The Infinity system is designed for scalability, enabling the addition or removal of LITECOM CCDs as needed.

In summary, with its user-friendly web application, scalability options, and open platform approach, LITECOM offers an intuitive and customisable solution for lighting management. By combining technological advancements, LITECOM provides users with a highly adaptable and efficient lighting control system for a wide range of applications.

2.2 Indoor positioning using a mobile phone

Indoor positioning consists of determining an individual's or object's location within a confined indoor space. Mobile devices like smartphones have become popular for accomplishing this [3].

In the past, locating a person required additional sensors. With the widespread use of mobile phones equipped with advanced sensors, people now carry everything they need to locate themselves at all times.

However, the unique challenges that indoor environments present, such as the absence of GPS signals and the presence of obstacles, make indoor positioning a difficult task.

This chapter aims to thoroughly examine the most recent indoor positioning systems, focusing on the chosen technology, Ultra-Wideband, concerning mobile phone usage.

2.2.1 Indoor Positioning System

An Indoor Positioning System (IPS) is a technology that uses various wireless technologies to determine an individual's or object's location within a confined indoor space. Bluetooth Low Energy (BLE), Wi-Fi, and Ultra-Wideband (UWB) are the most commonly used wireless technologies for IPS.

Indoor positioning differs from outdoor positioning regarding technology and location-based information. GPS positioning, the most common outdoor positioning technology, uses satellite signals to determine a person's or object's location. GPS signals are typically strong and accurate, making outdoor positioning a simple process. However, natural and manufactured objects, such as tall buildings, can block or distort GPS signals, making outdoor positioning less accurate in specific environments [2].

On the other hand, achieving high accuracy with indoor positioning can be challenging since indoor environments can be complex and dynamic. Instead of satellite signals, indoor positioning systems use other wireless technologies to determine location, such as BLE, Wi-Fi, and UWB.

The type of location-based data provided is another significant distinction between indoor and outdoor positioning. Outdoor positioning is commonly used to provide information about a person's or object's location in terms of geographical coordinates such as longitude and latitude. Indoor positioning, however, is typically used to provide information about a person's or object's location within a specific indoor environment, for instance, a building or a shopping mall, and requires a higher level of precision [2].

Despite the challenges that indoor positioning faces, IPS is a rapidly growing technology with many potential applications. With the advancement of IoT and Machine Learning, IPS is expected to become even more powerful and sophisticated and be integrated into various applications and industries.

2.2.2 Localisation Techniques

IPSs rely on various localisation techniques to determine the location of a device or person within an indoor environment. In this subsection, some of these techniques are discussed, including Received Signal Strength Indicator (RSSI), Fingerprinting, Time of Arrival (ToA) or Time of Flight (ToF), Angle of Arrival (AoA), and Time Difference of Arrival (TDoA). Each method has its own advantages and limitations that should be considered when selecting the appropriate technique for a given technology.

Before delving into the various techniques used in IPS, it is crucial first to understand the concepts of Line of Sight (LOS) and Non Line of Sight (NLOS). LOS refers to the straight path between the transmitter and receiver without obstacles or barriers. In contrast, NLOS occurs when there is no direct path between the transmitter and receiver, resulting in multipath propagation where the signal bounces off walls or other surfaces before reaching the receiver.

2.2.2.1 Received Signal Strength Indicator

Received Signal Strength Indicator (RSSI) measures the power level in a received radio signal. It is usually expressed in decibel-milliwatts (dBm) or milliWatts (mW) [34] and is used to determine the distance between a wireless device and the wireless access point or other devices emitting wireless signals.

When a wireless device, such as a smartphone, receives a wireless signal from a wireless access point, the strength of the signal is measured and recorded as RSSI. The higher the RSSI value, the stronger the signal is and the closer the device is to the wireless access point. Conversely, the lower the RSSI value, the weaker the signal is, and the farther the device is from the wireless access point [29]. By measuring the RSSI from multiple access points, it is possible to triangulate the device's position based on the relative distances between it and each access point and determine its location [34]. The diagram for this technique can be seen in Figure 2.11.

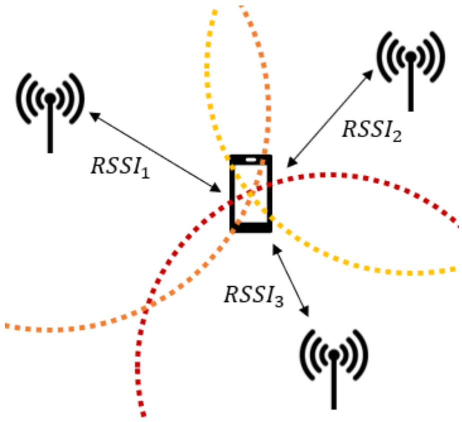

Figure 2.11: Estimating position using RSSI measurements from three access points

RSSI has the advantage of being relatively simple to implement and requiring no additional hardware. However, it has some limitations. Obstacles (for example, walls and furniture) and other wireless devices can affect signal strength [29]. Furthermore, the type of wireless technology used and the environment in which the device is located can impact RSSI accuracy [22]. Finally, complex algorithms likely need to be used to obtain highly accurate localisation [34].

2.2.2.2 Fingerprinting

Fingerprinting is a method based on the distinct properties of wireless signal strength from different access points in an environment. This process involves collecting measurements from a device at a known location. It is commonly accomplished by moving a mobile device around the area to collect measurements from various access points, which are then stored in a database, creating a "fingerprint" of the environment.

While locating a device, measurements are collected and compared to the fingerprints in the database. The fingerprint that most closely matches the measurements taken by the device is used to determine the device's location [22].

Since it considers the environment's specific properties, changes in the environment, for instance, adding or removing access points, can influence the method's accuracy. Moreover, gathering and storing fingerprints can be time-consuming and resource-intensive.

2.2.2.3 Time of Arrival or Time of Flight

Time of Arrival (ToA) or Time of Flight (ToF) measures the time required for a signal to reach the receiver from the transmitter. The distance between the transmitter and receiver can be calculated with $d = v(t_f - t_i)$, where t_i represents the time the transmitter sends the message, t_f the time the receiver receives it, and v the speed of the signal. This distance measurement, obtained from at least three reference nodes, can be used to triangulate a device's position through a trilateration process (see Figure 2.12) [34].

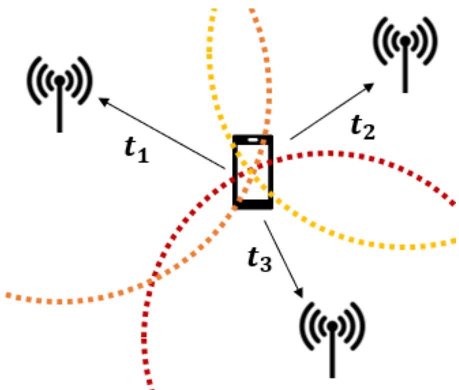

Figure 2.12: Estimating position using ToA and three access points

Although ToA is a simple method and requires only one measurement, when the direct LOS path is unavailable, localisation errors can occur due to obstacles deflecting the emitted signals and causing a longer path for the signal to propagate [22, 34].

Furthermore, the implementation of ToA requires clock synchronisation between all devices, increasing the hardware cost. An alternate two-way communication approach, two-way ranging (TWR), represented in Figure 2.13, uses the round trip duration of a signal to determine the distance between the transmitter and the measuring equipment. This technique does not require clock synchronisation but takes longer to process and consumes more power [22].

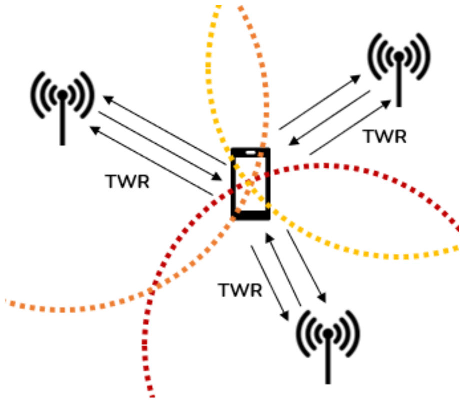

Figure 2.13: Estimating position using TWR and three access points

2.2.2.4 Time Difference of Arrival

Time Difference of Arrival (TDoA) is a technique used to determine the location of a transmitter by using multiple receivers placed at known locations. The transmitter sends out a signal, commonly referred to as a *blink*, which is received by all the receivers. Each receiver then measures the time the signal arrives (ToA) and calculates the difference between the ToA at each receiver. This difference in ToA is then used to calculate the distance between the transmitter and each receiver through multilateration. By comparing the distances calculated from each receiver, the precise location of the transmitter can be determined, as demonstrated in Figure 2.14.

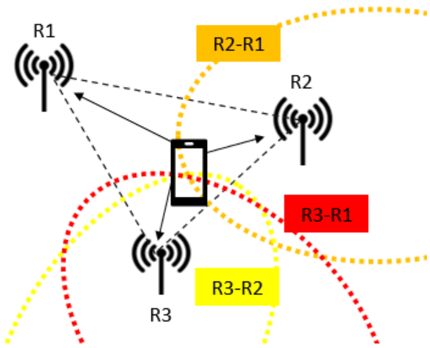

Figure 2.14: Estimating position using TDoA and three access points

Mathematically, the system of equations 2.1 can be used to calculate the exact location of the transmitter by determining the coordinates (x, y) [33], knowing that the other pairs of coordinates are the coordinates of each receiver.

$$\begin{cases} R2 - R1 = \sqrt{(x_2 - x)^2 + (y_2 - y)^2} - \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \\ R3 - R1 = \sqrt{(x_3 - x)^2 + (y_3 - y)^2} - \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \\ R3 - R2 = \sqrt{(x_3 - x)^2 + (y_3 - y)^2} - \sqrt{(x_2 - x)^2 + (y_2 - y)^2} \end{cases} \tag{2.1}$$

This method requires all the receivers to be synchronised to ensure accurate results [22], as even a slight difference in timing can lead to significant errors in the final location calculation.

Also, it is important to note that the accuracy of TDoA is strongly reliant on the signal bandwidth and sampling rate [33].

2.2.2.5 Angle of Arrival

Angle of Arrival (AoA) calculates the transmitter's position using antenna arrays installed on the receiving side. The transmitter sends a signal received by the antenna array at the receiver. The AoA algorithm then measures the angle at which the signal was received by calculating the TDoA at specific antenna array elements. Finally, the intersection of the angle line for each signal source can be used to estimate the transmitter's location [33, 34].

Although they can determine an accurate location, AoA-based approaches are vulnerable to various circumstances that can result in incorrect target location determination. AoA accuracy decreases with increasing transmitter/receiver distance [2]. In addition, it can be challenging to obtain accurate results due to multipath effects in indoor environments caused by NLOS situations [33]. Besides, AoA estimate techniques are more complex than other methods and necessitate exact calibration [2].

2.2.3 Localisation Technologies

As mobile devices have advanced, their built-in sensors have become increasingly sophisticated, enabling various applications to harness their capabilities. Among these, indoor positioning technology stands out as a crucial tool for various applications, such as indoor navigation, asset tracking, and wayfinding.

Despite the availability of numerous indoor positioning options, none have achieved the same level of widespread adoption as GPS for outdoor location determination.

In the following sections, the most important technologies used with mobile phones for indoor positioning will be discussed.

2.2.3.1 Wi-Fi Positioning Technology

Wi-Fi is a widely-used wireless local area network (WLAN) technology that allows users to access the internet without needing wires. This technology is standardised under the IEEE 802.11[1] standard, widely recognised for its versatility and ease of deployment [3]. Due to its prevalence, Wi-Fi is very often utilised for indoor positioning with mobile phones.

Over the years, Wi-fi-based IPSs have been the subject of extensive research [18]. Microsoft RADAR [4], developed in 2001, was the first Wi-fi-based IPS and used both RSSI and fingerprinting. Although it is easy to implement, the system's accuracy was not particularly high, leading to the further developing of more sophisticated systems.

Despite Wi-fi-based IPSs have been improving, and other techniques have been explored (see [18] for more details), they still have some limitations, one being the impact of access point positioning on accuracy. The position of access points, which are not always strategically placed for

[1] https://www.ieee802.org/11/

precise indoor positioning, and a failure in one of them can lead to a significant error in the position [24]. Additionally, the system needs to be recalibrated each time the environment changes, which can be a problem in dynamic environments [9, 24].

2.2.3.2 Bluetooth Positioning Technology

Bluetooth, a short-range wireless communication technology, is increasingly being explored as an option for IPSs. Compared to Wi-Fi, Bluetooth operates over a smaller range and uses less power, making it ideal for mobile devices [3, 24].

Bluetooth-based IPSs have been proposed in the past decades, usually based on fingerprint and RSSI techniques with different models to match the measurements (see examples of that in [6, 7, 8]). Their performance has been less than satisfactory, with low accuracy and slow location identification times, which is not ideal for real-time indoor positioning applications [9].

Recent advances in Bluetooth technology have made it possible to use Bluetooth Low Energy (BLE) beacons to create a more accurate and faster IPS than traditional Bluetooth [9]. BLE beacons are small, low-power devices that broadcast their identity and location information to nearby devices and can reach up to 50 meters [27]. BLE beacons are suitable for indoor positioning applications because they can be placed strategically and overcome better interference issues and obstacles than Wi-Fi.

Despite the potential use of BLE for IPSs, some disadvantages still need to be overcome. It was observed some latency problems in the communication between the beacons and mobile devices, making it impractical for real-time tracking applications. Furthermore, multipath loss still affects the accuracy of BLE-based IPSs. Although it can handle obstacles better than Wi-Fi, the technology is still susceptible to signal degradation caused by multipath interference. Besides, even in a fixed environment, RSSI values exhibit high variability in space and time, further affecting the system's accuracy. Finally, the RSSI values are device-dependent and can vary significantly between different devices, making it challenging to create a general model for the system [27, 29].

2.2.3.3 Cellular Positioning Technology

Cellular positioning technology refers to the use of mobile networks to determine a device's location both indoors and outdoors. It is based on the principle of trilateration, which calculates the distance between the device and the cellular towers [24]. The technology has evolved over time, with the introduction of 2G, 3G, 4G, and now 5G.

Researchers, such as Lakmali and Dias [19] and Zhao [36], proposed techniques using this technology for indoor positioning. The results could have been more satisfactory, with an order of a hundred meters accuracy. Over the years, other researchers [15, 1] achieved better results, but it can be stated that 2G/3G/4G cellular technology is not optimal for indoor positioning.

With the introduction of 5G, cellular positioning technology is expected to see significant advancements [27]. The 5G network has a higher bandwidth, lower latency, and higher reliability. The requirement for a higher density of base stations in the same area due to the difficulty in penetrating walls results in closer proximity of the device to the base stations and a reduction in NLOS situations, further contributing to enhanced positioning accuracy [9, 27]. However, this high density of base stations is a cost-prohibitive for many, and the accuracy of the 5G technology currently available on mobile phones is still around a few meters [20].

2.2.3.4 Pedestrian Dead Reckoning

Mobile phones are equipped with low-cost MEMS sensors, such as accelerometer, magnetometer and gyroscope, allowing the device's movement and direction detection [3]. Pedestrian Dead Reckoning (PDR) uses these sensors to determine the device's location: the accelerometer measures the velocity of the movement, and the magnetometer and gyroscope are used to determine the direction of the movement [9].

While this method has the advantage of being simple and providing a high level of accuracy, it is often prone to errors and inaccuracies that need to be addressed using additional technologies. For example, it is necessary to estimate the initial position of the device with other technology since PDR cannot do that on its own [9]. Furthermore, the accuracy of PDR is also affected by factors such as the environment and the user's movement, as well as external factors such as the instability of the hand holding the phone [3].

Studies have shown that PDR with smartphones is not always effective in complex indoor environments. Many challenges and limitations exist, and while some solutions have been proposed to overcome these challenges (see [31] for examples), much more research and development are required to realise the potential of PDR in indoor positioning applications fully [3].

2.2.3.5 Magnetic Matching Positioning Technology

Magnetic Matching (MM) is a technology that uses magnetic fields to pinpoint the user's location. Using the smartphone's magnetometers, MM utilises the fingerprint method to match the current magnetic field readings with those stored in a pre-existing database.

This technology is known for its pervasiveness and cost-effectiveness. However, magnetic fields are affected by certain physical obstacles and external magnetic fields. On the other hand, magnetic field readings can vary based on the device being used, as presented in [3]. These challenges can be addressed by incorporating multiple devices to measure magnetic fields and supplementing MM with other technologies.

Ashfar *et al.* [3] had gathered several research studies and systems that improved MM using different fingerprint techniques and augmented with other technologies.

One practical optimised application of MM is the Indoor Atlas[2], which can achieve an accuracy range of 0.1 to 2 meters. It integrates MM with other built-in sensors, such as GPS and Wi-Fi, to produce even more accurate results [9].

[2] https://www.indooratlas.com/solutions/indooratlas-location/

2.2.3.6 Camera-based Positioning Technology

Camera-based positioning is a technique that uses a smartphone's camera to determine the user's location. The process starts by taking images of the current location and then comparing these images with a pre-existing database of images after extracting features. The extracted features are matched with the images in the database to determine the user's location [3, 9].

To ensure accuracy, the angle and orientation of the camera are also recorded and stored with the images [3]. This method has the potential to achieve high accuracy, with some studies reporting accuracy in the range of meters or even centimetres [9].

However, camera-based positioning has some significant drawbacks that must be considered. This image database must be comprehensive and up-to-date to ensure reliable results, which can be time-consuming and use a lot of memory.

In addition, the method requires a smartphone with a good camera and lighting conditions. It may not work well in environments with no distinguishable features, such as long corridors with white walls and brown doors. Some environments' complexity and dynamic nature can also confuse the algorithm, resulting in incorrect positioning results [3]. Another factor to consider is that camera-based positioning is not a low-power solution. It requires complex algorithms and large amounts of data, impacting battery life and overall performance [9].

2.2.3.7 Visible Light Positioning Technology

Visible light positioning (VLP) is a technology that leverages the light emitted by LED lights to determine a device's location. The technology works by using the light from LED lamps to transmit identified information to the device, which then processes this information to determine its position. Using an appropriate localisation technique, such as ToF or RSSI, the device then uses this data to triangulate its position based on multiple LED lamps' signal strength and arrival time [3].

VLP offers various advantages, including enhanced indoor precision, lower power consumption, and increased privacy and security. It can also be integrated into established LED lighting infrastructure, making it an affordable alternative for various applications [3].

Liu *et al.* [21] studied the use of VLP in indoor positioning and comprehensively explained the system's functioning. They also highlighted other relevant research papers and identified the main challenge with VLP as the LOS requirement between the device and LED lamps. In other words, the device must have a direct view of the LED lamps without any obstructions, which can be a challenge in indoor environments with many obstacles, such as walls, furniture, and people. EyeLight, proposed by Nguyen *et al.* [25], attempted to address this issue by exploring shadows, but the accuracy was still lower than in a LOS environment.

2.2.4 Ultra-Wideband

Ultra-Wideband (UWB) technology has been a promising innovation for decades, with potential applications in various fields. Recently, UWB has been integrated into smartphone models, unlocking new possibilities for indoor positioning. In 2019, smartphone manufacturers began integrating UWB chips, but it was not until last year, 2022, that Apple[3] and Google[4] made their APIs available for third-party access.

UWB operates based on the IEEE 802.15.4a [5] standard, a low-power, low-data-rate wireless communication technology that uses a radio frequency band of 3.1 to 10.6 GHz to transmit data at high speed [35]. Given its high bandwidth and low frequency, UWB is resistant to interference and can function in NLOS environments, making it a desirable technology for indoor positioning. Also, the technology has a low power consumption compared to other indoor positioning technologies, which is particularly important for mobile devices where the battery is a critical factor [2].

UWB technology stands out for its highly accurate positioning capabilities, often capable of reaching centimetre-level accuracy [35]. UWB can use the following algorithms to pinpoint the device's location: AoA, ToA, TWR, TDoA and RSSI. Alarifi *et al.* [2] gathered multiple UWB systems developed and proposed over the years and analysed the limitations of each one of those algorithms. They concluded the following:

- AoA is cost-intensive and requires the maintenance of large equipment, making it less practical than other algorithms. It is prone to error accumulations and requires strong collaboration between sensors;
- RSSI is ideal for narrowband signals but lacks accuracy for UWB signals. In contrast, ToA demonstrates improved accuracy in UWB systems;
- ToA and TDoA are the ones that have better accuracy;
- ToA relies on clock synchronisation and is affected by clock jitter;
- TDoA and AoA can be combined to reduce the errors of the two methods;
- TDoA is a better option when the nodes are out of sync with the synchronised reference nodes;
- Hyper algorithms, which merge the benefits of multiple algorithms, prove the most effective for UWB positioning.

