UniLOCK: A Mobile Application-Monitored Locker Kiosk for Marinduque State University Using Raspberry Pi and Arduino

by

Riego, Raven Bryle Hexter O.

Mayores, Jelaica Mae J.

Limpiada, Chrisnell Joy C.

Gonzales, Regina G.

A Thesis Submitted to the College of Engineering in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Engineering

Marinduque State University

February 2025

APPROVAL SHEET

This thesis entitled **'UniLOCK: A Mobile Application-Monitored Locker Kiosk for Marinduque State University Using Raspberry Pi and Arduino,'** prepared and submitted by **Raven Bryle Hexter O. Riego, Jelaica Mae J. Mayores, Chrisnell Joy C. Limpiada, and Regina G. Gonzales,** in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Engineering,** has been examined and is recommended for acceptance and approval for oral examination.

ANA FE N. MOLATO, PCpE, MPA
Thesis Adviser
Date Signed: _____

PANEL OF EXAMINERS

Approved and accepted by the committee on Oral Examination with a grade of _____.

JAN ERROL B. MAMPUSTI, CpE
Member

BERNIE JR. M. OSINSAO, CpE
Member

DESIREE T. MALALAD, PCpE, MIS
Member

JONAH MARIE E. RICAMARA, CpE
Member

ANTONIO JEROLD R. LANTORIA, PCpE, MEng
Chairperson

Accepted as partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Engineering.**

MELITO L. HIRONDO, CE, MEng
Dean, CEng
Date Signed: _____

MA. EDELWINA M. BLASÈ, PhD
Vice President for Research Extension and Development
Date Signed: _____

LANGUAGE CRITIC CERTIFICATION

This is to certify that the thesis entitled **'UniLOCK: A Mobile Application-Monitored Locker Kiosk for Marinduque State University Using Raspberry Pi and Arduino,'** prepared and submitted by **Raven Bryle Hexter O. Riego, Jelaica Mae J. Mayores, Chrisnell Joy C. Limpiada, and Regina G. Gonzales,** in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Engineering, has been examined, grammatically criticized, reviewed, checked, and corrected.

Issued this 11th day of February 2025 at WAH PHIL BUILDING, 1321 E Rodriguez Sr. Ave, Cubao, Quezon City.

Digitally signed
by Reginio
John Robert Omalay

MR. JOHN ROBERT O. REGINIO
Language Critic

TURNITIN CERTIFICATION

STATISTICIAN CERTIFICATION

This is to certify that the thesis entitled **'UniLOCK: A Mobile Application-Monitored Locker Kiosk for Marinduque State University Using Raspberry Pi and Arduino,'** prepared and submitted by **Raven Bryle Hexter O. Riego, Jelaica Mae J. Mayores, Chrisnell Joy C. Limpiada, and Regina G. Gonzales,** in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Engineering, has been examined, statistically criticized, reviewed, checked, and corrected.

Issued this 11th day of February 2025 at Malanday, Valenzuela City.

Digitally signed
by Laderas
Jepthe Madrigal

ENGR. JEPTHE M. LADERAS

Statistician

CERTIFICATE OF ORIGINALITY

This is to certify that all sources used in this research study have been properly acknowledged or duly cited.

This is to certify further that the study is an original research undertaking and has not been copied from any previous work.

Signed this __th day of February 2025 at Boac, Marinduque.

RAVEN BRYLE HEXTER O. RIEGO
Researcher

JELAICA MAE J. MAYORES
Researcher

CHRISNELL JOY C. LIMPIADA
Researcher

REGINA G. GONZALES
Researcher

ACKNOWLEDGEMENT

The researchers extend their heartfelt appreciation to **Engr. Pamela Jane R. Agapay**, their former thesis adviser, for the invaluable guidance and wisdom she provided during the early stages of their research. During the research proposal stage, she played a significant role in shaping their work and ensuring they remained on the right path. Her patience and dedication greatly contributed to their progress, and they are deeply grateful for her support.

They also express their sincerest appreciation to **Engr. Ana Fe N. Molato**, their current thesis adviser, for accepting them as her advisee when Engr. Agapay had to pursue other career opportunities outside the university. Throughout the research process, Engr. Molato guided and supported them even when the path was challenging. Her comments and patience helped them refine their work, and the researchers are grateful for the time and effort she has given to help them move forward with their thesis.

The researchers would also like to acknowledge **Engr. Jonah Marie M. Ricamara, Engr. Desiree T. Malalad, Engr. Jepthe M. Laderas** (who also served as a statistician), **Engr. Jan Errol B. Mampusti, Engr. Bernie Jr. M. Osinsao, and Engr. Antonio Jerold R. Lantoria Jr.**, the faculties who provided insightful suggestions and continuous support throughout the research. Their understanding and patience, especially during the researchers' most challenging moments, were extremely helpful in improving the quality of this study.

The researchers would also like to extend their special thanks to the **security guards of Marinduque State University**, who willingly assisted them during the data-

gathering process by accommodating their questions, allowing the movement of the locker prototype in and out of the university premises, and helping with its setup within the gate. Their willingness to participate has been a great help to the completion of the research.

The researchers are extremely grateful to **Engr. Clarence S. Madrigal** for his undeniable expertise and kindness in his assistance, especially during late-night consultations due to his busy schedule. Despite that, he generously shared his knowledge and provided valuable recommendations that helped resolve crucial issues in the prototype.

They also extend their gratitude to **Mr. John Robert O. Reginio**, for his assistance in critiquing the language aspect of the thesis paper despite the limited timeframe. His efforts in refining the manuscript repeatedly due to necessary edits of the paper are truly appreciated.

The researchers would also like to thank the **respondents** who generously provided their time to test and evaluate the prototype. Their participation is the most significant in analyzing the data and guiding the research towards its completion. The researchers truly appreciate their generous act of willingly participating in the data-gathering procedure as it led to bringing the thesis into its final stage.

The researchers also extend their sincerest gratitude to **everyone** who helped, guided, and supported them throughout the development and implementation of the locker prototype. Their efforts and contributions never went unnoticed and meant a lot to the researchers, especially during the toughest moments of the journey. Having people to lend a hand during those times made a significant difference and kept them moving forward.

To their **parents**, the researchers express their deepest gratitude for their unwavering support—both financial and emotional—throughout the stages of this research. Their patience, encouragement, and belief in the capabilities of the researchers helped them continue, even when challenges arose and timelines were not met.

Above all, the researchers acknowledge **the Almighty God** for His boundless grace, wisdom, and strength. During moments of doubt and uncertainty, He provided them with the hope and perseverance needed to continue. His guidance carried them through every challenge, leading them to this milestone, even if the journey took longer than expected.

This research is a result not only of the researchers' efforts but also of the support, guidance, and encouragement from the individuals mentioned. For their invaluable contributions, the researchers extend their sincerest gratitude.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

UniLOCK: A Mobile Application-Monitored Locker Kiosk for Marinduque State University Using Raspberry Pi and Arduino

Raven Bryle Hexter O. Riego, Jelaica Mae J. Mayores, Chrisnell Joy C. Limpiada, Regina G. Gonzales

ABSTRACT

The increasing need for secure and efficient storage of items within university settings has highlighted issues related to parcel deliveries, personal item storage, and security concerns at the guardhouse of Marinduque State University (MarSU) Boac Campus. The traditional systems of managing deliveries, items left at the guards for later retrieval, and lost-and-found items create inefficiencies, security risks, and additional workload on the security guards. This study addresses the following challenges by developing a mobile application-monitored locker kiosk system to enhance security, efficiency, and user convenience. The system integrates Raspberry Pi and Arduino, including IR proximity sensors for item detection, solenoid locks for security, and a GSM module for OTP-based retrieval and real-time notifications. Key objectives of this study include evaluating the system's average execution time, accuracy in detecting the item to be stored as well as the accuracy of both the mobile application and kiosk screen in reflecting the compartment's availability, and user acceptability.

The study includes execution time analysis, accuracy testing, and user experience assessments (UEQ). The OTP notification via SMS executes within 25.7 seconds, while the system demonstrates high accuracy in item detection and effective real-time synchronization with Firebase. Performance evaluation using the mean absolute error (MAE) confirms the system's reliability in accurately displaying compartment information. Moreover, the user experience assessments (UEQ) indicate that the system significantly improves security and operational efficiency.

The findings suggest that implementing this smart locker system reduces security risks, streamlines item retrieval, and lessens the workload of on-campus security personnel. By leveraging technology to optimize campus operations, this system aligns with Marinduque State University's (MarSU) vision of becoming a smart campus.

Keywords: *locker kiosk system, storage solutions, locker, OTP-based, storage*

CHAPTER I

THE PROBLEM AND ITS BACKGROUND

1.0 Background of the Study

The increasing mobility of individuals and growing concerns about the safety of belongings in public places have emphasized the need for secure and convenient storage solutions. As people move through various environments, ensuring the security of their belongings has become a critical issue. The traditional storage methods often lack the necessary security and accessibility of items. The need for storage solutions is particularly noticeable in higher education institutions, where individuals have busy schedules. It is important to assess current practices, such as reliance on guardhouses or centralized storage locations, which can pose security risks and create additional responsibilities for security guards. When people leave their personal belongings in the guardhouse, it raises concerns regarding the security of their items. Additionally, non-security-related tasks can burden the security guards, diverting their attention from their primary responsibilities. The concept of secure storage, such as lockers, became a solution to the challenges of keeping items safe.

The concept of secure storage dates back thousands of years, with the earliest known lockers appearing in ancient Egypt around 2000 BCE. These early versions mostly consisted of wooden compartments or woven baskets secured with locks, which laid the foundation for future developments. In 1977, a huge step was taken with the invention of "The Plato Hotline," the first machine where people could interact directly. This machine,

built at the University of Illinois, was not for deliveries, but it showed the potential of user-operated systems. This was an early hint that machines could do more than just give out information. In 2018, LAUNDRAFE, a cafe and laundromat all rolled into one, showed how technology can be used to make everyday services more convenient. Then, in September 2023, the Philippines got its first-ever smart locker system called Qube. This system highlights the growing trend of using technology to make deliveries and keep things safe and easier to manage.

The growing trend of technology has led to the development of locker kiosks that are essentially self-service automated lockers integrated with a user interface. These locker kiosks have the same purpose, secured and convenient storage of items. In a US patent US8509944B1 by Kranyec Stephen, titled “Self-Storage Kiosk”, it is a locker kiosk that focuses on the secured storage of prohibited items at checkpoints while in another patent titled “Smart Locker Kiosk System”, this is the type of locker kiosk that allows users to order products through the kiosk and receive them via the lockers. While these existing locker kiosks share the core purpose of secure storage, their functionalities differ significantly and neither directly addresses the challenges of managing on-campus deliveries and personal item storage specific to universities.

Marinduque State University (MarSU) in the Philippines is the province's sole state university and college. Focusing specifically on its Boac Campus, traditional parcel delivery methods present several problems. One major issue is the requirement for recipient presence, which can disrupt busy schedules and potentially lead to missed classes for students or work interruptions for faculty and staff. An alternative way is to

leave deliveries unattended at the guardhouse. However, this practice raises security concerns and creates a significant logistical burden for school guards who are already tasked with safeguarding the campus. A recent survey conducted at MarSU revealed that on average, 16-20 items are left at the guardhouse daily. Additionally, there have been cases of items not being received, and items being lost, highlighting the inefficiency of the current system.

In addition to deliveries, there were also external sources such as people from outside the campus who are related to individuals inside the campus. In a situation where an individual such as a student, a teacher, or a staff member inside the campus requests someone from outside to bring them something, the person from outside would usually just leave the item at the guardhouse until the owner or the receiver inside the campus retrieves it. This method presented challenges, this imposed security concerns as the items may get lost or destroyed due to the inefficiency of the current system whereas, when a large number of items are left at the guardhouse it can take a significant amount of time for the recipient to locate their belongings, especially if the guard who facilitated the drop-off transaction is different from the guard on duty when they come to retrieve it.

To address the growing demand for deliveries and items being left at the guardhouse, this study developed a mobile application-monitored locker kiosk system at Marinduque State University (MarSU). Public universities in the Philippines, such as MarSU, often lack the infrastructure to efficiently manage the surge in deliveries or items left at the guardhouse, leading to missed deliveries, security concerns, and operational burdens on school guards (Dela Cruz & Reyes, 2021). Traditional methods including

requiring the presence of the recipient or leaving deliveries or items unattended are no longer sustainable. This study aimed to bridge this gap by developing a user-friendly locker kiosk system integrated with a mobile application designed for school guards to monitor transactions. This mobile application-monitored locker kiosk system served as a potential solution to address these inefficiencies at MarSU. This locker kiosk aimed to provide a secure, user-friendly, and innovative alternative for the university community. The locker kiosk system provided a secure storage solution, reducing security concerns associated with unattended packages or items. Thus, in return, has lessened the operational burden on school guards, allowing them to focus on their core duties of ensuring the university's security. Furthermore, the locker kiosk system aligned perfectly with Marinduque State University's pursuit of becoming a smart campus. A smart campus uses technology to create a more efficient, secure, and interconnected learning environment (Cox Business, 2023). By adopting this innovative solution, MarSU can take a crucial step towards achieving its vision of becoming a smart campus.

The development of a locker kiosk system with a mobile application designed for guards to monitor transactions significantly relates to Computer Engineering (CpE) as it addresses a real-world problem in universities, particularly in resource-constrained settings. The project integrated mobile technology for school guards with locker kiosks, which required a core CpE skill, and expertise in both software and hardware development. By aiming to improve efficiency, security, and user experience, the project is aligned with CpE's focus on optimizing processes, designing secure systems, and creating user-centered technology (Burnell, 2019).

The current system of managing deliveries and personal items at the guardhouse of Marinduque State University (Boac Campus) presented several challenges, such as inconvenience and security risks. The unavailability of guards that facilitate the transaction at some periods, the risk of theft or damage to unsecured items at the guardhouse, and the absence of a tracking system with detailed descriptions of items for item-related incidents were some of the drawbacks of the current system of item management at the guardhouse of Marinduque State University (Boac Campus). To address these inefficiencies and security concerns, the researchers aimed to implement a mobile application-monitored locker kiosk system. This system provided a secure, user-friendly, and efficient alternative for managing parcels and personal belongings at Marinduque State University (Boac Campus).

2.0 Statement of the Problem

The current on-campus delivery and item storage system is experiencing inefficiencies, security risks, and challenges for the school guards of the MarSU-Boac Campus. Delivery riders frequently encounter missed deliveries due to the recipient's unavailability, and lost and found items create an additional workload. School guards experience peak workload pressures during deliveries, and managing lost items further strains their resources.

This study was conducted to develop a solution by investigating the effectiveness of a locker kiosk with a real-time monitoring system for improving delivery management and user experience on campus. Specifically, the study ought to determine the following:

1. **What is the average execution time of the Locker Kiosk System in terms of:**
    a. Sending an OTP to the item-retriever after a drop-off transaction
    b. Opening of Locker Kiosk Compartment using Electric Solenoid Lock
2. **What is the accuracy of the Locker Kiosk System in terms of:**
    a. Detecting the presence of the item to be stored inside the compartment using an IR Proximity sensor
    b. Displaying compartment's availability in the Kiosk
    c. Displaying the compartment's availability in the Mobile Application
3. **What is the acceptability rate of the mobile application-monitored locker kiosk at Marinduque State University (Boac Campus)?**

3.0 Objectives of the Study

The main purpose of this study was to develop a secure locker kiosk, which was monitored by security personnel through a mobile application. It aimed to simplify the process of dropping off and retrieving items at the guardhouse. Its specific objectives were the following:

1. **To determine the average execution time of the Locker Kiosk System in terms of:**
   a. Sending an OTP to the item retriever after a drop-off transaction
   b. Opening of Locker Kiosk Compartment using Electric Solenoid Lock
2. **To determine the accuracy of the Locker Kiosk System in terms of:**
   a. Detecting the presence of the item to be stored inside the compartment using an IR Proximity sensor
   b. Displaying compartment's availability in the Kiosk
   c. Displaying the compartment's availability in the Mobile Application
3. **To determine the acceptability rate of the mobile application-monitored locker kiosk at Marinduque State University (Boac Campus).**

4.0 Significance of the Study

The current system of dropping off and retrieving items at the MarSU-Boac Campus guardhouse presented various challenges such as inefficiencies, security concerns, and increased workload for school guards. The study explored a novel solution, which focused on developing a locker kiosk with a mobile application monitored by security personnel, to address these issues. No records regarding locker kiosks with

mobile app integration have been extensively studied in the context of campus security and item management. This study may contribute valuable insights to the field of campus operations and security at the guardhouse and may offer potential benefits for the following:

Marinduque State University (MarSU). This study may align perfectly with MarSU's vision of becoming a smart campus by utilizing technology to improve campus operations. A successful implementation of this study may pave the way for further technological advancements, ultimately improving the overall campus experience for everyone. This may also serve as an evaluation tool, helping MarSU assess the need and effectiveness of such a product in addressing the current challenges.

People from outside the campus. The study may allow them to securely drop off items using the authorized system which eliminates the need for negotiation with the school guards and the associated fear of the items being lost, damaged, or not retrieved by the recipient.

Students, Faculty, and Staff. The kiosk may offer flexibility by allowing users to retrieve parcels or leave items for them at their convenience, eliminating disruptions to their schedules. It also provided a secure option for leaving items for others on the campus, reducing the risk of loss or damage. An additional benefit was the "lost and found" feature, allowing people to report lost items and reunite them with their rightful owners.

Delivery Riders. This study may eliminate their need to call or find parcel recipients, leading to faster deliveries and fewer missed deliveries within the campus. This led to a smoother workflow for both riders and recipients.

School Guards. This study may reduce their burden of storing items, allowing them to focus on their primary duty which is ensuring campus security. The mobile application can keep them informed about item transactions without the hassle of managing them directly.

Future Researchers. This study may serve as their reference for future research studies on optimizing locker kiosks, mobile app integration, and evaluation of smart campus technology.

5.0 Scope and Limitations

The study focused on developing a locker kiosk at the guardhouse of Marinduque State University-Boac Campus (MarSU Boac Campus) in Tanza, Boac, Marinduque. The locker kiosk is equipped with a mobile application designed for guards for real-time monitoring of the compartment's availability and items' status. This aimed to address the security risks of items left at the guardhouse and the inconvenience experienced by both users and guards during item transactions. The target users include students, faculty members, and delivery riders specifically delivering parcels within the campus. To achieve this, the study involved the development of a locker kiosk with 9 secure compartments featuring individual locks and an LED indicator of the compartments' availability, where items such as academic materials, packages, personal belongings, and electronic devices are allowed to be stored. The kiosk operated within the guardhouse, where power outlets were required. The study also included a manual locker opening mechanism for emergency purposes, such as power outages. The entire development process was conducted within the first semester of A.Y. 2024-2025, involving collecting data on user experience.

This study was limited to developing and implementing a functional locker kiosk within the guardhouse of Marinduque State University (Boac Campus) only. It has three main limitations. First, the fixed number of compartments in the locker kiosk system may not accommodate all items if fully occupied, potentially inconveniencing users. However, considering the collected data during the needs assessment process which showed 16-20 items being left on the guardhouse daily, the locker kiosk still reduced the guards'

workload. Second, the locker kiosk cannot accommodate some of the items such as perishable goods, wet items, and those that may impose harm or compromise the security of the campus. Third, the exclusive access to the mobile app by security personnel restricts real-time tracking for students, faculty members, and delivery riders, which hinders their ability to manage deliveries or locker usage effectively. Despite limiting real-time tracking for other users, exclusive access by security personnel minimized the risk of unauthorized access and promoted a more secure delivery system. The system may not have processed quickly if there was an unstable or no internet connection available. This issue could impact the overall user experience, as the system relies heavily on a stable internet connection to function efficiently. Additionally, the IR proximity sensor may have difficulty detecting items made of materials that do not reflect infrared light effectively. In such cases, users may need to use their hands to trigger detection before placing the item inside the compartment.

6.0 Definition of Terms

The following terms were defined according to their operation in this paper to ensure a common understanding.

Compartment - referred to a designated, lockable unit with a fixed size for storing deliveries and other items left for individuals inside the MarSU Boac Campus.

Drop-off - the transaction with the locker kiosk involved leaving or storing items such as belongings, parcels, and lost and found ones inside the locker.

Locker Kiosk - a self-service electronic system equipped with lockable compartments where users can store and retrieve items including lost and found objects. It is integrated with a mobile application designed for the school guards for real-time monitoring of items stored.

Lost and Found - the transaction that involves leaving and retrieving a lost item inside the locker kiosk.

Mobile Application - a software program designed for smartphones or tablets that allowed the school guards of MarSU Boac Campus to monitor the status of stored items in the locker kiosk in real time.

Retrieve - the transaction with the locker kiosk which involved getting back or accessing the previously stored or left item inside the locker.

CHAPTER II

REVIEW OF RELATED LITERATURE

1.0 Related Literature

This section focused on exploring related literature to have an understanding of the concept of the study, which encompassed several aspects of locker systems and related technologies. By examining existing works, the aim was to comprehend the landscape of locker systems and identify any gaps in the current body of knowledge.

1.1 Locker System

A locker system is a secure storage solution designed to temporarily store personal or shared items. Locker systems are utilized in various settings, such as schools, gyms, workplaces, transportation hubs, libraries, and retail environments.

1.1.1. Locker

A locker is a lockable storage unit used in places like schools, gyms, or train stations to keep personal items safe. Modern lockers are created from various materials, including wood, aluminum, steel, metal, and fabricated wood and they come in different sizes, shapes, colors, and designs, and can be manufactured, customized, or ordered to fit various budgets and needs (The Bulletin, 2001). School lockers are essential for enhancing the educational experience by providing security, teaching responsibility, ensuring health, and fostering focus and privacy (Louise, 2018).

1.1.2. Smart Locker

A smart locker is a storage unit that utilizes Internet of Things (IoT) technology to offer secure, keyless, and data-driven storage solutions, as defined by the eLocker company. The primary concern regarding smart lockers is ensuring their security and preventing unauthorized access.

Research by the UG Students of the Department of Electronics and Instrumentation Engineering, Prof R. Srinivasa, T. Mettilda, D. Surendheran, K. Gopinath, and P. Sathishkumar highlighted the implementation of advanced locker security systems, incorporating RFID, PASSWORD, CONVEYER, GSM, and heat sensor technologies, particularly in smart lockers for enhanced security in high-risk environments such as banks and offices. According to them, smart lockers utilize RFID technology for automatic identification and tracking of objects, streamlining access and reducing waiting times, PASSWORD verification to add an extra layer of security, ensuring only authorized users can access lockers, GSM technology that sends random passwords to users' mobile phones for secure access, CONVEYOR system in smart lockers that brings the designated locker to the user after security verification, improving operational efficiency, and heat sensors that detect unauthorized access attempts, triggering alarms to prevent theft.

1.2. Kiosk

A kiosk is a creative and adaptable stand-alone booth that businesses and organizations use in busy areas like malls, airports, and train stations to engage with

customers in dynamic ways. Digital kiosks offer various features for interactive customer engagement through touch screens, payment processing, intuitive user interface, printing capabilities, and the capability of being customizable in both hardware and software that meets specific business needs (Farrukh, 2023). According to KIOSK Information Systems, the kiosk is efficient in delivering comprehensive turnkey deployment services, flexible field and managed services, and a user-friendly locker software platform, showcasing a commitment to delivering efficient and effective self-service solutions for various retail settings. Thus, incorporating a kiosk in the locker system is essential for ensuring that lockers function as an effective storage solution.

1.3. Infrared (IR) Sensors

Infrared (IR) sensors are small electronic devices that detect specific features of their surroundings by either emitting or detecting infrared radiation. These sensors are widely used in various applications. These include object or item detection, motion sensing, and even proximity measurements. A light sensor is being utilized by IR sensors to function where it determines a specific wavelength in the infrared spectrum. The IR sensor detects the reflected IR light and measures the change in the received signal when a particular item is present (Ajmera, 2021).

The sensor's output can be run by a microcontroller, such as an Arduino, to display the status on an LCD screen or transmit the data to a central monitoring system (Maharaja et al., 2021). IR sensors are proven to be accurate, affordable, and

versatile. Thus, they are widely used in automation systems, security systems, and consumer electronics.

IR sensors can be integrated into locker systems to guarantee the presence of an item. According to the paper "MINI IR RADAR FOR UNAUTHORIZED OBJECT DETECTION" by Maharaja et al. (2021), IR sensors can monitor the occupancy of the locker by detecting the presence or absence of an item inside, whether it is occupied or empty. When an item is put inside the locker, the IR sensor detects its presence through the status of the received IR signal. This information can be used to determine the availability of a locker.

1.4 Solenoid Lock

A solenoid lock is a locking mechanism that uses the principles of electromagnetism to function. According to EFY Bureau (2020), solenoid locks are commonly used in electronic and biometric password-based security systems because of their strong metal plungers and ability to lock and unlock operations effectively. A solenoid lock can work with a relay module, which is configured by the microcontroller to control the lock's operation, emphasizing the solenoid locks' difference from manual locks due to their electrical operation (M. Shahid, 2021). The design allows for precise control over the lock mechanism, making it a reliable choice for securing items.

1.5 Evaluating the Prototype

1.5.1 Average Execution Time

Rajagpal, S. M., M, S., & Buyya, R. (2023) explained that the entire duration from the moment a request is initiated to the finalization of all tasks was taken to get the execution time, summing up even the completion time of the last machine in the process. This extensive measure is important, particularly in complex computing environments where the optimization of task processing times can substantially raise system performance. With the stated definition of execution time (ET) for an individual task, it can be calculated using formula 1:

$$ET = T_{end} - T_{start} \quad (1)$$

Average execution time is an essential measure in computing and information systems, serving as an assessment of a system's efficiency in processing tasks. This is defined as the sum of execution times for a series of tasks divided by the total number of tasks, providing a mean value indicative of the typical duration required for a system to complete a single task from start to last. Thus, the formula for solving the average execution time (ETavg) is:

$$ET_{avg} = \frac{\sum_{i=1}^{n} ET_i}{n} \quad (2)$$

Hurley, M., & Tenny, S. (2023) further explained the significance of the average execution time as it directly impacts system performance. The relationship between performance and execution time is inverse, as described

by Parthsarathi, R. (n.d.) in the formula Performance = 1/Execution Time. Therefore, minimizing average execution time is essential for enhancing speed, efficiency, and user experience.

With the stated insights, it is therefore evident that the average execution time plays a big role in computing performance. The efficiency assessment of computing systems across various applications, from IoT computing environments to more direct computational tasks undergo the process of getting the average execution time. Thus, it proves that the average execution time serves as a basis for assessing the efficiency and performance of the system. By focusing on reducing the average execution time, developers and engineers will be able to ensure that systems are optimized for performance, which improves the overall system throughput and user experience. In essence, the lower the average execution time of a system, such as a locker kiosk, the higher its performance, which highlights the important role of aiming for minimal average execution times in system design and optimization where performance is mainly important.

1.5.2 Confusion Matrix

The confusion matrix offers a clear visualization of the predictive accuracy of an algorithm, usually within machine learning. It is an organized table that compares the predicted class cases against the actual, or ground truth, cases for a dataset (Ting, 2018). A standard binary classifier's confusion matrix consists of four representations of different prediction outcomes: the true

positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) (Ting, 2018). One such metric is accuracy, which provides a measure of the total accuracy by comparing the number of correct predictions to the total number of predictions. The formula for accuracy is shown below:

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})} \quad (3)$$

Aside from obtaining accuracy, the matrix can also be utilized in computing other metrics such as precision, recall, and the F1 score (Zhang & Zhang, 2018). Precision, or the positive predictive value (PPV), measures the proportion of correct positive predictions, while recall, also known as sensitivity or true positive rate (TPR), assesses the percentage of actual positive instances identified by the model (Kuhn & Johnson, 2016). The F1 score coordinates precision and recall, particularly beneficial for datasets with discrepancies where the exchange between the precision and recall is significant (Goodfellow, Bengio, & Courville, 2016). However, some researchers argue that the F1 score's equal emphasis on precision and recall may not suit all datasets, leading to the development of modified variants (Hand & Christen, 2018). In addition to these conditional measures, the confusion matrix also informs unconditional metrics such as the negative predictive value (NPV), which predicts the possibility of an instance not belonging to a specific class (Kuhn & Johnson, 2016). The confusion matrix is a complex but critical instrument for machine learning experts to evaluate and improve model performance through the comparison between the predicted and actual values.