A recent project described in [14] combined the ToA algorithm with an SVM-based classification method and CNN-based error elimination method to mitigate NLOS issues and achieve an accuracy of approximately 1-2 centimetres, demonstrating the potential of UWB technology for accurate indoor positioning.

UWB's short-range nature enhances its security compared to other wireless technologies. However, this property also limits its applicability to outdoor or large indoor spaces. While the

[3] https://developer.apple.com/documentation/nearbyinteraction/
[4] https://developer.android.com/guide/topics/connectivity/uwb
[5] https://standards.ieee.org/ieee/802.15.4a/3571/

cost of UWB devices may be higher compared to other wireless technologies, this value gradually decreases as the technology becomes more widely researched and utilised [2].

2.3 Related Technologies

Smart lighting management systems, such as Philips Hue[6], Wipro[7], and GE Link[8], offer many benefits, including convenience, energy efficiency, and customisation. Despite these advantages, the system's reliance on user input to adjust lighting can limit its ability to respond to environmental changes and the user's location in real-time.

While motion sensors can be used for some level of automation, they lack the accuracy of indoor positioning technologies like UWB. Integrating UWB into smart lighting management systems would significantly enhance their capabilities, providing a more dynamic and efficient lighting experience. Currently, there are no known smart lighting management systems that incorporate UWB technology for user location detection and lighting adjustments.

Fluid One[9] is a potential and related technology that could be launched this year. It is a type of hub that centralises all compatible smart devices through a single iPhone app. Currently, in the midst of a funding campaign, the app will only be released if it reaches its required funding goal.

Fluid One utilises UWB technology to allow users to control compatible smart devices, such as humidifiers, vacuum cleaners, consoles, doorbells, and more, simply by pointing their mobile phones at the device in question. It should be noted that while the app aims to work with a variety of smart devices, it may not offer the level of control and customisation specifically tailored towards lighting systems.

2.4 Summary

This chapter delved into the advancements in lighting management systems, including lighting controls, smart lighting management systems, and the principles of human-centric lighting, showcasing their important features, advantages and limitations. Following that, it was presented various indoor positioning technologies that can be used with mobile phones, with a specific emphasis on UWB technology.

Also, it was highlighted the limitations of current lighting management systems and the potential for incorporating indoor positioning technologies to enhance their performance and provide a more dynamic and efficient lighting experience.

[6] https://www.philips-hue.com/
[7] https://www.wiproconsumerlighting.com/
[8] https://www.wink.com/products/ge-link-connected-led-bulbs/
[9] https://www.geeky-gadgets.com/home-automation-07-09-2022/

Chapter 3

Problem and Proposed Solution

In this section, the existing problems with current lighting management systems are described, along with a discussion of how the proposed solution aims to resolve them. This includes analysing the requirements for the implementation. Finally, a high-level overview of the system is presented.

3.1 Existing Problems

In a contemporary world characterised by the increasing integration of technology and automation into everyday life, ensuring that these advancements are designed with user-centricity in mind is essential. This principle holds particular significance for smart lighting systems, as they greatly influence user well-being and productivity.

Within the context of Tridonic, there were outlined some problems regarding these topics. While their current systems incorporate automation features such as presence and daylight sensors, as well as time scheduling, there is a recognised need for a more personalised experience through the utilisation of user context. User context refers to the information that can be gathered about the user and environment, such as current location and preferences.

Despite their rapid evolution in recent years, smart lighting systems continue to face several challenges that impede their efficacy and user satisfaction. These include:

Lack of context awareness and personalisation: Although current LMSs can detect user presence or movement, they fail to identify the specific user or their preferences. In large office settings, for instance, it is common for individuals to have different preferences regarding lighting. Failure to address these differences means that the lighting system may not provide the optimal lighting conditions for each user, resulting in discomfort and reduced productivity;

Inaccurate indoor positioning: Current LMSs rely on technologies such as BLE and Wi-Fi, which are not accurate enough to provide precise indoor positioning. Consequently, these systems may mistakenly alter lighting conditions in the wrong room or overlook adjustments altogether. This discrepancy can result in user frustration and dissatisfaction with the system;

Complex and technical user interfaces: Existing LMSs often have interfaces that demand a certain level of technical expertise from users in order to operate them effectively. This requirement is a barrier for many individuals, particularly those with limited technological familiarity. Consequently, users may experience frustration and dissatisfaction when navigating and interacting with the system. A truly context-aware system should minimise the need for extensive user input and instead offer a simple interface for effortless adjustment of settings.

In an effort to address these problems, Tridonic initially explored the use of BLE technology for precise indoor user localisation due to its wide availability on mobile phones and potential for integration with lighting systems. Several tests were conducted in various scenarios, distances, and with different boards to assess the limitations of this technology (refer to Appendices A and B for more details). Unfortunately, the results of these tests proved to be unsatisfactory. The accuracy of BLE relied heavily on the positioning of the board antennas, and there were errors of up to five meters in some instances. This presented a significant challenge that needed to be overcome.

By addressing these challenges, smart lighting systems can provide users with a more comfortable and customised experience, taking advantage of the latest technological advancements to ensure optimal performance.

3.2 Proposed Solution

The proposed solution addresses the abovementioned challenges by integrating context awareness into Tridonic LMS. This Proof of Concept (PoC) is designed to test and evaluate the feasibility of incorporating context awareness into the system. By doing so, it aims to determine whether this idea can achieve the desired outcome.

The PoC takes advantage of the recent integration of UWB into mobile phones, which makes the technology more accessible and affordable. UWB offers precise indoor positioning, allowing the system to determine the user's location within a few centimetres accurately. Consequently, the system can identify the appropriate luminary to adjust the lighting conditions accordingly. It revolves around dynamically adapting lighting conditions based on user preferences and the current environmental context, ensuring not only a comfortable but also an appropriate lighting environment.

This solution aligns with human-centred lighting principles, prioritising productivity and well-being. It delivers a comfortable and suitable lighting environment that caters to individual needs and preferences.

To ensure the successful implementation of the solution, the system's main requirements have been meticulously defined and categorised into two groups: functional and non-functional. Both are described in the tables 3.1 and 3.2, respectively. These requirements serve as guidelines, guaranteeing the system's intended functionality and optimal performance throughout development.

Table 3.1: Functional Requirements Description

ID	**Title**
F-R01	The system must accurately detect changes in the user's position.
F-R02	The system must adjust the lighting within a defined area accordingly to estimated user's position/preferences.
F-R03	The system must use UWB technology to estimate the user's location.
F-R04	The system must allow users to store their lighting preferences.
F-R05	The system must allow users to create customised lighting scenes.
F-R06	The system must allow users to manually adjust the lighting conditions at any given time, overriding any pre-set preferences or automated adjustments.
F-R07	The system must allow the commissioner to enter information about a new installation.
F-R08	The system must allow administrators to edit the information about an existing installation.
F-R09	The system must allow administrators to manage users' access privileges.
F-R10	The system must establish a communication with the LMS to manage luminaires.
F-R11	The system must retrieve necessary information about the installation from the LMS.
F-R12	The application must be able to run in the background.
F-R13	The system should calibrate the current lighting using the mobile phone's light sensors.

Table 3.2: Non-Functional Requirements Description

ID	Title
NF-R01	The application must be compatible with iOS operating systems.
NF-R02	The application must be compatible with Android operating systems.
NF-R03	The application should be optimised to minimise battery consumption, not exceeding 10 percent of battery usage per hour under normal operating conditions.
NF-R04	The system must be able to estimate the distance between the UWB anchor and the user's device, with a maximum acceptable margin of error of 1 meter.

To fully understand the solution, consider the following scenario:

1. A user enters a room with the proposed application installed on their mobile phone, which is equipped with a UWB chip. The room has already been identified and registered in the system;
2. Once the user's position is determined using the UWB technology, the system retrieves the user's preferences associated with the specific zone from the database;
3. The LMS listening to the system receives the necessary information and adjusts the lighting conditions accordingly.

By showcasing this POC to premium customers and stakeholders, Tridonic can effectively communicate this solution's unique selling points and competitive advantages. They can highlight the enhanced user experience, increased efficiency, comfort, and personalisation of the context-aware lighting system. This demonstration enables Tridonic to showcase the potential impact and value of the solution to its target audience, positioning it as an innovative and desirable option in the market.

3.3 System Overview

To comprehend the solution's architecture and its components, it is essential to have an overview of the system. The system architecture, illustrated in Figure 3.1, is categorised into three main sections: on-premise components, cloud components, and user interfaces. Each section comprises several components that communicate with each other and are responsible for different tasks.

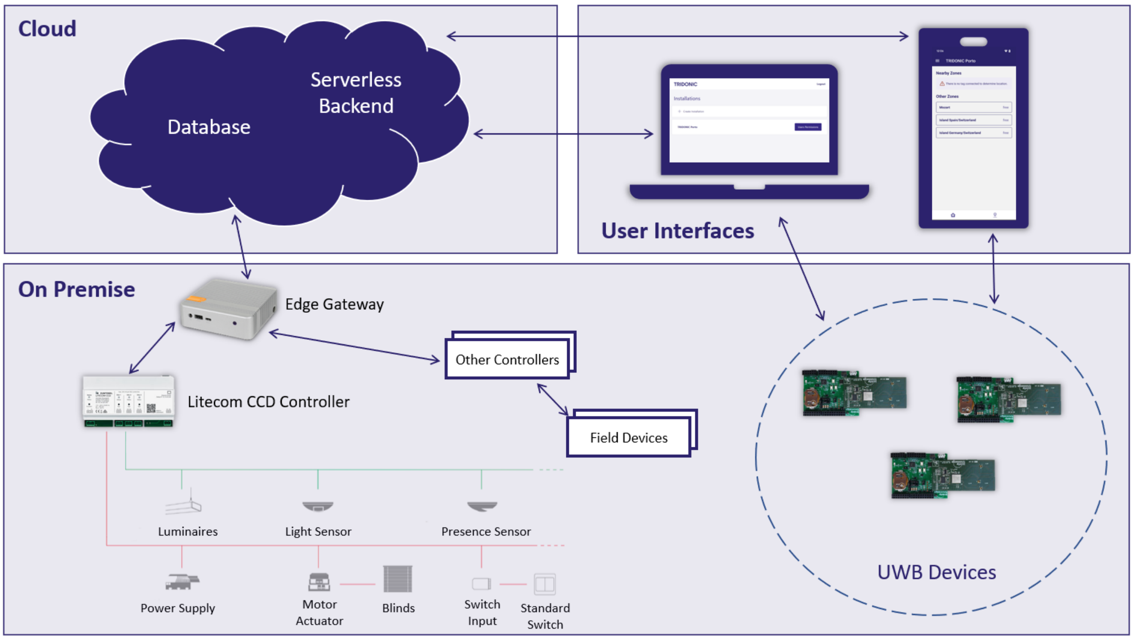

Figure 3.1: System Overview

On-premise components are installed within the customer's premises, including the UWB devices, edge gateway, and LITECOM CCD, along with its constituents. UWB devices estimate the distance between themselves and the mobile phone, enabling the system to determine the user's position. These devices communicate with the user interfaces to exchange the necessary information. The edge gateway acts as a bridge between the LITECOM CCD and the cloud components. Its primary functions are to transmit relevant installation data to the cloud and send commands to the CCD to adjust the lighting conditions. Finally, the LITECOM CCD is responsible for controlling the luminaires and other devices. While this specific controller was chosen for the system, the system is designed to be compatible with other controllers.

Proceeding to the cloud components, there are two primary elements. The serverless backend handles all the backend functionalities and incorporates triggers that prompt the edge gateway to update the lighting settings. Additionally, the database stores essential information about the luminaires and the users. Both cloud components communicate with the user interfaces and the edge gateway to seamless information exchange and trigger necessary actions.

Lastly, the user interfaces comprise the mobile application and the web application. Although initially not part of the plan, the decision was made to include a website during the development phase. The mobile application serves as the primary interface for users, enabling them to interact with the system, adjust preferences, and access real-time information. The web application, on the other hand, is primarily designed for administrators and commissioners, granting them system management capabilities and user administration functionalities. Both user interfaces establish communication with the cloud components to ensure up-to-date information retrieval, storage, and execution of necessary actions.

3.4 Summary

In this chapter, the problems with current smart lighting management systems were analysed, followed by a description of the proposed solution. This solution aims to address the identified challenges and enhance the performance of smart lighting management systems by providing more precise and context-aware lighting control.

This chapter also outlined the requirements of the proposed system, establishing the necessary functionalities for its successful implementation. Finally, an overview of the system was provided, offering a comprehensive visualisation of its components and interactions. With this overview, the reader can understand the system as a whole and how each component interacts with the others.

Chapter 4

Solution Implementation

To ensure the feasibility of the project since UWB is still recent on mobile devices, a kit was purchased to ensure the viability of this project - specifically, the MK UWB Kit Mobile edition 2.0[1]. This kit includes two UWB anchors along with their documentation and simple native mobile applications for both iOS and Android platforms that demonstrate the basic features of the UWB.

The initial phase of the development process focused on exploring the functionalities of the UWB Jetpack library and gaining a deeper understanding of the tags provided in the kit. This involved studying the protocols and mechanisms required to interact effectively with the UWB anchors. Throughout the project, only the UWB anchors and the associated protocol were utilised.

The project's implementation was supported by Firebase, a cloud-hosted NoSQL database, and the LITECOM system, a lighting management system. The implementation was divided into four main parts: the implementation of UWB ranging, the development of the web application, the creation of the mobile application, and the integration with the LITECOM system. Each part focused on a specific aspect of the overall functionality.

The following section analyses the software architecture of the solution, describing the different components and their interactions. Subsequent sections provide a detailed description of each part, highlighting their respective contributions to the project's overall functionality and user experience.

4.1 Software Architecture

The software architecture of the proposed solution, as depicted in Figure 4.1, exhibits a clear separation of components and their interactions.

[1] https://www.themobileknowledge.com/product/mk-uwb-kit-mobile-edition-2-0/

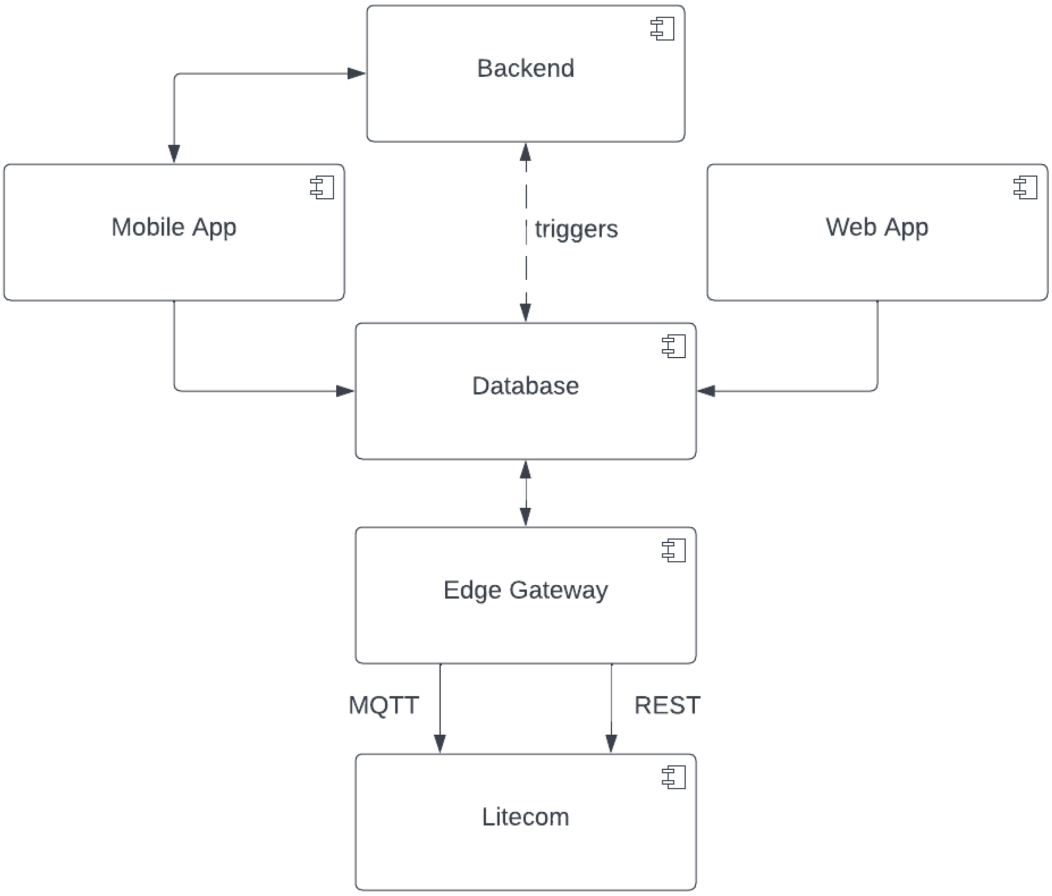

Figure 4.1: High-level Software Architecture

At the highest level, the system comprises the mobile and web applications (explained in sections 4.4 and 4.3, respectively). The mobile application is the primary user interface and communicates directly with the serverless backend. The web application, on the other hand, is a secondary user interface designed for administrators and commissioners. Both applications interact with the NoSQL database, Firestore, for data retrieval and storage. Modifications to the database can trigger specific actions in the backend, and the backend can also update the database with relevant information. Section 4.2 provides a more detailed explanation of the serverless backend using Cloud Functions and the utilisation of Firestore as the database.

The applications do not directly communicate with the LMS. Instead, the edge gateway acts as an intermediary, as described in section 4.5. The edge gateway listens for specific updates in the database and triggers actions in the LMS accordingly. It retrieves relevant information from the database and also stores data in the database. To interact with the LITECOM CCD, the edge gateway uses a REST and an MQTT protocols. Both protocols are utilised due to certain limitations of the LITECOM CCD, which will be explained in more detail in the mentioned section.

This software architecture allows for a clear separation of responsibilities and ensures a more scalable solution. For instance, if the LITECOM CCD needs to be replaced with a different controller, only the edge gateway would need to be modified, while the mobile and web applications can remain unchanged.

4.2 Firebase

Firebase[2], founded in 2011 by James Tamplin and Andrew Lee, has evolved into a comprehensive app development and infrastructure management tool suite. Firebase is now a powerful Backend-as-a-Service (BaaS) platform that provides developers with a wide range of tools and services for fast app development.

The platform offers various features, spanning user authentication, a real-time database, cloud storage, hosting, and other functionalities. By providing these services, Firebase simplifies backend management and significantly accelerates time to market for developers.

Operating within a serverless environment, it becomes an attractive choice for developers aiming to create scalable applications. Its serverless architecture removes the burden of managing server infrastructure, configuring setups, and planning for capacity. Firebase automatically handles scaling, resulting in improved efficiency and cost-effectiveness.

Overall, Firebase empowers developers by providing a robust and user-friendly platform that covers various aspects of app development, allowing them to build scalable and efficient applications while streamlining backend management and reducing time to market.

4.2.1 Authentication

Firebase Authentication[3] offers a secure and user-friendly authentication system, enabling seamless sign-in functionality for app users. It supports multiple authentication methods, including email and password sign-in, Google Sign-In and Facebook Login.

In this project's specific context, the chosen user authentication method was email and password sign-in. Users can sign in by providing their email address and password combination. Firebase Authentication handles potential errors that may arise during the sign-in process, such as invalid email, wrong password, and user not found.

By leveraging Firebase Authentication, it is ensured a secure and reliable sign-in experience. This authentication method offered a convenient and familiar way for users to authenticate themselves and gain access to the app's features and functionalities.

4.2.2 Cloud Firestore

Cloud Firestore[4] is a cloud-based NoSQL document database that offers a flexible data model and real-time capabilities. It is a serverless database that automatically scales to meet application

[2] https://firebase.google.com/
[3] https://firebase.google.com/docs/auth
[4] https://firebase.google.com/docs/firestore

demands. It also provides offline support for web and mobile applications, allowing users to access data even when offline.

Firestore stores data in JSON-like documents organised into collections. Each document contains key-value pairs, allowing for a schema-less structure. Documents can be queried by any of their fields and can be updated, deleted, and retrieved individually or in batches.

This database was chosen for this project due to its flexibility and scalability. It is a NoSQL database, meaning it does not require a predefined schema and allows for a flexible data model.