1.5.3 Mean Absolute Error

Mean Absolute Error (MAE) is a key statistical measure in the field of predictive analytics and data accuracy measurement where it represents the average significance of errors in a set of predictions, without considering their direction (Alexander, 2020). The MAE is calculated by obtaining the average of the absolute differences between the predicted values and the observed true values. The Absolute Error is also called the Absolute Accuracy Error where it measures the discrepancy between a measured value and the true value. The formula for absolute error is:

$$\Delta x = x_i - x \tag{4}$$

In this context, Absolute Error is the discrepancy between a measured value and the true value. The Mean Absolute Error (MAE), as defined by Alexander (2020), is the average of the total absolute errors across a dataset. According to Alexander (2020), this metric is particularly useful because it measures the accuracy of predictions in the same units as the data, making it easy to interpret. The formula for MAE is not complex and involves taking the total of the absolute values of the individual errors and then dividing them by the total number of observations. The formula for calculating the Mean Absolute Error (MAE) is the following:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - x| \tag{5}$$

This measure provides a clear indication of the average significance error and is particularly crucial when comparing the performance of different predictive classes or measurement systems.

1.5.4 Likert Scale

According to Joshi, A., Kale, S., Chandel, S., & Pal, D. (2015), the Likert scale is one of the most important and frequently used psychometric tools in various research, such as educational and social sciences. It was introduced by Rensis Likert in 1932 and has been widely utilized by many researchers. The wide use of the Likert scale led to debate and controversy regarding the number of response options and its impact on reliability and validity.

Clark and Watson (1995, 2019) highlighted the simplicity of the Likert Scale which analyzes collected data more easily. Due to its simplicity, Jebb, Ng, and Tay (2021) emphasize the significance of using the Likert scale for data collection in psychology. However, the researchers cannot continuously use the same psychometric technique considering the significant advancements in the past 25 years. The progress in psychometric technique must undergo a review and revision if necessary. This has been addressed by the authors who stated that their review focuses on the initial development and validation of self-report Likert scales, excluding broader topics like general psychometrics and response biases. The scale typically involves presenting respondents with a series of statements related to the topic with a range of response options that reflect varying degrees of agreement or disagreement. The most common formats are

5-point and 7-point scales, with options ranging from “Strongly Disagree” to “Strongly Agree.”

One of the key characteristics of the Likert Scale is its use of ordinal data. This means that while the responses reflect a rank order, they do not quantify the exact distance between points on the scale. A notable feature of the Likert Scale is the inclusion of a neutral option, such as "Neither Agree nor Disagree," which allows respondents to indicate a neutral or indifferent position. The scale is typically balanced, with an equal number of positive and negative options, to avoid biasing the responses.

However, the Likert Scale is not without its limitations. One issue is central tendency bias, where respondents may avoid extreme options and prefer middle choices. Another is acquiescence bias, where some respondents may agree with statements regardless of their actual opinions. Additionally, there can be variability in how different respondents interpret the scale points.

In terms of analysis, the data collected from Likert Scales can be subjected to both descriptive and inferential statistical techniques. Researchers can calculate mean, median, and mode to summarize the responses, and apply methods like t-tests, ANOVA, and regression analysis to explore relationships and differences among groups. Despite its limitations, the Likert Scale remains a valuable tool for capturing and analyzing people's attitudes and opinions in a structured and quantifiable manner.

1.5.5 User Experience Questionnaire

The User Experience Questionnaire (UEQ) is a valuable tool to assess user perceptions across key dimensions, such as usability, practicality, attractiveness, and stimulation (Schrepp, 2017). By getting both pragmatic and hedonic aspects of the user experience, the UEQ provides a comprehensive understanding of user interactions and perceptions of a product or system. The UEQ typically used a Likert Scale where respondents rate their agreement with statements on a scale (e.g., 1-7, strongly disagree to strongly agree) (Lauer & Albrecht, 2019). Data analysis involves calculating mean scores for each dimension and comparing them across different user groups or conditions. For instance, high scores on the "Attractiveness" dimension would indicate that users find the product aesthetically pleasing. By combining scores across dimensions, researchers can obtain an overall UX score, providing a comprehensive evaluation of the user experience. By analyzing these responses, researchers can compute the mean scores for each dimension, determine areas of strength and weakness, compare different prototypes, and use the data to guide design decisions and improve the prototype. This data-driven approach ensures that the final product is not only functional but also user-centered and enjoyable.

The User Experience Questionnaire (UEQ) family offers a suite of standardized instruments for assessing various aspects of user experience (UX) (Lauer & Albrecht, 2019). It includes questionnaires such as the UEQ-S (short version), UEQ-Basic, UEQ-S3 (simplified version), UEQ-HMI (for Human-Machine Interfaces), and UEQ-A (for assessing aesthetic and emotional

responses). Choosing the appropriate UEQ depends on factors such as the specific research objectives, the target audience, the context of use, and available resources. For instance, if researchers primarily aim to evaluate overall user satisfaction, the UEQ-S might be suitable. Conversely, if the focus is on assessing the user experience of a specific human-machine interface, the UEQ-HMI would be more appropriate.

The UEQ family offers several advantages, it provides a standardized and validated framework for measuring various UX aspects, ensuring reliable and comparable results across studies (Lauer & Albrecht, 2019). Also, the questionnaires are relatively easy to administer and understand, making them accessible to a wide range of researchers and practitioners. However, limitations also exist. For instance, social desirability bias and other response biases may influence user responses. Moreover, UEQ may require adaptation for use in different cultures.

The UEQ family offers several advantages. Firstly, it provides a standardized and validated framework for measuring various UX aspects, ensuring reliable and comparable results across studies (Lauer & Albrecht, 2019). Secondly, the questionnaires are relatively easy to administer and understand, making them accessible to a wide range of researchers and practitioners. However, limitations also exist. For instance, social desirability bias and other response biases may influence user responses. Moreover, the UEQ may require adaptation for use in different cultural contexts.

2.0 Related Studies

Lockers serve as secure storage, but traditional options often lack advanced features. Smart lockers offer enhanced security and functionality. This section reviews relevant studies on smart lockers to identify gaps and position the current study within the existing body of knowledge.

2.1 Foreign Studies

In a study conducted by Srinivasan, Mettilda, Surendhran, Gopinath, & Sathishmar (2015), a locker security system was developed that uses RFID, password, Conveyer, GSM, and heat sensors for enhanced security of lockers located in banks, offices, and homes as the technology allows only the authorized users to access the items inside the locker. The same kind of locker system was developed by Ramani, Selvajaru, Valarmathy, & Niranjan (2012) which has an additional two-password feature and a log containing the log-in and log-out of users along with their basic information. The first password is entered into the locker's keypad and once verified, the password will be entered again through a mobile phone, this ensures that only the authorized user of the locker can access the specific locker. Another study conducted which has the same feature as the previously mentioned study is a locker system used for storing money wherein the two-password feature is present, but the sequence is different as the inputted password through SMS comes first which is followed by entering the password through the locker's keypad (Menaka Devi, Kausalya, Kaviya, Manoranjitha, Monisha, 2020). Another feature added in this study that fosters more security is the alert system which is a buzzer that turns on

when one of the inputted passwords is incorrect. These studies focus on developing locker systems that are for offices, homes, and more specifically, banks.

Smart Lockers or modern locker systems may have been designed for different settings, but they are also designed to minimize human intervention while their activities are monitored and communicated in a server that contains the database for the said storage (Mahadik, Latif, & Khot, 2023). Another minimized intervention is the delivery of parcels, this is seen in the study conducted by Yao (2018) wherein a code is sent to the receiver which will be entered into the screen of the locker once the user retrieves the parcel from the locker. In line with the sending of notifications to the user that contain the details to be inputted in the screen of the locker is the study about an IOT-based smart locker that sends an OTP to the email of the user which will be used to unlock the locker (Mostakim, Sarkar, & Hossain, 2019). A layer of security is incorporated with this locker as it has features of face detection and a notification sent to the user when unauthorized use through the heat of the locker is detected.

The following studies have relevance to the current study in terms of the function of the lockers. Technologies such as GSM are used in recent studies and were used in the current study for notification purposes which included details needed for the retrieval process to ensure the security of the item inside the locker. However, none of the recent studies have focused on a school setting, specifically at the guardhouse or the main gate, unlike the current study. Aside from this, Raspberry Pi has never been used in the following study, allowing the current study to explore

how the use of the said microcomputer (Raspberry Pi) can be used in developing the locker kiosk.

2.2 Local Studies

Several studies have been also conducted in the Philippines regarding lockers with more advanced features compared to generic lockers. The study conducted by Bonifacio, Cale, Ferrer, Gomera, Gomez, Manaois, Picar, Pili, Simon, Sison, Turla, & Valencia (2008), developed a locker system that offers a user-friendly storage solution within educational institutions through its card swiping and pin code access control features with a database to ensure that the lockers are accessed only by the authorized user who rented them. Along with this is the study that aimed to modernize generic lockers in schools by developing a student modern locker using Arduino (Sebastian, Rivete, Villegas, Perez, Hernandez, & Lope, 2021). This locker ensures the security of lockers through biometric security, usage of RFID, and SMS notification as well as an alarm system with the use of a buzzer when unauthorized access is detected.

While studies about school lockers have been conducted, Capre, Reyes, & Tolentino (2022) have conducted a study about developing a smart locker for a company. The smart locker uses IOT and RFID for secure access and to replace the traditional locks used in generic lockers. Each employee has their own locker and wearable device from which the notification will be received once unauthorized access to the locker is detected.

The studies presented were related to the current study as they all have the same functionality, secure storage of items with advanced features. However, the following studies have focused on personal lockers and none of them focused on lockers being monitored by guards, as well as the temporary storage of items. The lockers mentioned in the studies mentioned earlier were more likely to be for personal use, allowing the current study to explore different features, such as lost and found items. This showed that there were still areas for discovery of how far features of lockers can go.

The reviewed foreign and local studies explored secure locker systems with features like RFID, passwords, and GSM for access control and user notifications (Srinivasan et al., 2015; Ramani et al., 2012; Menaka Devi et al., 2020; Bonifacio et al., 2008; Sebastian et al., 2021; Capre et al., 2022). However, these studies focused on personal lockers in various settings such as offices, banks, and schools. In contrast, the current study addressed temporary item storage in a school guardhouse setting, allowing for features like lost and found, usage of an IR proximity sensor to detect the presence of an item inside the compartment, and a notification sent to every retriever of the item when class hours end each day. The usage of a Raspberry Pi was also explored in this study to see how it works in a locker kiosk system.

3.0 Conceptual Framework

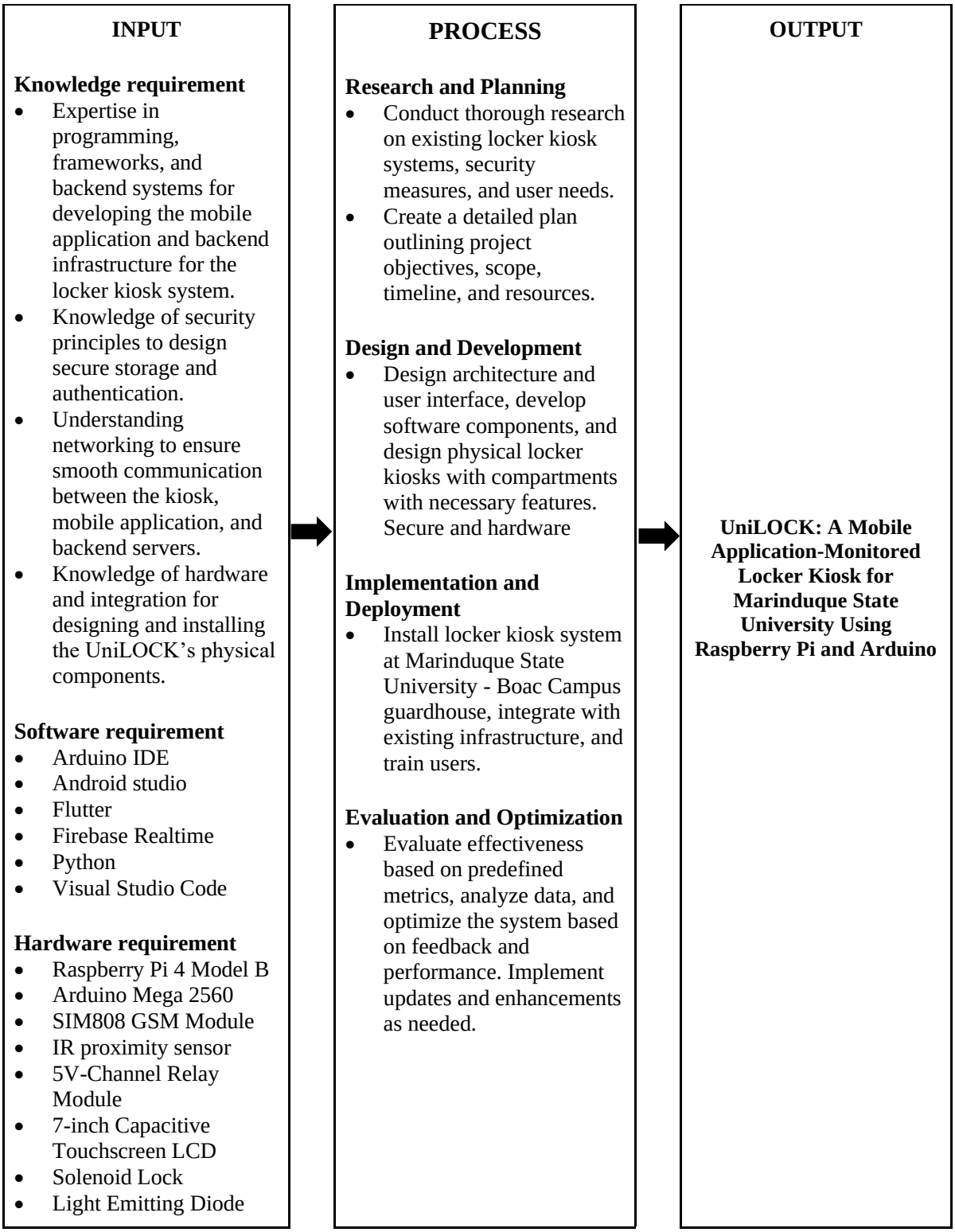


Figure 1. Conceptual Framework

Figure 1 shows the conceptual framework of the study, consisting of the input, process, and output. The input includes the knowledge, software, and hardware requirements necessary for the development of the prototype. These requirements were used as the foundation for the process, which started with the research and planning phase, followed by the design and development phase, then the implementation and deployment phase, and finally, the evaluation and optimization phase.

The final output is the UniLOCK, a mobile application-monitored locker kiosk for Marinduque State University which uses Raspberry Pi and Arduino, designed to improve delivery management and user experience on campus.

CHAPTER III

METHODOLOGY

This chapter outlines the methods and techniques that were used to accomplish the objectives of this study. It contains the hardware and software design, the system design, testing and evaluation procedures, and the methods of research used in this study.

1.0 Hardware Design

This section discusses the hardware specifications as well as illustrations, such as diagrams, on the circuitry, connectivity, and integration of the hardware component, which are block diagrams, circuit diagrams, and pictorial designs that were used in the development of the locker kiosk.

1.1 Hardware Specification

The hardware specification explains the technical and functional requirements of the hardware components utilized in the locker kiosk.

1.1.1 Raspberry Pi 4 Model B

Figure 2. Raspberry Pi 4 Model B

Image Source: https://images-na.ssl-images-amazon.com/images/I/71IOISwSYZL._AC_SL1400_.jpg

Figure 2 shows the Raspberry Pi 4 Model B that was used by the researchers as the central control unit of the kiosk system, which enabled seamless communication, data processing, and management of the other hardware components. Powered by the Broadcom BCM2711 processor with a quad-core Cortex-A72 ARMv8 64-bit SoC running at 1.5GHz, it provided the computational power needed to handle real-time operations. Its flexible memory options made it efficient in running the system's software and managing multiple tasks simultaneously.

The Raspberry Pi was used to communicate with Firebase, enabling two-way data exchange. It was used to send data such as user inputs, system status, and locker usage details to Firebase while receiving updates or commands to ensure real-time synchronization. For user interaction, the Raspberry Pi

managed the LCD, which facilitated two-way communication to show prompts, instructions, and status updates. Additionally, it was used to control the SIM808 GSM module, enabling SMS-based communication for notifications, verification, or reminder alerts.

1.1.2 Arduino Mega 2560

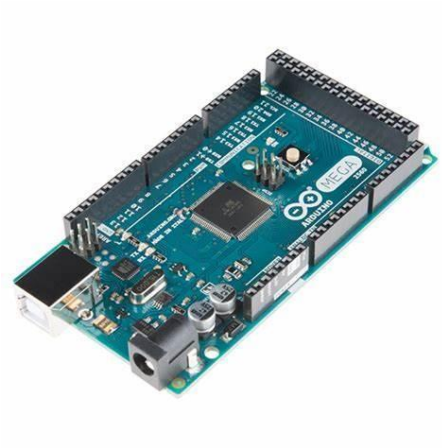


Figure 3. Arduino Mega 2560

Image Source: https://cdn.littlebird.com.au/images/files/000/055/499/large/SF-DEV-11061.jpg?1535765325

Figure 3 shows the Arduino Mega 2560 which was an essential component of the locker kiosk system as it was responsible for managing the hardware operations and maintaining seamless communication with the Raspberry Pi. In the system, the Arduino Mega was the one controlling the solenoid locks through the commands it sent to the relays. It was also responsible for the operation of the RGB LED which indicates the compartment's availability. Additionally, the Arduino Mega processed inputs

from the IR proximity sensors for the detection of items to be placed inside the compartments.

The Arduino Mega facilitates two-way communication with the Raspberry Pi, allowing it to send sensor data and system status updates while receiving commands from it to execute specific tasks. For instance, the Raspberry Pi can send a command to the Arduino to change the RGB LED indicator from red to green when a user successfully retrieves their item or from green to red when a drop-off is completed. As for the IR Proximity sensor, when it detects an item, it sends this information to Arduino. This is then forwarded to the Raspberry Pi and with this, the system displays a confirmation message, such as "Item detected successfully," on the LCD screen via the Raspberry Pi. The Arduino also controls the relay, which in turn operates the locking and unlocking of the solenoid lock based on the command sent by the Raspberry Pi from the input of the user in the LCD.

With 256 KB of flash memory for program storage, 8 KB of SRAM for handling dynamic data, and 4 KB of EEPROM for non-volatile storage, the Arduino Mega can easily support the complex codebase and operations of the locker kiosk system. Its compatibility with a wide range of shields and libraries simplified the integration of essential components such as IR sensors, LCDs, relays, and LEDs. By efficiently managing hardware and enabling real-time coordination with the Raspberry Pi, the Arduino Mega 2560 ensured the locker kiosk system was reliable, secure, and user-friendly.

1.1.3 SIM808 GSM Module

Figure 4. SIM808 GSM Module

Image Source: https://robotechshop.com/wp-content/uploads/2017/08/sim-808-blue-1.jpg

Figure 4 shows the SIM808 GSM Module which enabled communication between the kiosk and the user in the locker kiosk system. Once SIM808 receives data from the Firebase backend, which is the Raspberry Pi, it uses its GSM/GPRS functionality to send SMS notifications to the user such as a one-time-password (OTP) and a reminder that there is still an item that needs to be retrieved, ensuring that users were promptly informed via reliable cellular communication.

1.1.4 IR Proximity Sensor

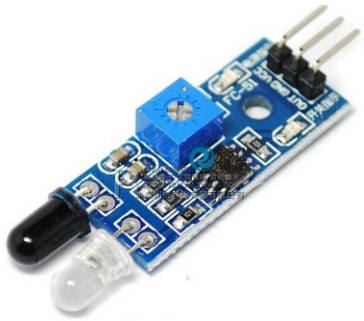


Figure 5. IR Proximity Sensor

Image Source: IR+Proximity+Sensor.jpg (1000×832)

Figure 5 shows the IR Proximity which was an essential component that detects the presence of an item inside the locker's compartment in the locker kiosk system. It sends data to the Arduino Mega 2560, which processes the information. The IR sensor corresponds with its task to detect an item inside the compartment due to its reliability, affordability, and ease of integration. In addition, an IR sensor can measure distances from 2 to 80 cm with an accuracy of 2-3 cm, making it ideal for detecting objects inside the locker's compartment. With a simple 3-pin interface: VCC, GND, and OUT, it connects excellently to both the Arduino and Raspberry Pi for real-time monitoring and control. Its good structure and accessibility ensured consistent performance and easy maintenance, making it a reliable component of the system.

1.1.5 5V- Channel Relay Module

Figure 6. 5V-Channel Relay Module

Image Source: https://alexnld.com/wp-content/uploads/2019/02/5961420c-1380-4593-936d-2a37b6ef3803.jpg

Figure 6 shows the 5V channel relay module which was responsible for the operation of the solenoid lock, whether to lock or unlock the compartment based on the user's input. The relay module receives commands from the Arduino Mega 2560, which processes the data. When a user interacts with the system, the Arduino transmits a signal to the relay that will either activate or deactivate the solenoid to lock or open the compartment. Also, the relay module is compatible with both the Arduino and Raspberry Pi because it can operate on a 5V signal.

1.1.6 7.0” Capacitive Touchscreen LCD

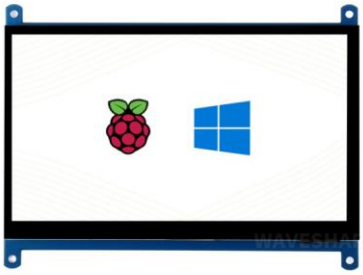


Figure 7. 7.0” Capacitive Touchscreen LCD

Image Source:7inch Capacitive Touch Screen LCD (C), 1024×600, HDMI, IPS, Low Power Consumption

Figure 7 shows the 7.0’’ Capacitive Touchscreen LCD which worked as the main interface for user interaction. It communicates with the Raspberry Pi and provides a clear and responsive display for the user interface. With a resolution of 1024x600, the screen ensured clear and detailed visuals, making it easy for users to interact with the system. The capacitive touch technology enabled an intuitive and responsive experience, supporting multi-touch gestures for enhanced usability. The large 7.0-inch screen offers ample space for interactive elements while maintaining a compact form factor, ideal for the kiosk design. Its durability and reliability make it well-suited for public-facing applications like this locker kiosk system.

1.1.7 12V Solenoid Lock

Figure 8. 12V Solenoid Lock

Image Source: https://i5.walmartimages.com/asr/97278666-6359-4417-ac65-fc0aa272a316_1.39017b074d0f48f506e295936e165f72.jpeg

Figure 8 shows the 12V Solenoid Lock which benefits the development of a mobile app-monitored locker kiosk. It has a dependable and secure locking mechanism, which is crucial for safeguarding the contents of each compartment. Typically, it has a 12V DC working voltage, uses little power (generally 1-2A), and is made of sturdy materials that can sustain repeated usage. A straightforward digital output pin can be used to remotely lock and unlock the lock using a mobile app when using the Raspberry Pi or Arduino Mega 2560. Furthermore, to accommodate varying security requirements, these locks frequently have fail-safe (which unlocks when the power is cut) or fail-secure (which locks when the power is cut) configurations. Their compact size allows for seamless integration into the kiosk's design, ensuring both functionality and security in a streamlined package.

1.1.8 RGB Light Emitting Diode

Figure 9. RGB Light Emitting Diode (LED)

Image Source: https://usercontent.one/wp/www.okuelectronics.com/wp-content/uploads/2020/02/61DdW9-R5wL._AC_SL1200_.jpg

Figure 9 shows the 5mm RGB LED which was used to visually indicate the status of each locker compartment. It received data from the Arduino Mega, which controlled the LED to display one of the two different colors based on the compartment's availability. The LED turns green to indicate that a compartment is available or does not have any item inside it and red to imply that it was occupied or not available due to the presence of an item inside it. This configuration provided a clear, real-time visual indicator for users that improves their overall experience and the operational efficiency of the locker kiosk system. The RGB LED where RGB stands for red, green, and blue color channels had a 5mm diameter and used up around 20mA per color channel.

1.2 Block Diagram

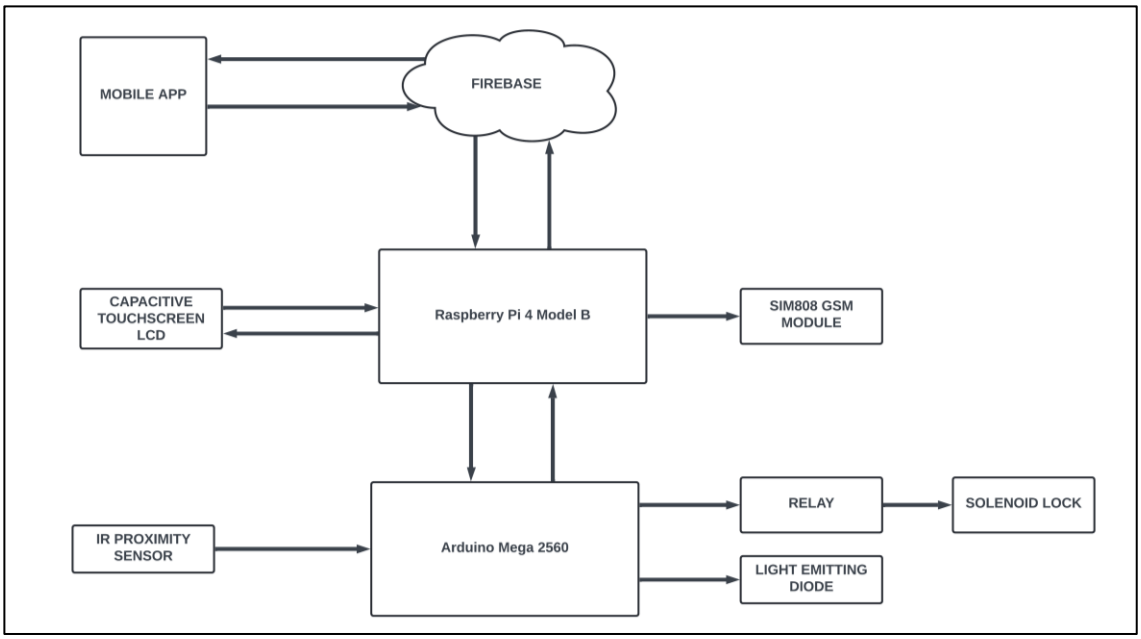


Figure 10. Block diagram of the Mobile Application-Monitored Locker Kiosk

Figure 10 shows the block diagram of the study, where the hardware components were integrated to create the locker kiosk system.

The system is composed of two main processing units, namely the Raspberry Pi 4 Model B and Arduino Mega 2560. The Raspberry Pi is the primary and central processing unit of the system, where all commands originate and where information is processed. It interfaces with Firebase, a cloud-based platform, to handle the storage of data and to facilitate communication between the kiosk system, which is shown in the capacitive touchscreen LCD, and the mobile application. The Raspberry Pi receives power to operate its connected peripherals and modules.

The Raspberry Pi manages inputs from a capacitive touchscreen LCD that provides a local interface for users to interact with the system directly at the kiosk to view status information and enter commands. Also, it communicates with the

Arduino Mega 2560, controlling and monitoring connected devices such as sensors, locks, and actuators. It interfaces with the SIM808 GSM module to enable communication via SMS.

Meanwhile, the Arduino functions as a microcontroller, managing direct interactions with hardware components. It receives commands from the Raspberry Pi and executes control over connected devices. The Arduino interacts with various input and output devices, such as the IR proximity sensors for detecting the items to be stored in the compartment, the solenoid locks for securing the compartment, a relay that acts like a switch to control the solenoid locks, and LED for providing visual indicators regarding the status of the compartments.

1.3 Circuit Diagram

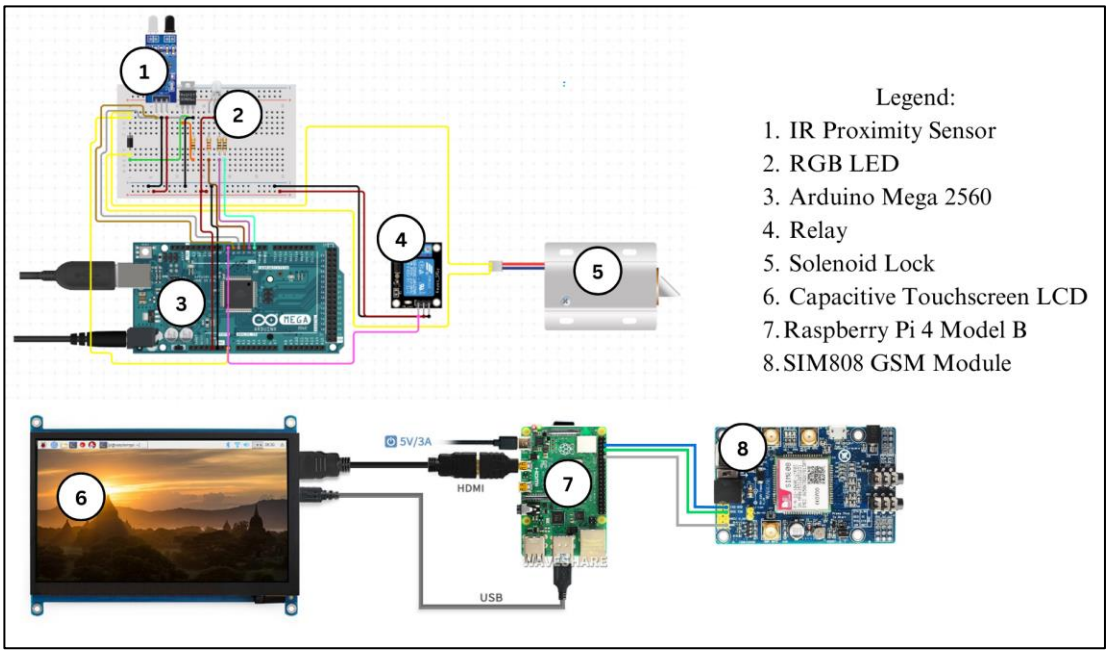


Figure 11. Circuit diagram of the Mobile Application-Monitored Locker Kiosk

Figure 11 shows the circuit diagram of the system which refers to the interconnection of the hardware components, which are the following: (1) IR Proximity Sensor, (2) RGB LED, (3) Arduino Mega 2560, (4) Relay, (5) Solenoid Lock, (6) Capacitive Touchscreen LCD, (7) Raspberry Pi 4 Model B, and (8) SIM808 GSM Module.

1.4 Pictorial Design

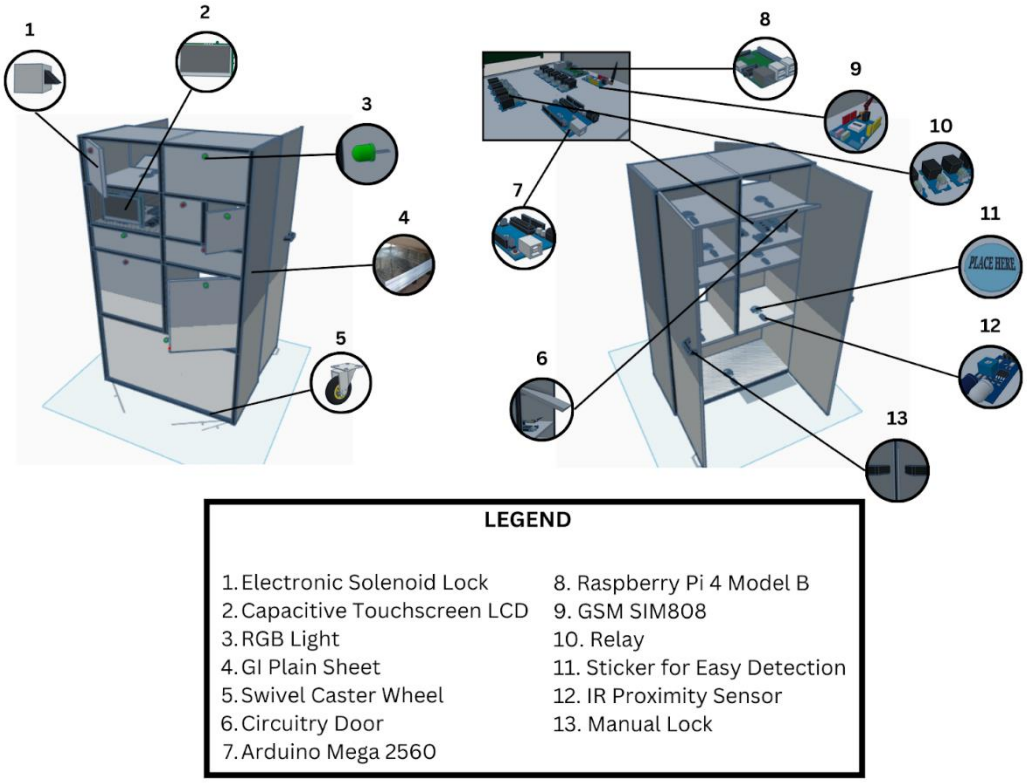


Figure 12. Pictorial diagram of the Mobile Application-Monitored Locker Kiosk

Figure 12 presents the hardware designs, components, and sensors used in the development of a mobile application-monitored locker kiosk system, with the integration of Raspberry Pi 4 Model B and an Arduino Mega 2560. This locker system comprises 9 compartments with various sizes to accommodate different items: compartments 1 and 2 (medium, 12” × 9” × 7”) for medium-sized packages and belongings; compartments 3 and 4 (small, 5” × 9” × 7”) for smaller items like gadgets and accessories; compartments 5 and 6 (extra-small, 12” × 9” × 3”) for compact items such as documents and wallets; compartments 7 and 8 (large, 12” × 9” × 15”) for bulkier items or larger packages; and compartment 9 (extra-large, 24” × 9” × 15”) for suitcases and oversized packages.

2.0 Software Design

This section involves the software design for the locker kiosk system, where details on the software tools used, user requirements, system specifications, data flow, use cases, and development methods are shown.

2.1 Software Requirements Specification

This section presents the specific software tools and platforms used in the development of the locker kiosk. Essential software components were explained, such as programming environments and databases, which were essential for the system's successful implementation.

2.1.1 Arduino IDE

Figure 13. Arduino IDE

Source:

https://www.arduino.cc/wiki/370832ed4114dd35d498f2f449b4781e/arduino.svg

The researchers used Figure 13, Arduino IDE, which is an open-source platform, to program the microcontroller used in the locker kiosk's hardware component. This allowed an easier way to write code and upload it to the

Arduino board through its user-friendly interface, pre-written library codes, and a combination of editing, compiling, and uploading codes into a single software package. This software has a large community, so tutorials, examples, and support can be found easily online if there are problems encountered in using this. This is also widely supported, allowing it to work with any Arduino board with compatibility and a smooth development experience.

2.1.2 Flutter

Figure 14. Flutter

Source: https://storage.googleapis.com/cms-storage-bucket/847ae81f5430402216fd.svg

Flutter, shown in Figure 14, was used in the development of the mobile application. It is an open-source framework by Google used to build beautiful, natively compiled, multi-platform applications from a single codebase. It provides a wide range of built-in widgets, which simplifies the creation of the mobile application's user interface. Also, it has a hot reload functionality for instant feedback on code changes to the running application. Flutter has a large and active developer community, which helps in finding plenty of online resources, tutorials, and libraries for the development of mobile applications.

2.1.3 Firebase Realtime Database

Figure 15. Firebase Realtime Database

Source:

https://miro.medium.com/v2/resize:fit:1400/format:webp/1*HFlYgB6gVLc4Su9HsB9MZg.png

Firebase Realtime Database, shown in Figure 15, is a cloud-hosted service used to store and sync JSON data in real time. It is designed to work with mobile and web software development kits (SDKs) with no server setup. It allowed the storage of information on the transaction within the locker kiosk in the cloud in JSON format, a flexible and widely used data structure, making it accessible to the mobile application as well as the kiosk itself. Its real-time synchronization feature allowed any changes made to the data instantly reflected in the database, as well as the mobile application and the kiosk to ensure that the latest information about the locker kiosk is available to all connected devices.

2.1.4 Python Programming Language

Figure 16. Python Logo

Source: https://www.python.org/static/community_logos/python-logo.png

Python, shown in Figure 16, is a widely used high-level programming language with a general purpose, capable of supporting object-oriented programming. It offered an easy-to-build user interface, and rapid development, and reusing code were used in developing the kiosk application. Also, it runs well on Raspberry Pi, allowing a streamlined interaction with the hardware components and communicating with the mobile application.

2.2 User Requirements

1. The user should be able to start drop-off and retrieve transactions of an item in the locker kiosk through the kiosk system.
2. The user (security guard) should be able to monitor the number of transactions and compartment availability in the locker kiosk and unlock the compartment, if necessary, through the manual lock at the back of the locker.

2.3 System Requirements

1. The system should be able to automatically store the data when a drop-off and retrieval transaction occurs in the database.
2. The system should be able to send notifications to both the sender and retriever of the item once a transaction (drop off and lost & found-retrieve) is made.
3. The kiosk system should have an intuitive and user-friendly user interface that guides the user throughout the transaction process.

2.4 Data Flow Diagram

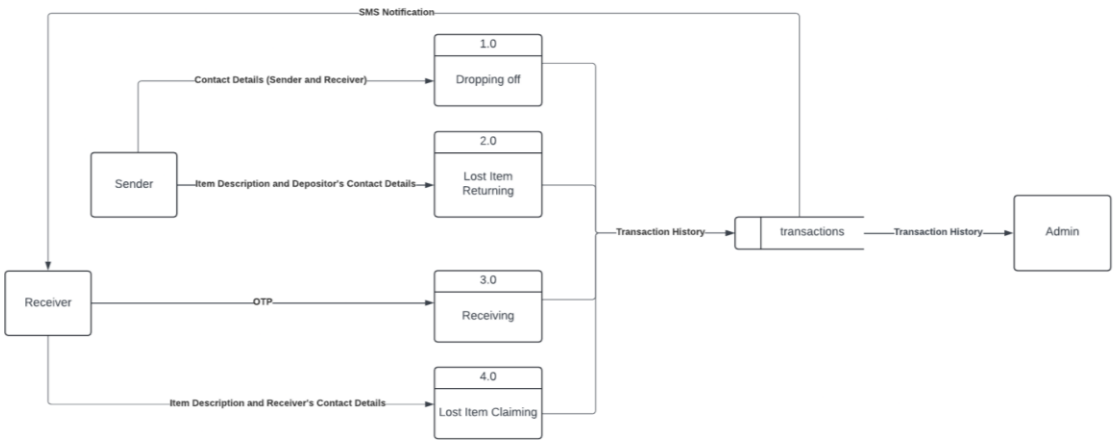


Figure 17. Data Flow Diagram

Figure 17 shows the data flow diagram of the system comprising three external entities (sender, receiver, and admin), and four processes (Dropping off, Lost Item Returning, Receiving, and Lost Item Claiming), all of which interact with a central Database (transactions) to store transaction histories. In the dropping off (1.0) process, the sender inputs contact details (of both the sender and receiver) and selects a compartment for storage. This data is processed by the system and stored in the

Database (transactions). Subsequently, the system sends an SMS notification containing an OTP to the receiver. For the receiving (3.0) process, the receiver uses the OTP received from the SMS notification to retrieve the stored item. The system verifies the code, facilitates the retrieval, and records this transaction in the database. For the lost and found, the data flow would start with the sender who will input the item description and its contact details in the Lost Item Returning (2.0) process, then this transaction will be saved in the database. In the Lost Item Claiming (4.0) process, the receiver enters a description of the lost item and its contact details. The system matches this description with the data stored in the database. If a match is found, the receiver retrieves the item; the transaction is stored in the database, and the receiver of the lost item will be notified of the contact details of the sender who stored the lost item in the compartment. All the transaction history stored in the database will be reflected in the mobile application, which is accessed by the admin or the security personnel.

2.5 Use Case Diagram

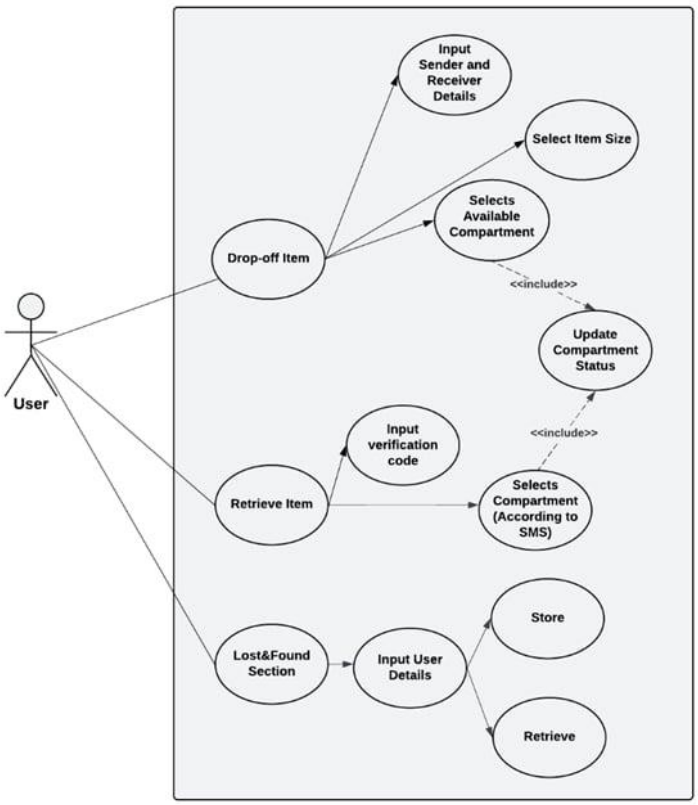


Figure 18. Use Case Diagram of the Kiosk

Figure 18 represents a user interacting with a locker kiosk system to perform various actions, such as dropping off items, retrieving items, and managing lost and found items. The diagram is divided into different use cases within the user interface, showing the processes and steps involved in each interaction.

1.) **Drop-off** - allows the user to deposit an item into the kiosk. This process includes the following steps, represented by `<<include>>` relationships:

Input Sender and Receiver Details: The user provides details about the sender and receiver, ensuring the system has the information to notify the recipient and manage the item appropriately.

Select Item Size: The user selects the size of the item to assist them in which compartment they should place their item. Each size also provides item suggestions to the user.

Selects Available Compartment: The system guides the user to select an empty or available compartment, ensuring secure storage until retrieval. Then, the system will update the compartment status after the transaction.

2.) **Retrieve -** involves retrieving an item from the locker compartments. This process includes:

Input Verification Code: The user must enter the One Time Password (OTP) sent by the system via SMS. This ensures that only allowed individuals can access the item inside the compartment.

Selects Compartment (According to SMS): The system will check if the One Time Password (OTP) entered by the user matches the selected compartment before it unlocks, ensuring security and accuracy. After the transaction, the system will update the selected compartment's status.

3.) **Lost & Found** - designed to handle misplaced or unclaimed items. It comprises of two main activities:

Store: When a user finds a lost item, they can store it in the kiosk. The system provides the option to categorize the item by selecting item categories, aiding in effective cataloging and retrieval.

Retrieve: Owners searching for their lost items can retrieve them by matching the descriptions and categories provided during storage.

Selects Compartment: The user can select a compartment depending on its transaction: store or retrieve. Then, the system will update the selected compartment's status.

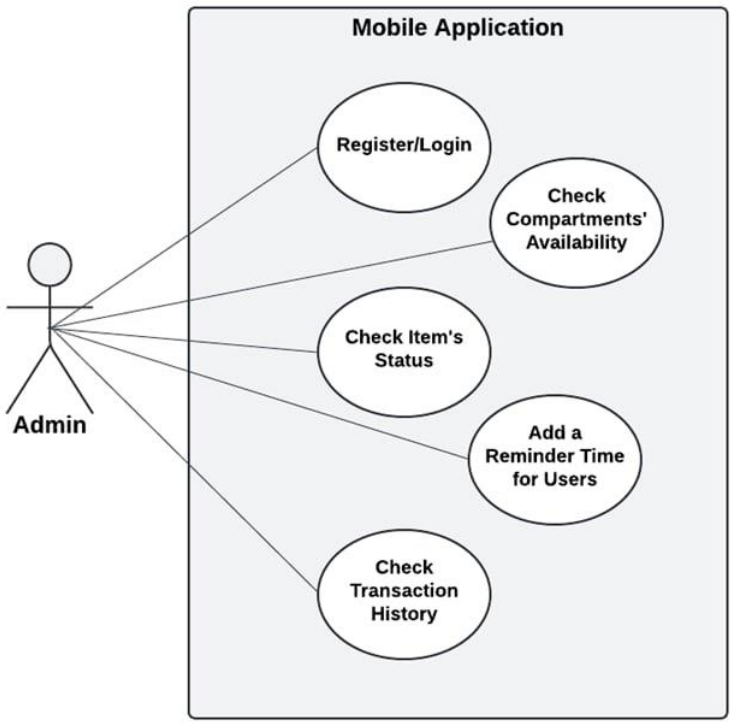


Figure 19. Use Case Diagram of the Mobile Application

Figure 19 illustrates the interactions between an admin (specifically security personnel) and a mobile application dedicated to monitoring and managing a locker system. Each use case within the mobile application defines specific functionalities that the admin can perform, ensuring the efficient operation and security of the locker system.

1.) **Register/Login** - represents the initial step for the admin to access the mobile application. This step ensures that only allowed personnel can log into the system, maintaining security and control over who can monitor and manage the lockers. By logging in, the admin verifies their identity and gains access to the application's features.

2.) **Check Item's Status** - allows the admin to monitor the real-time status of every item stored inside each locker's compartment. This functionality enables the admin to easily check whether the item is still pending or completed. The information shown in the item status can be used to address users' concerns regarding their items.

3.) **Check The Compartment's Availability** - the compartment's availability information allowed the admin to verify which compartments in the locker system are currently available for use.

4.) **Add a Reminder Time for Users** - allows the admin to set a specific time to remind users, particularly receivers, who still have pending items to be retrieved. This sends a message to the retriever to remind them about their items in the

locker. Admin can use this, specifically when school hours are about to end, to ensure that no items remain stored in the locker for extended periods.

5.) **Check Transaction History** - allows the admin to review the transactions' history of the locker system. This includes viewing records of completed transaction details, such as the type of transaction, sender's and receiver's name and contact, drop-off date, and the received date. Monitoring transaction history helps in auditing, resolving issues, and ensuring transparency in the system's operations.

Overall, this use case diagram outlines the specific interactions that an admin will encounter with the mobile application, solely, for monitoring the locker system. Each use case is designed to provide critical functionality that improves the security, efficiency, and management of the lockers, ensuring that the system operates effectively, securely, and reliably.

2.6 Software Development Method

2.6.1 Requirement Analysis

To ensure the system meets user demands, a thorough analysis of its functional and non-functional requirements was conducted at this stage. Identifying the hardware and software components was essential to facilitate real-time data collection, processing, and transmission.

2.6.1.1 Functionality

The system was designed to manage the processes of drop-off, storage, and retrieval of items. Through the kiosk interface, users can interact with the system and start transactions where they can drop off items into available compartments. The availability of the compartments is visible to users through the LED indicator of the locker and it is also displayed on the kiosk interface. The system generates a unique verification code, or a one-time password (OTP) that will be sent to the intended recipient via SMS notification. While retrieving an item, the receiver inputs the verification code at the kiosk that will authorize them to open the compartment that matches the code entered. Moreover, the system extends to a mobile applicated designated for security personnel that enables them to monitor the compartment's availability, view transaction history, and set a reminder time for pending transactions. These functionalities ensured a secure, efficient, and user-friendly process for managing and retrieving items from the locker.

2.6.1.2 Usability

The system was designed to be user-friendly, ensuring easy and intuitive interaction with the user. Simple and direct prompts were featured in the user interface and the user manual to prevent transactions' complexity. The system is accessible to users of all ages

and technical proficiency, including the mobile application for security personnel.

2.6.1.3 Reliability

The system was designed to provide security for users and high reliability for consistent and dependable performance given a stable internet connection. The system ensures security through its feature of sending a one-time password (OTP) to the recipient only and the use of a locking mechanism that can only be opened through the received one-time password (OTP) by the intended recipient.

2.6.1.4 Performance

The system was designed to deliver optimal performance while dependent on a stable internet connection. Given a stable internet connection, the system ensures fast and efficient operations, providing a smooth user experience. However, with unstable internet connections, the system might also encounter delays in executing tasks.

2.6.1.5 Supportability

The system was designed to ensure ease of maintenance and further updates, making it also flexible for new features to be integrated with it. These features ensure the system's adaptability with maintenance and the evolving user needs.

3.0 System Design

3.1 System Architecture

The study integrated hardware and software components to develop the functional locker kiosk system. The Raspberry Pi 4 Model B served as the main controller that interfaced with several components such as the Arduino Mega 2560 and Capacitive Touchscreen LCD. It also communicates with the Firebase Realtime Database to ensure synchronization of real-time data between the mobile application and the kiosk system. Users interact with the system through the kiosk, while the security personnel interact with the mobile app, which connects to Firebase to manage locker transactions. The Arduino Mega handled the hardware operations, including controlling the solenoid locks through the relays for securing compartments, using the IR proximity sensor to detect item presence, and managing the RGB LED as visual feedback of the compartment's availability. The SIM808 GSM Module, connected to the Raspberry Pi, sends SMS notifications to users with access codes. The system is designed for robust functionality and user-friendly interaction, providing a seamless experience for dropping off, retrieving items, and handling lost-and-found operations while ensuring security through real-time monitoring and controlled access.

3.2 System Flowchart

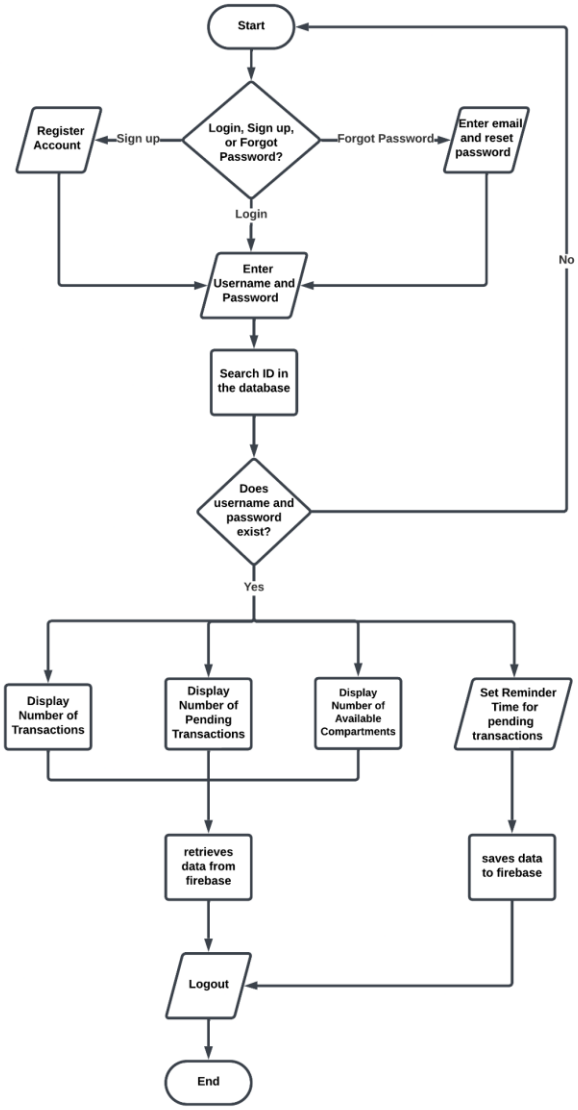


Figure 20. Flowchart Diagram of the Mobile Application

Figure 20 shows the flowchart diagram of the mobile application, outlining the process for managing the kiosk system by the security guards using the mobile application. The process starts with the user being shown three options: login, sign up, or forget a password.

If the user does not have an existing account, they can select a sign-up to register. Once the account is created, the user can proceed to log in. For cases where the user forgets their password, they can choose the forgot password option, where they will be asked to enter their email to set a new password. After resetting their password, the user can use their new credentials to log in.

Upon selecting login, the user is prompted to input their username and password. The system then searches for the inputted credentials in Firebase. If the username and password match an existing account, the mobile application brings them to the dashboard that shows the following: the number of transactions, pending transactions, and available compartments. Choosing any of the three options retrieves the relevant data from Firebase and shows the necessary information, this enables the user to manage and monitor the locker kiosk effectively.

Additionally, the user can set a reminder time for pending transactions. This reminder is saved in Firebase, which sends automated messages to notify the recipients of their pending items at the specified time. If the user has no further concerns, they can choose to log out.

This process ensures that the security guards can efficiently access and manage the locker kiosk by monitoring item statuses, compartment availability, and transaction histories, and setting a reminder time for pending transactions—all through the mobile application.

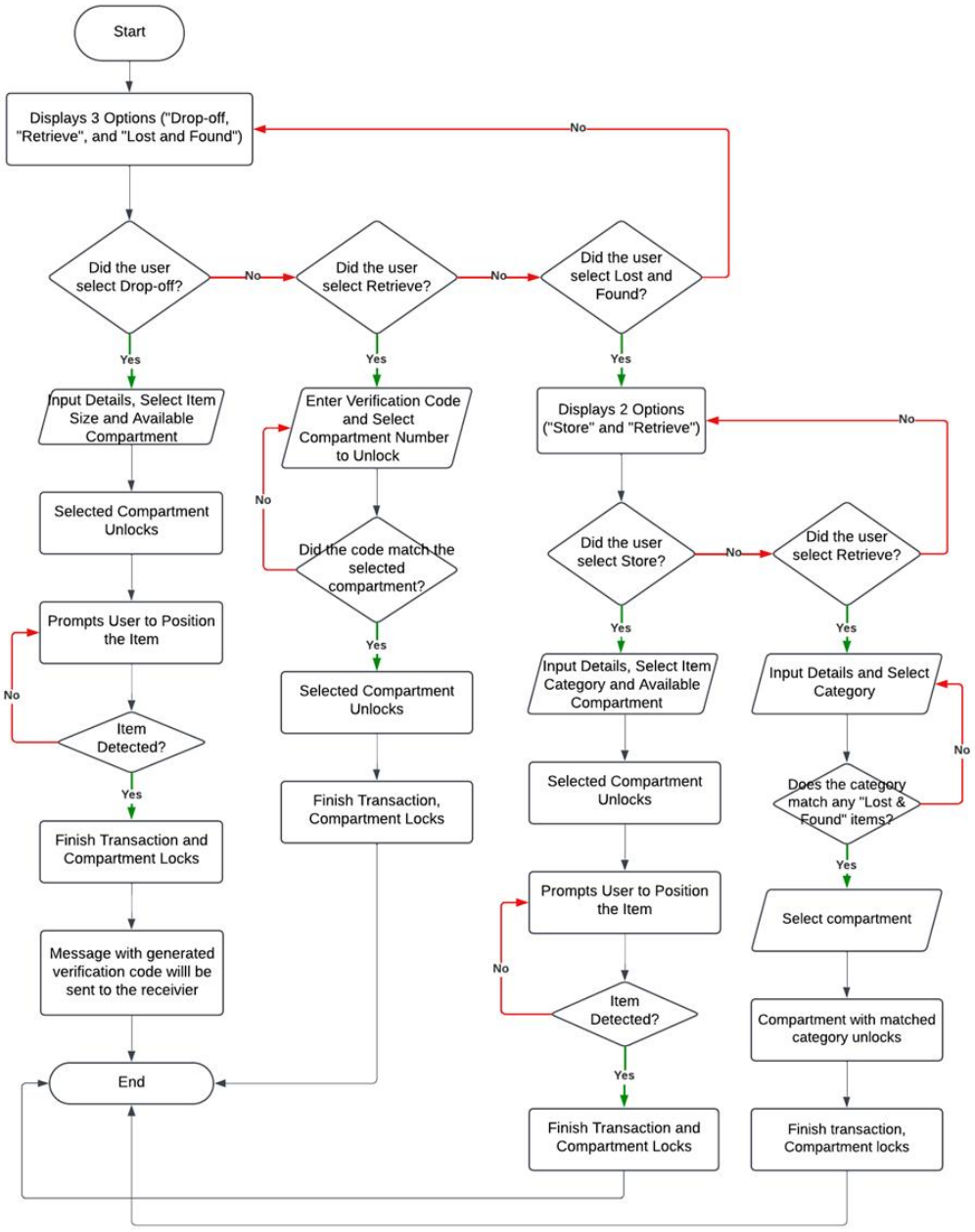


Figure 21. Flowchart Diagram of the Kiosk's General Process

Figure 21 presents the general flowchart of the system which illustrates the overall process of the three transactions; drop-off, retrieve, and the transaction in the lost and found section.