Firestore also offers real-time capabilities, which means that it can notify users of any changes to the data in real time. This feature was leveraged in the project to provide users with a seamless and responsive experience. For example, the lighting management system listens to changes in specific documents to update the lighting system's state accordingly. From a programming perspective, real-time data synchronisation and updates in Firestore can be accomplished using the *onSnapshot()* method.

The database structure in the project comprises four collections, each serving a specific purpose:

- ***users* collection**: stores user-related data, including the user's email and the installations they have access to;
- ***installations* collection**: holds installation-specific data such as the installation's name, floor plan's image and scale, ceiling height, information necessary to establish a connection to the lighting system, and a list of users with access to the installation. It also has two subcollections:
  - ***tags* subcollection**: indicates where each UWB tag is in the map;
  - ***zones* subcollection**: stores the information about different zones in the installation, including the zone type, the map coordinates of each zone, and the zones of the lighting system they represent.
- ***lmsController* collection**: has information retrieved from the lighting system for each installation. It includes the GPS coordinates, the ID, and the Controller's name. It contains two subcollections:
  - ***zones* subcollection**: organises subzones at different levels by hierarchy, with a maximum of three levels, along with their associated light devices;
  - ***allzones* subcollection**: stores all the zones and subzones at the same level, which one having the respective devices.
- ***scenes* collection**: responsible for storing lighting scenes. This includes details such as intensity, name, whether the scenes are part of the system or created by a user, and the user's ID if the case.

Despite their similar content, it is essential to maintain a clear distinction between the *zones* and *allzones* collections. These collections serve different purposes within the project's architecture. The *allzones* collection is utilised by the lighting management system to facilitate the process of changing the light settings. The system requires the zone identifier and the device ID to execute the necessary actions. Therefore, it actively listens for changes in the *allzones* collection to receive updates and respond accordingly.

On the other hand, the *zones* collection plays a crucial role in displaying the zones to users in a structured and hierarchical manner. By organising the zones based on their hierarchy, the project ensures a comprehensive and intuitive presentation of the zone system to users. This hierarchical organisation enables users to navigate and understand the relationships between different zones and subzones, enhancing their overall user experience.

The decision to structure the database into these four collections ensures an organised and comprehensive storage of relevant data, promoting efficient management and retrieval processes.

By duplicating specific data across multiple collections, the project can optimise query performance and minimise the need for complex joins or multiple queries to retrieve related information. This redundancy supports faster data retrieval and improves the application's overall responsiveness.

4.2.3 Cloud Storage

Cloud Storage[5] is a robust and secure object storage solution that empowers developers to store and deliver user-generated content, such as images and videos, efficiently and at a reasonable cost. It offers integrated security features that provide control over access to stored content, ensuring data privacy and protection.

Upon uploading an object to Cloud Storage, developers can generate a URL that provides access to the stored content. Saving this URL in a document field in Firestore allows seamless retrieval and access to the stored content.

In the context of this project, Cloud Storage was leveraged to store the floor plan images associated with each installation. To ensure the security and controlled access to the stored images, user authentication is required. Only authenticated users with the appropriate permissions are granted access to view and interact with the images. These images are subsequently retrieved and displayed within the web application, eliminating the need for additional server infrastructure solely dedicated to image storage. This approach optimises resource allocation and reduces costs.

4.2.4 Cloud Functions

Cloud Functions[6] provide a highly integrated serverless backend solution, offering the ability to create HTTP endpoints and trigger functions based on various events within the project. These events can include database writes (*@on_document_updated*), new document creations

[5] https://firebase.google.com/docs/storage
[6] https://firebase.google.com/docs/functions

(*@on_document_created*), and more. The functions can be written in different programming languages to cater to specific requirements.

In this project, Python was utilised to implement Cloud Functions in four distinct scenarios:

checkInstallations function is triggered when a new user is created in the database. It verifies if the user has access to any installations by checking if their email is listed in the installation's users. If access is granted, the function adds the corresponding installation to the user's list of accessible installations;

updateUserPermissions function is triggered when an installation document is updated. It ensures that changes to the list of users in the installation document are reflected in the respective user's list of installations. If a user is added to an installation, the function adds the installation to the user's list. Conversely, if a user is removed, the function removes the installation from their list;

updateZonesDevice function is triggered when a device within the *allzones* collection is updated. This function guarantees data consistency by updating the corresponding zone within the "zones" collection with the new device intensity. This ensures that the application and the lighting system have synchronised light information;

trilateration function, shown in Listing 4.1, is a callable function to calculate the user's position based on the positions of UWB tags, their respective distances to the user, the floor plan scale, and the ceiling height. Python was chosen for this function due to the availability of specific packages necessary for accurate position calculations that are not readily available in JavaScript.

```python
import math
from typing import Any
from easy_trilateration.model import Circle
from easy_trilateration.least_squares import easy_least_squares
from firebase_admin import initialize_app
from firebase_functions import https_fn

initialize_app()

@https_fn.on_call()
def trilateration(req: https_fn.CallableRequest) -> Any:

    try:
        points = req.data.get('points')

        if len(points) < 2:
            return None

        ceiling_height = req.data.get("ceiling-height")
```

```
        cellphone_height = req.data.get("cellphone-height", None)
        scale = req.data.get("scale")

        if cellphone_height is None:
            cellphone_height = 110 / scale

        distances = req.data.get("distances")

        arr = []
        for r, d in zip(points, distances):
            # adjust distance considering the height difference
            d /= scale
            d = math.sqrt(math.pow(d, 2) + math.pow(ceiling_height -
                cellphone_height, 2))
            arr.append(
                Circle(x=r[0], y=r[1], r=d),
            )

        result, meta = easy_least_squares(arr)

        return {
            "x": round(result.center.x, 2),
            "y": round(result.center.y, 2),
        }

    except Exception as e:
        raise https_fn.HttpsError(
            code=https_fn.FunctionsErrorCode.UNKNOWN, message="Error",
                details=e
        )
```

Listing 4.1: Trilateration Method

Using Cloud Functions in these scenarios, the project achieves seamless integration, event-driven functionality, and efficient data consistency.

4.3 Web application

The initial plan aimed to integrate the commissioning phase within the mobile app, which involves installing and setting up the system and saving all the information necessary to configure the lighting system properly. However, through the development, it was necessary to separate this phase and create a web application.

The LITECOM system, as explained in Section 2.1.4, is a lighting management system for controlling luminaires, blinds and other building devices. It operates on a zone-based architecture, where the installation is divided into distinct zones, each associated with specific devices. However, the system lacks awareness of the physical space in which it is installed, meaning that

LITECOM does not have information about the installation's floor plan or the physical location of each zone within the space. While it knows that a specific zone exists, it does not know where it is located and which zones are adjacent. This presents a challenge as the system needs to establish a connection between the lighting system and the actual layout of the installation. Without access to the floor plan and precise location information, the proposed solution cannot accurately determine the spatial context in which it operates and consequently cannot estimate the user's location.

During the commissioning phase, the commissioning personnel are responsible for locating the tags, creating zones on the map, and associating them with the physical tags and zones in the lighting system. A thorough research was conducted to explore various techniques for overlaying drawings on images within the mobile app, necessary for the zones part. Nevertheless, it became evident that implementing this functionally with React Native could be extremely challenging and time-consuming. Additionally, considering the limited interface dimensions of mobile phones, it was realised that it would be more practical and user-friendly to implement this functionality on a larger screen, such as a tablet or a computer. As a result, it was decided to transfer the commissioning phase to a web application, where administrators and commissioning personnel are empowered to create and modify installations more efficiently and comprehensively.

The web application was developed using React[7], a JavaScript library for building user interfaces. React is a component-based library that allows developers to create reusable UI components and build complex user interfaces from these components. React is also compatible with other libraries and frameworks, enabling developers to leverage a vast ecosystem of tools and packages to enhance their applications.

In the following sections, the four main components of the web application, namely the authentication, home page, user permissions page and installation flow are described in detail, highlighting their key functionalities and the technologies utilised.

4.3.1 Authentication

Authentication is crucial in this web application, as it ensures that only authorised users can access the system. It is important to guarantee that only users with the necessary permissions can create and edit installations in order to maintain data integrity and consistency. To achieve this, Firebase Authentication was used to implement a secure authentication system.

Users can sign in and register using their email and password (see Figure 4.2). The only restriction is that the email must be associated with a company domain (in this case, @*tridonic.com*).

[7] https://react.dev/

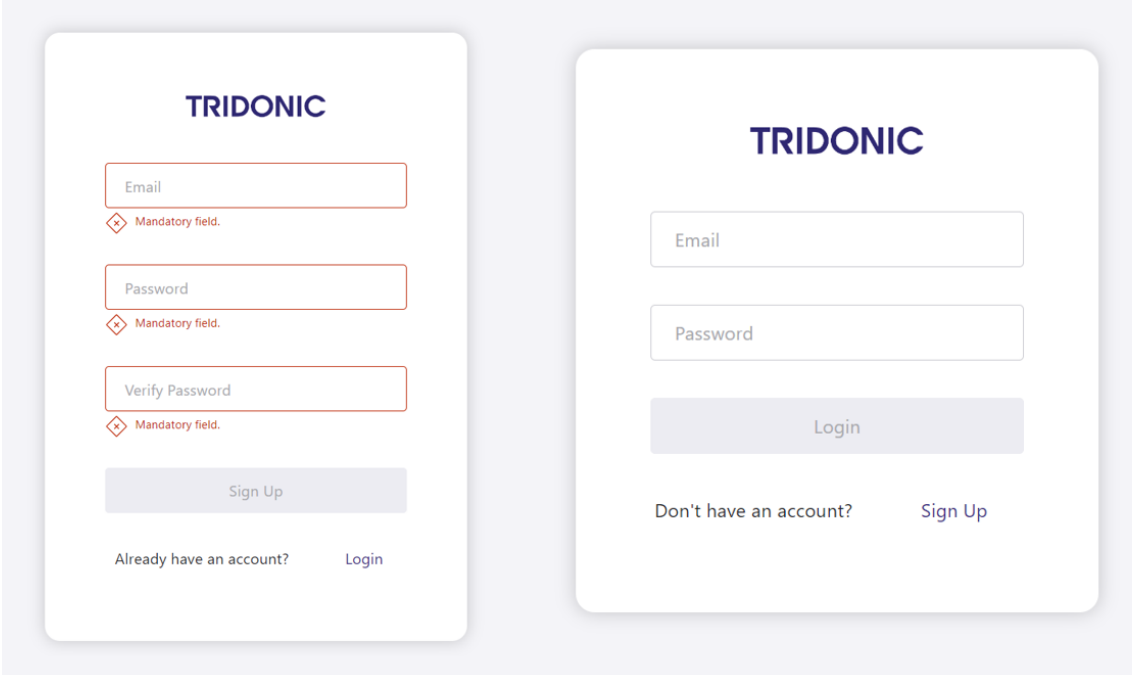

Figure 4.2: Authentication Pages: Sign Up in and Login (from left to right)

Upon successful authentication, users are redirected to the home page.

4.3.2 Home Page

The main page of the web application (shown in Figure 4.3) features a list of pre-existing installations. Additionally, it provides the option for users to create a new installation by clicking on the "Create Installation" button.

Each installation is displayed as a card with the installation name and a button to access the user permissions page. These cards are interactive. When a card is clicked, users are redirected to the installation flow page to proceed with the edition process.

Figure 4.3: Home Page

4.3.3 User Permissions Page

The user permissions page (see Figure 4.4) is a simple page that displays the list of users associated with the installation. It also provides the option to add or remove users from the installation. To simplify the process, users are identified by their email addresses. It is not necessary for them to already exist in the system. Whenever an authorised user creates an account, this particular installation will be automatically added to their list of installations.

Figure 4.4: User Permissions Page

4.3.4 Installation Flow

The installation flow is a multi-step process that guides users through creating or editing an installation. In this flow, an important React package, react-konva[8], was used to create complex canvas graphics. This package was used to display the floor plan image and to enable interactive functionalities such as polygon and line drawing, image dragging, and more. To further enhance user experience, a zoom-in, zoom-out, and scaling feature (shown in Figure 4.5) was incorporated to ensure seamless navigation and manipulation of the floor plan image.

Figure 4.5: Zoom-in, Zoom-out and Scaling Buttons

Users provide basic information about the installation on the first page of the installation flow, as visible in Figure 4.6. This information includes the installation name, the floor plan image and the data necessary to establish a connection with LITECOM (id, port, IP address and password token). All the information is stored in Firestore, and the floor plan image is uploaded to Cloud Storage. After submitting these details, a connection is established with the LITECOM system to retrieve the list of zones and devices associated with the installation. This information is stored in Firestore and used in the following steps of the installation flow.

[8] https://www.npmjs.com/package/react-konva

Figure 4.6: Installation Information

Afterwards, users are redirected to the measurements page, where they can define the scale of the floor plan image. Users draw a line in the map and specify the line length in meters, which is used to calculate the scale of the floor plan image. As demonstrated in Figure 4.7, the line is represented in the map in red, and the vertexes are represented as blue circles. Additionally, it is necessary to input the ceiling height since the tags are installed on the luminaries, and the ceiling height is used in the trilateration algorithm to calculate the user's position accurately.

Figure 4.7: Measurements Page

The subsequent step is to locate the tags on the map by dragging a pin image to the desired position. As shown in Figure 4.8, when pressed on the tag, the pin becomes red to facilitate the identification of the tag's position. For each tag, it is necessary to connect with the physical device using Web Bluetooth API[9]. This API enables BLE functionality, allowing scanning, connecting,

[9] https://developer.mozilla.org/en-US/docs/Web/API/Web_Bluetooth_API

and interacting with Bluetooth devices directly from the browser. The coordinates of each tag and the corresponding UWB device are stored in Firestore.

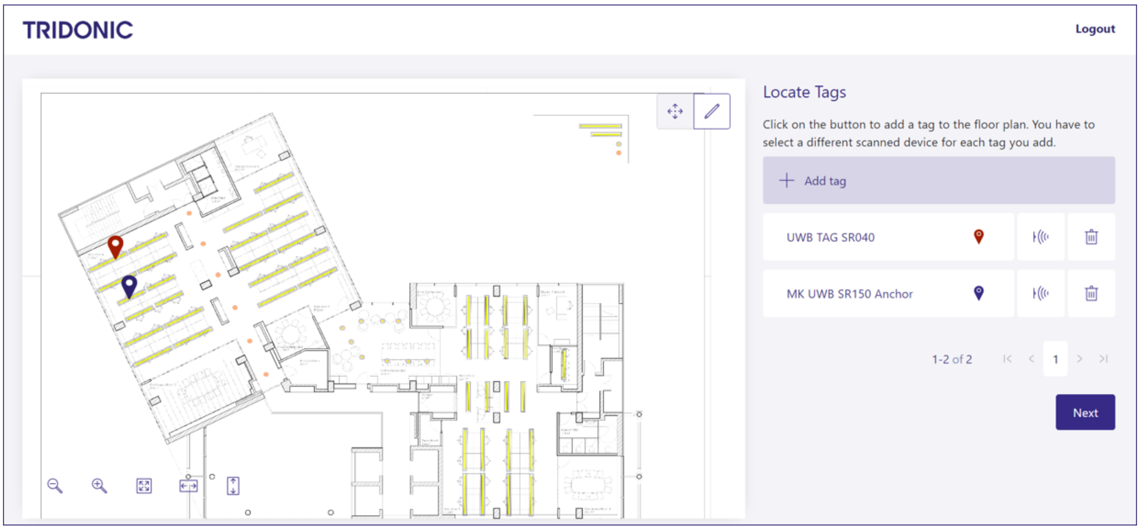

Figure 4.8: Tags Page

The last step involves creating the zones on the map, as shown in Figure 4.9. Users can draw polygons on the map, with a maximum of twelve vertices per polygon. Each zone has three mandatory fields: zone name, type, and the corresponding LITECOM zone. The last field is used to associate virtual zones in the LITECOM system with the physical zones in the application. Given that the LITECOM zones can have three levels, it is possible to designate a maximum of three LITECOM zones for each drawn zone. As seen in Figure 4.10, each LITECOM zone that has luminaries associated with it displays a button with a light bulb icon. When it is clicked, the luminaries associated with the zone blink, allowing users to verify if the LITECOM zones are correctly associated with the drawn zones.

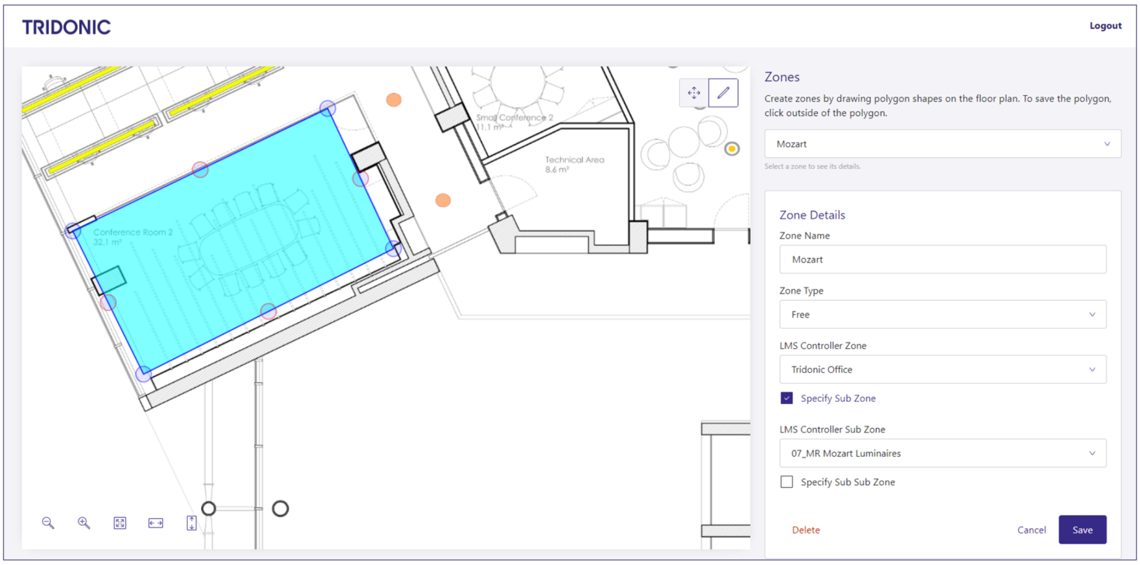

Figure 4.9: Zones Page

Figure 4.10: Light Bulb Button

After all these steps are completed, authorised users can access this installation and interact with the lighting management system using the mobile application.

Note that the same flow is utilised when editing the installation, with the fields pre-filled with the current values.

4.4 Mobile Application

The mobile application is responsible for establishing a connection with the UWB tags and providing users with a user-friendly interface to interact with the lighting system. With a focus on user experience, the interface was designed to be intuitive and straightforward, allowing users to navigate the application and perform the desired actions efficiently.

Currently, there is no package compatible with React Native to interact with UWB devices. Consequently, developing a native module became essential to enable such functionality on both iOS and Android platforms. However, due to time constraints, a decision was made to prioritise Android devices for implementation. The native module was developed in Java and used a Jetpack library. It was then integrated into the React Native application, thereby enabling the development of a cross-platform application capable of interacting with UWB devices.

The following sections describe the main components of the mobile app, giving particular attention to the UWB-related functionalities.

4.4.1 UWB Technology

Android Jetpack offers a collection of components and tools that ensure uniform and consistent functionality across different Android versions and devices. In 2022, Google added a Jetpack library[10] to allow third-party integration with UWB-enabled devices. Despite certain limitations and the partial availability of certain functionalities, the library provides a solid foundation for developing UWB-based applications.

UWB communication involves two devices: the Controller and the Controlee. The Controller assumes the initiator role, taking responsibility for determining the complex channel, which represents the specific channel on which a UWB device is currently active. On the other hand, the Controlee serves as the responder in the communication process.

To initiate ranging, the Controlee and Controller devices must establish mutual identification and exchange ranging parameters. However, the Jetpack library does not encompass this functionality, nor does it define an Out-of-Band (OoB) protocol for this purpose. Hence, it was decided to use Bluetooth Low Energy (BLE), the recommended technology, to accomplish this task. To ensure code reusability and avoid redundancy, this particular aspect was implemented using React Native rather than native code, thus providing a cross-platform solution that functions seamlessly on both iOS and Android platforms.

4.4.1.1 OoB Protocol

The OoB protocol is a BLE-based protocol that enables the exchange of ranging parameters between the Controller and Controlee devices.