If the user selects **Drop-off,** they are prompted to input details, including the contact number of the receiver that is necessary to retrieve the item. After several instructions, such as selecting the item size and available compartment, the selected compartment unlocks and then prompts the user to position the item to ensure that the user has an item to drop off. The locker system uses an IR sensor to accurately detect the item and ensure that the item has been successfully placed inside the compartment. If the item is detected, the user is prompted with a few simple instructions to finish the transaction and lock the compartment. On the other hand, if no item is detected, the system loops back to prompt the user to position the item again until it gets detected. The process of drop-off will be completed once the message with the generated verification code or the one-time password (OTP) is sent to the receiver. Once the process in drop-off transaction is done, all the data will be updated in the database.

If the user selects **Retrieve**, the transaction will be quicker than the drop-off transaction. The system only requires the user to input the one-time password (OTP) included in the SMS sent to them. Once the code is entered, the user is prompted to select the compartment number included in the message as well. The system retrieves the compartment and code details from the database to verify the match. If the code entered by the user does not match the selected compartment, it will remain unlocked

and the system displays an incorrect code message. The user is allowed to type the code again. If the code matches the selected compartment, it unlocks, allowing the user to retrieve the item. Further instructions to the user such as locking the compartment will be displayed to the user. After the transaction, the database is updated to mark the compartment as currently available.

If the user selects **Lost and Found**, the system provides two options: **Drop-off and Retrieve.** If the user chooses the **Drop-off** or to store a found item, the process mirrors the drop-off flow. The user is prompted to select from general to specific categories to acquire the description and identify features of the item. Then, the user selects an available compartment. Similar to the Drop-off process, the user positions the item inside the compartment, and once the item is detected, further instructions to lock the compartment will be displayed on the user interface of the kiosk. The transaction will be saved in the database. While, if the user selects **Retrieve,** they will select from general to specific categories as well, similar to the categories presented during the Drop-off process. The user will then select an occupied compartment to retrieve their item. If the selected compartment has the record of their selected categories and description on the system, the compartment unlocks, allowing the user to retrieve the item. However, if the selected categories do not match any of the compartments, all of them will remain locked and the user is allowed to refine their search or try to change selected categories.

The system integrates error handling by providing prompts and loops to address invalid inputs or incomplete actions. All user actions, including drop-offs,

retrievals, and lost-and-found transactions, are securely recorded in the real time firebase for tracking and accountability. This ensures smooth and secure operations while enabling users to efficiently manage locker transactions.

Figure 22 to Figure 24 shows the breakdown of the three primary transactions of the system. Figure 22 displays the flowchart for the drop-off transaction, Figure 23 for the retrieval transaction, and Figure 24 for the lost & found transaction.

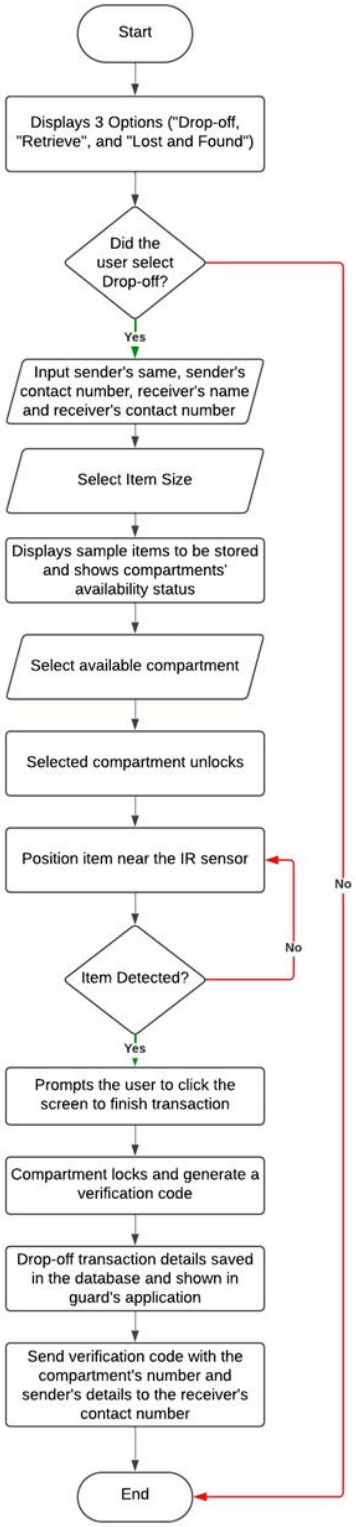


Figure 22. Flowchart Diagram of the Kiosk: Drop-off Process

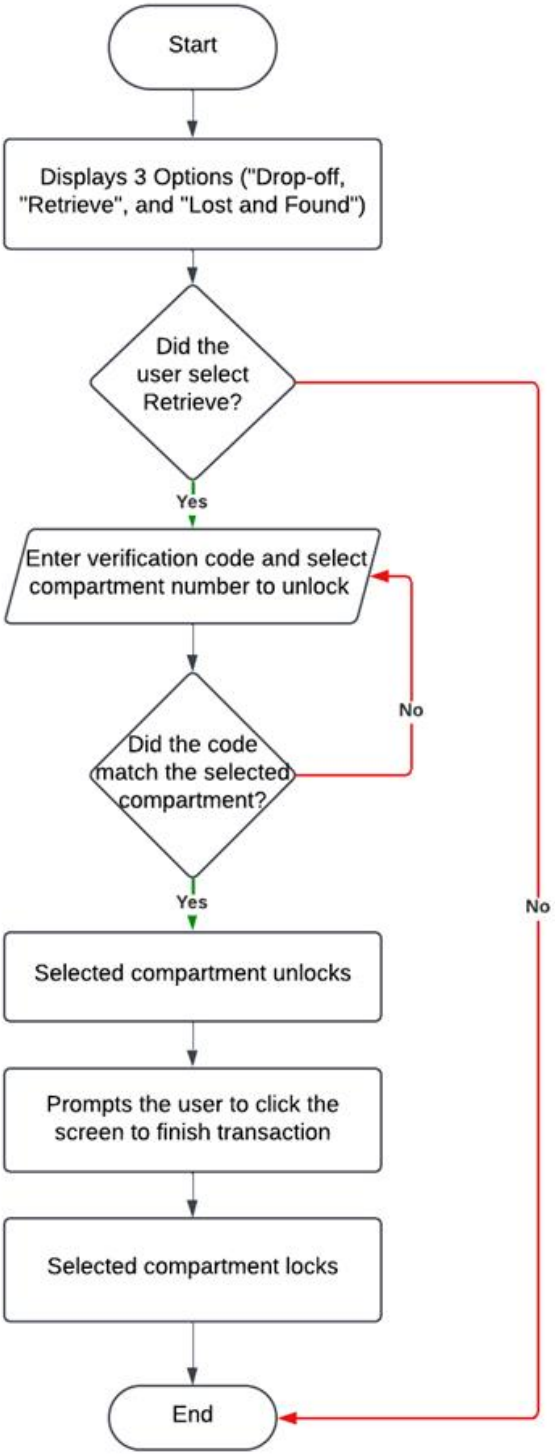


Figure 23. Flow Chart Diagram of the Kiosk: Retrieve Process

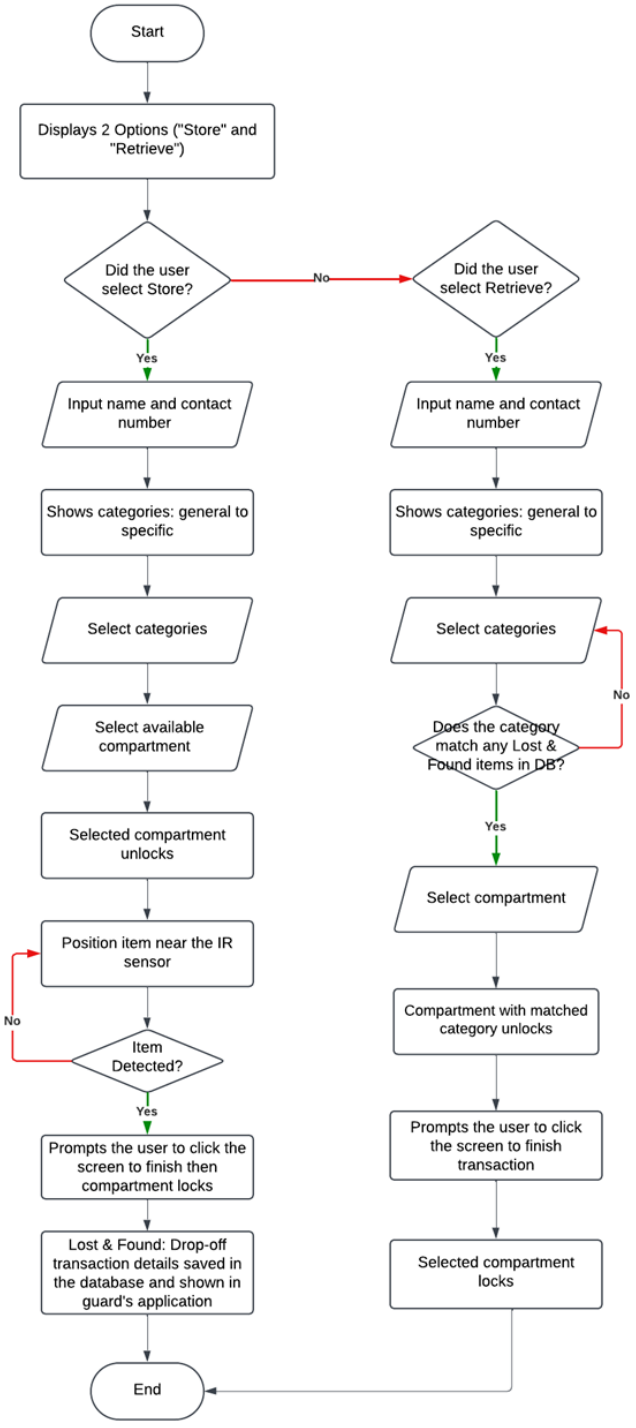


Figure 24. Flowchart Diagram of the Kiosk: Lost & Found Process

4.0 Testing & Evaluation Procedure

The developed locker kiosk system was tested and evaluated in the aspects of both hardware and software. Multiple tests were conducted to ensure the proper functionality of the system. This section presents the testing and evaluation procedure done to the hardware and software components of the locker kiosk system.

Hardware

The hardware components of the system were tested to ensure that they function correctly and meet the required specifications. The evaluation of the components is as follows:

Solenoid Locks

The solenoid locks were assessed individually to ensure that every lock in each compartment operates correctly and accurately unlocks each compartment according to the user's transaction. The evaluation included:

- **Accuracy Tests**: Ensuring that only the compartment that matched the one-time password (OTP) entered will be unlocked. Multiple trials were conducted to verify the reliability of the locking mechanism.
- **Response Time**: Measuring the duration of time between the confirmation of the user about the selected compartment and the actual unlocking of the solenoid lock.

IR Proximity Sensors

The IR proximity sensors positioned in each compartment were tested to determine their effectiveness and accuracy in detecting items inside the compartment. The assessment involved:

- **Accuracy Tests**: Verifying whether the sensor correctly detects the item to be stored inside the compartment. Objects of different sizes and materials were used to evaluate the sensor's detection capabilities.

Touchscreen

The touchscreen was tested multiple times to ensure its responsiveness, accuracy, and durability. The evaluation needed:

- **Responsiveness Tests**: Measuring how quickly and accurately the touchscreen registers user inputs. This test ensured that the system provided smooth transactions with the user.

- **Accuracy Tests**: Verifying that the touchscreen accurately detects touch points and translates them correctly on the display.

Software

The software component of the system has undergone various tests to ensure proper interaction with the hardware, ensuring optimal functionality and desired outputs. The testing procedures for the software component were as follows:

The software in our system included programs for each microcontroller. These consist of an Arduino Mega with a SIM808 GSM Module and a Raspberry Pi 4 Model B. Programs for the Arduino microcontroller were developed using the Arduino IDE, while Google Firebase Realtime Database stores data from the kiosk and mobile application. The mobile application is developed using Flutter. Python, a widely used high-level programming language, supports object-oriented programming and provides an easy-to-build user interface, rapid development, and code reusability, which was leveraged in developing the kiosk application. Additionally, Python runs efficiently on the Raspberry Pi, facilitating seamless interaction with hardware components and communication with the mobile application.

These tests helped to ensure that the software component of the system was functioning correctly, interacting well with the hardware, and producing accurate data.

Functional Testing

Functional testing aims to verify the proper functionality of the software tools used in the system. This involves testing each feature of the software to correctly ensure that it works accordingly.

- **User Authentication:** In the mobile application, users can successfully log in and out of their accounts. In the locker system, only a valid one-time password (OTP) can unlock a particular compartment.
- **Drop-off and Retrieval:** Multiple tests were conducted to ensure that the correct compartments opened and closed based on the input codes.

- **Lost and Found Management:** The system was tested to verify the correctness of item drop-off and retrieval processes.
- **Notifications:** The system's ability to send notifications, such as codes for item retrieval, was evaluated for accuracy and timeliness.

Usability Testing

Usability testing focused on evaluating the system's ease of use and overall user experience. Real users were observed as they interacted with the system, and feedback was collected.

- **Interface Navigation:** Users were assessed on how easily they could navigate the mobile app and kiosk touchscreen.
- **User Instructions:** The clarity of instructions for tasks like dropping off and retrieving items was evaluated
- **Feedback Mechanisms:** The system's ability to provide confirmation messages and error alerts was tested.
- **User Satisfaction:** Surveys and interviews were conducted to gather feedback on user experience and satisfaction.

Reliability Testing

Reliability testing ensured that the system functioned consistently over time without failures.

- **Stress Testing:** The system was subjected to high usage conditions to identify potential weaknesses.
- **Error Handling:** The response to unexpected inputs and errors was tested to ensure the system provided clear error messages and recovered properly.
- **Uptime Monitoring:** The system's availability was tracked over an extended period to assess its reliability and stability.
- **Transaction Integrity:** Verify that all transactions (such as item drop-offs and retrievals) are accurately recorded and processed without loss of data.

Performance Testing

Performance testing was conducted to evaluate how well the software operates under different conditions.

- **Response Time:** The time required for the system to process user inputs, such as entering codes and unlocking compartments, was measured.
- **Load Testing:** The system's performance was assessed under varying usage conditions, including multiple users accessing it in sequence.
- **Resource Usage:** System resource consumption, including CPU, memory, and network bandwidth, was monitored to ensure efficient operation.

- **Scalability:** The ability of the software to handle an increasing number of users and transactions without performance issues was tested.

These tests were performed to confirm that the smart locker system functions correctly, remains user-friendly, operates reliably, and maintains high performance across different devices and environments.

5.0 Method of Research

5.1 Research Design

The study used a quantitative research design in developing and evaluating the locker kiosk system. Its objectives include measuring the execution time of the system, assessing its accuracy, and evaluating the acceptability of the system. Quantitative data were collected through testing of the system. The testing includes a timing test for the SMS notification and unlocking of the compartment, an accuracy test on the IR proximity sensor's detection of the item and the reflected compartment availability in the kiosk screen and mobile application, and a quantitative assessment of acceptability using user experience questionnaires. Statistical methods were used to comprehensively evaluate the system's performance and acceptability rate.

5.2 Research Instrument

To assess the execution time and accuracy of the system, system observation was used by the researchers while it was tested by the respondents. Timestamps were taken from the log entry to compute the execution time of the system in sending an SMS after a drop-off transaction and the unlocking of the compartment upon

confirmation in the system. As for accuracy, the behavior of the IR proximity sensor was observed on the output shown on the screen by the system while the mobile application and the kiosk were observed if they reflected correctly the compartment availability saved in Firebase.

To determine the acceptability rate of the locker kiosk system using Raspberry Pi and Arduino, a survey was administered among various selected individuals at Marinduque State University (Boac Campus), which included the use of a User Experience Questionnaire employing a 7-point Likert Scale to provide a nuanced and detailed approach to measuring user attitudes and satisfaction. A 7-point Likert scale can help capture the full range of their experiences and perceptions regarding the system. By asking students, staff, faculty members, security personnel, and riders who deliver at Marinduque State University (Boac Campus) to rate their agreement with statements about the ease of use, reliability, security, and convenience, researchers gathered detailed insights into how well the system meets their needs.

The tools and methods stated above were used to identify useful information for the numerical data gathered from the trials and testing that were conducted.

5.3 Data Gathering Procedure

Data were gathered to assess the locker kiosk system. Several 130 respondents were recruited through convenience sampling. This sampling involves

selecting respondents from those who are readily available and willing to participate in the study for practicality and efficiency.

After being recruited, the respondents were able to try and experience how the system works and were then asked to fill out the User Experience Questionnaire. The questionnaire was administered both in person and online, based on the preferences of the respondents. The researchers provided clear instructions and a comprehensive set of questions in the questionnaire, which made it easier for the respondents to understand and complete the questionnaire. This data collection provided valuable insights into the attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty of the locker kiosk system, aiding in its assessment and possible improvement.

5.3.1 Data Collection

The data collection for the assessment of the system's average execution time and accuracy was conducted through ten (10) trials.

To compute the average execution time of the system in sending SMS notifications through the GSM module after a drop-off transaction, the researchers recorded timestamps by reviewing the log entry to identify the moment the user confirms the drop-off transaction and the moment the SMS is sent to the retriever of the item. By calculating the execution time between the two timestamps and analyzing the data from ten (10) user interactions, the researchers were able to determine the average execution time for sending

OTPs via SMS notifications after a drop-off transaction. Similarly, to compute the average execution time of opening a locker compartment using the electric solenoid lock, the researchers reviewed the log entry to identify the moment the user confirms the opening of the correct compartment and the moment the solenoid lock unlocks. The execution times for unlocking the compartments were calculated and averaged across the ten (10) trials.

To assess the accuracy of the system in detecting an item's presence using IR proximity sensors, the researchers conducted ten (10) trials – five (5) of them were instances where there was an item in the compartment while the remaining five (5) trials were the instances where there was no item placed inside the compartment. During these sessions, the sensor's binary output was observed, with the infrared light turning on to indicate an item's presence and turning off to indicate its absence. The researchers tracked the system's ability to correctly detect an item to be placed in the compartments, as well as instances of incorrect detections. The percentage of correct detections of the presence and absence of an item from the total number of trials was the measure of the accuracy of the system's item detection ability. Similarly, the system's accuracy was also assessed in terms of displaying compartment availability in both the kiosk and mobile applications correctly. The tests include the manual tracking of the compartments' actual status and comparing it with the displayed status to record discrepancies. The percentage of the discrepancies was used to measure the system's accuracy in displaying the correct compartment availability in both the kiosk and mobile applications.

To evaluate the acceptability rate of the locker kiosk system among different user groups at Marinduque State University (Boac Campus), the researchers recruited respondents from students, faculty members, security guards, and delivery riders using convenience sampling. The respondents were given the User Experience Questionnaire (UEQ) after testing the locker kiosk. The UEQ included questions about attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty of the Locker Kiosk System. Respondents answered the UEQ by rating their level of agreement or satisfaction with the system on a scale that ranged from "strongly disagree" to "strongly agree." The responses were collected and analyzed to compute the acceptability rate of the system.

5.4 Data Analysis Procedure

The gathered data were analyzed by the researchers to address the research questions in the statement of the problem of the study.

For question 1a, the average execution time for sending an OTP to the item-retriever after a drop-off transaction was computed by recording two (2) timestamps: T_{start} (the moment the user confirms drop-off on the kiosk) and T_{end} (the moment the recipient receives OTP via SMS after the drop-off transaction). To solve for the execution time (ET), T_{start} will be subtracted from T_{end}, which is shown in Equation 1:

$$ET = T_{end} - T_{start} \quad (1)$$

There will be ten (10) test cases, after the execution time for the test cases is computed, their average will be computed using Equation 2 to get the average execution time for the drop-off process.

$$ET_{avg} = \frac{\sum_{i=1}^{n} ET_i}{n} \quad (2)$$

For question 1b, the average execution time of the Locker Kiosk System in terms of Opening of Locker Kiosk Compartment using an Electric Solenoid Lock was computed by recording two timestamps: T_{start} (the moment the user confirms the unlocking of the compartment) and T_{end} (the moment the solenoid lock unlocks). To solve for the execution time, T_{start} was subtracted from T_{end}.

$$\text{ET for Compartment Unlocking} = T_{end} - T_{start}$$

There were ten (10) test cases, after the execution time for the test cases was computed, their average was computed to get the average execution time for Opening of the Locker Kiosk Compartment using an Electric Solenoid Lock.

$$ET_{avg} \text{ for Compartment Unlocking} = \frac{\sum_{i=1}^{n} \text{ET for Compartment Unlocking}}{\text{number of trials}}$$

For question 2a, a confusion matrix was used to evaluate the IR proximity sensor's accuracy in detecting the presence of the item stored inside the compartment.

Table 1.
Tabulation of the IR proximity sensor's accuracy in detecting the presence of the item stored inside the compartment

IR Proximity Sensor's Accuracy in Compartment n					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

Table 1 was employed across all compartments to aggregate the total values required for calculating the accuracy of the IR proximity sensors, using Equation 3 as the basis for the computation.

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (3)$$

Where:

TP: True Positives (correct occupied detection) - Instances where the system correctly detected the presence of an item.

TN: True Negatives (correct empty detection) - instances where the system correctly detected the absence of an item.

FP: False Positives (incorrect occupied detection) - Instances where the system incorrectly detected the presence of an item when the compartment was empty.

FN: False Negatives (incorrect empty detection) - Instances where the system incorrectly detected the absence of an item when the compartment was occupied.

For questions 2b and 2c, the actual and displayed information regarding compartment availability in both the kiosk and mobile application was documented across ten (10) test cases. The recorded statuses are represented as follows: 0 = Available and 1 = Unavailable:

Table 2.
Tabulation of the Accuracy of The Compartment's Information in the Kiosk and Mobile Application for nth Test Case

Trial n					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1					
2					
3					
4					
5					
6					
7					
8					
9					
Total					

Table 2 was used ten (10) times for the ten test cases. A binary value was used to determine whether the compartment was available or unavailable. When the actual and displayed status in the kiosk of the compartments are the same, the absolute error is 0. Otherwise, the opposite status of the compartment or the status has 0 and 1, and vice versa, there is an absolute error of 1. To get the absolute error, equation 4 was used:

$$\Delta x = x_i - x \tag{4}$$

where:

x_i = measured value (displayed status)

x = true value (actual status)

After ten (10) test cases, equation 5 was used to find the mean absolute error.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - x| \tag{5}$$

where:

n = number of errors (data points)

$x_i - x$ = absolute errors for each measurement

The mean absolute error (MAE) was then multiplied by 100 to get the percentage. A lower percentage of mean absolute error indicates a higher degree of accuracy in the kiosk and mobile application's representation of the compartment's availability information, while a higher percentage may indicate potential issues

within the system, such as data synchronization errors or communication problems between the database, the kiosk, and the mobile application.

For question 3, to determine the acceptability rate of the system, the researchers used the User Experience Questionnaire (UEQ) framework, which employs a 7-point Likert scale for collecting user feedback. The Likert scale is composed of verbal descriptions ranging from "Strongly Disagree" to "Strongly Agree" and is mapped to corresponding numerical values on the UEQ scale, which ranges from negative three (-3) to positive three (+3), as shown in Table 3. This allowed the quantitative measurement of user experience dimensions such as Attractiveness, Efficiency, Dependability, Stimulation, and Novelty.

Table 3.
Mapping Between Likert Scale and UEQ Scale

Likert Scale		UEQ Scale Value
Numerical Representation	**Verbal Description**	
1	Strongly Disagree	-3
2	Disagree	-2
3	Somewhat Disagree	-1
4	Neutral	0
5	Somewhat Agree	1
6	Agree	2
7	Strongly Agree	3

Several steps were involved to calculate the acceptability of the system. First, user responses were collected and assigned their respective UEQ values. These values were then used to calculate the weighted scores for each question, reflecting

the importance of each dimension based on user feedback. The weighted score for a question was calculated using formula 6:

$$\text{Weighted Score (WS)} = \sum_{i=1}^{n} (\text{Frequency of Response} \times \text{UEQ Value}) \quad (6)$$

Where:

n = total number of Likert scale options (typically 7 for UEQ).

Frequency of Response = number of responses for a specific Likert scale option

After calculating the weighted score for each question, the mean score for each dimension was determined by dividing the total weighted score by the total number of responses using Equation 7:

$$\text{Mean Dimension Score} = \text{Total Responses} / \text{Weighted Score} \quad (7)$$

Finally, the overall acceptability rate was determined by computing the average of all mean dimension scores across all dimensions using equation 8:

$$\text{Overall Dimension Score} = \sum^{k}_{i=1} \text{Mean Dimension Score} / k \quad (8)$$

Where:

k = total number of dimensions (e.g., Attractiveness, Efficiency, etc.).

The computed scores were interpreted using the verbal interpretation scale that was defined by Martin Schrepp and his collaborators. According to this framework, scores are categorized as "Very Bad," "Bad," "Below Average," "Neutral," "Good," "Very Good," or "Excellent," based on their range within the -3 to +3 scale. This is shown in table 4:

Table 4.
Verbal Interpretation of UEQ Scale

Score Range (UEQ Scale)	Interpretation
-3.00 to -2.01	Very Bad
-2.00 to -1.01	Bad
-1.00 to -0.01	Below Average
0.00 to 0.80	Neutral
0.81 to 1.50	Good
1.51 to 2.00	Very Good
2.01 to 3.00	Excellent

For instance, if a dimension such as Attractiveness receives responses like 10 for "Strongly Agree" (+3), 8 for "Agree" (+2), and 5 for "Neutral" (0), the weighted score is calculated. Dividing this by the total responses gives the mean score, which is then interpreted using Schrepp's verbal scale. By combining statistical computation with the standardized UEQ interpretation framework, this methodology ensures a robust and reliable approach to assessing the system's acceptability.

CHAPTER IV

PRESENTATION, ANALYSIS, AND INTERPRETATION OF DATA

This chapter presents a detailed framework and design of the prototype, as well as its integration with the accompanying kiosk and mobile application. It also includes the analysis of the data collected and tabulated during the evaluation process (refer to the Appendices for detailed results). These insights offer a comprehensive assessment of the prototype, highlighting its usability and acceptability ratings based on the evaluation scale.