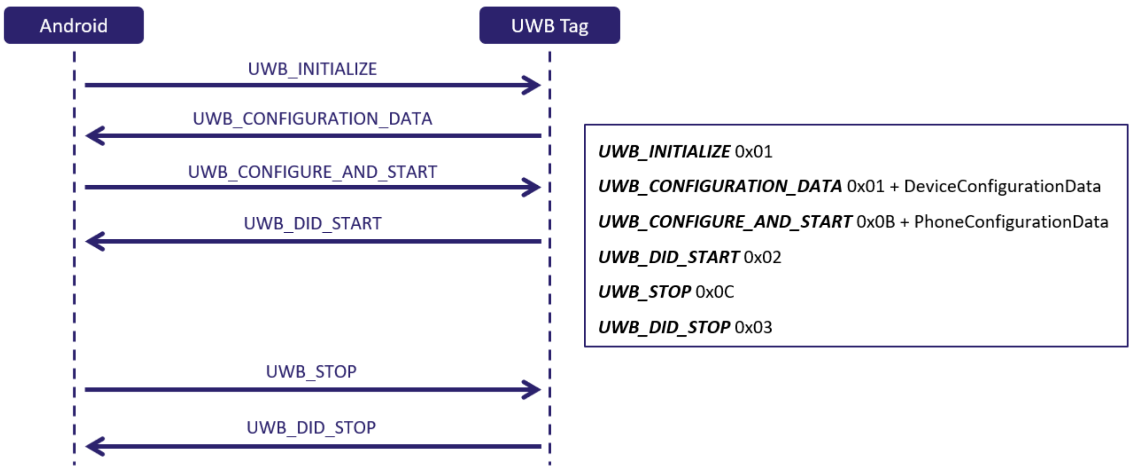

Figure 4.11: OoB Protocol

Once a connection is established with the UWB tag, it is possible to exchange messages and start the UWB ranging. As shown in Figure 4.11, the following sequence of messages is involved:

[10] https://developer.android.com/guide/topics/connectivity/uwb

1. The mobile phone initiates the process by sending an initialise message (*UWB_INITIALIZE*) to the UWB tag, including the device type information (*0x01* for Android devices);

2. In response, the UWB tag sends a *UWB_CONFIGURATION_DATA* message to the mobile phone. This message contains the supported capabilities and preferred parameters for the UWB ranging session configuration (*DeviceConfigurationData*, see Figure 4.12);

3. The mobile phone informs that it is ready to start the ranging session and transmits a *UWB_CONFIGURE_AND_START* message that includes the selected UWB ranging session configuration parameters, referred to as *PhoneConfigurationData* (see Figure 4.12). The *PhoneConfigurationData* is generated based on the received *DeviceConfigurationData*;

4. To confirm the successful start of the ranging session, the UWB tag sends a *UWB_DID_START* message to the mobile phone.

```
struct DeviceConfigurationData {
  uint8_t spec_ver_major[2];
  uint8_t spec_ver_minor[2];
  uint8_t chip_id[2];
  uint8_t chip_fw_version[2];
  uint8_t mw_versions[3];
  uint32_t supported_config_types;
  uint8_t supported_device_ranging_roles;
  uint8_t device_mac_addr[2];
}
```

```
struct PhoneConfigurationData {
  uint8_t spec_ver_major[2];
  uint8_t spec_ver_minor[2];
  uint32_t session_id;
  uint8_t preamble_id;
  uint8_t channel_number;
  uint8_t selected_config_type;
  uint8_t selected_device_ranging_role;
  uint8_t phone_mac_addr[2];
}
```

Figure 4.12: OoB Protocol Data Structures

To conclude the ongoing ranging session at any point, the Controller sends a *UWB_STOP* message to the tag. Subsequently, the UWB tag acknowledges the termination by transmitting a *UWB_DID_STOP* message.

4.4.1.2 UWB Ranging

The UWB Jetpack library provided by Android allows developers to implement UWB ranging sessions. However, one of the challenges encountered during implementation was the limited documentation provided for this library. Certain important aspects were either not adequately described or lacked proper examples. Additionally, it is worth noting that this library is relatively new, and, as a result, there is limited availability of error reports or community discussions regarding its usage on the internet, which made it difficult to identify and resolve specific issues encountered during the development process. Even though it was time-consuming, it was possible to overcome these challenges and implement the desired functionality.

The UWB Jetpack library can only be used in devices with Android 12 or higher and with hardware that supports UWB functionality. This library has a major limitation: the app initiating new UWB ranging sessions must remain in the foreground. If the app moves to the background while a session is ongoing, the ranging reports are suspended until the app returns to the foreground.

To include the UWB Jetpack library in the project, the following dependencies need to be added to the project's build.gradle file:

```
implementation "androidx.core.uwb:uwb-rxjava3:1.0.0-alpha05"
implementation "androidx.core.uwb:uwb:1.0.0-alpha05"
```

Before starting a ranging session, the app must have the required permission to perform UWB ranging, which is "*android.permission.UWB_RANGING*".

To initiate a ranging session using the UWB Jetpack library, the app needs to follow a specific sequence of steps. First, an instance of the UWB Manager needs to be created. Then, a ranging session configuration must be defined, including the necessary parameters described in *PhoneConfigurationData*, as mentioned in Section 4.4.1.2. These parameters are built based on the *DeviceConfigurationData*.

Once the ranging session configuration is prepared, it is created a *Flowable* object that emits ranging reports. This object is created by calling the ranging method of the *UWB Manager* instance, passing the ranging session information and client session scope as parameters. The *Flowable* object is then subscribed to an observer responsible for handling the received ranging reports. Additionally, the observer manages any errors that may occur during the ranging session. Listing 4.2, provided by the Jetpack library documentation, illustrates this flow.

```
private final UwbManager uwbManager;

// Retrieve uwbManager.clientSessionScope as a Single object
Single < UwbClientSessionScope > clientSessionScopeSingle =
    UwbManagerRx.clientSessionScopeSingle(uwbManager);
UwbClientSessionScope uwbClientSessionScope = clientSessionScopeSingle.
    blockingGet();

// Retrieve uwbClientSessionScope.prepareSession Flow as a Flowable object
Flowable < RangingResult > rangingResultFlowable =
    UwbClientSessionScopeRx.rangingResultsFlowable(clientSessionScope,
        rangingParameters);

// Consume ranging results from Flowable using Disposable
Disposable disposable = rangingResultFlowable
    .delay(1, TimeUnit.SECONDS)
    .subscribeWith(new DisposableSubscriber < RangingResult > () {
        @Override public void onStart() {
            request(1);
        }

        @Override public void onNext(RangingResult rangingResult) {
            doSomethingWithRangingResult(rangingResult);
            request(1);
```

```
            }


            @Override public void onError(Throwable t) {
                t.printStackTrace();
            }


            @Override public void onComplete() {
                doSomethingOnEventsCompleted();
            }
        });

    // Stop subscription
    disposable.dispose();
```

Listing 4.2: UWB Ranging

The RangingResult object in the provided code snippet plays a crucial role in the UWB ranging session implementation. It holds important information about the UWB device, including the distance in meters between the mobile phone and the UWB tag, as well as the azimuth and elevation angles relative to the other device. This distance is calculated using a Two-Way Ranging (TWR) method, explained in section 2.2.2.3, and, to determine the azimuth, it is used the Angle of Arrival (AoA) method, described in section 2.2.2.5. This information is sent to the Javascript code to be further processed and used to calculate the user's position.

Despite the challenges encountered, the UWB Jetpack library was successfully integrated into the project, and the desired functionality was implemented. The UWB ranging session was successfully initiated, and the ranging reports were received and processed. The next step was to integrate this functionality into the React Native framework.

4.4.1.3 Android Native Module

An Android Native Module[11] serves as a bridge between native APIs and React Native code, allowing the execution of native APIs code within JavaScript and facilitating data exchange between native APIs and React Native code. It is used when native capabilities that are not available in React Native are required or when native performance is superior. In this particular case, a native module was used to implement the UWB ranging session since there is no React Native package that supports this functionality.

The native module was implemented in Java. To effectively utilise it within the React Native framework, a series of steps were followed to integrate it seamlessly. Firstly, the native module was created for the desired platform, which in this case was Android. The module was designed to interact directly with the UWB tag using the native APIs provided by the platform. To expose the native module to React Native, a package was created extending the *ReactPackage* class.

[11] https://reactnative.dev/docs/native-modules-android

Within this package, the native module was added as a component. The native module extends the *ReactContextBaseJavaModule* class, which allows it to communicate with the React Native environment. Methods within the native module that need to be accessible from JavaScript are annotated with *@ReactMethod*, which allows React Native to identify and invoke these methods from JavaScript. Finally, the created package containing the native module was added to the *MainApplication.java* file, ensuring that the native module was registered correctly and ready for use within the React Native application.

Since the beginning of the project, it was decided to implement native code only for functionalities that were not readily available in React Native. Consequently, while the OoB protocol was implemented in React Native, the native module took responsibility for initiating the ranging session and receiving the ranging data. Note that in the development kit, the OoB protocol is implemented in the native code. However, it was decided to implement it in React Native to ensure code reusability and avoid redundancy.

To initiate the OoB protocol, it is mandatory to establish a BLE connection with the UWB tag. A specific React Native package called ***react-native-ble-manager***[12] was used to accomplish this, which enables functionalities such as scanning, connecting, and discovering BLE peripherals.

Within the React Native environment, a constant Bluetooth scanning method was employed to continuously monitor and store information about all available UWB tags within the installation. Whenever a new tag is found, or the user moves, a function responsible for handling the OoB protocol is invoked. User movement is detected using the mobile phone's accelerometer sensor data, as shown in Listing 4.3. Although the accelerometer sensor data might not be precise for determining the user's position, it proved adequate for detecting user movement.

```
import {
    SensorTypes,
    accelerometer,
    setUpdateIntervalForType
} from 'react-native-sensors';

setUpdateIntervalForType(SensorTypes.accelerometer, 1000);

accelerometer.subscribe(({ x, y, z }) => {
  handleAccelerometer({ x, y, z });
});
```

Listing 4.3: Acceleremoter Data Updates

The core component of the UWB ranging session is the *ranging()* function. This function consists of a loop that iterates over all the reachable UWB tags. For each UWB tag encountered, the mobile phone attempts to establish a BLE connection. Upon successfully establishing a connection, the initial OoB messages are exchanged between the mobile phone and the UWB tag. Once

[12]https://www.npmjs.com/package/react-native-ble-manager

the initiation of the ranging session is confirmed, the method responsible for starting the ranging session and receiving the ranging data of the native module (described in the Section 4.4.1.2) is called. The collected ranging information is then used to calculate the user's position. After getting a defined number of ranging positions (*RANGING_TIMES*), the ranging session is terminated, and the BLE connection is closed.

4.4.1.4 Integration of UWB in the Application

Within the application, the UWB ranging session is utilised by implementing a React context[13]. This approach ensures that every React component can access the UWB ranging session and constantly receives the most up-to-date information. By using a context, data can be passed through the component tree without manually passing props at every level. Additionally, any component dependent on the context will automatically rerender when the context's state changes. This is particularly beneficial for this project, as the user's position is continuously changing, and the application needs to be aware of the user's current position at all times.

During the initialisation of the context, a function is called to retrieve the zones and tags associated with the installation. On the one hand, the zones are used to determine where the user is in the installation. On the other hand, the retrieved tags are the only ones that the UWB ranging session will search for. This is done to avoid the ranging session from searching for all available UWB tags, which would be inefficient and time-consuming.

Whenever a new position is received, it is averaged with the previous position to ensure a more stable and consistent user position. Subsequently, the retrieved zones are ordered based on their distance from the user's position. The ordering process takes into consideration the number of tags connected to the application. If only one tag is connected, the azimuth angle and the distance between the tag and the mobile phone are utilised to calculate the user's coordinates. However, for improved accuracy, it is preferable to have multiple tags connected. In such cases, the trilateration algorithm determines the user's position based on the distances from multiple tags. Once the user's position is obtained, the zones are arranged in ascending order based on their distance from the user.

To accurately calculate the current user's coordinates, the distance between the user and each zone is calculated. This is accomplished by calculating the minimum distance between the user's coordinates and each polygon's segment. Using this information, the zones are categorised as nearby or far. If the distance is smaller than a predetermined value, the zone is classified as nearby; otherwise, it is categorised as a far zone. The ordered zones are stored within the context and can be accessed by any component in the application.

By utilising the UWB ranging session within the React context, the application ensures that all relevant components have access to real-time user position information, enabling seamless integration of the user's location data into various features and functionalities throughout the application.

[13] https://react.dev/learn/passing-data-deeply-with-context

4.4.2 Mobile Application Flow

The application comprises multiple screens, each serving a specific purpose. The following sections describe the application's main screens, functionalities, and the navigation flow between them.

4.4.2.1 Navigation

Navigation is a fundamental part of any mobile application. It enables users to move between different screens and access the application's various functionalities. To achieve smooth navigation, the mobile application leverages the capabilities of React Navigation[14], a powerful library that offers a range of navigators to cater to specific requirements.

Three main navigators are used in the application:

Stack Navigator [15] maintains a stack of screens, where new screens are added to the top of the stack and can be popped off to reveal the previous screens. This navigator is the default choice for the application and is used to navigate between the register and login screens and the zones and zone screens. It usually has an arrow on the top left corner to go back to the previous screen;

Bottom Tabs Navigator [16] uses a tab bar positioned at the bottom of the screen to navigate between screens. As shown in Figure 4.13, it was explicitly employed to navigate between the zones screen (left icon) and the current zone screen (right icon);

Figure 4.13: Bottom Tabs Navigator

Drawer Navigator [17] presents a navigation drawer on the side of the screen, accessible through gestures, that provides additional navigation options. Within the drawer navigator (see Figure 4.14), users can access their profile screen. Additionally, the drawer navigator is used to navigate between different installation workspaces. Each installation is represented by its name, and, if the user is physically in that installation, a pin is displayed next to the installation name.

[14] https://reactnavigation.org/
[15] https://reactnavigation.org/docs/stack-navigator
[16] https://reactnavigation.org/docs/bottom-tab-navigator
[17] https://reactnavigation.org/docs/drawer-navigator

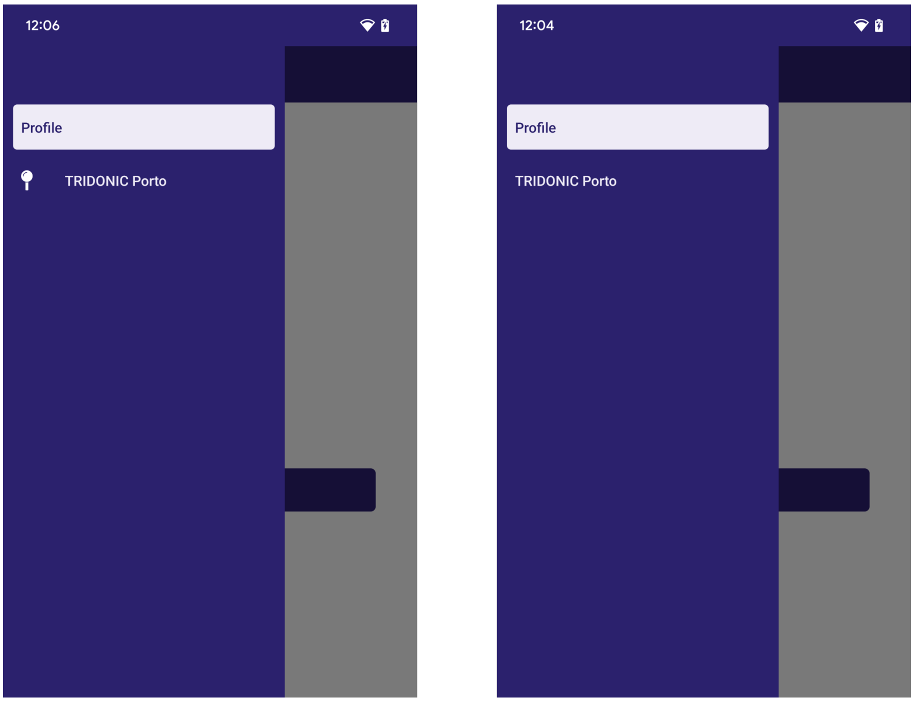

Figure 4.14: Drawer Navigator (on the left, the user is not in the installation and, on the right, the user is within the installation)

By employing these navigators, the application ensures a smooth and intuitive navigation experience for the user.

4.4.2.2 Authentication

Similar to the web application mentioned earlier, the mobile application also features two authentication screens: the login and sign-up screens (shown in Figure 4.15). Firebase Authentication is utilised to handle the authentication process.

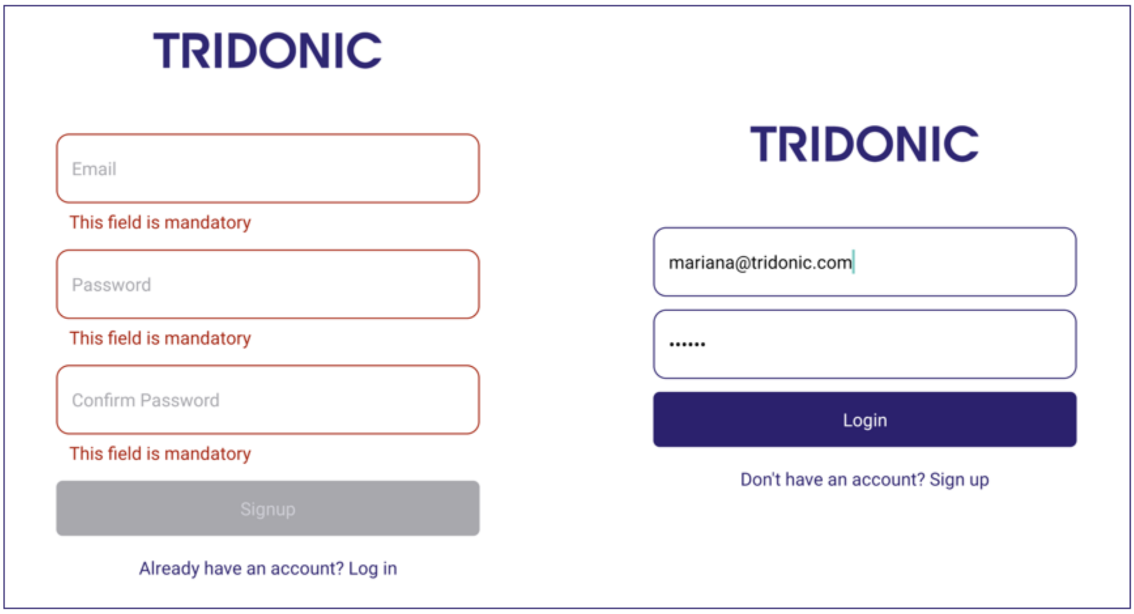

Figure 4.15: Authentication Screens

On the login screen, users can log in by entering their email address and password. However, access is restricted to email addresses with the domain "*@tridonic.com*". If a user does not have an account, they can create one by selecting the "Sign Up" button. This action redirects the user to the sign-up screen, where they are required to provide an email address and password. After clicking the "Sign Up" button, the user is authenticated and subsequently redirected to the profile screen.

4.4.2.3 Profile

Each user is assigned a profile that is used across all installations. This profile includes all the scenes that the user has created.

The profile screen, shown in Figure 4.16, can be accessed from the drawer menu. It provides a dedicated zone for managing scenes and a logout button.

The scenes are displayed in a scrollable list, presenting the name of each scene along with its corresponding lighting intensity. Users can click on each scene in the list to access the edit modal, where they have the option to modify the scene's name and adjust the lighting intensity. The lighting intensity is controlled using a slide bar with a range of values from 0 to 100. Additionally, the edit modal includes a bin icon that allows users to delete a scene if needed.

To create new scenes, users can simply click on the designated button. This action triggers a modal that allows them to input the desired name and set the lighting intensity for the scene.

All the scenes created by the user are stored in the database, ensuring real-time updates of the information. These scenes are accessible and can be utilised across all installations that the user has access to.