1.0 Prototype Structure and Design

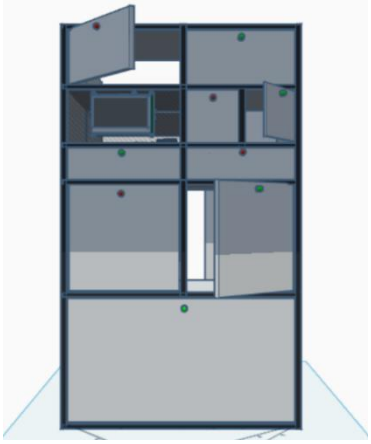


Figure 25. Front View of the Prototype

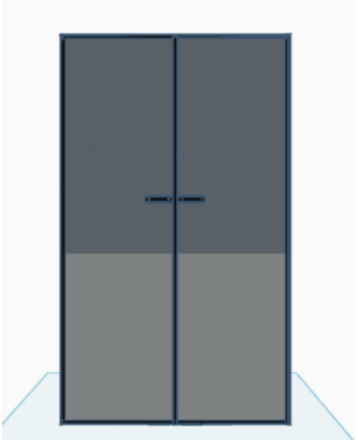


Figure 26. Back View of the Prototype

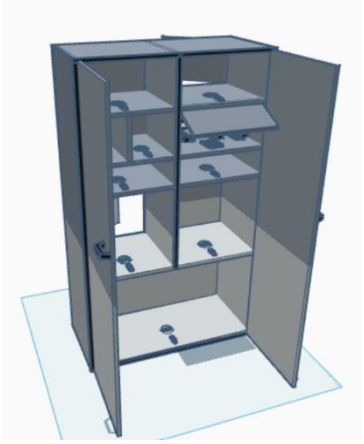


Figure 27. Inner Structure of the Prototype

The three figures above show the fully constructed prototype, built using GI plain sheets and tubular materials painted in a sleek combination of gray and black, aligning with the outlined software and hardware design specifications. The front view of the prototype, shown in Figure 25, showcases multiple compartment doors of the locker, each equipped with a solenoid lock, an IR sensor for item detection, and an LED light for

compartment status. One designated compartment houses the system's circuitry, which includes a soldered configuration of the Arduino Mega, Raspberry Pi, GSM SIM 808 module, relays, and other essential connections assembled on a PCB board. An external LCD touchscreen attached to this compartment enables user interactions and transaction processing, such as tapping for authentication and managing transactions at the kiosk.

The relay and solenoid lock ensure secure unlocking of compartments, while the IR sensor detects items placed inside. At the back of the prototype, shown in Figure 26 is a manually operated door for emergencies, allowing access to retrieve items when the front doors are inaccessible. Additionally, the circuitry compartment has a separate back door, shown in Figure 27, providing controlled access to the internal components to prevent unauthorized tampering.

2.0 Kiosk Structure and Design

Figure 28. UniLOCK Kiosk Screen Welcome Page

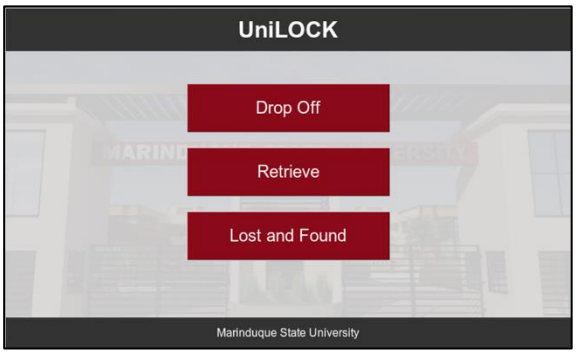


Figure 29. UniLOCK Kiosk Screen Transactions Option Page

Figure 28 displays the welcome page, where users can touch anywhere on the screen to initiate a transaction. Figure 29 prompts users to select their desired transaction type: drop-off, retrieval, or actions related to lost and found items. Once a transaction type is chosen, subsequent screens provide step-by-step instructions to guide users through completing their transaction. The screen serves as the primary interface, allowing users to control the locking or unlocking of compartments by providing the necessary information or details.

3.0 Mobile Application Structure and Design

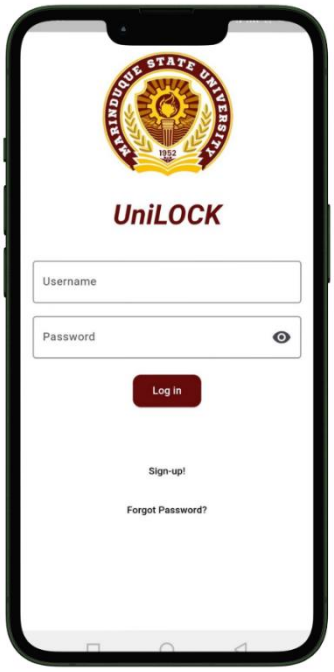


Figure 30. UniLOCK Mobile Application Log-in Page

Figure 30 shows the login page of the mobile application designed for security guards. To log in, users are required to enter their username and password. For first-time users, the application prompts them to sign up by clicking the "Sign Up" button, which redirects them to a registration page. Here, they must provide their phone number, email address, username, password, and password confirmation to create an account. Once the account is successfully created, they can log in using their credentials.

If a user forgets their password, they can click the "Forgot Password" button. This action prompts them to enter their registered email address. The system sends an email containing a link to reset their password. By clicking the link, users are redirected to a secure page where they can set a new password. Once the password is successfully updated, they can log in using their new credentials.

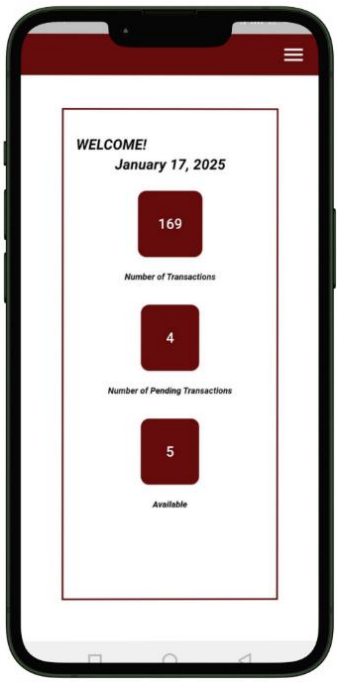


Figure 31. UniLOCK Mobile Application Dashboard Page

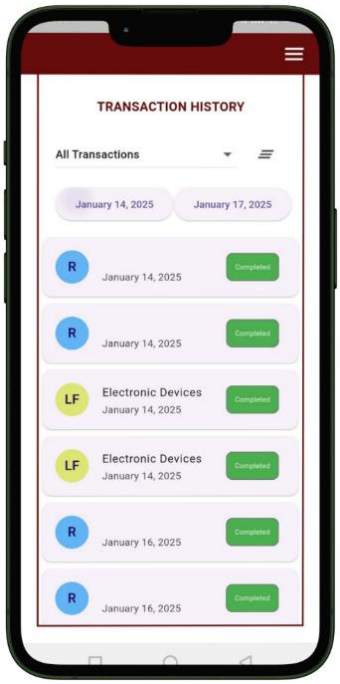


Figure 32. UniLOCK Mobile Application Transaction History

Figure 33. UniLOCK Mobile Application Compartments Page

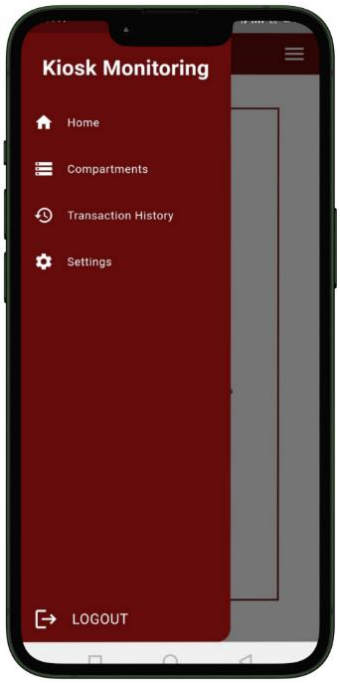


Figure 34. UniLOCK Mobile Application Sidebar

Figure 35. UniLOCK Mobile Application Setting of Reminder Time Page

Figure 31 depicts the dashboard of the app, displaying key information, such as the number of transactions, pending transactions, and available compartments. Clicking on any of these items redirects the user to a more detailed page. Selecting "Number of Transactions" or "Pending Transactions" opens a comprehensive transaction history which is shown in Figure 32, while selecting "Available Compartments" navigates to a page displaying all compartments, with their status indicated by specific colors: red for unavailable, green for available, and yellow for under maintenance which is shown in figure 33.

Users can also access these features through the sidebar, which is shown in Figure 34, which provides direct links to the transaction history and compartment status pages, mirroring the dashboard options. Additionally, the sidebar includes a settings option, leading to a page where users can configure reminder notifications for pending transactions, which is shown in Figure 35. This feature sends notifications to recipients, reminding them to retrieve their pending items. If the user has no further actions to perform, they can choose to log out via the sidebar.

4.0 Results and Discussions

Research Objective 1: To determine the average execution time of the locker kiosk system in terms of sending an OTP to the item-retriever after a drop-off transaction and opening of locker kiosk compartment using an electric solenoid lock.

To determine the average execution time of the system to send an OTP to the item-retriever after a drop-off transaction and unlocking the Locker Kiosk Compartment using the Electric Solenoid Lock, the researchers conducted ten (10) trials. Code snippets were used to record timestamps for each transaction, allowing the researchers to calculate the execution time for each trial and the average execution time across all 10 trials.

The average execution time for sending an OTP to the item-retriever after a drop-off transaction using the SIM808 module primarily depends on the stability of the internet signal. As shown in the table below, all notifications were successfully received, but the execution times varied depending on the strength and stability of the internet connection.

Table 5.
Summary of Execution Time for Sending OTP via SMS Notification Using SIM808 Module After Drop-Off Transaction

Sending OTP via SMS Notification	
Trials	**Execution Time (secs)**
1	25
2	13
3	65
4	14
5	58
6	14
7	16
8	23
9	13
10	16
Average Execution Time	25.7
Standard Deviation	19.3795

Table 5 summarizes the execution times for sending OTPs via SMS notifications using the SIM808 Module after a drop-off transaction across ten trials. All OTP notifications were successfully sent, with execution times varying between 13 and 65 seconds. Six notifications were delivered within 13–16 seconds, reflecting efficient performance under stable conditions. One trial recorded 23 seconds, another took 25 seconds, while the longest delays were 58 and 65 seconds. The average execution time was 25.7 seconds, highlighting the system's overall responsiveness.

A study by Amusa K.A., Olanipekun M.U., & Nuga O.O. (2017) demonstrated average SMS delivery times using a GSM module across three network providers (A, B, C) as 31, 30, and 33 seconds, respectively. Meanwhile, Braga A.M., Neto, R.Z., Vannucci, A.L., & Hiramatsu, R.S. (2015) indicated an average SMS delivery time of approximately 20 seconds. In our study, the average delivery time is 25.7 seconds, which

demonstrates that the system is responsive as it falls within the range established by these studies.

Despite some trials experiencing longer delays of 58 and 65 seconds, the system remains reliable as all SMS messages were successfully delivered, regardless of the time taken. This demonstrates its ability to handle network variability while maintaining an average execution time of 25.7 seconds, which is generally responsive.

Most notifications were sent within 13–16 seconds, reflecting efficient performance under stable conditions. Longer delays highlight the impact of network instability, suggesting that improving network reliability is key to reducing deviations and enhancing overall SMS delivery performance.

Table 6.
Summary of Execution Time for Opening of Locker Kiosk Compartment using Electric Solenoid Lock

Opening of Locker Kiosk Compartment using Electric Solenoid Lock	
Trials	**Execution Time (secs)**
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
Average Execution Time	1

Table 6 shows the summary of execution times for opening the Locker Kiosk Compartment using the Electric Solenoid Lock after ten (10) trials. The system

successfully unlocked the compartments in 1 second for each trial, demonstrating exceptional consistency in performance.

Out of the ten (10) trials, all notifications had an execution time of exactly 1 second, demonstrating the system's efficient and reliable performance. The uniformity of execution time across all trials highlights the solenoid lock's quick operation and suggests that the system operates effectively under standard conditions, ensuring a seamless user experience.

Research Objective 2: To determine the accuracy of the Locker Kiosk System in terms of detecting the presence of the item stored inside the compartment using an IR proximity sensor and displaying the compartment's availability in the kiosk and the mobile application.

To determine the accuracy of the developed Locker Kiosk System to detect the presence of the item to be stored inside the compartment using an IR Proximity sensor, displaying the compartment availability on the kiosk, and reflecting the same information in the mobile application, the researchers conducted ten (10) trials per IR proximity sensor for each compartment, as well as for the kiosk and mobile application's displays. Detecting items using IR sensors and displaying compartment availability in the kiosk and mobile application was taken from ten (10) respondents' participation in performing the Locker Kiosk System process.

Table 7.
Accuracy of IR Proximity Sensors Across Nine (9) Compartments

IR Proximity Sensor's Accuracy					
Compartment No.	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	5	5	0	0
2	T	5	5	0	0
3	T	5	5	0	0
4	T	5	5	0	0
5	T	5	5	0	0
6	T	5	5	0	0
7	T	5	5	0	0
8	T	5	5	0	0
9	T	5	5	0	0
Total	**True**	45	45	0	0
Accuracy			1.0000		

Based on Table 7 the confusion matrix confirms that the IR proximity sensors achieved a flawless accuracy of 1.0000, equivalent to 100%, over 10 trials. Throughout the testing process, the sensors consistently demonstrated their reliability by accurately detecting the presence or absence of items in all compartments without any discrepancies or errors.

The True Positives (TP) indicate that the sensors successfully identified the presence of an item in every trial, while the True Negatives (TN) show the system's perfect ability to recognize when a compartment was unoccupied. Furthermore, no False Negatives (FN) or False Positives (FP) were observed across the 10 trials, meaning the system never failed to detect an item when present and never mistakenly identified an item in an empty compartment. This performance across multiple trials highlights the accuracy, consistency, and dependability of the sensors in ensuring precise detection across all compartments.

Table 8.
Accuracy of the Compartments' Availability in the Kiosk and Mobile Application Across Nine (9) Compartments

Ten (10) Trials Across All Compartments					
		Displayed Status: Unavailable (x)		Absolute Error	
Trials	**Actual Status: Unavailable (x)**	**Kiosk**	**Mobile Application**	**Kiosk**	**Mobile Application**
1	9	9	9	0	0
2	8	8	8	0	0
3	7	7	7	0	0
4	6	6	6	0	0
5	5	5	5	0	0
6	4	4	4	0	0
7	3	3	3	0	0
8	2	2	2	0	0
9	1	1	1	0	0
10	0	0	0	0	0
Total	**45**	**45**	**45**	**0**	**0**

Table 8 presents the aggregated results of ten (10) trials conducted across nine (9) compartments to evaluate the accuracy of the system in displaying the availability of compartments on both the kiosk interface and the mobile application. The analysis was performed using the mean absolute error (MAE) equation, a widely recognized metric for quantifying prediction accuracy by measuring the average magnitude of errors in a dataset.

The findings indicate that the absolute error was consistently recorded as zero (0) across all trials. This absence of deviation between the observed and expected values resulted in a perfect accuracy rate of 100%. Such a result underscores the system's high reliability and precision in accurately reflecting the real-time availability status of

compartments across both platforms, thereby validating its effectiveness in maintaining consistency and accuracy in user interactions and system feedback.

Research Objective 3: To determine the acceptability rate of the mobile application-monitored locker kiosk at Marinduque State University (Boac Campus).

To assess user acceptability in terms of user experience, the researchers utilized the User Experience Questionnaire (UEQ), developed by Dr. Martin Schrepp. The UEQ consists of a 7-point scale for rating various components and includes six primary dimensions: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty, with a total of twenty-six items.

In this study, the researchers specifically focused on individuals who were willing to try the prototype of the mobile application-monitored locker kiosk, which was stationed at the university's guardhouse. Meanwhile, the mobile application, specifically designed for monitoring, is tried and evaluated by the security guards. The target respondents for this assessment were primarily students, faculty members, and delivery riders who use or might potentially use the locker kiosk system for storing their belongings. As well as the security personnel/guards who were tasked with the mobile application to monitor the locker and help users in case of emergency.

The use of the UEQ allowed for a comprehensive evaluation of how these different user groups perceived the system in terms of its overall appeal, ease of use, efficiency, and novelty, among other factors. By focusing on this group, the researchers

aimed to gather valuable feedback on the usability and acceptability of the mobile application-monitored locker kiosk, which can be used for further system enhancements or potential implementation across the campus.

Figure 36. Mean Value of Responses per Item for the Locker Kiosk

Figure 36 presents the range of ratings across different aspects of usability and user experience of the locker kiosk system. In terms of attractiveness, users generally found it more appealing than unappealing, with a rating of two and three hundred sixty-nine thousandths (2.369). It was also considered slightly more attractive than unattractive, scoring two and two hundred sixty-nine thousandths (2.269). Additionally, the kiosk was seen as more pleasant than unpleasant, receiving a rating of two and three hundred ninety-two thousandths (2.392). However, while users leaned toward finding it pleasing, it was not rated as highly enjoyable, as reflected in the scores of two and three hundred fifty-three thousandths (2.353) for annoying versus enjoyable and two and three hundred ninety-two thousandths (2.392) for unlikable versus pleasing. These results suggest that while the kiosk's appearance was generally well-received, there is room to make it feel more inviting and engaging.

When it comes to efficiency, users found the kiosk to be reasonably fast, with a score of two and three hundred fifty-three thousandths (2.353). However, its overall efficiency was rated slightly lower at two and two hundred thousandths (2.200), indicating that while speed was not a major issue, certain processes may have felt inefficient or in need of optimization. Similarly, the organization of the kiosk's interface was rated at two and two hundred thousandths (2.200), suggesting that while it was not entirely disorganized, the layout could be clearer and better structured to help users navigate more easily.

In terms of usability, the kiosk was seen as somewhat understandable but not entirely intuitive, receiving a score of two and one hundred forty-four thousandths (2.144)

for not understandable versus understandable and two and one hundred seven thousandths (2.107) for confusing versus clear. Learning how to use it was not completely effortless, as reflected in ratings of two and two hundred forty-six thousandths (2.246) for difficult to learn versus easy to learn and two and thirty-eight thousandths (2.038) for complicated versus easy. These scores indicate that while users were able to operate the kiosk, improvements in instructions, guidance, or overall design could make the experience smoother and more intuitive.

Regarding dependability and security, users found the kiosk to be somewhat supportive, giving it a score of two and three hundred seventy-six thousandths (2.376). It was also considered moderately secure, with a rating of two and three hundred twenty-three thousandths (2.323). However, it only slightly met expectations, receiving a score of two and three hundred thirty-two thousandths (2.332). This suggests that while the kiosk functioned as expected, it did not surpass expectations, and some users may have anticipated better performance or reliability.

In terms of stimulation and novelty, users found the kiosk to be somewhat engaging, with a score of two and four hundred ninety-two thousandths (2.492) for boring versus exciting and two and four hundred sixty-nine thousandths (2.469) for not interesting versus interesting. While the kiosk was perceived as somewhat modern and inventive, its ratings remained moderate, with scores of two and two hundred sixty-one thousandths (2.261) for conventional versus inventive, two and four hundred twenty-three thousandths (2.423) for conservative versus innovative, and two and two hundred ninety-two thousandths (2.292) for usual versus leading edge. This suggests that while the kiosk

was not seen as outdated, it also was not considered highly creative or groundbreaking. Small improvements to its design and functionality could enhance its sense of innovation and engagement.

In terms of practicality and value, the kiosk was viewed as moderately useful, scoring two and four hundred fifteen thousandths (2.415), and somewhat practical, with a rating of two and two hundred thirty thousandths (2.230). This indicates that while users found it functional, there are opportunities to enhance its overall convenience and usefulness.

With a maximum possible rating of three, the kiosk received generally positive but not outstanding ratings across most categories. It was perceived as pleasant, functional, and somewhat modern, but certain areas—such as usability, efficiency, and engagement—could benefit from improvements. Enhancing clarity, ease of use, and the overall user experience would likely result in greater satisfaction. Optimizing the layout, improving efficiency, and introducing more engaging elements could help create a smoother and more enjoyable experience for users.

Figure 37. Mean Value of Responses per Item for the Mobile Application

Figure 37 presents an evaluation of the user experience for the mobile application across different categories, including Attractiveness, Efficiency, Perspicuity, Dependability, Stimulation, and Novelty, following the UEQ framework. Each category reflects user ratings for opposing attributes (e.g., "annoying/enjoyable"), with higher scores indicating more positive evaluations. The recorded scores range from two (2) to three (3), with higher values suggesting a better user experience. Based on the UEQ benchmark, scores around two point five (2.500) or higher are generally considered positive, while lower scores indicate areas that may need improvement.

The system performed well in multiple areas, particularly in terms of Attractiveness, Efficiency, and Perspicuity. Users found the application engaging, as reflected in a rating of two point eight (2.8) for "boring/exciting." The system was also perceived as innovative, with a perfect score of three (3) for "conventional/inventive." Additionally, it was rated as highly learnable, with "difficult to learn/easy to learn" receiving three (3). These scores suggest that the application creates a positive first impression and is easy to understand and navigate. Users also rated the system favorably in terms of efficiency, with a score of three (3) for "inefficient/efficient," indicating smooth functionality and responsiveness. Emotional responses were generally positive, with ratings of two point eight (2.8) for "bad/good" and three (3) for both "unlikable/pleasing" and "unpleasant/pleasant," suggesting that users found the application visually appealing and pleasant to interact with.

However, the evaluation also highlights areas requiring improvement, particularly in Dependability, Clarity, and Novelty. The lowest rating, two (2) for

"cluttered/organized," indicates that users found the interface somewhat disorganized, which could impact usability. Similarly, the rating of two (2) for "unfriendly/friendly" suggests that users did not perceive the application as particularly welcoming or intuitive in terms of interaction. Dependability aspects, such as "not secure/secure" and "obstructive/supportive," both received scores of two and five hundred thousandths (2), showing that while users found the system somewhat reliable, it does not fully inspire confidence.

In terms of Novelty, there is an inconsistency in user perception. While "conventional/inventive" received a perfect score of three (3.000), "conservative/innovative" scored only two (2.000), suggesting that while the application introduces modern elements, it may not be seen as particularly groundbreaking or forward-thinking. Additionally, "not interesting/interesting" received a score of two (2.000), indicating that the system may not sustain long-term engagement for all users.

According to the UEQ benchmark interpretation, the application scores above average in Attractiveness, Efficiency, and Learnability, making it easy and enjoyable to use. However, the lower ratings in Dependability, Clarity, and Stimulation indicate areas that need refinement. To improve the user experience, efforts should focus on enhancing the organization of the interface, creating a more user-friendly interaction, and reinforcing the system's security and reliability. Increasing engagement elements and refining innovative aspects could also help maintain user interest over time. While the application has several strong points, addressing these concerns would make it a more well-rounded and satisfying experience for users.

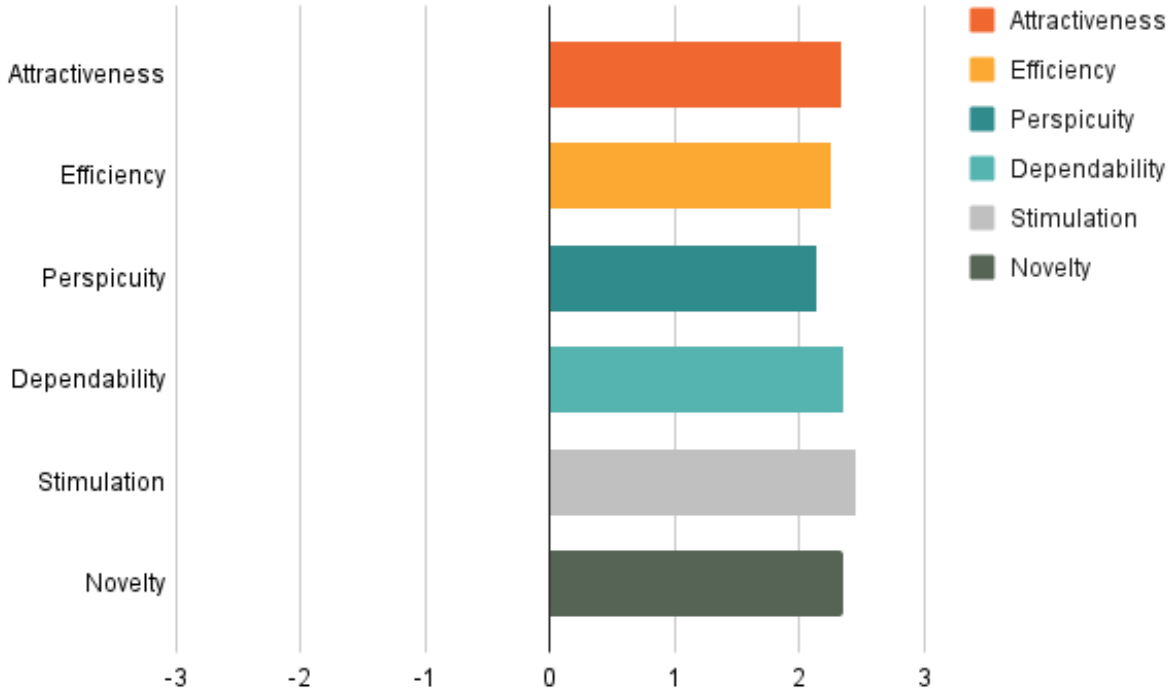


Figure 38. Mean User Response per UEQ Scale for the Locker Kiosk

Figure 38 provides an overview of the user experience for the system based on six key categories: Attractiveness, Efficiency, Perspicuity, Dependability, Stimulation, and Novelty. The scores range from two and one hundred thirty-four thousandths to two and four hundred forty-four thousandths, with three being the highest possible rating. According to the User Experience Questionnaire benchmark interpretation, scores above two point five (2.5) are generally seen as positive, while those below this level indicate areas that may need improvement.

The system performed best in stimulation, receiving a score of two and four hundred forty-four thousandths (2.444). This suggests that users found the system somewhat engaging and enjoyable, though there is still potential to make it more immersive. Dependability also received a fairly strong rating of two and three hundred

fifty-one thousandths (2.351), meaning users viewed the system as somewhat reliable, though not entirely without issues.

Attractiveness, which reflects the overall impression and visual appeal, was rated at two and three hundred forty-three thousandths (2.343). This indicates that while users generally found the system pleasant, it was not particularly striking or visually impressive. Novelty, with a score of two and three hundred forty-four thousandths (2.344), suggests that while the system is perceived as somewhat modern, it does not stand out as highly innovative or unique.

The lowest-rated category was perspicuity, with a score of two and one hundred thirty-four thousandths (2.134). This indicates that some users may have found the system difficult to understand or navigate, highlighting the need for clearer guidance and a more intuitive design. Efficiency was also rated relatively low at two and two hundred forty-five thousandths (2.245), suggesting that some processes might feel slow or require unnecessary effort.

Based on the benchmark analysis, the system falls into the below-average range in most categories, meaning that while it is functional, it does not excel in user experience. The highest ratings in stimulation and dependability suggest that users find the system somewhat enjoyable and reliable. However, the lower ratings in perspicuity and efficiency indicate that usability and performance need refinement.

To improve the user experience, adjustments should focus on making the system clearer and easier to use. Enhancing the layout, simplifying navigation, and providing more intuitive instructions could address perspicuity concerns. Optimizing processes to

ensure a smoother workflow and reducing any unnecessary steps could improve efficiency. Additionally, refining the design to make it more visually appealing and adding engaging features could enhance attractiveness and stimulation. Introducing more innovative elements may also help the system feel more modern and dynamic.

While the system performs adequately, there is significant potential for improvement in usability, efficiency, and engagement. Addressing these areas could lead to a more satisfying and well-rounded user experience.