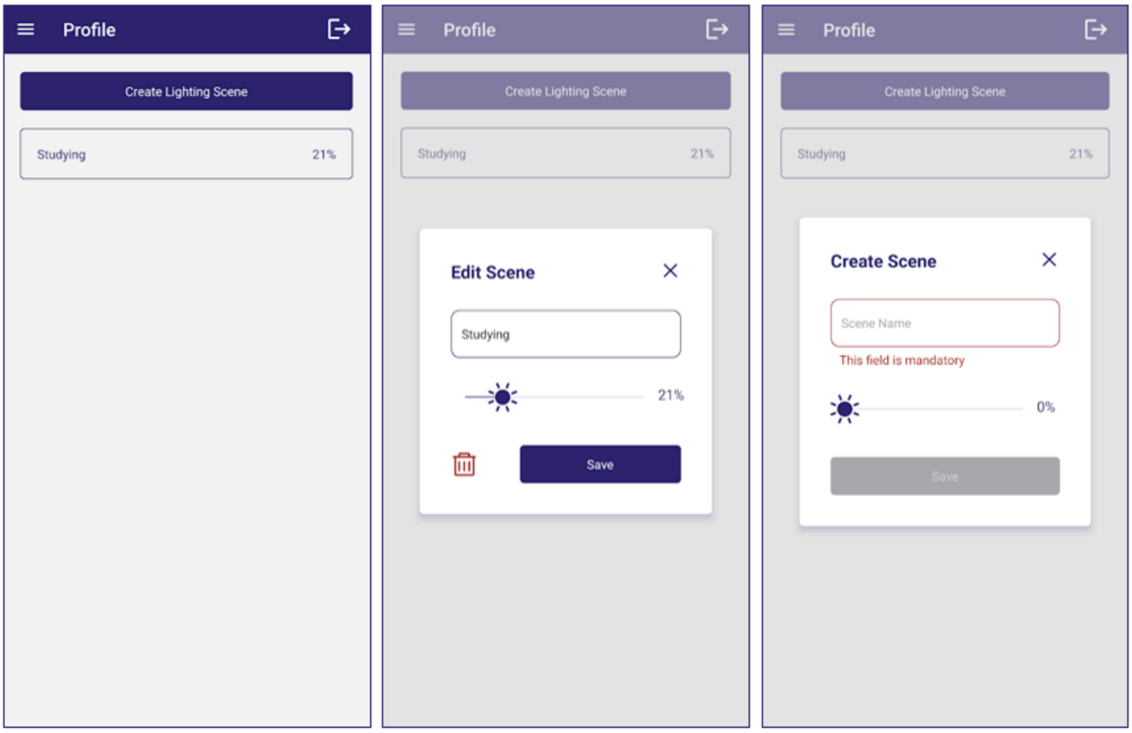

Figure 4.16: Profile Screen

4.4.2.4 Installation Workspace

The installation workspace serves as a central component of the mobile application, offering various functionalities tailored to the user's current location and preferences.

Upon successful authentication, the system retrieves the installations the user has permission to access and displays them in the navigation drawer. Simultaneously, the user's GPS coordinates are tracked to determine their current location and the installation they are in. The function used to retrieve the user's location is shown in Listing 4.4, and it is executed every 10 minutes.

```
import Geolocation from 'react-native-geolocation-service';
Geolocation.getCurrentPosition(
        ({ coords }) => {
                setPosition({ latitude: coords.latitude, longitude: coords.
                    longitude });
        },
        (error) => {
                console.error(error.code, error.message);
                setPosition({ latitude: null, longitude: null });
        },
        { enableHighAccuracy: true, timeout: 15000, maximumAge: 10000 },
);
```

Listing 4.4: Method used to retrieve the user's location

If the user is within an installation, a pin icon is displayed alongside the installation name in the navigation drawer, clearly indicating their current location. This feature facilitates easy identification of the active installation. The Ranging Context, explained in Section 4.4.1.4, is initialised only when the user is within an installation to optimise battery consumption and avoid unnecessary BLE scanning.

Moreover, the user's interaction with the luminaires is limited to the installation they are currently in. The application's primary focus is to enable context-based lighting control rather than direct control of individual luminaries. As a result, interaction with the luminaires of other installations is not necessary.

The installation workspace encompasses two key screens: the zones screen and the current zone screen. The latter is only displayed when the user is within an installation.

As demonstrated in Figure 4.17, the zones screen showcases all the zones within the installation, ordered based on their proximity to the user. These zones correspond to the ones described earlier in Section 4.4.1.4. Some messages can be displayed on the screen, depending on the user's location and the number of tags connected to the system. The messages are the following:

- "There is no tag connected to determine location.": if no tags are connected to the system, the user's location cannot be determined. Therefore, the zones cannot be ordered based on their proximity to the user;
- "There are no nearby zones.": if the user is not near any zone, the nearby zones list is empty;
- "You are not within the installation area, so you are currently in viewing mode.": in this case, it is similar to the first message, so the zones cannot be ordered based on their proximity to the user. Additionally, the user knows they cannot interact with the luminaires in the installation.

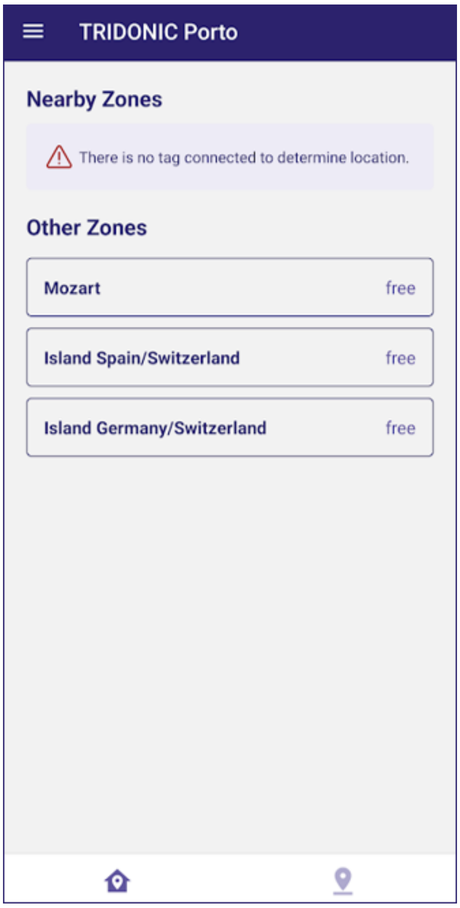

Figure 4.17: Zones Screen

Upon clicking on a zone, the user is redirected to the zone screen (refer to Figure 4.18). This screen provides various options for managing the zone, including selecting subzones and individual luminaires within the zone. Light settings can be modified for individual or all luminaires in the zone. These settings are disabled if the user is not within the installation area.

The zone screen also allows users to set their preferences, including the default scene, default zone, and default luminaire. These preferences enhance the user experience by preselecting the desired settings for quicker access. The dropdown menus display a star icon next to the user's favourite scenes, zones, and luminaires to identify them quickly. In the case of scenes, if a scene is a default system scene, it is labelled with a tag displaying the name "system." This labelling helps users differentiate between user-created scenes and predefined system scenes.

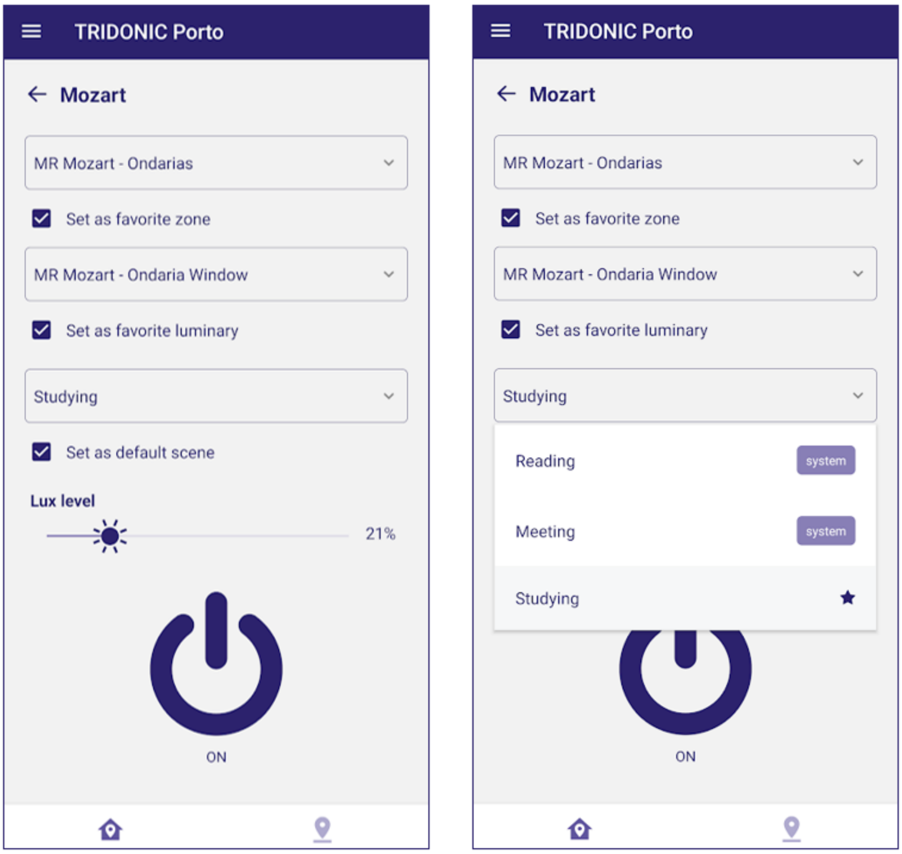

Figure 4.18: Zone Screen

The current zone screen is similar to the zone screen described above, but it explicitly represents the zone where the user is currently located. The screen automatically applies the user's set preferences to adjust the lighting accordingly when they enter the respective zone. Note that an error message is displayed to inform the user if the user is not near a known zone.

4.5 Edge Gateway

Two protocols were explored to facilitate communication with the LITECOM system: Message Queuing Telemetry Transport (MQTT) and Representational State Transfer (REST). The implementation of both methods is described in the following sections.

4.5.1 MQTT

Message Queuing Telemetry Transport (MQTT) is a lightweight messaging protocol designed to facilitate efficient communication between devices, particularly in scenarios with limited bandwidth or resources. MQTT operates on a publish/subscribe model, where devices can publish messages to a central broker, which then distributes these messages to interested subscribers.

In this model, publishers generate messages and send them to the broker without direct knowledge of the subscribers. Subscribers, on the other hand, express their interest in specific topics or types of messages by subscribing to them on the broker. The broker acts as an intermediary, receiving messages from publishers and forwarding them to the appropriate subscribers based on the topic of each message.

In this project, the LITECOM controller incorporates the broker for message handling. An asynchronous client in Python is used to establish a connection with the mentioned broker and receive relevant messages. The client subscribes to topics related to the intensity of luminaries. More specifically, it subscribes to topics with the format "*zones/+/devices/+/services/lighting/intensity*". The plus sign act as a single-level wildcard, which means that the client subscribes to all topics with the pattern "*zones/zoneId/devices/deviceId/services/lighting/intensity*", where *zoneId* and *deviceId* are the ids of the zone and device respectively. The client also ensures that it receives retained messages, in other words, the last messages published in a topic. This is useful because, upon connecting to the broker, it allows the client to obtain the last intensity values of each luminary.

Whenever a message is published on the specified topic, the client receives the message. If the received intensity value differs from the current value stored, the client updates the intensity value for the corresponding luminary in the Firestore database. This mechanism allows for real-time synchronisation of the luminary intensity values between the broker and the application, ensuring accurate and up-to-date information.

4.5.2 REST API

The LITECOM system offers a REST API that facilitates interaction with the luminaries. This API allows system integrators and service technicians to seamlessly integrate lighting management system functionality with third-party systems by providing a digital interface for executing control operations.

The REST API is utilised in two specific scenarios within this project:

- During the creation of an installation, a system function is invoked to retrieve all the relevant information, including devices and zones, from the LITECOM system. This retrieval process involves multiple *GET* requests to the REST API. The obtained data is then organised hierarchically in the Firestore database, ensuring efficient storage and retrieval;
- A Python function constantly monitors changes in the *allzones* collection of the corresponding installation in Firestore. When a document is updated, specifically when the intensity of a device is modified, this function is triggered. In response, a *PUT* request is made to the REST API, allowing for the adjustment of the intensity of the luminary or zone, depending on whether the change occurred at the luminary or zone level. To ensure that the function is only triggered by user-initiated changes in light intensity, it verifies the value of a flag called *updated*. If the flag is false, indicating that the change was made by the system, the function takes no action. Conversely, if the flag is true, denoting a user-initiated change, the function modifies the luminary's intensity or zone accordingly.

By leveraging the REST API provided by the LITECOM system, this project facilitates seamless integration with the luminaries, allowing for the retrieval of relevant information and the execution of control operations.

4.6 Summary

In this chapter, the focus was on the implementation of the proposed solution, beginning with a high-level architecture of the software. The architecture provided an overview of the system's components and their interactions.

Each system component was described in detail, outlining the tools required for its implementation and the challenges encountered during the development process. The descriptions mentioned the technical aspects of each component, highlighting the key functionalities and how they contribute to the overall system.

Throughout the implementation, various challenges were faced, which are discussed in this chapter. Among these challenges, the most prominent one was the limited documentation and exploration of the UWB Android API, making its integration with the application time-consuming.

By presenting the implementation details and discussing the challenges faced, this chapter comprehensively explained how the proposed solution was realised.

Chapter 5

System Validation

The validation of a system is crucial to ensure its accuracy, reliability, and adherence to the defined requirements. This chapter presents the validation process conducted for the implemented system, focusing on two key aspects: UWB technology validation and requirements validation. The UWB validation involved a series of tests to assess the accuracy and performance of the UWB technology in different scenarios. On the other hand, the requirements validation aimed to determine whether the system meets the functional and non-functional requirements defined earlier in the project.

5.1 UWB Validation

To validate the accuracy of the UWB technology, two sets of tests were conducted. The first set used two UWB anchors, while the second set used a UWB anchor and a UWB tag (in this case, an iPhone).

In the first set of tests, the focus was on assessing the performance of the UWB anchors in different scenarios, including open space and a setup with a wooden door in between. The results obtained from these tests are visually represented in two separate plots, namely Figures 5.1 and 5.2, where the distance is plotted along the y-axis in meters. In each plot, the blue line represents the median of the measurements obtained from all positions. The positions labelled as *a*, *b*, *c*, and *d* indicate different positions of the UWB antenna. For each position, 50 measurements were taken.

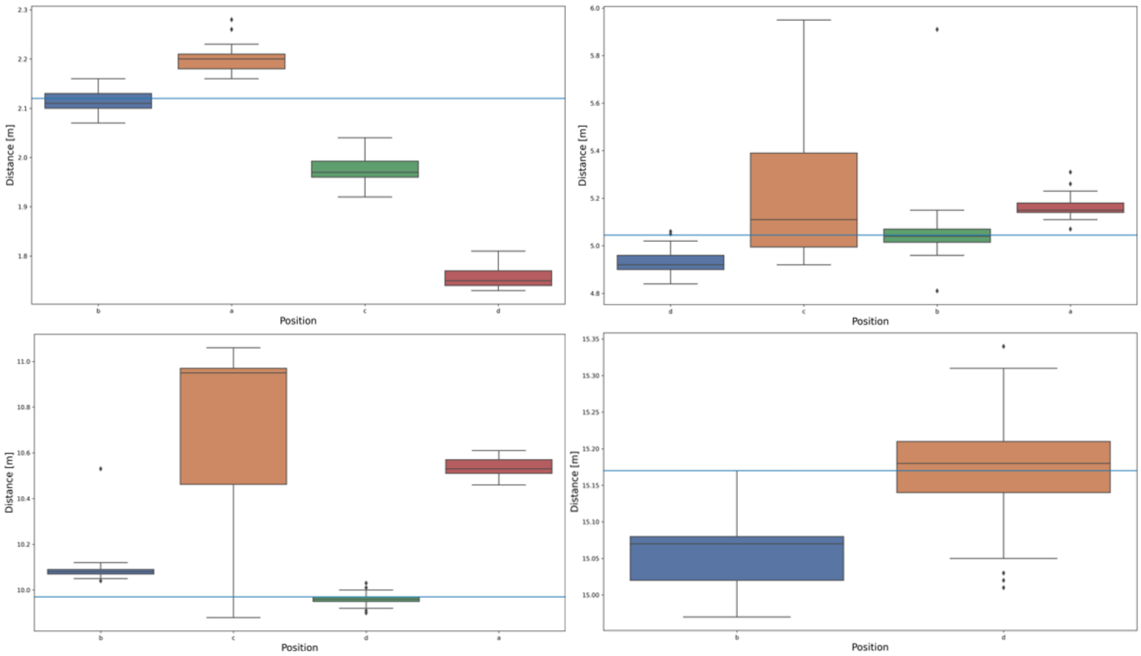

Figure 5.1: UWB Performance in an Open Space (from left to right, top to bottom: 2, 5, 10 and 15 meters)

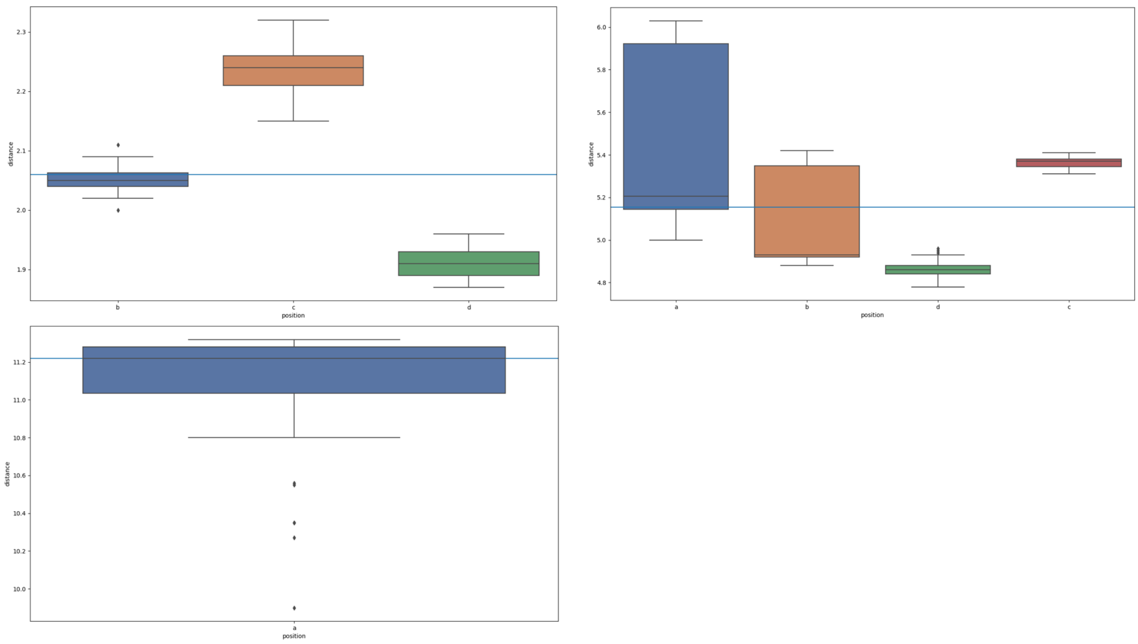

Figure 5.2: UWB Performance with Wooden Door (from left to right, top to bottom: 2, 5 and 10 meters)

To interpret the results, it is important to understand how to analyse box plots. Box plots provide a visual summary of the data distribution. The rectangular box represents the interquartile range and the median, while the whiskers extend from the box and indicate the data range. Outliers, shown as individual data points beyond the whiskers, indicate extreme values.

Based on the analysis of the plots, the following conclusions can be drawn:

- For the **Open Space Setup**:
  - UWB technology demonstrates high accuracy for distances ranging from 2 to 10 meters;
  - The rotation of the UWB antenna has some impact on the measured distance but does not significantly affect the overall accuracy;
  - UWB measurements beyond 10 meters are meaningful only when the transmit (Tx) power is close to the maximum. Otherwise, accurate measurements are limited to specific positions of the antenna;
  - The maximum range for UWB measurement in an open space was not precisely measured, but indications suggest it does not exceed 15 meters.
- For the **Door Setup**:
  - Introducing a wooden door between the UWB devices reduces the quality of the measurements, particularly for distances equal to or greater than 5 meters;
  - The devices appear to become more sensitive to the rotation of the antennas in this setup;
  - The range of accurate UWB measurements is reduced to slightly more than 10 meters. However, the precise range was not accurately measured.

The second set of tests focused on evaluating the performance of the UWB anchor and UWB tag, specifically using an iPhone as the UWB tag. This choice was made based on the availability of the iPhone as the only mobile phone with UWB capability during the initial phase of the tests. The tests were conducted in an open space without any obstacles between the devices.

To understand the behaviour of the UWB tag with different anchors, three tests were conducted, each involving different anchors. In total, there were 28 scenarios for each anchor, with each scenario involving a different position of the mobile phone. During these tests, the distances between the mobile phone and the anchors were deliberately varied, ranging from 0.5 to 14 meters, with an incremental step of 0.5 meters. The results of these tests are presented in Figure 5.3.