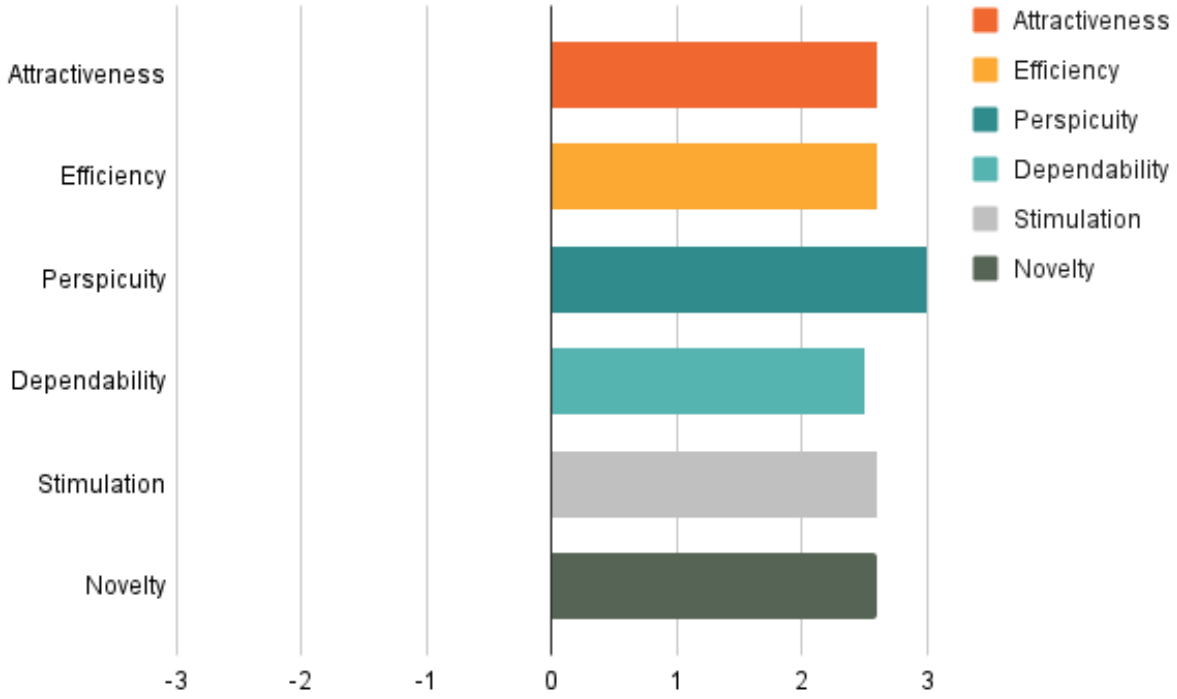


Figure 39. Mean User Response per UEQ Scale for the Mobile Application

Figure 39 presents an overview of participant responses regarding six key dimensions of the User Experience Questionnaire (UEQ) as applied to the Mobile Application: Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty. Each dimension assesses a specific aspect of user interaction and satisfaction.

Attractiveness received a rating of two point six (2.6), indicating that users generally find the application visually appealing and emotionally engaging. This suggests that the interface creates a positive first impression and maintains user interest throughout the interaction. Efficiency was also rated at two point six (2.6), showing that users perceive the application as functional and capable of completing tasks effectively. This score reflects a satisfactory level of usability and performance.

Perspicuity received the highest rating, with a score of three (3), signifying that users consider the application highly intuitive and easy to understand. This indicates that navigation and interaction are seamless, reducing the learning curve and enhancing overall usability. Dependability, with a score of two point five (2.5), suggests that users find the application relatively reliable but recognize areas for improvement, such as stability or security. Addressing these aspects could increase trust and confidence in the system.

Stimulation also received a rating of two point six (2.6),), suggesting that the application provides an engaging and enjoyable experience. This reflects its ability to maintain user motivation and interest. Novelty, similarly rated at two point six (2.6), indicates that users perceive the application as somewhat innovative and unique. However, while this score suggests an appreciation for originality, there is room to introduce more groundbreaking features to further enhance the system's perceived innovation.

Overall, the findings suggest a generally positive user experience, with particularly strong ratings for perspicuity (3), efficiency (2.6), and stimulation (2.6). However, areas such as dependability (2.5) could benefit from enhancements to further build user trust. Additionally, increasing the application's level of innovation may improve its appeal. By refining these aspects while maintaining the system's clarity and ease of use, the overall user experience can be further improved.

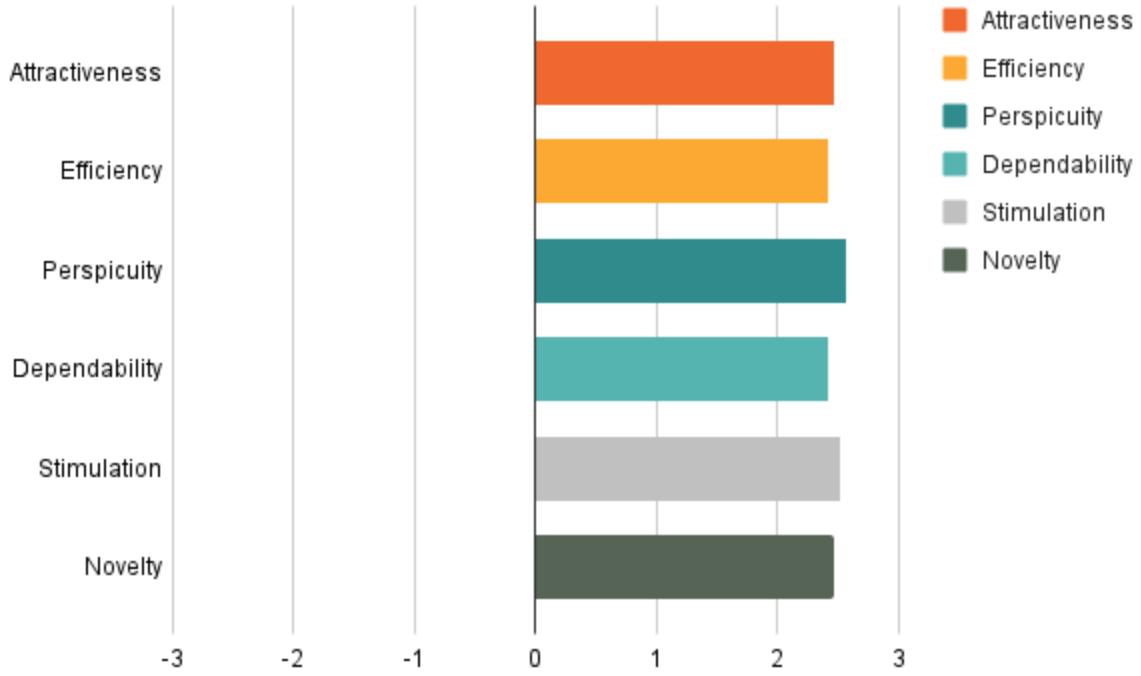


Figure 40. Overall Mean User Response per UEQ Dimension for both the Locker Kiosk and the Mobile Application

Figure 40 presents the overall interpretation of user experience ratings for the Locker Kiosk and the Mobile Application based on the six key dimensions of the User Experience Questionnaire (UEQ): Attractiveness, Perspicuity, Effectiveness, Dependability, Stimulation, and Novelty. Each dimension reflects a specific aspect of usability and user perception. Higher scores indicate a more positive user experience. The

table also includes the average dimension scores and an overall acceptability score, which summarize the general user impressions of both systems.

Attractiveness received a rating of two and three hundred forty-two thousandths (2.342) for the Locker Kiosk and two point six (2.6) for the Mobile Application, with an average score of two and four hundred seventy-one thousandths (2.471). These scores suggest that while both systems are generally viewed as visually appealing, users found the Mobile Application slightly more engaging in terms of design and overall appearance.

Perspicuity, which measures how easy a system is to understand and navigate, was rated two and one hundred thirty-four thousandths (2.134) for the Locker Kiosk and three (3) for the Mobile Application, resulting in an average score of two and five hundred sixty-seven thousandths (2.567). The higher score for the Mobile Application indicates that users find it significantly more intuitive and easier to use compared to the Locker Kiosk.

Effectiveness, reflecting how efficiently users can complete tasks, scored two and two hundred forty-five thousandths (2.245) for the Locker Kiosk and two point six (2.6) for the Mobile Application, with an average score of two and four hundred twenty-three thousandths (2.423). This suggests that while both systems perform their intended functions effectively, the Mobile Application is perceived as slightly more efficient.

Dependability, which assesses reliability and trustworthiness, received ratings of two and three hundred fifty-one thousandths (2.351) for the Locker Kiosk and two point five (2.5) for the Mobile Application, leading to an average score of two and four hundred

twenty-five thousandths (2.425). While both systems are regarded as relatively dependable, there is still room for improvement in ensuring a more stable and reliable experience.

Stimulation, which gauges how engaging and enjoyable a system is, was rated two and four hundred forty-four thousandths (2.444) for the Locker Kiosk and two point six (2.6) for the Mobile Application, resulting in an average score of two and five hundred twenty-two thousandths (2.522). These ratings suggest that both systems provide a reasonably engaging experience, though the Mobile Application appears slightly more stimulating.

Novelty, measuring originality and innovation, received a rating of two and three hundred forty-four thousandths (2.344) for the Locker Kiosk and two point six (2.6) for the Mobile Application, with an average score of two and four hundred seventy-two thousandths (2.472). While both systems offer some level of innovation, there is potential to introduce more distinctive and creative features.

The overall acceptability score across all dimensions is two and four hundred eighty thousandths (2.480), indicating a generally positive user experience for both systems. Although the Mobile Application consistently received higher ratings across multiple dimensions, improvements in dependability and innovative features could further enhance overall satisfaction.

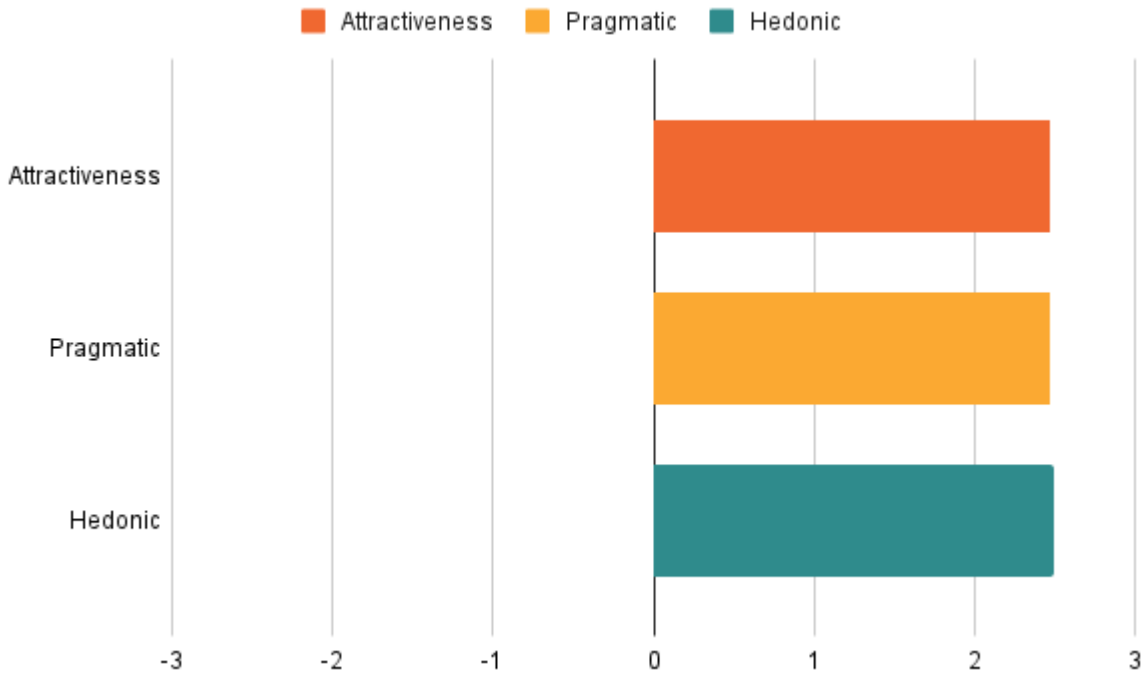


Figure 41. Mean User Response per UEQ Category/Group

Figure 41 presents the mean user response for each category in the User Experience Questionnaire (UEQ), grouping the dimensions into three primary aspects: Attractiveness, Pragmatic Quality, and Hedonic Quality. These categories provide a broader understanding of how users perceive and interact with the system.

Attractiveness, rated at two and four hundred seventy-one thousandths (2.471), is a standalone category because it reflects an overall emotional reaction to the system. It measures users' immediate impression of the design, aesthetics, and appeal. A high score in this category suggests that users find the system visually pleasing and enjoyable to use, which can significantly influence their willingness to engage with it.

Pragmatic Quality, which includes Perspicuity, Effectiveness, and Dependability, also received a mean rating of two and four hundred seventy-one thousandths (2.471).

This category evaluates how practical and functional the system is, focusing on usability, task efficiency, and reliability. A strong score here indicates that users can easily understand, use, and trust the system to perform as expected.

Hedonic Quality, which consists of Stimulation and Novelty, received a slightly higher rating of two and four hundred ninety-seven thousandths (2.497). This category measures how engaging, enjoyable, and innovative the system feels. It reflects the extent to which the system provides an exciting and modern experience that keeps users interested over time. A high hedonic score suggests that users find the system not only useful but also enjoyable and unique.

The breakdown of individual dimensions further highlights these trends. Attractiveness was rated two and three hundred forty-three thousandths (2.343) for the Locker Kiosk and two point six (2.6) for the Mobile Application, averaging two and four hundred seventy-one thousandths (2.471). This confirms that users generally find both systems visually appealing, though the Mobile Application is perceived as slightly more attractive.

Pragmatic Quality is derived from three dimensions: Perspicuity, Effectiveness, and Dependability. Perspicuity, which assesses how easy the system is to learn and navigate, scored two and one hundred thirty-four thousandths (2.134) for the Locker Kiosk and three (3) for the Mobile Application, averaging two and five hundred sixty-seven thousandths (2.567). Effectiveness, measuring how well users can accomplish tasks, was rated two and two hundred forty-five thousandths (2.245) for the Locker Kiosk

and two point six (2.6) for the Mobile Application, resulting in an average of two and four hundred twenty-three thousandths (2.423). Dependability, evaluating system reliability and trustworthiness, received scores of two and three hundred fifty-one thousandths (2.351) for the Locker Kiosk and two point five (2.5) for the Mobile Application, with an overall average of two and four hundred twenty-five thousandths (2.425). These scores indicate that while both systems provide a functional and dependable experience, the Mobile Application is generally regarded as more efficient and intuitive.

Hedonic Quality consists of Stimulation and Novelty, which together determine how engaging and modern the system feels. Stimulation was rated two and four hundred forty-four thousandths (2.444) for the Locker Kiosk and two point six (2.6) for the Mobile Application, with an average of two and five hundred twenty-two thousandths (2.522). Novelty, measuring innovation, was rated two and three hundred forty-four thousandths (2.344) for the Locker Kiosk and two point six (2.6) for the Mobile Application, leading to an average of two and four hundred seventy-two thousandths (2.472). The higher scores in these dimensions suggest that users appreciate the systems' engaging features, though additional innovations could further enhance user interest and excitement.

To conclude, the results indicate a balanced user experience, with the Mobile Application consistently scoring higher across all categories. The strong ratings in Attractiveness and Hedonic Quality highlight the importance of visual appeal and engagement in shaping user satisfaction. Meanwhile, improvements in Pragmatic Quality—particularly in dependability and effectiveness—could enhance the overall usability and trustworthiness of both systems.

5.0 Interpretation of Final Result of Values

This section provides an interpretation of the results obtained from the conducted trials, focusing on the performance of key components in the kiosk system.

Table 9.
Average Execution Time for Sending OTP via SMS Notification Using SIM808 Module

Sending OTP via SMS Notification	
Trials	**Average Execution Time**
10	25.7

As shown in Table 9, the system's average execution time of 25.7 seconds demonstrates responsiveness and reliability in sending OTPs via SMS. While most notifications were delivered promptly within 13–16 seconds, delays of up to 65 seconds occurred due to network instability. These delays highlight the dependence of SMS delivery times on the consistency of network conditions.

Despite the variability, the system's ability to successfully deliver all notifications indicates it is not time-restricted, making it reliable for applications where precise timing is not critical. This ensures the OTP notifications are received even under less favorable network conditions, providing consistent functionality and supporting user satisfaction.

Table 10.
Average Execution Time of Electric Solenoid Lock

Opening of Locker Kiosk Compartment using Electric Solenoid Lock	
Trials	**Average Execution Time (secs)**
10	1

As shown in Table 10, the electric solenoid lock had a consistent 1-second unlocking time which shows the reliability and efficient performance of the electric solenoid lock across all trials. This quick and uniform response time ensures that the system operates smoothly, contributing to a seamless user experience. The system's ability to unlock the compartments almost instantaneously is crucial for maintaining operational efficiency and user satisfaction in the kiosk system.

Table 11.
IR Proximity Sensors Across Nine (9) Compartments

	Item detected	No item detected
There is an item	45 (TP)	0 (FP)
There is no item	0 (FN)	45 (TN)

As shown in Table 11, the IR Proximity sensors exhibited consistent and reliable performance in detecting the presence and absence of items within the compartment throughout 10 trials. The sensors recorded 45 True Positive (TP) and 45 True Negative (TN) results, showing that they correctly identified an item when it was present (True

Positive) and correctly identified the absence of an item when it was not present (True Negative). These results yielded a 100% accuracy rate, which suggests that the IR Proximity sensors functioned with complete precision in detecting the items within the compartment, with no false positives or false negatives recorded. This high level of accuracy underscores the effectiveness of the sensors in their designated task.

Table 12.
Compartments' Availability in the Kiosk and Mobile Application Across Nine (9) Compartments

Ten (10) Trials Across All Compartments					
		Displayed Status: Unavailable (x_i)		Absolute Error	
Trial No.	**Actual Status: Unavailable (x)**	**Kiosk**	**Mobile Application**	**Kiosk**	**Mobile Application**
10	45	45	45	0	0

Table 12 shows that for ten (10) trials, the kiosk and mobile application showed perfect synchronization with the actual status of the system, resulting in zero (0) absolute error. This shows that the data displayed on the mobile application precisely matched the real-time status of the kiosk in all trials. The absence of any discrepancies or errors further highlights the high level of accuracy and reliability in the communication between the kiosk and mobile application, ensuring seamless and correct system operation throughout the testing process.

Table 13.
Overall Mean Value Response for the System's Acceptability and Its Interpretation

Dimension	Locker Kiosk	Mobile Application	Average Dimension Score	Interpretation
Attractiveness	2.343	2.6	2.471	Good
Perspicuity	2.134	3	2.567	Excellent
Effectiveness	2.245	2.6	2.423	Good
Dependability	2.351	2.5	2.425	Good
Stimulation	2.444	2.6	2.522	Excellent
Novelty	2.344	2.6	2.472	Good
Overall Mean Value Response			**2.48**	**Good**

The evaluation of user experience across six dimensions, as shown in Table 13, shows that the Mobile Application generally outperforms the Locker Kiosk. Attractiveness (2.471) and Novelty (2.472) indicate a good design and a moderately innovative system, with the Mobile Application scoring higher in both. Perspicuity (2.567) received the highest rating, highlighting the Mobile Application's ease of use, while the Locker Kiosk scored lower, indicating room for improvement.

Effectiveness (2.423) and Dependability (2.425) were rated as good, suggesting that both systems function well but could benefit from enhancements in reliability and efficiency. Stimulation (2.522) also received a high score, showing that users find the system engaging. The overall mean value response of 2.48 indicates a good user experience, with the Mobile Application performing better across all dimensions. Future improvements should focus on refining the Locker Kiosk's usability, dependability, and engagement features.

CHAPTER V

SUMMARY OF FINDINGS, CONCLUSION, AND RECOMMENDATION

This chapter presents a summary of the gathered and analyzed data, addressing the research questions outlined in the statement of the problem. Based on this summary, the conclusions of the study are drawn and discussed. Furthermore, the chapter provides recommendations for enhancing the study and guiding future research in this area.

1.0 Summary of Findings

The study was conducted to develop and assess the effectiveness of a mobile application-monitored locker kiosk system for managing deliveries and personal items at Marinduque State University (Boac Campus). The following key findings were observed:

- Execution Time:
    - The ten (10) test cases revealed that the average time for sending an OTP via SMS notification using the SIM808 module was 25.7 seconds. While most notifications were delivered within 13 to 16 seconds, some experienced delays, with the longest recorded time reaching 65 seconds because of network instability.
    - The average time for opening the locker kiosk compartment using the Electric Solenoid Lock was 1 second across all trials, demonstrating quick and reliable operation.

- Accuracy of the Locker Kiosk System:
    - IR Proximity Sensors: The IR proximity sensors used to detect items in the compartments performed with flawless accuracy, achieving a perfect 100% accuracy rate across all 10 trials.
    - Compartment Availability in Kiosk and Mobile Application: The system's accuracy in reflecting compartment availability on both the kiosk and mobile application was equally impressive, with no discrepancies recorded in any trial, resulting in a perfect synchronization rate of 100%.

- System Acceptability:
    - The overall acceptability of both the Locker Kiosk and the Mobile Application, based on user feedback, is categorized as "good" according to the UEQ framework by Martin Schrepp. Both systems demonstrate positive user experiences, though there are areas that require improvement to enhance usability and engagement.
        - The Locker Kiosk received feedback from a diverse group of users, including university students, faculty, delivery riders, and external users. The system was rated well in attractiveness, indicating that users find it visually appealing. However, some users, particularly delivery riders, noted that efficiency and dependability could be improved to streamline service and enhance reliability.
        - The Mobile Application, evaluated by security personnel, scored higher in perspicuity. This suggests that users find the system easy to

understand and navigate. However, lower ratings in novelty and stimulation indicate that users may not find it particularly engaging or innovative. Enhancing interactive and dynamic features could help boost user interest and satisfaction.

- Overall, both the Locker Kiosk and the Mobile Application achieved a mean score of two and forty-eight hundredths (2.48), maintaining a "good" level of acceptability. While both systems perform well in their core functions, improvements in efficiency, dependability, and novelty could further elevate user experience and ensure a higher level of satisfaction across all user groups.

2.0 Conclusion

Based on the findings, the mobile application-monitored locker kiosk system proved to be an effective solution for managing deliveries and personal items at Marinduque State University (Boac Campus). The system showed high performance in terms of efficiency, accuracy, and user satisfaction, successfully addressing the challenges identified in the current delivery management system. The following key conclusions were drawn from the study:

- Efficiency: The system demonstrated high efficiency, with reliable SMS notifications and quick unlocking of locker compartments. SMS notifications had an average execution time of 25.7 seconds, while the solenoid lock opened the compartments in 1 second.

- Accuracy:
  - The IR proximity sensors showed flawless performance with 100% accuracy in detecting items in the compartments.
  - The synchronization of compartment availability between the kiosk and mobile application was also 100%, ensuring real-time accuracy.
- High Acceptability: The acceptability of both the Locker Kiosk and the Mobile Application is Highly Acceptable, based on their mean score of 2.48, which falls within the "Good" category according to the UEQ framework. While both systems provide a positive user experience, enhancements in efficiency, dependability, and novelty could further improve usability and engagement, ensuring a more seamless and satisfying interaction for all users.

3.0 Recommendations

For future studies, the researchers recommend exploring possible improvements and advancements by enhancing both the software and hardware of the system to promote its efficiency and sustainability. This includes integrating a payment feature before initiating retrieval transactions, which provides a way for the implementing institution to recover costs and support the long-term maintenance of the locker system.

Another potential improvement involves adding a CCTV camera linked to the system's mobile application, allowing the security guards to monitor locker activity in real-time. Also, the footage could be stored for future reference to ensure that any incidents or concerns can be reviewed if needed. Finally, incorporating an individual lock

or door at the back of each locker compartment, accessible only by security personnel, would allow for emergency access or maintenance while maintaining the security of users' belongings.

Finally, to improve the system's usability, the researchers recommend enhancing the LED indication feature by introducing a dedicated color, such as blue, for compartments containing lost-and-found items. Since an RGB LED is used, this modification would make it easier for users and security personnel to distinguish compartments designated for lost-and-found items, ensuring better organization and functionality.

BIBLIOGRAPHY

12V Solenoid Lock 0.6A / Electric Door Lock. (n.d.). Retrieved from https://quartzcomponents.com/products/12v-solenoid-lock-0-6a-electric-door-lock

7inch capacitive touch screen LCD (H), 1024×600, HDMI, IPS, various devices & systems support. (n.d.). Retrieved from https://www.waveshare.com/7inch-hdmi-lcd-h.htm

Agarwal, T. (2021, August 3). 5V Relay Module : PIN configuration, circuit, working & its applications. Retrieved from https://www.elprocus.com/5v-relay-module/

AJER. (n.d.). Retrieved from https://www.ajer.org/

Ajmera, P. (2021). A Review Paper on Infrared Sensor. International Journal of Engineering Research & Technology (IJERT).

Alexander. (2020, December 28). Absolute Error & Mean Absolute Error (MAE) - Statistics How to. Retrieved from https://www.statisticshowto.com/absolute-error/

Application of GSM for home device control. (n.d.). Retrieved from https://www.researchgate.net/publication/317609888_Application_of_GSM_for_Home_Devices_Control

Aswini, D., & Nandakumar, S. (2019, September). A smart locker. *International Journal of Innovative Technology and Exploring Engineering, 8*(11S).

Balapour, A., Nikkhah, H. R., & Sabherwal, R. (2020). Mobile application security: Role of perceived privacy as the predictor of security perceptions. *International Journal of Information Management*, *52*, 102063. https://doi.org/10.1016/j.ijinfomgt.2019.102063

Bonifacio, I. S., Cale, D. E. V., Ferrer, B. C., Gomera, E. M., Gomez, Capre, L. M. S., Reyes, R. A. D., Tolentino, E. M. (2022). INTERNET-OF-THINGS BASED SMART LOCKER WITH ALERT NOTIFICATION FEATURES AND LOGS MONITORING SYSTEM FOR GRUPPO EMS. Retrieved on April 22, 2024 from https://www.studocu.com/ph/document/trimex-colleges/criminology/final-approved-internet-of-things-based-smart-locker-with-alert-notification-features-and-logs-monitoring-system-for-gruppo-ems-group-1-1/29932954

Bulletin. (2001, July 21). The evolution of school lockers: A look at the past, present, and future. Retrieved from https://thebulletin.net.au/news/32767-the-evolution-of-school-lockers-a-look-at-the-past-present-and-future

Burnell, D. (2019). Computer engineering: Skills for the future workforce. *Journal of Information Technology Education: Innovations in Practice, 18*(2), 1–13.

Circuit Design. (n.d.). Cirkit Designer Tutorials. Retrieved from https://docs.cirkitdesigner.com/component/2bdb7b22-701f-4643-b026-88c2a024fe6c/sim808-gsm-gps-gprs-module

Cox Business (2023). What is a Smart Campus and the Benefits to College Students and Faculty? Retrieved on March 28, 2024, from cox.com/business/enterprise/higher-education.html

Dela Cruz, R., & Reyes, M. (2021). Investment in logistics infrastructure in the Philippines: Challenges and opportunities. *Philippine Journal of Public Administration, 55*(2), 187–212.

Dimalanta, A. (2023, September 28). QUBE introduces PH's first 360° digital smart locker tech for deliveries, and safekeeping. Retrieved from https://manilamillennial.com/2023/09/23/qube-smart-lockers/

Eladmin. (2025, January 22). What is a Smart Locker? A Guide to Smart Locker Technology | eLocker.com. Retrieved from https://www.elocker.com/knowledge-base/what-is-a-smart-locker/

Eusin, J. (2021). *Smart locker kiosk system* (Patent No. KR102257014B1).

Farnell. (n.d.-a). Retrieved from https://www.farnell.com/datasheets/2213335.pdf

Farrukh, Q. (2024, August 16). What is a kiosk? A guide to digital kiosks in 2024. Retrieved from https://www.wavetec.com/blog/what-is-a-kiosk/

Firebase Realtime Database | Store and sync data in real-time. (n.d.). Retrieved from https://firebase.google.com/products/realtime-database

Flutter - Build apps for any screen. (n.d.). Retrieved from https://flutter.dev/

Fonzeloffice, Fonzeloffice, & Fonzeloffice. (2023, September 18). How smart locker systems benefit the education sector. Retrieved on April 20, 2024, from https://www.fonzel.com/blog/smart-lockers/how-smart-locker-systems-benefit-the-education-sector/

Hassan, M. (2024, November 16). Convenience Sampling - Method, types, and examples. Retrieved from https://researchmethod.net/convenience-sampling/

Hurley, M., & Tenny, S. (2023, July 17). Mean. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK546702/

IACLEA. (n.d.). The State of Campus Safety 2021: International Association for Campus Safety and Security. Retrieved on March 28, 2024, from https://www.iaclea.org/

IBM. (2024, January 19). What is a confusion matrix? Retrieved from https://www.ibm.com/topics/confusion-matrix

Implementation issues in the construction of an application framework for secure SMS messages on Android smartphones. (n.d.). Retrieved from https://www.researchgate.net/publication/281204059_Implementation_Issues_in_the_Construction_of_an_Application_Framework_for_Secure_SMS_Messages_on_Android_Smartphones

Industries, A. (n.d.). SIM808 GSM + GPRS + GPS Cellular Module. Retrieved from https://www.adafruit.com/product/2637

Jebb, A. T., Ng, V., & Tay, L. (2021). A review of key Likert scale development advances: 1995–2019. *Frontiers in Psychology*, *12*. https://doi.org/10.3389/fpsyg.2021.637547

KIOSK Information Systems. (2024, January 31). KIOSK: Self-Service Kiosks Manufacturer & supplier. Retrieved from https://kiosk.com/

KIOSK Information Systems. (n.d.). Kiosk LockSpot Brochure. Retrieved from https://kiosk.com/wp-content/uploads/kiosk-lockspot-brochure.pdf

Kranyec, S. (2013). *Self-storage kiosk* (U.S. Patent No. 8509944B1).

Lockertek. (2024, July 3). The history of lockers: from ancient times to the present. Retrieved from https://www.lockertek.co.uk/the-history-of-lockers-from-ancient-times-to-the-present/

Louise, C. (2018, September 20). The proven benefits of school lockers. RCR Education. Retrieved from https://www.rcreducation.com/the-proven-benefits-of-school-lockers/

Ltd, R. P. (n.d.). Raspberry Pi 4 Model B specifications – Raspberry Pi. Raspberry Pi. Retrieved from https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

Mahadik, S. R., Latif, S. S., & Khot, P. A. (2023). IoT-based smart locker system. *International Journal of Research Publication and Reviews*, *4*(5), 2798–2800.

Maharaja, M., Dhanasekar, M., Govintharaj, N., Madhu, M., & Poovarasan, S. (2021). Mini IR radar for unauthorized object detection. *International Research Journal of Engineering and Technology (IRJET)*.

Menaka Devi, T., Kausalya, K., Kaviya, M., Manoranjitha, M., & Monisha. (2020). Arduino-based bank locker security system using RFID and GSM. *International Journal of Innovative Research in Science, Engineering and Technology*, *9*(Special Issue 1), 95–99.

Mzero_Admin. (n.d.). MZero Software – the Plato hotline. http://www.mzerosoftware.com/the-plato-hotline/

Narkhede, A., Mapari, V., & Karande, A. (2021). IOT smart locker. In *Algorithms for intelligent systems* (pp. 1–7). https://doi.org/10.1007/978-981-33-6691-6_1

Niaz Mostakim, Ratna R Sarkar, Md. Anowar Hossain, "Smart Locker: IOT-based Intelligent Locker with Password Protection and Face Detection Approach", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.9, No.3, pp. 1-10, 2019.DOI: 10.5815/ijwmt.2019.03.01

Parthasarathi, R. (n.d.). Performance Metrics – Computer architecture. Retrieved from https://www.cs.umd.edu/~meesh/411/CA-online/chapter/performance-metrics/index.html

Rajagopal, S. M., M, S., & Buyya, R. (2023). FedSDM: Federated learning-based smart decision-making module for ECG data in IoT integrated Edge–Fog–Cloud computing environments. *Internet of Things*, *22*, 100784. https://doi.org/10.1016/j.iot.2023.100784

Ramani, R., Selvaraju, S., Valarmathy, S., & Niranjan, P. (2012). Bank locker security system based on RFID and GSM technology. *International Journal of Computer Applications*, *57*(18), 15–20.

Sebastian, K. A. P., Rivete, K. C., Villegas, J. P. R., Perez, R., Hernandez, J. R., Lope, C. M. (2021). Student Modern Arduino Locker. Retrieved on April 22, 2024, from https://www.scribd.com/document/549379066/local-media7697233756487054568

Shahid, M. (2021, July 2). Solenoid Lock Interfacing with Arduino | Arduino Tutorial. Mechatronic. Retrieved from https://techatronic.com/solenoid-lock-interfacing-with-arduino/

SHENZHEN IFAN DISPLAY TECHNOLOGY CO., LTD. (2025, January 9). 6-inch TFT LCD With Capacitive Touch Panel 720x1440 - IPS Panel. Retrieved from https://ifan-display.com/product/6-inch-tft-lcd-ips-with-capacitive-touch-panel-720x1440/

Sinha, A. (2024, November 25). Solenoids - Definition, Types, circuit, working, Advantages. Retrieved from https://www.electronicsforu.com/technology-trends/learn-electronics/electromagnet-solenoid-basics-working

Software. (n.d.). Arduino. Retrieved from https://www.arduino.cc/en/software

Srinivasan, R., Mettilda, T., Surendhran, D., Gopinath, K., & Sathishkumar, P. (2015). Advanced locker security system. In *Proceedings of the International Conference on Information Engineering, Management and Security* (pp. 12–16).

Ultrasonic ranging module HC - SR04. (n.d.-b). Retrieved from https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

User Experience Questionnaire (UEQ). (n.d.). Retrieved from https://www.ueq-online.org/

Van Huet, J. (2023, December 19). Education Smart Lockers: Learning about the Benefits. Keynius. Retrieved from https://keynius.eu/education-smart-lockers-learning-about-the-benefits/

What is Python? Executive Summary. (n.d.). Python.org. Retrieved from https://www.python.org/doc/essays/blurb/

Yao, D. (2018). Public Locker System Web API. Retrieved on April 22, 2024, from https://scholarworks.calstate.edu/downloads/4q77ft56b

APPENDICES

Appendix A
Project Cost

Table A.
Bills of Materials and Prototype's Circuit

Materials	No. of Unit	Unit	Unit Price	Total Price
8 GB Raspberry Pi 4 Model B	1	pc	6,800.00	6,800.00
Arduino Mega R3	1	pc	1,050.00	1,050.00
LCD 7" for Raspberry Pi	1	pc	2,914.00	2,914.00
GSM SIM808 Module	1	pc	745.00	745.00
12V Power Supply	1	pc	225.00	225.00
Sim Card	1	pc	59.00	59.00
64 GB SanDisk SD card	1	pc	363.00	363.00
Male & Female Barrel Jack	5	pairs	119.00	119.00
Relay	9	pc	42.00	378.00
IR Proximity Sensors	9	pcs	25.00	225.00
RGB LED	9	pcs	15.00	135.00
¼W Resistors	18	pcs	2.00	36.00
Solenoid Lock (big)	1	pc	249.00	249.00
Solenoid Lock (small)	8	pcs	130.00	1,040.00
Jumper Wires	6	strip	57.00	342.00
Screw Driver	1	pc	165.00	165.00
Raspi Power Supply	1	pc	574.00	574.00
Raspi Casing	1	pc	119.00	199.00
Plastic Molding (1/4)	1	pc	100.00	100.00
Plastic Molding (5/8)	1	pc	65.00	65.00
PVC molding	2	pcs	60.00	120.00
Plain Sheet (22)	2	pcs	665.00	1,130.00
Tubular (1x1)	7	pcs	318.00	2,226.00
Drill bit (1/8)	9	pcs	65.00	585.00
Blind Rivets	200	pcs	0.50	100.00
Welding Rod	1	pc	65.00	65.00
Cutting Disc	2	pcs	25.00	50.00
Hinges	16	pairs	15.00	240.00
Rivet	116	pcs	1.00	116.00
Drill bit (5/32)	1	pc	100.00	100.00
Labor for the casing ***(including the Additional materials bought for the casing)***	-	-	-	6,600.00
Wheels for the casing (with lock)	3	pcs	150.00	450.00
Wheels for the casing (w/o lock)	1	pc	120.00	120.00
Wires	132	meters	7.00	924.00
Lead	10	meters	15.00	150.00
Soldering Paste	1	pc	38.00	36.00
Stick Glue	5	pcs	8.00	40.00
Electrical tape	1	pc	35.00	35.00
Transportation Fee of Casing	5	times	100.00	500.00
Marker	1	pc	35.00	35.00
Masking Tape	2	pcs	38.00	76.00
Shrinkable Tube	7	strip	15.00	105.00
Tape	1	pc	11.00	11.00
Scissor	1	pc	15.00	15.00
Spring	10	pcs	19.00	190.00
Spray paint	3	pcs	130.00	650.00
Adhesive Tape	1	pc	86.00	86.00
Barrel Bolt	1	pc	60.00	60.00
Sandpaper	3	pcs	20.00	60.00
Sticker Paper	1	pack	80.00	80.00
Hacksaw blade	1	pc	75.00	75.00
9V Adapter	1	pc	225.00	225.00
GSM Sim Card Load	-	-	-	300.00
30x20 cm PCB	1	pc	153.00	153.00
Ferric Chloride Solution	3	pcs	40.00	120.00
JST	3	packs	45.00	225.00
JST Kit	1	pc	126.00	126.00
Extension Wire	1	pc	107.00	107.00
USB HUB	1	pc	88.00	88.00
Padlock	3	pcs	58.00	174.00
Fare and Shipping Fees	-	-	-	1,160.00
TOTAL				**33,491.00**

Appendix B
Data Presentation

Evaluation Sheet 1

Good day!

We are students from BSCpE 4 at Marinduque State University. As part of our research, we would greatly appreciate your feedback regarding the performance of our developed prototype while you were using it. Your feedback is incredibly valuable to us, and it plays a significant role in the completion of our thesis.

The survey will only take a few minutes of your time, and we assure you that all responses will remain strictly confidential.

Thank you so much for taking the time to share your thoughts with us. Your participation means a lot to us, and we truly appreciate your support.

Warm regards,
Riego, Raven Bryle Hexter O.
Mayores, Jelaica Mae J.
Limpiada, Chrisnell Joy C.
Gonzales, Regina G
The Researchers

https://forms.gle/56XVFeapKC5WLyVk6


User Experience Questionnaire (UEQ) for Locker Kiosk at Marinduque State University Using Raspberry Pi and Arduino

UniLOCK: A Mobile Application-Monitored Locker Kiosk System for Marinduque State University using Raspberry Pi and Arduino

User Experience Questionnaire (UEQ) for Locker Kiosk at Marinduque State University Using Raspberry Pi and Arduino

1. The purpose of this questionnaire is to evaluate your experience with the **Locker Kiosk system** designed for Marinduque State University, which uses **Raspberry Pi and Arduino** technologies.
2. Please rate each of the following 26 statements based on your experience with the Locker Kiosk system using the **7-point Likert scale**:
   - **(1) Strongly Disagree**
   - **(2) Disagree**
   - **(3) Slightly Disagree**
   - **(4) Neutral**
   - **(5) Slightly Agree**
   - **(6) Agree**
   - **(7) Strongly Agree**
3. Answer each statement honestly based on your interactions with the Locker Kiosk system.
4. The questionnaire should take about 5-10 minutes to complete.

* Indicates required question

Email *

Your email

Next

Page 1 of 8

Clear form

Attractiveness

The Locker Kiosk system is enjoyable to use. *
Masayang gamitin ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

My overall overall experience with the system is good. *
Ang kabuuang karanasan ko sa paggamit ng sistema ay maganda.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I like the Locker Kiosk system. *
Gusto ko ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system is pleasant to use. *
Kaaya-ayang gamitin ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system is attractive. *
Kaakit-akit ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I feel comfortable using the Locker Kiosk system. *
Kumportable akong gamitin ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Next Page 2 of 8 Clear form

Perspicuity (Ease of Learning and Use)

I think the system is understandable. *
Sa tingin ko, ang sistema ay madaling maintindihan

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I think the Locker Kiosk system is easy to use. *
Sa tingin ko, madaling gamitin ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I quickly understood how to use the Locker Kiosk system. *
Mabilis kong naintindihan kung paano gamitin ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I think the Locker Kiosk system is clear. *
Sa tingin ko, malinaw ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Efficiency

I can complete my tasks quickly with the Locker Kiosk system. *
Mabilis kong natatapos ang aking mga gawain gamit ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I can efficiently perform my tasks with the Locker Kiosk system. *
Epektibo kong nagagawa ang aking mga gawain gamit ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system helps me accomplish my goals in a short time. *
Tinutulungan ako ng Locker Kiosk system na mabilis na maabot ang aking mga layunin.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system is time-saving. *
Nakatitipid sa oras ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Next Page 4 of 8 Clear form

Dependability (Reliability)

The Locker Kiosk system behaves as expected. *
Gumagana ang Locker Kiosk system ayon sa inaasahan.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system is trustworthy. *
Mapagkakatiwalaan ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I feel secure when using the Locker Kiosk system. *
Kampante akong gamitin ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system works reliably. *
Maasahan ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Next Page 5 of 8 Clear form

Stimulation (Motivation)

The Locker Kiosk system provides new ideas. *
Nagbibigay ang Locker Kiosk system ng mga bagong ideya.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system is exciting. *
Nakakapukaw ng interes ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system stimulates my interest. *
Pinupukaw ng Locker Kiosk system ang aking interes.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system is engaging. *
Kapana-panabik gamitin ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Next Page 6 of 8 Clear form

Novelty (Innovation)

The Locker Kiosk system is creative. *
Malikhain ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system has a fresh design. *
Bago ang disenyo ng Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system is unique. *
Natatangi ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The Locker Kiosk system is innovative. *
Makabago ang Locker Kiosk system.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Next Page 7 of 8 Clear form

Evaluation Sheet 2

User Experience Questionnaire (UEQ) for UniLOCK System Mobile Application

Purpose of the Questionnaire:
The purpose of this questionnaire is to evaluate your experience with the **UniLOCK: A Mobile Application-Monitored Locker Kiosk for Marinduque State University Using Raspberry Pi and Arduino**. Your responses will help us understand how users perceive the app and identify areas for improvement.

Instructions for Participants:
Please rate each of the following 26 statements based on your experience with the Locker Kiosk system using the **7-point Likert scale**:

(1) Strongly Disagree
(2) Disagree
(3) Slightly Disagree
(4) Neutral
(5) Slightly Agree
(6) Agree
(7) Strongly Agree

Answer each statement honestly based on your interactions with the Locker Kiosk system.
The questionnaire should take about 5-10 minutes to complete.

User Experience Questionnaire (UEQ) for Locker Kiosk

1. Attractiveness

1. The UniLOCK app is enjoyable to use.
   Ang UniLOCK app ay nakakatuwang gamitin.
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

2. My overall overall experience with the system is good.
   Ang kabuuang karanasan ko sa paggamit ng sistema ay maganda.
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

3. I like this product (UniLOCK app).
   Gusto ko ang produktong ito (UniLOCK app).
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

4. The UniLOCK app is pleasant to use.
   Ang UniLOCK app ay kaaya-ayang gamitin.
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

5. The UniLOCK app is attractive.
   Ang UniLOCK app ay kaakit-akit.
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

6. I feel comfortable using the UniLOCK app.
   Komportable ako sa paggamit ng UniLOCK app.
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

---

2. Perspicuity (Ease of Learning and Use)

7. I think the system is understandable.
   *Sa tingin ko, a*ng sistema ay madaling maintindihan
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

8. I think the UniLOCK app is easy to use.
   Sa tingin ko, ang UniLOCK app ay madaling gamitin.
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

9. I quickly understood how to use the UniLOCK app.
   Madali kong naintindihan kung paano gamitin ang UniLOCK app.
   (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

10. I think the UniLOCK app is clear.
    Sa tingin ko, ang UniLOCK app ay malinaw.
    (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

---

3. Efficiency

11. I can complete my tasks quickly with the UniLOCK app.
    Madali kong natatapos ang aking mga gawain gamit ang UniLOCK app.
    (1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

12. I can efficiently perform my tasks with the UniLOCK app.
    Mabisang nagagawa ko ang aking mga gawain gamit ang

UniLOCK app.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

13. The UniLOCK app helps me accomplish my goals in a short time.
Tinutulungan ako ng UniLOCK app na makamit ang aking mga layunin sa maikling panahon.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

14. The UniLOCK app is time-saving.
Ang UniLOCK app ay nakakatipid ng oras.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

---

4. Dependability (Reliability)

15. The UniLOCK app behaves as expected.
Ang UniLOCK app ay gumagana ayon sa inaasahan.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

16. The UniLOCK app is trustworthy.
Ang UniLOCK app ay mapagkakatiwalaan.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

17. I feel secure when using the UniLOCK app.
Pakiramdam ko ay ligtas ako kapag ginagamit ko ang UniLOCK app.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

18. The UniLOCK app works reliably.
Ang UniLOCK app ay maaasahang gumagana.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

---

5. Stimulation (Motivation)

19. The UniLOCK app provides new ideas.
Ang UniLOCK app ay nagbibigay ng mga bagong ideya.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

20. The UniLOCK app is exciting.
Ang UniLOCK app ay nakakatuwa.

(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

21. The UniLOCK app stimulates my interest.
Ang UniLOCK app ay nakakapukaw ng aking interes.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

22. The UniLOCK app is engaging.
Ang UniLOCK app ay nakaka-engganyo.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

---

6. Novelty (Innovation)

23. The UniLOCK app is creative.
Ang UniLOCK app ay malikhain.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

24. The UniLOCK app has a fresh design.
Ang UniLOCK app ay may sariwang disenyo.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

25. The UniLOCK app is unique.
Ang UniLOCK app ay kakaiba.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

26. The UniLOCK app is innovative.
Ang UniLOCK app ay makabago.
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

Rating Scale:
For each question, please rate your agreement with the statement using the following scale:
(1) Strongly Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ (7) Strongly Agree

Additional Comments:
Please provide any additional feedback or suggestions you have regarding the **UniLOCK mobile app**:

Tabulated Data and Result for Average Execution Time of the Locker Kiosk System in terms of Sending an OTP to the Item-retriever after a Drop-off Transaction and Opening of Locker Kiosk Compartment using Electric Solenoid Lock

Table B.
Tabulated Data and Result of Execution Time for Sending OTP via SMS Notification Using SIM808 Module After Drop-Off Transaction

Sending OTP via SMS Notification	
Trials	**Execution Time (secs)**
1	25
2	13
3	65
4	14
5	58
6	14
7	16
8	23
9	13
10	16

Table C.
Tabulated Data and Result of Execution Time for Opening of Locker Kiosk Compartment using Electric Solenoid Lock

Opening of Locker Kiosk Compartment using Electric Solenoid Lock	
Trials	**Execution Time (secs)**
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

Tabulated Data and Result for accuracy of the Locker Kiosk System in terms of detecting the presence of the item to be stored inside the compartment using IR proximity sensor, displaying compartment's availability in the kiosk, and displaying compartment's availability in the mobile application

Table D.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 1

IR Proximity Sensor's Accuracy in Compartment 1					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table E.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 2

IR Proximity Sensor's Accuracy in Compartment 2					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table F.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 3

IR Proximity Sensor's Accuracy in Compartment 3					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table G.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 4

IR Proximity Sensor's Accuracy in Compartment 4					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table H.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 5

IR Proximity Sensor's Accuracy in Compartment 5					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table I.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 6

IR Proximity Sensor's Accuracy in Compartment 6					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table J.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 7

IR Proximity Sensor's Accuracy in Compartment 7					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table K.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 8

IR Proximity Sensor's Accuracy in Compartment 8					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table L.
Tabulation of the IR Proximity Sensor's Accuracy in Compartment 9

IR Proximity Sensor's Accuracy in Compartment 9					
Trials	**True/False**	**TP**	**TN**	**FP**	**FN**
1	T	1	0	0	0
2	T	1	0	0	0
3	T	1	0	0	0
4	T	1	0	0	0
5	T	1	0	0	0
6	T	0	1	0	0
7	T	0	1	0	0
8	T	0	1	0	0
9	T	0	1	0	0
10	T	0	1	0	0
Total	**True**	5	5	0	0

Table M.
Trial 1: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 1					
		Displayed Status (x_i)		**Absolute Error**	
Compartment No.	**Actual Status (x)**	**Kiosk**	**Mobile Application**	**Kiosk**	**Mobile Application**
1	1	1	1	0	0
2	1	1	1	0	0
3	1	1	1	0	0
4	1	1	1	0	0
5	1	1	1	0	0
6	1	1	1	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
Total	**9**	**9**	**9**	**0**	**0**

Table N.
Trial 2: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 2					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	1	1	1	0	0
3	1	1	1	0	0
4	1	1	1	0	0
5	1	1	1	0	0
6	1	1	1	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
Total	8	8	8	0	0

Table O.
Trial 3: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 3					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	1	0	0
4	1	1	1	0	0
5	1	1	1	0	0
6	1	1	1	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
Total	7	7	7	0	0

Table P.
Trial 4: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 4					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	1	1	1	0	0
5	1	1	1	0	0
6	1	1	1	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
Total	6	6	6	0	0

Table Q.
Trial 5: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 5					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	1	1	1	0	0
6	1	1	1	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
Total	5	5	5	0	0

Table R.
Trial 6: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 6					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	1	1	1	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
Total	4	4	4	0	0

Table S.
Trial 7: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 7					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
Total	3	3	3	0	0

Table T.
Trial 8: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 8					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	1	1	1	0	0
9	1	1	1	0	0
Total	**2**	**2**	**2**	**0**	**0**

Table U.
Trial 9: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 9					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	1	1	1	0	0
Total	**1**	**1**	**1**	**0**	**0**

Table V.
Trial 10: Accuracy of the Compartments' Availability in the Kiosk and Mobile Application

Trial 10					
		Displayed Status (x_i)		Absolute Error	
Compartment No.	Actual Status (x)	Kiosk	Mobile Application	Kiosk	Mobile Application
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
Total	**0**	**0**	**0**	**0**	**0**

Table W.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Attractiveness Dimension (Locker Kiosk)

Likert Scale Values	UEQ Scale Values	Frequency of Responses						Weighted Score					
		Q1	Q2	Q3	Q4	Q5	Q6	Q1	Q2	Q3	Q4	Q5	Q6
1	**-3**	0	0	0	0	0	0	0	0	0	0	0	0
2	**-2**	0	0	0	2	0	0	0	0	-4	0	0	0
3	**-1**	2	3	2	1	3	2	0	-2	-3	-2	-1	-2
4	**0**	8	2	6	5	4	7	0	0	0	0	0	0
5	**1**	9	8	6	6	9	10	9	8	6	6	9	10
6	**2**	34	48	41	38	53	44	68	96	82	76	106	88
7	**3**	77	69	75	78	61	67	231	207	225	234	183	201
Total		**130**	**130**	**130**	**130**	**130**	**130**	**306**	**308**	**311**	**311**	**295**	**297**

Table X.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Attractiveness Dimension (Mobile Application)

Likert Scale Values	UEQ Scale Values	Frequency of Responses						Weighted Score					
		Q1	Q2	Q3	Q4	Q5	Q6	Q1	Q2	Q3	Q4	Q5	Q6
1	**-3**	0	0	0	0	0	0	0	0	0	0	0	0
2	**-2**	0	0	0	0	0	0	0	0	0	0	0	0
3	**-1**	0	0	0	0	0	0	0	0	0	0	0	0
4	**0**	0	0	0	0	0	0	0	0	0	0	0	0
5	**1**	0	0	0	0	0	0	0	0	0	0	0	0
6	**2**	5	1	0	0	0	6	10	2	0	0	0	12
7	**3**	1	5	6	6	6	0	3	15	18	18	18	0
Total		**6**	**6**	**6**	**6**	**6**	**6**	**13**	**17**	**18**	**18**	**18**	**12**

Table Y.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Perspicuity Dimension (Locker Kiosk)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q7	Q8	Q9	Q10	Q7	Q8	Q9	Q10
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	1	1	0	1	-2	-2	0	-2
3	**-1**	2	2	3	2	-2	-2	-3	-2
4	**0**	8	6	3	5	0	0	0	0
5	**1**	8	6	17	16	8	6	17	16
6	**2**	58	55	49	56	116	110	98	112
7	**3**	53	60	51	50	159	180	153	150
Total		**130**	**130**	**130**	**130**	**279**	**292**	**265**	**274**

Table Z.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Perspicuity Dimension (Mobile Application)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q7	Q8	Q9	Q10	Q7	Q8	Q9	Q10
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	0	0	0	0	0	0	0	0
3	**-1**	0	0	0	0	0	0	0	0
4	**0**	0	0	0	0	0	0	0	0
5	**1**	0	0	0	0	0	0	0	0
6	**2**	0	0	0	0	0	0	0	0
7	**3**	6	6	6	6	18	18	18	18
Total		**6**	**6**	**6**	**6**	**18**	**18**	**18**	**18**

Table AA.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Efficiency Dimension (Locker Kiosk)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q11	Q12	Q13	Q14	Q11	Q12	Q13	Q14
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	0	0	1	2	0	0	-2	-4
3	**-1**	2	3	3	2	-2	-3	-3	-2
4	**0**	5	7	3	3	0	0	0	0
5	**1**	11	15	12	11	11	15	12	11
6	**2**	39	41	50	58	78	82	100	116
7	**3**	73	64	61	55	219	192	183	165
Total		**130**	**130**	**130**	**130**	**306**	**286**	**290**	**286**

Table AB.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Efficiency Dimension (Mobile Application)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q11	Q12	Q13	Q14	Q11	Q12	Q13	Q14
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	0	0	0	0	0	0	0	0
3	**-1**	0	0	0	0	0	0	0	0
4	**0**	0	0	0	0	0	0	0	0
5	**1**	0	0	0	0	0	0	0	0
6	**2**	2	0	1	6	4	0	2	12
7	**3**	4	6	5	0	12	18	15	0
Total		**6**	**6**	**6**	**6**	**16**	**18**	**17**	**12**

Table AC.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Dependability Dimension (Locker Kiosk)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q15	Q16	Q17	Q18	Q15	Q16	Q17	Q18
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	0	0	0	0	0	0	0	0
3	**-1**	2	2	2	2	-2	-2	-2	-2
4	**0**	3	4	3	3	0	0	0	0
5	**1**	8	5	12	12	8	5	12	12
6	**2**	48	51	44	46	96	102	88	92
7	**3**	69	68	69	67	207	204	207	201
Total		**130**	**130**	**130**	**130**	**309**	**309**	**302**	**303**

Table AD.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Dependability Dimension (Mobile Application)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q15	Q16	Q17	Q18	Q15	Q16	Q17	Q18
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	0	0	0	0	0	0	0	0
3	**-1**	0	0	0	0	0	0	0	0
4	**0**	0	0	0	0	0	0	0	0
5	**1**	0	0	0	0	0	0	0	0
6	**2**	1	3	3	2	2	6	6	4
7	**3**	5	3	3	4	15	9	9	12
Total		**6**	**6**	**6**	**6**	**17**	**15**	**15**	**16**

Table AE.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Stimulation Dimension (Locker Kiosk)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q19	Q20	Q21	Q22	Q19	Q20	Q21	Q22
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	0	0	1	0	0	0	-2	0
3	**-1**	2	4	3	2	-2	-2	-2	-2
4	**0**	3	1	0	5	0	0	0	0
5	**1**	6	7	5	8	6	7	5	8
6	**2**	47	33	42	39	94	66	84	78
7	**3**	72	85	79	76	216	255	237	228
Total		**130**	**130**	**130**	**130**	**314**	**324**	**321**	**312**

Table AF.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Stimulation Dimension (Mobile Application)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q19	Q20	Q21	Q22	Q19	Q20	Q21	Q22
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	0	0	0	0	0	0	0	0
3	**-1**	0	0	0	0	0	0	0	0
4	**0**	0	0	0	0	0	0	0	0
5	**1**	0	0	0	0	0	0	0	0
6	**2**	2	1	6	0	4	2	12	0
7	**3**	4	5	0	6	12	15	0	18
Total		**6**	**6**	**6**	**6**	**16**	**17**	**12**	**18**

Table AG.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Novelty Dimension (Locker Kiosk)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q23	Q24	Q25	Q26	Q23	Q24	Q25	Q26
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	2	0	0	3	-4	0	0	-6
3	**-1**	1	3	2	0	-1	-3	-2	0
4	**0**	2	3	6	1	0	0	0	0
5	**1**	6	12	8	5	6	12	8	5
6	**2**	46	51	50	47	92	102	100	94
7	**3**	73	61	64	74	219	183	192	222
Total		**130**	**130**	**130**	**130**	**312**	**294**	**298**	**315**