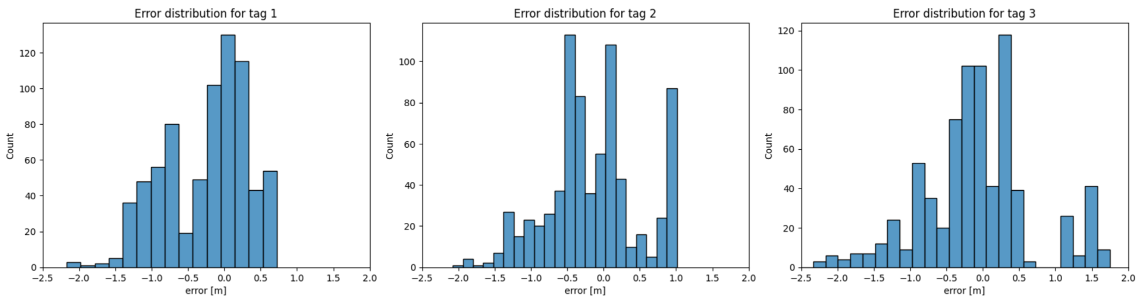

Figure 5.3: UWB Performance with mobile phone and three different tags

A histogram was utilised to analyse the accuracy of the measurements since it provides a better understanding of the magnitude and distribution of errors in the UWB measurements. This graph represents the distribution of errors between the actual distance (the distance between the anchors and the mobile phone) and the measured distance (estimated by UWB technology). The error is calculated as the difference between these two distances. The error distribution chart illustrates the frequency of errors falling within specific ranges.

Upon examining the histograms, it becomes evident that the UWB technology provides very accurate measurements, as evidenced by a maximum error of 2 meters. Furthermore, the majority of errors fall within the range of 0 to 1 meter, indicating a high level of precision in the measurements.

5.2 Requirements Validation

In a previous section (Section 3.2), the requirements for the system were defined. Some were mandatory to ensure the system's functionality, while others were optional and could be implemented if time allowed. In this section, the requirements are validated based on the implementation of the system and the results of the tests conducted.

5.2.1 Functional Requirements

In each section, the functional requirements are presented and validated. The status of each requirement is presented in Table 5.1.

Table 5.1: Functional Requirements Description and Status

ID	Title	Status
F-R01	The system must accurately detect changes in the user's position.	**Validated**
F-R02	The system must adjust the lighting within a defined area accordingly to estimated user's position/preferences.	**Validated**
F-R03	The system must use UWB technology to estimate the user's location.	**Validated**
F-R04	The system must allow users to store their lighting preferences.	**Validated**
F-R05	The system must allow users to create customised lighting scenes.	**Validated**
F-R06	The system must allow users to manually adjust the lighting conditions at any given time, overriding any pre-set preferences or automated adjustments.	**Validated**
F-R07	The system must allow the commissioner to enter information about a new installation.	**Validated**
F-R08	The system must allow administrators to edit the information about an existing installation.	**Validated**
F-R09	The system must allow administrators to manage users' access privileges.	**Validated**
F-R10	The system must establish a communication with the LMS to manage luminaires.	**Validated**
F-R11	The system must retrieve necessary information about the installation from the LMS.	**Validated**
F-R12	The application must be able to run in the background.	**Partially Validated**
F-R13	The system should calibrate the current lighting using the mobile phone's light sensors.	**Not Validated**

5.2.1.1 Requirement F-R01

The system must accurately detect changes in the user's position.

The system incorporates the use of the accelerometer sensor in the mobile phone to detect changes in the user's position. As described in section 4.4.1.3, the accelerometer sensor measures the acceleration experienced by the mobile phone in three dimensions: X, Y, and Z. By analysing the changes in acceleration values over time, the system can determine if the user has moved and trigger the estimation of the user's position. This enables the system to track the user's position and provide real-time updates continuously.

Therefore, **the requirement F-R01 is validated**.

5.2.1.2 Requirement F-R02

The system must adjust the lighting within a defined area accordingly to the estimated user's position/preferences.

Based on the previously estimated position of the user and their preferences, the system can adapt and regulate the lighting conditions within a designated area, as explained in the section 4.4.2.4. By comparing the user's estimated position with the positions of the zones and luminaires, the system can determine whether the user is situated within a particular zone.

When the user is within a specific zone, the system can proceed to adjust the lighting conditions for all the luminaires within that particular zone. This ensures a cohesive lighting experience throughout the zone, promoting consistency and desired ambience.

Additionally, if the user has expressed a preference for a specific luminaire, the system can adjust the lighting conditions solely for that particular luminaire. This individualised approach allows the user greater control over their lighting experience and aligns with their specific preferences.

Thus, **the requirement F-R02 is validated**.

5.2.1.3 Requirement F-R03

The system must use UWB technology to estimate the user's location.

Out of all the available technologies, UWB was chosen as the most suitable option for the system. As discussed in section 2.2.4, UWB is a wireless technology known for its exceptional accuracy in object localisation within a defined area. So, this system uses UWB to estimate the user's position accurately.

To implement UWB functionality, UWB chips are integrated into the user's mobile phone, while UWB anchors are strategically installed in the ceiling of the designated installation area. The UWB chips within the mobile phone establish communication with the anchors, allowing for the estimating of the user's position. The detailed process of this integration is elaborated upon in section 4.4.1.

Consequently, **the requirement F-R03 is validated**.

5.2.1.4 Requirement F-R04

The system must allow users to store their lighting preferences.

The system indeed offers users the ability to store their lighting preferences. As outlined in section 4.4.2.4, the mobile application features an intuitive interface that allows users to input and save their lighting preferences. These preferences include the default zone, default luminary, and preferred lighting scene. By storing these preferences, the system ensures that the automatic lighting adjustments align with the user's preferences.

Therefore, **the requirement F-R04 is validated**.

5.2.1.5 Requirement F-R05

The system must allow users to create customised lighting scenes.

The system allows users to create customised lighting scenes. As described in section 4.4.2.3, the mobile application has a section dedicated to the creation and edition of lighting scenes. With this feature, users can easily switch between different scenes to specific zones and luminaires without manually adjusting the intensity each time.

The ability to create and apply custom lighting scenes enhances user convenience and allows for efficient management of lighting settings throughout the installation.

Thus, **the requirement F-R05 is validated**.

5.2.1.6 Requirement F-R06

The system must allow users to manually adjust the lighting conditions at any given time, overriding any pre-set preferences or automated adjustments.

The system provides users with the capability to manually adjust the lighting conditions at any time. As detailed in section 4.4.2.4, an intuitive interface is available to users for this purpose in the mobile application. The interface includes a slider that enables users to control the brightness of the lighting conditions according to their preferences. Furthermore, users have the option to choose from a predefined set of lighting scenes, offering different lighting configurations. These manual adjustments take precedence over any pre-set preferences or automated adjustments, ensuring that the user's specific preferences are prioritised and implemented in real-time.

For this reason, **the requirement F-R06 is validated**.

5.2.1.7 Requirement F-R07

The system must allow the commissioner to enter information about a new installation.

The system includes a functionality that allows the commissioner to enter information about a new installation. This feature is facilitated through the web application, as described in section 4.3.4.

Within the web application, the commissioner can input various information related to the installation, including details about zones and UWB anchors. The commissioner can define and organise different zones within the installation area, specifying their boundaries and characteristics. Additionally, the commissioner can place UWB anchors in strategic locations within the installation area, which are crucial for the UWB positioning system to function accurately.

Therefore, **the requirement F-R07 is validated**.

5.2.1.8 Requirement F-R08

The system must allow administrators to edit the information about an existing installation.

The system utilises the flow described in the previous requirement to enable administrators to edit the information about an existing installation. The guide already contains the previously saved information, and administrators can access and modify this information as necessary. Once the modifications have been made, the administrator can save the updated information, ensuring the changes are stored and reflected accurately within the system.

Hence, **the requirement F-R08 is validated**.

5.2.1.9 Requirement F-R09

The system must allow administrators to manage users' access privileges.

The system includes functionality that enables administrators to manage user access privileges. Within the web application, administrators have access to a list of users who have been granted permission to access the installation. Administrators can add or remove users from the list through the interface presented in section 4.3.3, thereby managing their access privileges. This feature is crucial for ensuring that only authorised users can access the installation and make changes to the system.

In conclusion, **the requirement F-R09 is validated**.

5.2.1.10 Requirement F-R10

The system must establish a communication with the LMS to manage luminaires.

The system includes a communication mechanism with the LMS to manage luminaires effectively. This communication is established through a REST API, as detailed in section 4.5.2. Through the REST API, the system can send commands to the LMS, enabling various actions such as turning luminaires on or off and adjusting their brightness levels. The utilisation of a REST API enables efficient communication between the system and the LMS, ensuring that changes in the luminaires' settings can be executed in real-time and as needed.

Accordingly, **the requirement F-R10 is validated**.

5.2.1.11 Requirement F-R11

The system must retrieve necessary information about the installation from the LMS.

The system retrieves the necessary information from the LMS to have up-to-date data about the installation. As explained in section 4.5, the system utilises the REST API and the MQTT broker to retrieve information from the LMS.

The REST API allows the system to make specific HTTP requests to the LMS and obtain details such as the installation information, hierarchy of zones, existing luminaires, and other relevant data.

The MQTT broker enables the system to subscribe to specific topics and receive information about the luminaires' status. By subscribing to specific topics or channels within the MQTT broker, the system can receive updates and notifications from the LMS in real-time, ensuring that the system stays synchronised with the latest changes in the installation.

By combining the REST API and MQTT broker, the system can effectively retrieve and maintain up-to-date information from the LMS.

In consequence, **the requirement F-R11 is validated**.

5.2.1.12 Requirement F-R12

The application must be able to run in the background.

There are certain constraints regarding the UWB technology's functionality when the application is not actively running in the foreground.

The UWB technology used in the system requires the application to be in the foreground to function properly. If the application is closed, the UWB technology will stop working, and the application will not be able to detect the user's location. However, if the user keeps the application in the Recents screen (also known as the app switcher), it can continue running in the background, allowing the UWB technology to remain functional.

It is important to note that the term "background" here is sensitive and may refer to different states. In this context, the application being in the Recents screen allows it to continue running in a background-like state, enabling the UWB technology to operate.

Therefore, users need to be aware of this limitation and keep the application in the Recents screen if they want the UWB technology to continue functioning in the background. The system's design and functionality are constrained by the limitations of the UWB technology itself, and the application cannot run in the background without requiring user intervention to keep it in an active state.

For this reason and in this context, **the requirement F-R12 is partially validated**.

5.2.1.13 Requirement F-R13

The system should calibrate the current lighting using the mobile phone's light sensors.

This optional requirement, which involved calibrating the current lighting using the mobile phone's light sensors, was not implemented in the system. Time constraints primarily drove the decision not to include this functionality during the development process. While integrating this feature in React Native would not be overly complex, it would require additional time and effort to integrate it with the existing LMS.

Given the limited time available for the project, the decision was made to focus on implementing the system's core functionalities and leave this optional requirement for future work. The system's current design and architecture allow for the integration of this feature in the future, and it can be implemented without requiring significant changes to the existing system.

For this reason, **the requirement F-R13 is not validated**.

5.2.2 Non-Functional Requirements

In this section, the non-functional requirements are validated and discussed. The following table provides an overview of the non-functional requirements along with their current status.

Table 5.2: Non-Functional Requirements Description and Status

ID	Title	Status
NF-R01	The application must be compatible with iOS operating systems.	**Not Validated**
NF-R02	The application must be compatible with Android operating systems.	**Validated**
NF-R03	The application should be optimised to minimise battery consumption, not exceeding 10 percent of battery usage per hour under normal operating conditions.	**Validated**
NF-R04	The system must be able to estimate the distance between the UWB anchor and the user's device, with a maximum acceptable margin of error of 1 meter.	**Validated**

5.2.2.1 Requirement NF-R01

The application must be compatible with iOS operating systems.

As justified in section 4.4, it was not feasible to meet this requirement. Integrating the UWB technology into the system required developing the application using the native programming languages of the mobile operating systems. This means that even when using a cross-platform framework like React Native, the UWB functionality would still need to be implemented in native code specific to each platform.

Considering the limited time available for the project, the decision was made to prioritise the development of the application for Android devices and exclude iOS support. This decision was influenced by the fact that Apple does not allow the use of ambience light sensors in production applications, which would prevent the implementation of one of the system's requirements.

Therefore, due to technical limitations and time constraints, **the requirement NF-R01 is not validated**.

5.2.2.2 Requirement NF-R02

The application must be compatible with Android operating systems.

Section 4.4 provides detailed information on the development of the application using React Native and the implementation of UWB functionality through the Android Jetpack Library. The application was successfully installed and executed on Google Pixel 6 Pro, running Android 13.

In consequence, **the requirement NF-R02 is validated**.

5.2.2.3 Requirement NF-R03

The application should be optimised to minimise battery consumption, not exceeding 10 percent of battery usage per hour under normal operating conditions.

Battery consumption was a concern throughout the development of the application. The following strategies were implemented to minimise power consumption:

- The application only accesses the GPS module at intervals of 10 minutes. Since the GPS is only used to determine if a user is within an installation, continuous access is unnecessary;
- The UWB ranging is only activated when the user is within an installation. This avoids unnecessary battery consumption when the user is outside the range of the UWB anchors. In addition, the UWB ranging is only activated when the user is moving, which is detected through the accelerometer sensor.

Battery Historian[1] was chosen as the tool to analyse the battery consumption of the application. It provides a detailed analysis of the battery consumption, including the power usage of each system component. The tests were conducted under demanding conditions, with the application running in the foreground, the screen continuously turned on, and the mobile phone connected to the Wi-Fi network with Bluetooth enabled. The tests aimed to simulate real-world usage by frequently moving the mobile phone to trigger UWB distance estimation.

The test duration was 1 hour, and the results were recorded and presented in the following table.

Table 5.3: Battery Consumption Test Results

Metric Name	**Value**
Screen on Time	1h
Interactive Time	1h
Screen On Discharge Rate	7.34 %/hr (Discharged: 8%/hr)
Battery Percentage Consumed: React Native App	3.99%
Battery Percentage Consumed: SCREEN	3.82%
Battery Percentage Consumed: CELL	0.88%
Battery Percentage Consumed: IDLE	0.86 %
Battery Percentage Consumed: ANDROID_SYSTEM	0.13%

[1] https://developer.android.com/topic/performance/power/setup-battery-historian

Based on the conducted tests and analysis, it can be concluded that the application has a relatively low battery consumption, with a discharge rate of approximately 8% per hour. This value is considered low, especially considering the demanding test conditions that involved keeping the screen continuously turned on and the mobile phone consistently moving to trigger the UWB distance estimation.

Specifically, the application itself is responsible for 3.99% of the total battery consumption, indicating efficient power management. The screen, intentionally kept on throughout the test, accounts for 3.82%

Overall, the results obtained from the tests indicate that the battery consumption of the application is lower than the maximum acceptable value of 10% per hour. So, **the requirement NF-R03 is validated**.

5.2.2.4 Requirement NF-R04

The system must be able to estimate the distance between the UWB anchor and the user's device, with a maximum acceptable margin of error of 1 meter.

Several tests were made to validate the accuracy of the UWB distance estimation. The tests were performed in a controlled environment, with the UWB anchor placed on a table and the user's device at a distance of 2, 5, 10 and 15 meters. The results of the tests are presented in the following plots.

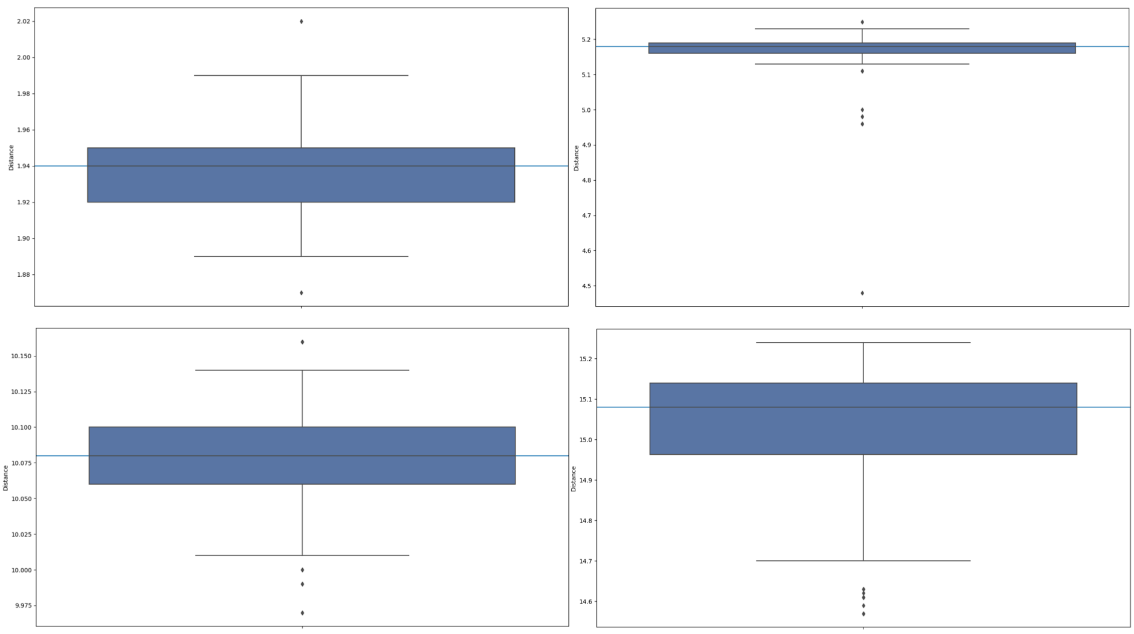

Figure 5.4: UWB Distance Estimation Measurements (from left to right, top to bottom: 2, 5, 10 and 15 meters)

The results obtained from the tests indicate a high level of accuracy in the distance estimation, with a maximum margin of error of 0.5 meters. The consistent accuracy is reflected in the median measurements, which fall within a range of 0.2 meters. These results are particularly promising in an optimal scenario without obstacles between the UWB anchor and the user's device.

In the app, we employ the average distance calculated from four measurements to ensure robustness in distance estimation. This approach helps mitigate the potential impact of outliers and enhances the system's overall accuracy.

While the accuracy may be slightly affected in real-world scenarios with obstacles, there are strategies to overcome this challenge. One practical approach is to increase the number of UWB anchors placed in the room. By utilising multiple anchors, the system can triangulate the user's position based on the distances measured from different anchors. This enhances the accuracy and robustness of the positioning system, even in the presence of obstacles.

So, despite the potential impact of obstacles and other factors on the accuracy of UWB distance estimation, the system can determine the user's location with a maximum margin of error that falls within an acceptable range of less than 1 meter. Therefore, **the requirement NF-R04 is validated**.

5.3 Summary

In summary, this chapter on system validation is divided into two main aspects: the validation of UWB technology and the assessment of system requirements. The UWB technology was thoroughly tested through various scenarios, demonstrating its high accuracy and precision, despite minor susceptibility to antenna rotation and obstacles.

Following that, the system requirements were analysed, carefully examining each requirement and determining its validation status based on the system's implementation and test results. Most functional requirements were successfully met; however, a few were not implemented due to time constraints but could be considered for future iterations.

Overall, the system validation ensures the implemented solution meets standards, provides accurate positioning using UWB technology, and satisfies the defined functional and non-functional requirements. As a result, the system provides an efficient and customisable context-based lighting solution.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The thesis presented a proof of concept for context-based lighting control, aiming to deliver users a personalised and automated lighting experience. By accurately estimating the user's location within an installation, the system automatically interacts with the appropriate luminaires based on the user's preferences.

In this document, chapter 2 provided the necessary concepts and knowledge to understand the solution, which was then described in detail in Chapters 3 and 4. The successful development of the solution is highlighted in Chapter 5, where the system is validated and tested.

A key design principle of the system is its modularity, allowing integration with any controller. This flexibility is particularly relevant in the current era of rapid technological evolution. As lighting management systems evolve, users often prefer to upgrade their existing installations by replacing old controllers rather than opting for entirely new installations. The proposed system can accommodate both wired and wireless controllers, providing a smooth transition and extending the lifespan of the existing infrastructure.

A significant achievement of this project is the successful implementation of UWB technology. Despite its limited documentation and recent introduction in mobile applications, a React Native Module was developed to incorporate UWB functionality and accurately estimate the user's location. The native code is reusable in other contexts, allowing the integration of UWB in other applications. UWB demonstrated high accuracy and reliability in the tests performed. The use of UWB in mobile applications is still in its early stages, and this project contributes to the growing body of work in this area. The successful implementation of UWB indicates its potential for a wide range of applications, particularly in precise indoor positioning systems.

Moreover, this project is inherently multidisciplinary, encompassing device, embedded, edge, cloud, web and mobile technologies. It required a lot of research and learning, which was a great challenge. The project was developed in a real environment, which allowed us to test the solution in a real scenario. The results obtained were very positive, demonstrating the solution's feasibility.

While the proposed solution is not yet market-ready, it serves as a robust proof-of-concept demonstrating the viability of context-based lighting control. The system successfully integrates various technologies and platforms to provide a customisable and user-centric lighting experience. Tridonic can leverage this achievement to showcase the potential of context-based lighting control to stakeholders and guide future development endeavours.

6.2 Future Work

Despite this project's significant achievements, a few areas could benefit from further improvement.

Firstly, the current system implementation only supports a single user within a zone, disregarding the preferences of other users present. To overcome this limitation and create a more inclusive experience, it is essential to implement a mechanism that addresses multiple user preferences or determines user control in shared spaces. One possible solution is to introduce a hierarchical system, assigning different levels of control based on user roles or privileges. Alternatively, a mechanism can be developed to recognise the first user in a zone as the primary controller of the lighting. Resolving this challenge will avoid conflicts and enable a seamless lighting experience for all users, considering their individual preferences.

Another area for future work is the implementation of light calibration using the mobile phone's light sensors. Although not integrated into the current version due to time constraints, incorporating this feature would be a valuable addition to the system's functionality. By calibrating the lighting based on the ambient conditions, the system can dynamically adapt and provide a more responsive and context-aware lighting experience without the need for extra hardware.

In conclusion, this thesis represents a significant advancement towards achieving context-based lighting control. The modular design, successful integration of UWB technology, and comprehensive validation of the system all contribute to showcasing the potential of this concept. With further development and refinement, this system has the potential to revolutionise our interaction with and control of lighting in various environments, leading to a more intuitive, responsive, and user-centric lighting experience.

References

[1] Belaabed Abdelghani and Gao Qiang. A novel fingerprinting positioning approach in urban cellular networks. In *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pages 155–159, June 2016.

[2] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A. Al-Ammar, and Hend S. Al-Khalifa. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors*, 16(5):707, May 2016. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[3] Imran Ashraf, Soojung Hur, and Yongwan Park. Smartphone Sensor Based Indoor Positioning: Current Status, Opportunities, and Future Challenges. *Electronics*, 9(6):891, June 2020. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[4] P. Bahl and V.N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 2, pages 775–784 vol.2, March 2000. ISSN: 0743-166X.

[5] Peter Boyce. Editorial: Exploring human-centric lighting. *Lighting Research & Technology*, 48(2):101–101, April 2016. Publisher: SAGE Publications Ltd STM.

[6] Liang Chen, Heidi Kuusniemi, Yuwei Chen, Jingbin Liu, Ling Pei, Laura Ruotsalainen, and Ruizhi Chen. Constraint Kalman filter for indoor bluetooth localization. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1915–1919, August 2015. ISSN: 2076-1465.

[7] Liang Chen, Heidi Kuusniemi, Yuwei Chen, Ling Pei, Tuomo Kröger, and Ruizhi Chen. Information filter with speed detection for indoor Bluetooth positioning. In *2011 International Conference on Localization and GNSS (ICL-GNSS)*, pages 47–52, June 2011. ISSN: 2325-0771.

[8] Liang Chen, Ling Pei, Heidi Kuusniemi, Yuwei Chen, Tuomo Kröger, and Ruizhi Chen. Bayesian Fusion for Indoor Positioning Using Bluetooth Fingerprints. *Wireless Personal Communications*, 70(4):1735–1745, June 2013.

[9] Ruizhi Chen and Liang Chen. Smartphone-based indoor positioning technologies. *Urban informatics*, pages 467–490, 2021. Publisher: Springer.

[10] Ivan Chew, Dilukshan Karunatilaka, Chee Pin Tan, and Vineetha Kalavally. Smart lighting: The way forward? Reviewing the past to shape the future. *Energy and Buildings*, 149:180–191, August 2017.

[11] Dominika Cupkova, Erik Kajati, Jozef Mocnej, Peter Papcun, Jiri Koziorek, and Iveta Zolotova. Intelligent human-centric lighting for mental wellbeing improvement. *International Journal of Distributed Sensor Networks*, 15(9):1550147719875878, September 2019. Publisher: SAGE Publications.

[12] Craig DiLouie. *Advanced Lighting Controls: Energy Savings, Productivity, Technology and Applications*. CRC Press, January 2021. Google-Books-ID: t3QWEAAAQBAJ.

[13] Marc Füchtenhans, Eric H. Grosse, and Christoph H. Glock. Smart lighting systems: state-of-the-art and potential applications in warehouse order picking. *International Journal of Production Research*, 59(12):3817–3839, June 2021. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207543.2021.1897177.

[14] Zefu Gao, Yiwen Jiao, Wenge Yang, Xuejian Li, and Yuxin Wang. A Method for UWB Localization Based on CNN-SVM and Hybrid Locating Algorithm. *Information*, 14(1):46, January 2023. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[15] Christian Gentner, Estefanía Muñoz, Mohammed Khider, Emanuel Staudinger, Stephan Sand, and Armin Dammann. Particle filter based positioning with 3GPP-LTE in indoor environments. In *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pages 301–308, April 2012. ISSN: 2153-3598.

[16] Kevin W. Houser and Tony Esposito. Human-Centric Lighting: Foundational Considerations and a Five-Step Design Process. *Frontiers in Neurology*, 12, 2021.

[17] KW Houser, PR Boyce, JM Zeitzer, and M Herf. Human-centric lighting: Myth, magic or metaphor? *Lighting Research & Technology*, 53(2):97–118, April 2021. Publisher: SAGE Publications Ltd STM.

[18] Ali Khalajmehrabadi, Nikolaos Gatsis, and David Akopian. Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges. *IEEE Communications Surveys & Tutorials*, 19(3):1974–2002, 2017. Conference Name: IEEE Communications Surveys & Tutorials.

[19] B.D.S. Lakmali and Dileeka. Dias. Database Correlation for GSM Location in Outdoor & Indoor Environments. In *2008 4th International Conference on Information and Automation for Sustainability*, pages 42–47, December 2008. ISSN: 2151-1810.

[20] Christos Laoudias, Adriano Moreira, Sunwoo Kim, Sangwoo Lee, Lauri Wirola, and Carlo Fischione. A Survey of Enabling Technologies for Network Localization, Tracking, and Navigation. *IEEE Communications Surveys & Tutorials*, 20(4):3607–3644, 2018. Conference Name: IEEE Communications Surveys & Tutorials.

[21] Xiangyu Liu, Lei Guo, and Xuetao Wei. Indoor Visible Light Applications for Communication, Positioning, and Security. *Wireless Communications and Mobile Computing*, 2021:e1730655, August 2021. Publisher: Hindawi.

[22] Fazeelat Mazhar, Muhammad Gufran Khan, and Benny Sällberg. Precise Indoor Positioning Using UWB: A Review of Methods, Algorithms and Implementations. *Wireless Personal Communications*, 97(3):4467–4491, December 2017.

[23] Francisco G. Montoya, Antonio Peña-García, Adel Juaidi, and Francisco Manzano-Agugliaro. Indoor lighting techniques: An overview of evolution and new trends for energy saving. *Energy and Buildings*, 140:50–60, April 2017.

[24] Khuong An Nguyen, Zhiyuan Luo, Guang Li, and Chris Watkins. A review of smartphones-based indoor positioning: Challenges and applications. *IET Cyber-Systems and Robotics*, 3(1):1–30, 2021. Publisher: Wiley Online Library.

[25] Viet Nguyen, Mohamed Ibrahim, Siddharth Rupavatharam, Minitha Jawahar, Marco Gruteser, and Richard Howard. Eyelight: Light-and-Shadow-Based Occupancy Estimation and Room Activity Recognition. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 351–359, April 2018.

[26] A Pandharipande and GR Newsham. Lighting controls: Evolution and revolution. *Lighting Research & Technology*, 50(1):115–128, January 2018. Publisher: SAGE Publications Ltd STM.

[27] Günther Retscher. Indoor Navigation—User Requirements, State-of-the-Art and Developments for Smartphone Localization. *Geomatics*, 3(1):1–46, March 2023. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[28] Smart Lighting Market Analysis - Industry Report - Trends, Size & Share. https://www.mordorintelligence.com/industry-reports/smart-lighting-market. Last accessed on 2nd February 2023.

[29] Santosh Subedi and Jae-Young Pyun. A Survey of Smartphone-Based Indoor Positioning System Using RF-Based Wireless Technologies. *Sensors*, 20(24):7230, January 2020. Number: 24 Publisher: Multidisciplinary Digital Publishing Institute.

[30] Editorial Team. What is Ultra-Wide Band (UWB) Technology?, July 2021.

[31] Nestor Michael Tiglao, Melchizedek Alipio, Roselia Dela Cruz, Fawaz Bokhari, Sammia Rauf, and Saad Ahmad Khan. Smartphone-based indoor localization techniques: State-of-the-art and classification. *Measurement*, 179:109349, July 2021.

[32] Tobias Roberts Writer, Rise. Smart Lighting: A Complete Guide. https://www.buildwithrise.com/stories/complete-guide-to-smart-lighting, April 2021. Last accessed on 2nd February 2023.

[33] Pengtao Wu. Comparison between the Ultra-wide Band based indoor positioning technology and other technologies. *Journal of Physics: Conference Series*, 2187(1):012010, February 2022. Publisher: IOP Publishing.

[34] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2599, 2019. Conference Name: IEEE Communications Surveys & Tutorials.

[35] Fusang Zhang, Zhaoxin Chang, Jie Xiong, Junqi Ma, Jiazhi Ni, Wenbo Zhang, Beihong Jin, and Daqing Zhang. Embracing Consumer-level UWB-equipped Devices for Fine-grained Wireless Sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4):194:1–194:27, January 2023.

[36] Yilin Zhao. Standardization of mobile phone positioning for 3G systems. *IEEE Communications Magazine*, 40(7):108–116, July 2002. Conference Name: IEEE Communications Magazine.

[37] Zumtobel. Impact of light. https://zumtobel.us/learn/impact-of-light/. Last accessed on 2nd February 2023.

[38] Zumtobel. Litecom software. https://litecom.zumtobel.com/. Last accessed on on 1st July 2023.

Appendix A

Indoor Positioning Accuracy: Bluetooth-Based Evaluation with 2 Anchors

A.1 Device Positioning

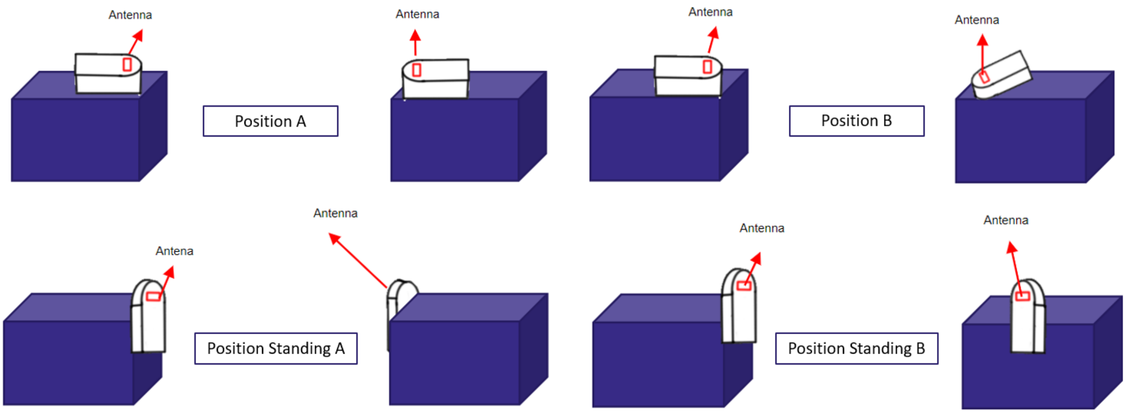

Figure A.1: Device Positioning Explanation

A.2 Measurements: Anchor 1 (DK)

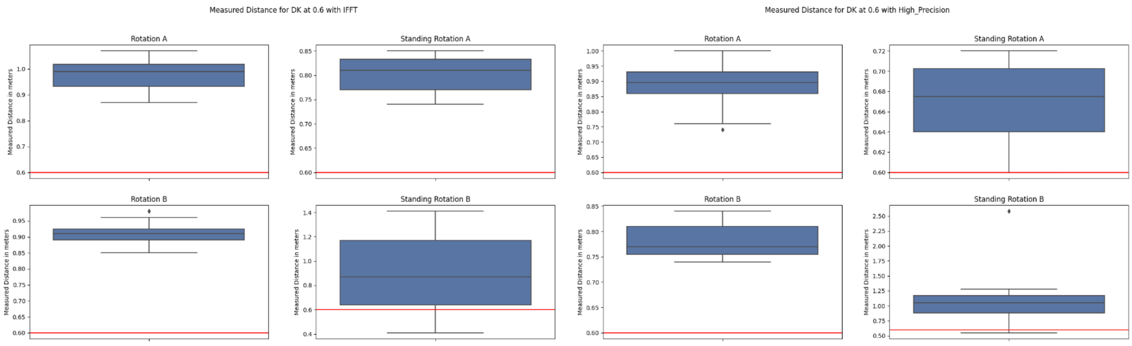

Figure A.2: DK Measurements - distance: 0.6m

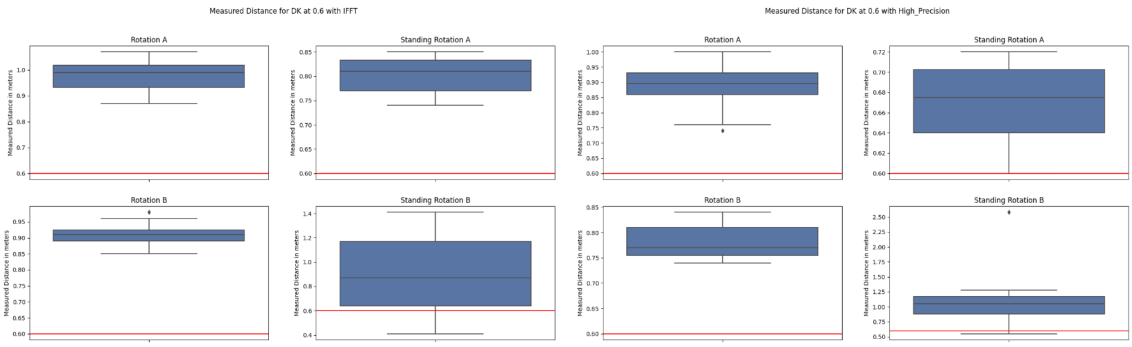

Figure A.3: DK Measurements - distance: 3m

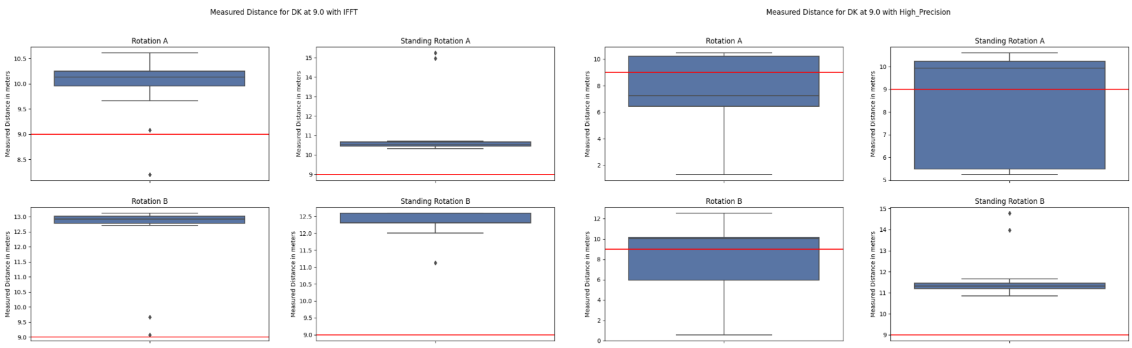

Figure A.4: DK Measurements - distance: 9m

A.3 Measurements: Anchor 2 (BDW)

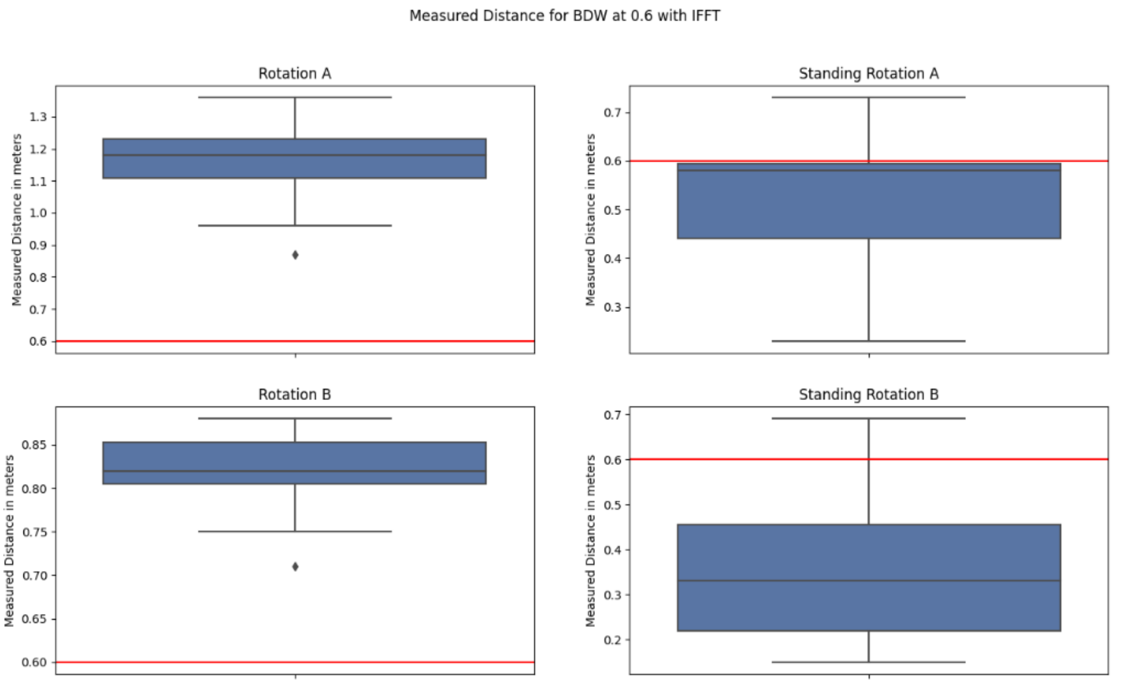

Figure A.5: BDW Measurements - distance: 0.6m

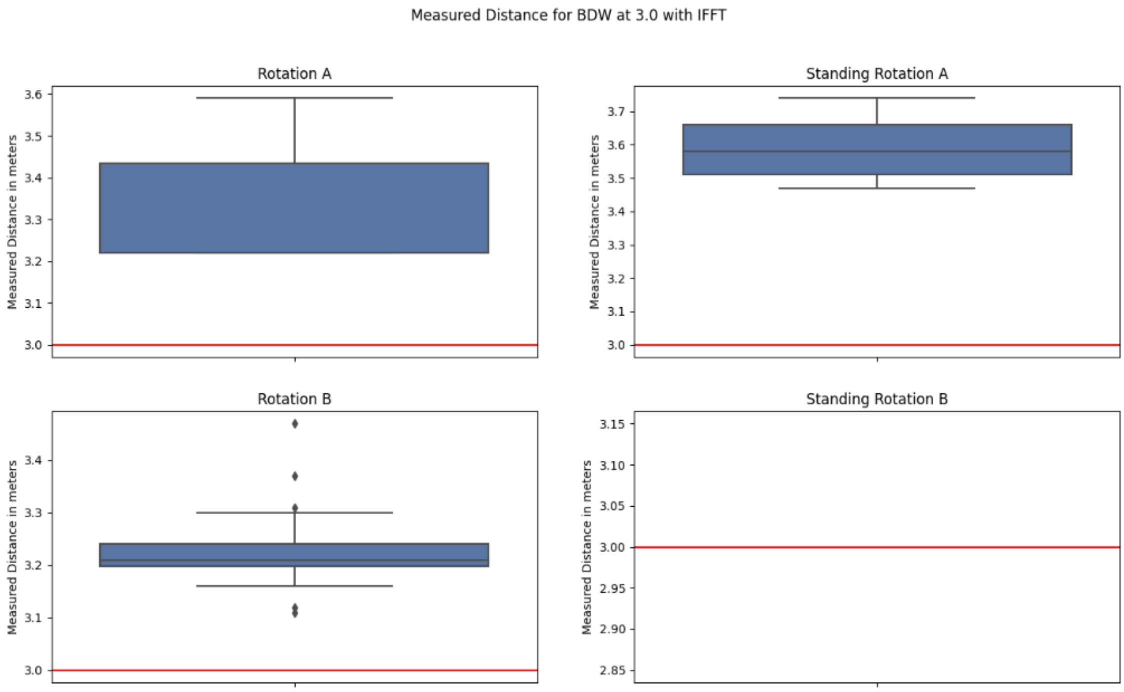

Figure A.6: BDW Measurements - distance: 3m

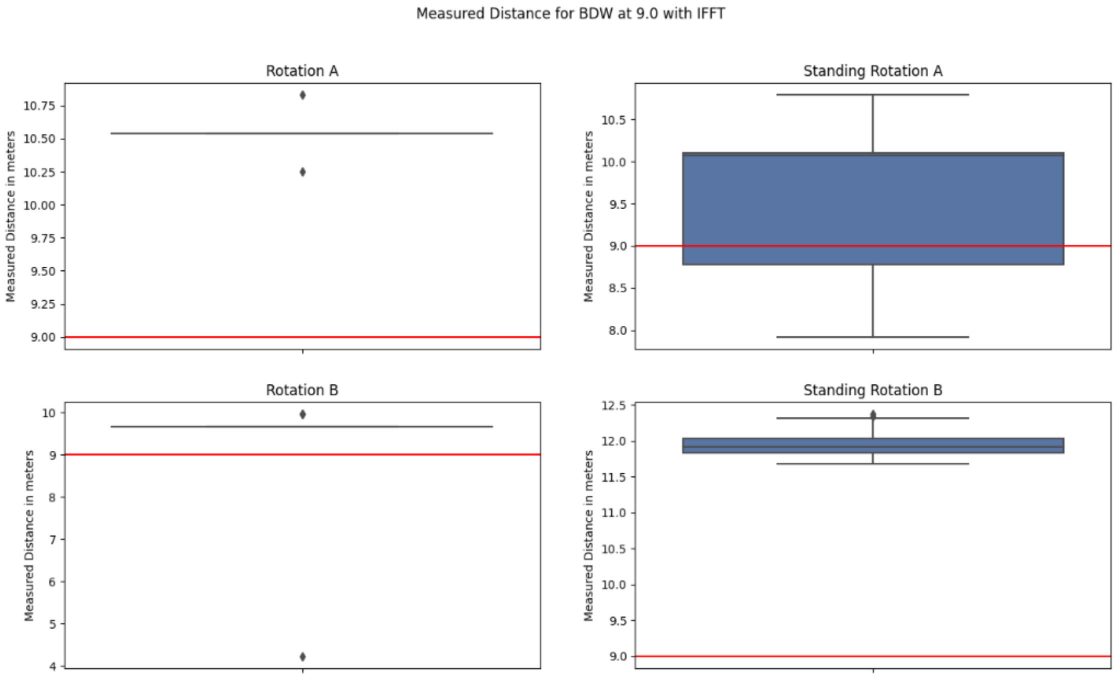

Figure A.7: BDW Measurements - distance: 9m

Appendix B

Indoor Positioning Accuracy: Bluetooth-Based Evaluation with Mobile Phone and Anchor

B.1 Measurements: Open Space

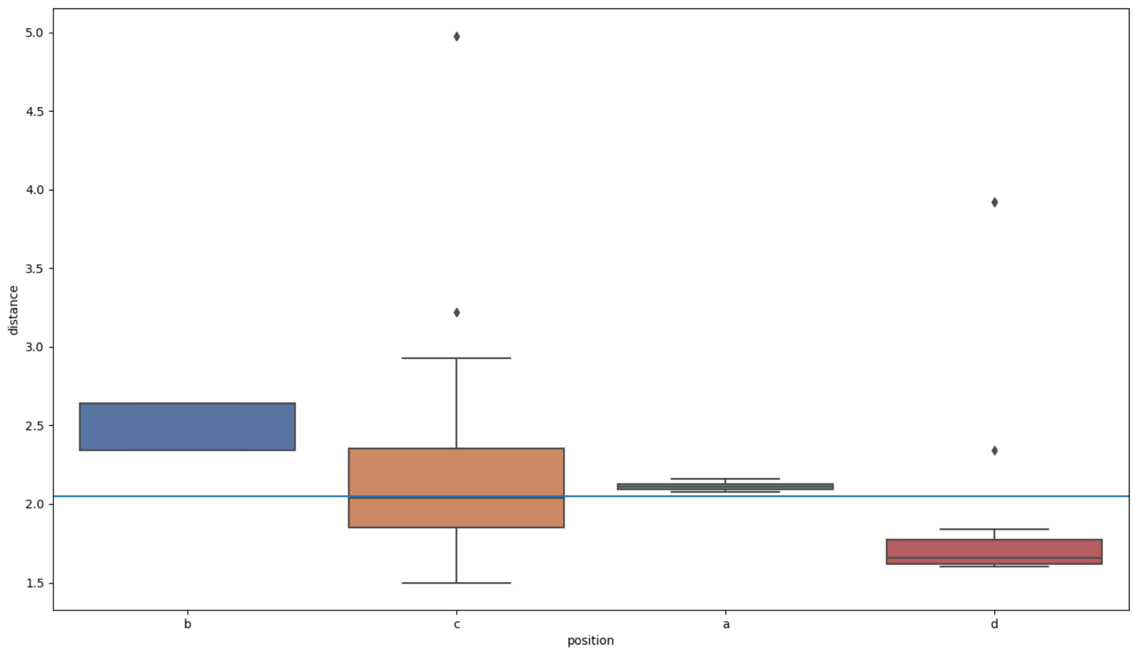

Figure B.1: Measurements Open Space - distance: 2m

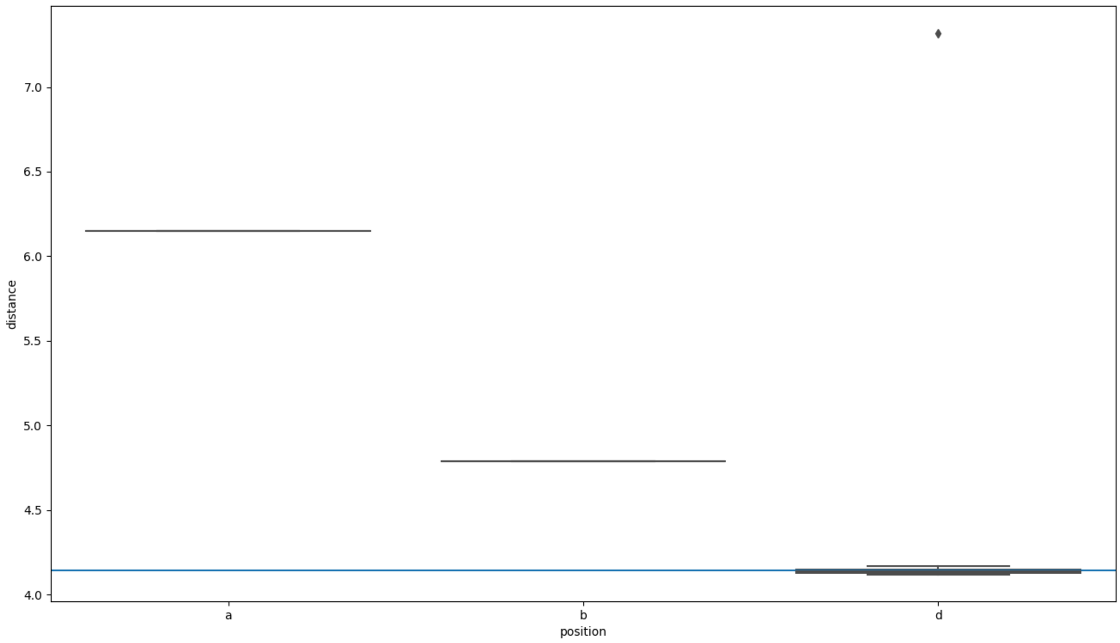

Figure B.2: Measurements Open Space - distance: 5m

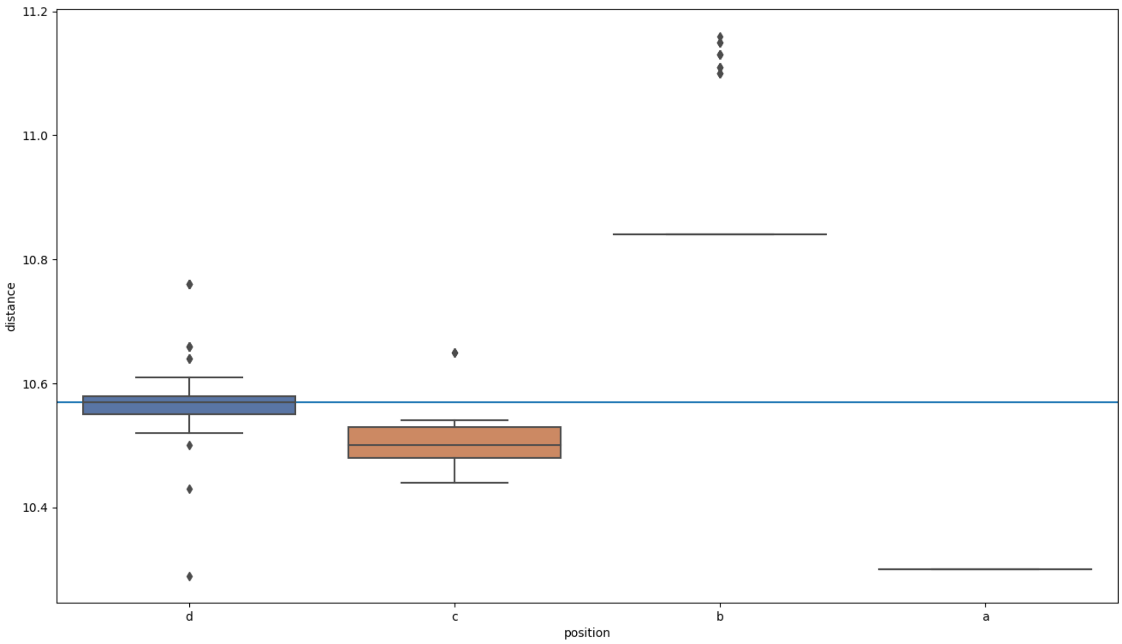

Figure B.3: Measurements Open Space - distance: 10m

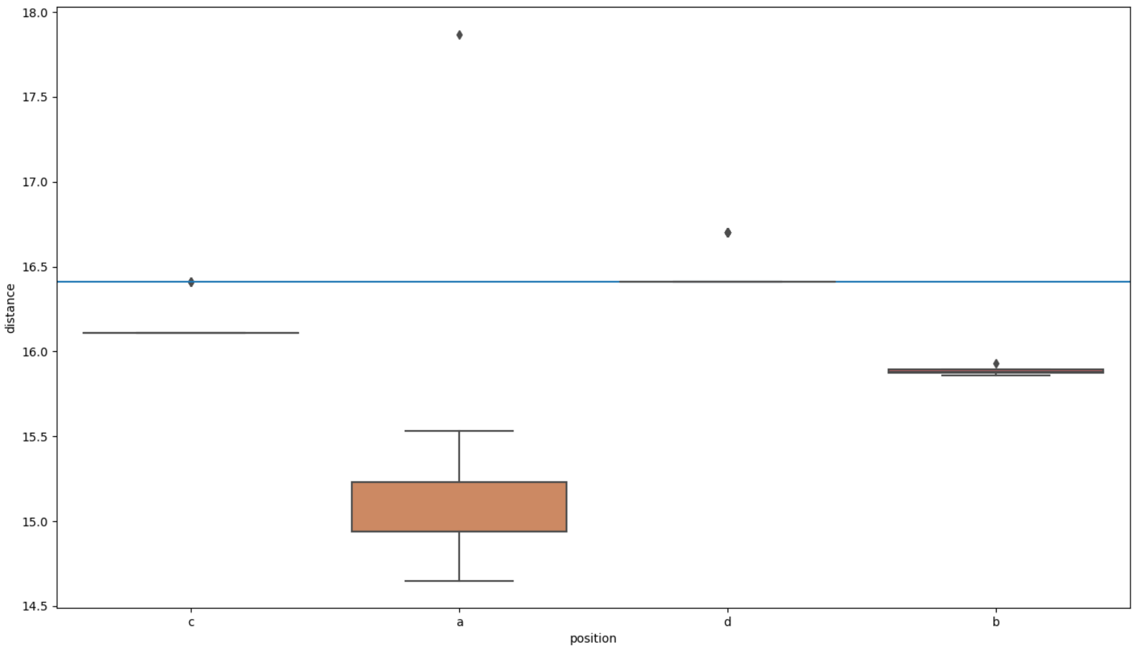

Figure B.4: Measurements Open Space - distance: 15m

B.2 Measurements: Wooden Door

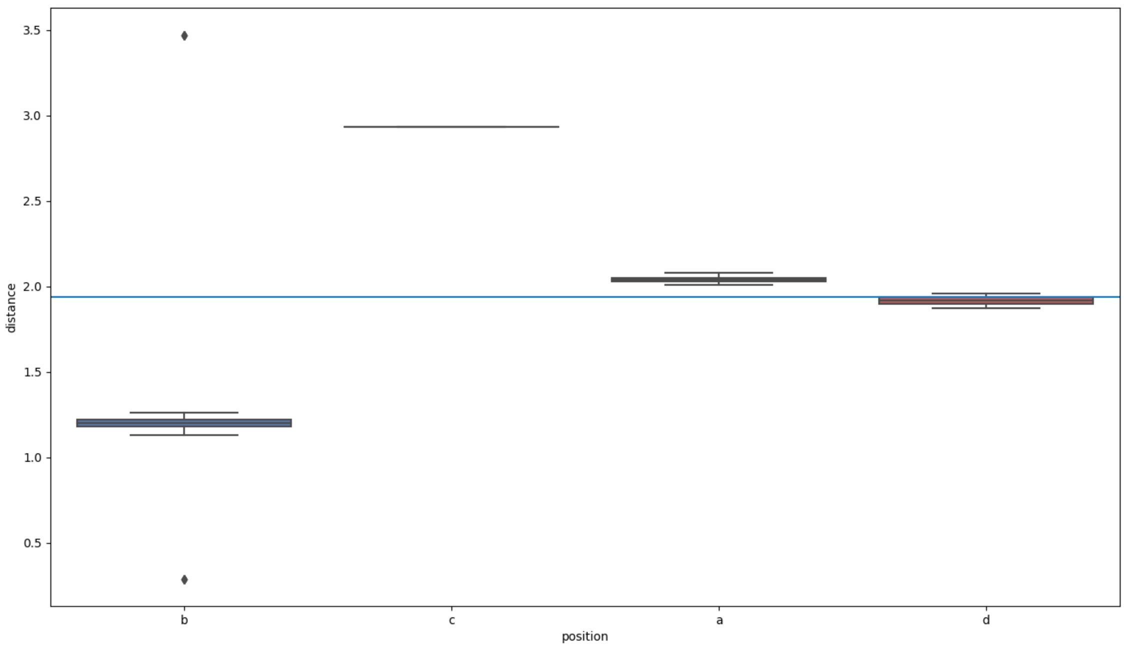

Figure B.5: Measurements Wooden Door - distance: 2m

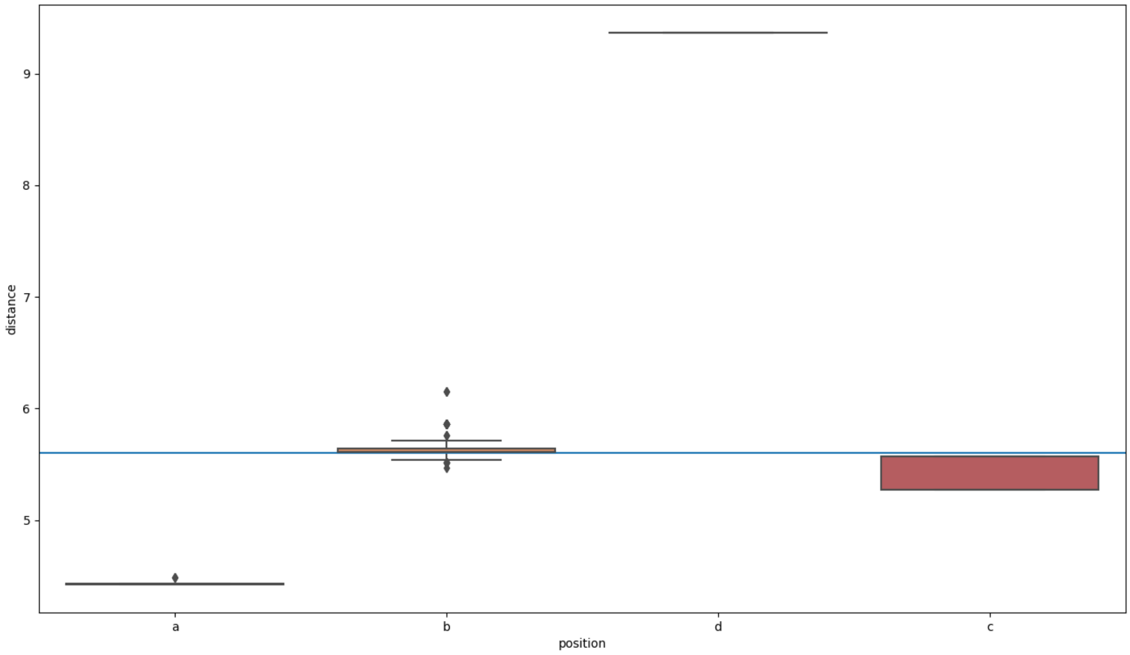

Figure B.6: Measurements Wooden Door - distance: 5m

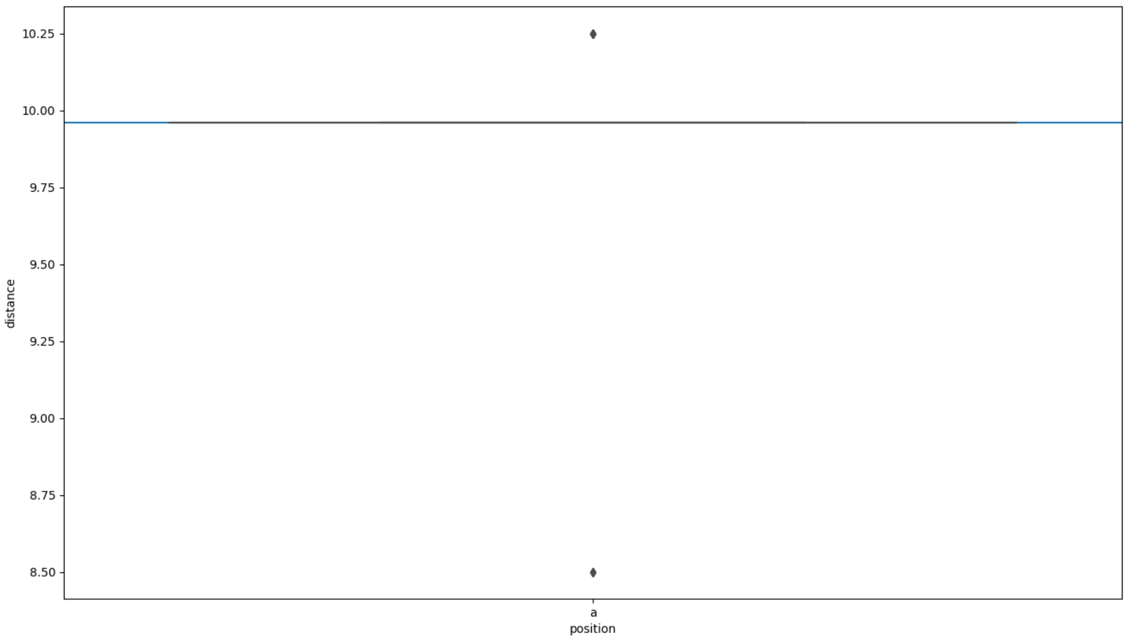

Figure B.7: Measurements Wooden Door - distance: 10m