Table AH.
Tabulated Data for the Frequency of Responses and its Corresponding Weighted Score for Novelty Dimension (Mobile Application)

Likert Scale Values	UEQ Scale Values	Frequency of Responses				Weighted Score			
		Q23	Q24	Q25	Q26	Q23	Q24	Q25	Q26
1	**-3**	0	0	0	0	0	0	0	0
2	**-2**	0	0	0	0	0	0	0	0
3	**-1**	0	0	0	0	0	0	0	0
4	**0**	0	0	0	0	0	0	0	0
5	**1**	0	0	0	0	0	0	0	0
6	**2**	2	0	1	6	4	0	2	12
7	**3**	4	6	5	0	12	18	15	0
Total		**6**	**6**	**6**	**6**	**16**	**18**	**17**	**12**

Table AI.
Tabulation of the Dimension Score for Attractiveness (Locker Kiosk)

Item	Total Weighted Score	Mean Score
Q1	306	2.353
Q2	308	2.369
Q3	311	2.392
Q4	311	2.392
Q5	295	2.269
Q6	297	2.284
	Dimension Score	**2.343**

Table AJ.
Tabulation of the Dimension Score for Attractiveness (Mobile Application)

Item	Total Weighted Score	Mean Score
Q1	13	2.2
Q2	17	2.8
Q3	18	3
Q4	18	3
Q5	18	3
Q6	12	2
	Dimension Score	**2.4**

Table AK.
Tabulation of the Dimension Score for Perspicuity (Locker Kiosk)

Item	Total Weighted Score	Mean Score
Q7	279	2.146
Q8	292	2.246
Q9	265	2.038
Q10	274	2.107
	Dimension Score	**2.134**

Table AL.
Tabulation of the Dimension Score for Perspicuity (Mobile Application)

Item	Total Weighted Score	Mean Score
Q7	18	3
Q8	18	3
Q9	18	3
Q10	18	3
	Dimension Score	**2.6**

Table AM.
Tabulation of the Dimension Score for Effectiveness (Locker Kiosk)

Item	Total Weighted Score	Mean Score
Q11	306	2.353
Q12	286	2.2
Q13	290	2.23
Q14	286	2.2
	Dimension Score	**2.245**

Table AN.
Tabulation of the Dimension Score for Effectiveness (Mobile Application)

Item	Total Weighted Score	Mean Score
Q11	16	2.6
Q12	18	3
Q13	17	2.8
Q14	12	2
	Dimension Score	**2.6**

Table AO.
Tabulation of the Dimension Score for Dependability (Locker Kiosk)

Item	Total Weighted Score	Mean Score
Q15	309	2.376
Q16	309	2.376
Q17	302	2.323
Q18	303	2.332
	Dimension Score	**2.351**

Table AP.
Tabulation of the Dimension Score for Dependability (Mobile Application)

Item	Total Weighted Score	Mean Score
Q15	17	2.8
Q16	15	2.5
Q17	15	2.5
Q18	16	2.6
	Dimension Score	**2.5**

Table AQ.
Tabulation of the Dimension Score for Stimulation (Locker Kiosk)

Item	Total Weighted Score	Mean Score
Q19	314	2.415
Q20	324	2.492
Q21	321	2.469
Q22	312	2.4
	Dimension Score	**2.444**

Table AR.
Tabulation of the Dimension Score for Stimulation (Mobile Application)

Item	Total Weighted Score	Mean Score
Q19	16	2.7
Q20	17	2.8
Q21	12	2
Q22	18	3
	Dimension Score	**2.6**

Table AS.
Tabulation of the Dimension Score for Novelty (Locker Kiosk)

Item	Total Weighted Score	Mean Score
Q23	**312**	**2.4**
Q24	**294**	**2.261**
Q25	**298**	**2.292**
Q26	**315**	**2.423**
	Dimension Score	**2.344**

Table AT. Tabulation of the Dimension Score for Novelty (Mobile Application)

Item	Total Weighted Score	Mean Score
Q23	**16**	**2.6**
Q24	**18**	**3**
Q25	**17**	**2.8**
Q26	**12**	**2**
	Dimension Score	**2.6**

Table AU. Tabulation of the Dimension Scores for the System's Acceptability

Dimension	Locker Kiosk	Mobile Application	Average Dimension Score
Attractiveness	2.342	2.6	2.471
Perspicuity	2.134	3	2.567
Effectiveness	2.245	2.6	2.423
Dependability	2.351	2.5	2.425
Stimulation	2.444	2.6	2.522
Novelty	2.344	2.6	2.472
Overall Acceptability Score			**2.48**

Table AV. Tabulation of the Mean User Response per UEQ Category/Group

UEQ Category/Group	Locker Kiosk	Mobile Application	Average UEQ Category Score
Attractiveness	2.342	2.6	2.471
Pragmatic	2.243	2.7	2.471
Hedonic	2.394	2.6	2.497

Table AW.
Tabulated Data for the Responses per Individual for Attractiveness, Perspicuity, and Effectiveness (Locker Kiosk)

Respon-dents	ATTRACTIVENESS						PERSPICUITY				EFFECTIVENESS			
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14
1	7	7	7	6	6	6	6	7	7	6	6	6	6	5
2	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	6	7	7	6	7	6	5	7	7	5	5	5	6	5
4	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	7	6	7	7	7	6	7	6	7	7	5	6	6	6
7	7	7	7	7	6	7	7	7	7	7	7	7	7	6
8	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	7	6	7	7	7	7	7	6	7	7	7	7	7	5
10	7	7	7	7	7	7	7	7	7	7	7	7	7	7
11	7	7	7	7	6	7	6	7	7	7	6	6	6	6
12	7	7	7	7	7	7	7	7	7	7	7	7	7	6
13	7	7	7	7	7	7	7	7	7	7	7	7	7	7
14	7	7	7	7	7	7	7	7	7	7	7	7	7	7
15	7	6	6	7	5	5	6	6	5	6	6	5	6	5
16	7	6	7	7	7	7	6	6	6	6	6	7	7	7
17	7	7	7	7	7	7	7	7	7	7	7	7	7	7
18	7	6	6	7	7	7	7	6	6	7	7	7	7	4
19	5	5	6	6	6	6	6	5	6	5	5	7	6	5
20	7	7	7	7	7	7	7	7	7	7	7	7	7	6
21	7	7	7	6	7	6	6	7	7	6	7	6	5	5
22	7	7	7	7	7	7	7	7	7	6	6	6	6	6
23	7	6	6	6	6	7	7	6	7	7	7	7	7	6
24	7	7	7	6	7	7	7	7	7	6	7	7	6	6
25	7	7	7	7	7	7	7	7	7	7	7	7	7	7
26	7	6	7	7	7	6	7	6	7	6	6	6	6	5
27	7	7	7	7	7	7	7	7	6	7	7	6	6	6
28	5	5	5	5	6	6	5	5	5	5	6	6	5	6
29	7	7	7	7	7	7	7	7	7	7	7	6	7	6

30	7	7	7	7	6	7	7	7	7	7	6	7	6	7
31	7	6	7	7	7	7	7	6	7	7	7	7	7	6
32	6	6	7	6	6	7	7	6	7	7	7	6	7	6
33	7	7	7	7	7	7	7	7	7	7	7	7	7	7
34	7	7	7	7	7	7	7	7	7	7	6	6	6	7
35	7	7	7	7	7	7	7	7	7	7	7	7	7	7
36	7	7	7	7	7	7	7	7	7	7	7	7	7	7
37	7	7	7	7	7	7	7	7	7	7	7	7	7	7
38	7	6	6	6	6	6	7	6	6	7	6	7	6	6
39	7	7	7	7	7	7	7	7	7	7	7	7	7	6
40	7	6	6	6	7	6	6	6	6	7	6	7	6	7
41	7	7	7	7	6	7	6	7	7	7	7	7	7	6
42	6	7	7	7	6	6	6	7	7	7	6	6	7	6
43	7	7	7	7	7	6	6	7	7	7	6	7	7	7
44	7	7	7	7	7	7	7	7	7	7	6	7	6	6
45	7	6	7	7	7	6	6	6	7	6	7	7	7	6
46	6	6	6	6	6	7	5	6	7	6	6	6	5	6
47	4	3	4	4	4	4	4	3	4	4	4	4	4	4
48	7	6	7	7	7	7	7	6	7	7	7	7	7	7
49	7	7	7	7	7	7	7	7	7	7	7	6	6	7
50	7	7	7	6	6	7	6	7	7	7	7	6	6	5
51	6	6	6	6	6	7	6	6	6	6	6	6	6	7
52	4	6	6	4	6	5	6	6	6	4	4	6	5	5
53	7	7	7	7	7	7	7	7	7	6	7	6	7	7
54	5	7	6	6	7	6	6	7	5	6	5	6	6	7
55	5	6	5	5	5	6	5	6	6	5	6	6	6	6
56	7	7	7	7	7	7	7	7	7	7	7	7	7	7
57	7	7	7	6	6	7	6	7	6	7	6	6	6	6
58	7	6	7	7	7	7	7	6	7	7	6	7	7	6
59	6	5	6	6	6	7	7	5	6	6	6	7	7	6
60	7	7	7	7	6	7	7	7	7	7	7	7	6	6
61	6	7	6	5	6	5	5	7	6	6	6	6	6	6
62	7	6	7	7	6	6	6	6	7	6	6	6	5	4

63	7	7	7	7	7	7	6	7	7	7	6	6	6	7
64	6	5	6	6	5	6	6	5	6	6	5	5	5	5
65	4	7	6	5	5	5	5	7	7	5	7	6	7	7
66	4	5	6	5	5	5	5	5	6	5	6	6	6	6
67	7	6	7	7	7	7	7	6	7	7	6	5	6	6
68	7	6	7	7	7	6	7	6	7	6	7	6	6	5
69	7	6	7	6	6	6	6	6	7	6	6	6	5	6
70	6	6	7	7	6	7	5	6	5	6	6	6	7	6
71	4	7	4	6	5	7	7	7	7	7	7	7	7	7
72	5	5	5	6	7	4	4	5	5	5	4	4	5	5
73	6	7	6	7	6	6	6	7	6	6	6	5	6	7
74	6	6	7	7	7	5	6	6	5	6	6	6	5	6
75	7	7	7	7	7	7	7	7	7	7	7	7	7	7
76	7	7	6	7	7	7	7	7	7	7	6	6	6	7
77	7	7	7	7	6	7	7	7	7	7	6	7	6	7
78	7	6	6	6	6	6	6	6	7	5	5	5	5	6
79	7	7	7	7	7	7	7	7	6	7	7	7	7	7
80	6	6	6	6	6	6	6	6	6	6	6	6	6	6
81	7	5	5	5	6	4	3	5	7	3	2	2	3	2
82	6	6	4	5	5	5	6	6	6	5	5	4	5	5
83	7	7	7	7	7	5	7	7	6	4	6	6	6	7
84	5	6	6	5	6	6	6	6	7	7	7	7	7	7
85	7	6	7	7	6	7	6	6	7	6	6	6	7	7
86	4	5	4	4	4	5	6	5	4	4	4	4	4	4
87	6	7	5	7	6	4	5	7	7	4	4	4	5	5
88	7	7	7	7	7	7	6	7	7	7	6	7	6	7
89	5	6	7	5	5	6	7	6	6	7	7	7	7	7
90	6	6	6	6	6	6	6	6	5	5	6	6	6	6
91	4	4	6	4	4	5	5	4	4	5	4	6	6	6
92	5	7	6	5	5	5	5	7	5	4	5	5	5	5
93	4	4	4	3	3	4	2	4	4	5	4	4	4	4
94	7	7	7	7	7	7	7	7	7	7	7	7	7	7
95	7	7	7	7	7	7	7	7	7	7	7	7	7	7

96	7	7	7	7	7	7	7	7	7	7	7	7	7	7
97	5	7	7	5	6	7	6	7	6	5	6	6	5	7
98	7	7	7	7	7	7	7	7	7	7	7	7	7	7
99	7	7	7	7	7	7	7	7	7	7	7	7	7	7
100	6	6	6	5	6	6	6	6	5	6	6	6	6	6
101	6	7	7	6	6	6	6	7	7	6	6	7	7	6
102	3	3	3	2	3	3	3	3	3	3	3	3	3	2
103	7	7	7	7	7	7	7	7	7	7	7	7	7	7
104	7	7	4	5	6	4	4	7	7	5	5	5	6	6
105	7	7	5	4	4	4	6	7	7	6	4	4	4	4
106	7	7	6	6	6	6	6	6	6	6	5	5	5	5
107	6	6	6	6	6	6	6	6	6	6	6	6	6	6
108	6	6	6	6	6	6	6	6	6	6	6	6	6	6
109	6	6	6	6	6	6	6	6	6	6	6	6	6	6
110	7	7	7	7	7	7	7	7	7	7	7	7	7	7
111	7	7	7	7	7	7	7	7	7	7	7	7	7	7
112	6	6	6	5	6	6	6	6	5	6	6	6	6	6
113	6	6	6	6	6	6	6	6	6	6	6	6	6	6
114	6	6	6	6	6	6	6	6	6	6	6	6	6	6
115	6	6	6	6	6	6	6	6	6	6	6	6	6	6
116	6	6	6	6	6	6	6	6	6	6	6	6	6	6
117	6	6	6	6	6	6	6	6	6	6	6	6	6	6
118	6	6	6	6	6	6	6	6	6	6	6	6	6	6
119	7	7	7	7	7	7	7	7	7	7	7	7	7	7
120	7	7	7	7	7	7	7	7	7	7	7	7	7	7
121	6	6	6	5	6	6	6	6	5	6	6	6	6	6
122	6	6	6	6	6	6	6	6	6	6	6	6	6	6
123	6	6	6	6	6	6	6	6	6	6	6	6	6	6
124	6	6	6	6	6	6	6	6	6	6	6	6	6	6
125	6	6	6	6	6	6	6	6	6	6	6	6	6	6
126	6	6	6	5	6	6	6	6	5	6	6	6	6	6
127	6	7	7	6	6	6	6	7	7	6	6	7	7	6
128	3	3	3	2	3	3	3	3	3	3	3	3	3	2

129	7	7	7	7	7	7	7	7	7	7	7	7	7	7
130	7	7	7	7	7	7	7	7	7	7	7	7	7	7

Table AX. Tabulated Data for the Responses per Individual for Dependability, Stimulation, and Novelty (Locker Kiosk)

Respon-dents	DEPENDABILITY				STIMULATION				NOVELTY			
	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26
1	7	7	7	7	7	7	7	7	7	7	7	7
2	7	7	7	7	7	7	7	7	7	7	7	7
3	6	7	7	6	6	7	7	7	5	5	7	7
4	7	7	7	7	7	7	7	7	7	7	7	7
5	7	7	7	7	7	7	7	7	6	7	6	6
6	7	7	6	6	5	7	7	7	6	5	4	5
7	7	7	7	7	7	7	7	7	7	7	7	6
8	7	7	7	7	7	7	7	7	7	7	7	7
9	6	7	6	7	7	7	7	7	6	6	6	6
10	7	7	7	7	7	7	7	7	7	7	7	7
11	7	7	7	7	7	7	7	7	7	7	7	7
12	7	7	7	7	6	7	7	7	7	6	6	7
13	7	7	7	7	7	7	7	7	7	7	7	7
14	7	7	7	7	7	7	7	7	7	7	7	7
15	7	6	6	7	7	6	7	6	7	6	7	7
16	6	6	7	6	7	6	6	6	7	7	6	7
17	7	7	7	7	7	7	7	7	7	7	7	7
18	7	7	7	7	6	7	7	7	7	6	5	7
19	5	6	6	6	6	6	6	6	6	5	6	6
20	7	6	7	7	6	7	7	7	6	6	6	5
21	6	6	7	6	7	7	6	6	7	7	7	7
22	7	7	7	7	7	7	7	7	7	7	7	7
23	6	6	7	6	7	7	7	7	7	7	7	7
24	6	6	7	7	6	7	6	7	6	7	6	7
25	7	7	7	7	7	7	7	7	7	7	7	7
26	6	6	6	6	6	7	7	7	6	6	6	6

27	7	6	7	6	7	5	7	6	7	7	6	7
28	5	6	5	6	6	6	6	5	6	6	6	6
29	7	7	6	7	5	7	6	7	6	5	6	6
30	6	7	7	7	6	7	7	7	7	7	7	7
31	6	6	6	6	6	7	6	6	6	4	4	6
32	6	6	7	7	6	7	7	7	6	5	6	6
33	7	7	7	7	7	7	7	7	7	7	7	7
34	7	7	7	7	7	7	7	7	6	7	7	7
35	7	7	7	7	7	7	7	7	7	7	7	7
36	7	7	7	7	7	7	7	7	7	7	7	7
37	7	7	7	7	7	7	7	7	7	7	7	7
38	7	7	6	6	6	6	6	6	6	6	7	6
39	7	7	7	7	7	7	7	7	7	7	7	7
40	7	7	6	6	7	6	7	7	7	6	7	6
41	7	7	7	7	7	7	7	7	7	7	7	7
42	6	7	7	7	6	7	7	7	7	6	7	7
43	6	7	7	7	7	7	7	7	7	6	7	7
44	7	7	6	7	7	7	7	7	6	6	6	6
45	7	7	7	7	7	7	7	7	7	7	7	7
46	7	6	6	7	6	7	6	6	6	5	6	6
47	4	4	4	4	4	4	4	4	4	4	4	4
48	7	7	7	7	6	7	7	7	6	6	6	6
49	7	7	7	7	7	7	7	7	7	7	7	7
50	6	7	7	6	7	7	7	7	7	6	7	7
51	6	6	7	6	6	6	7	6	6	6	6	6
52	6	5	6	5	7	5	5	6	7	7	6	7
53	7	7	7	7	7	7	7	7	7	7	7	7
54	6	6	6	6	6	6	6	6	7	6	6	7
55	6	6	6	5	6	5	6	5	6	6	6	6
56	7	7	7	7	7	7	7	7	7	7	7	7
57	7	6	7	7	7	7	6	7	7	7	7	7
58	7	7	7	7	6	7	7	7	6	6	6	6
59	6	7	6	6	6	6	5	5	7	6	6	7

60	7	7	7	7	7	7	7	7	7	6	6	7
61	6	6	6	7	6	6	7	7	6	6	6	7
62	6	6	6	6	6	7	7	6	5	5	5	5
63	7	7	7	7	7	7	7	7	7	7	7	7
64	6	6	6	6	6	7	7	6	6	5	5	6
65	7	6	6	7	7	7	6	6	6	6	7	7
66	7	6	6	6	7	5	6	6	7	6	5	7
67	7	6	7	7	5	6	5	6	5	6	7	5
68	6	7	6	7	6	7	7	7	6	6	6	6
69	6	6	6	6	7	7	7	7	7	6	6	7
70	7	6	6	5	7	7	4	6	7	7	7	7
71	7	7	7	7	7	7	7	7	7	7	7	7
72	5	5	5	5	5	5	5	5	5	5	5	5
73	6	7	6	6	6	7	6	6	6	6	7	6
74	6	5	5	5	6	6	6	5	7	7	6	6
75	7	7	7	7	7	7	7	7	7	7	7	7
76	6	7	7	7	7	7	7	7	7	7	7	7
77	7	6	6	7	7	7	7	7	7	7	7	6
78	7	7	7	7	7	7	7	7	6	6	6	6
79	7	7	7	7	7	7	7	6	7	6	7	7
80	6	6	6	6	6	6	6	6	6	6	6	6
81	5	6	6	5	6	7	7	7	7	6	5	7
82	7	6	7	6	7	6	6	4	7	7	7	6
83	6	6	4	5	7	7	7	7	7	7	7	7
84	7	7	7	7	7	7	7	7	7	7	7	7
85	7	7	7	7	6	7	7	7	6	6	6	6
86	5	4	4	5	5	3	3	4	5	5	4	5
87	5	6	5	5	7	7	7	7	7	6	4	7
88	6	6	6	6	7	7	7	7	7	7	7	7
89	7	6	6	7	7	6	6	6	7	4	7	7
90	5	6	6	5	6	6	6	5	5	5	6	6
91	6	6	6	6	7	6	6	4	7	7	5	6
92	5	5	5	5	5	5	5	5	6	6	7	7

93	4	4	5	4	4	3	2	4	3	3	4	2
94	7	7	7	7	7	7	7	7	7	7	6	7
95	7	7	7	7	7	7	7	7	7	7	7	7
96	7	7	7	7	7	7	7	7	7	7	7	7
97	7	7	7	7	7	7	6	6	5	6	6	6
98	7	7	7	7	7	7	7	7	7	7	7	7
99	7	7	7	7	7	7	7	7	7	7	7	7
100	6	6	5	6	6	6	6	6	6	6	6	6
101	7	7	7	6	7	7	7	7	7	7	7	7
102	3	3	3	3	3	3	3	3	2	3	3	2
103	7	7	7	7	7	7	7	7	7	7	7	7
104	6	5	5	5	7	7	6	6	7	7	6	7
105	4	4	4	4	4	5	5	5	4	7	7	7
106	7	7	7	6	5	7	7	7	6	5	5	6
107	6	6	6	6	6	6	6	6	6	6	6	6
108	6	6	6	6	6	6	6	6	6	6	6	6
109	6	6	6	6	6	6	6	6	6	6	6	6
110	7	7	7	7	7	7	7	7	7	7	7	7
111	7	7	7	7	7	7	7	7	7	7	7	7
112	6	6	5	6	6	6	6	6	6	6	6	6
113	6	6	6	6	6	6	6	6	6	6	6	6
114	6	6	6	6	6	6	6	6	6	6	6	6
115	6	6	6	6	6	6	6	6	6	6	6	6
116	6	6	6	6	6	6	6	6	6	6	6	6
117	6	6	6	6	6	6	6	6	6	6	6	6
118	6	6	6	6	6	6	6	6	6	6	6	6
119	7	7	7	7	7	7	7	7	7	7	7	7
120	7	7	7	7	7	7	7	7	7	7	7	7
121	6	6	5	6	6	6	6	6	6	6	6	6
122	6	6	6	6	6	6	6	6	6	6	6	6
123	6	6	6	6	6	6	6	6	6	6	6	6
124	6	6	6	6	6	6	6	6	6	6	6	6
125	6	6	6	6	6	6	6	6	6	6	6	6

126	6	6	5	6	6	6	6	6	6	6	6	6
127	7	7	7	6	7	7	7	7	7	7	7	7
128	3	3	3	3	3	3	3	3	2	3	3	2
129	7	7	7	7	7	7	7	7	7	7	7	7
130	7	7	7	7	7	7	7	7	7	7	7	7

Table AY.
Tabulated Data for the Responses per Individual for Attractiveness, Perspicuity, and Effectiveness (Mobile Application)

Respon-dents	**ATTRACTIVENESS**						**PERSPICUITY**				**EFFECTIVENESS**			
	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**
1	6	7	7	7	7	6	7	7	7	7	6	7	7	6
2	6	7	7	7	7	6	7	7	7	7	7	7	7	6
3	6	7	7	7	7	6	7	7	7	5	7	7	6	6
4	7	7	7	7	7	6	7	7	7	7	7	7	7	6
5	6	7	7	7	7	6	7	7	7	7	7	7	7	6
6	6	6	7	7	7	6	7	7	7	7	6	7	7	6

Table AZ.
Tabulated Data for the Responses per Individual for Attractiveness, Perspicuity, and Effectiveness (Mobile Application)

Respon-dents	**DEPENDABILITY**				**STIMULATION**				**NOVELTY**			
	Q15	**Q16**	**Q17**	**Q18**	**Q19**	**Q20**	**Q21**	**Q22**	**Q23**	**Q24**	**Q25**	**Q26**
1	7	7	7	7	7	7	6	7	7	7	7	6
2	7	7	7	7	7	6	6	7	7	7	7	6
3	6	6	6	6	6	7	6	7	7	7	7	6
4	7	7	7	7	7	7	6	7	7	7	7	6
5	7	6	6	7	7	7	6	7	6	7	6	6
6	7	6	6	6	6	7	6	7	6	7	7	6

Appendix C
User Manual and Source Code

UniLOCK: USER MANUAL

I. ABOUT UNILOCK

UniLOCK is a smart locker kiosk system developed by fourth-year Computer Engineering students at Marinduque State University. Powered by Raspberry Pi and Arduino, it offers secure locker for individuals inside and outside the campus, with a mobile app designed specifically for guards to monitor and manage lockers efficiently. This innovation demonstrates how technology can enhance campus security and convenience.

II. KIOSK USER GUIDE

1.Setting Up the Locker Kiosk

- Plug the system into any outlet.
- Turn on the GSM module for SMS notifications.
- Run the kiosk code to display the GUI.
- Follow the on-screen instructions for drop-off, retrieval, and lost & found.
- Note that compartments with a green LED are available, while those with a red LED are unavailable as they contain an item.

2. How to Use the Smart Locker (Kiosk)

- DROP OFF
  - Tap Anywhere on the Screen.
  - Input the necessary details (including the name and contact number of the sender and receiver.
  - Select the size that best suits the item you want to place inside the locker compartment. Recommended compartments will be shown on the screen based on the size of your item.
  - Choose an available compartment for the item and place the item carefully inside the selected compartment, ensuring it is positioned above the sticker inside for easy detection.
  - In the screen, confirm if you have placed the item in order to lock the compartment.
  - A notification will be sent to the receiver of the item.

- RETRIEVE
  - Tap Anywhere on the Screen.
  - Input the necessary details (including the name and contact number of the receiver).
  - Input the verification code in the screen from the SMS notification you received.
  - Select the compartment that indicates where your item is placed.
  - Confirm in the screen if you have retrieved the item to lock the compartment.

II. KIOSK USER GUIDE

2. How to Use the Smart Locker (Kiosk)
- LOST & FOUND
  - Input your name and contact details.
  - If you are going to store a lost item youve found, click drop off. To drop-off an item, refer to the 'DROP-OFF' instructions. (With the absence of an identified receiver, provide what information is being asked.)
  - In hopes that your belonging might be in the locker, refer to the 'RETRIEVE' instructions. (With the absence of the OTP, make sure to provide all the necessary details instead.)

- For the safety of the university, its community, and the locker kiosk system itself, the following items are strictly prohibited from being stored inside the compartments:
  - Perishable Goods
  - Harmful items/weapons
  - Toxic/corrosive materials
  - Wet items

III. SECURITY GUARD MOBILE APP GUIDE

1. Signing Up & Logging In
- Sign up if you don't have an existing account.
- After signing up, log in with your credentials.
- You will be directed to the dashboard.

2. Dashboard Overview
The dashboard displays the following:
- Number of transactions
- Number of available compartments
- Pending transactions

Each of these three sections is inside a dedicated box. Clicking on any box will show further details.

3. Sidebar & Reminder Settings
- Click the three-line menu at the top-right.
- Navigate to Settings.
- Set a reminder time for pending transactions.
- The system will automatically notify the retriever at the scheduled time to remind them to retrieve their item.

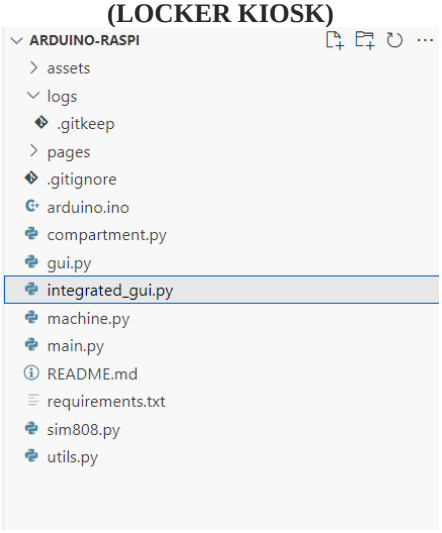
IV. RESEARCHERS & DEVELOPERS

The UniLOCK system was developed by the following researchers from BSCpE-4, Batch 2024-2025:
- Raven Bryle Hexter O. Riego (riegoraven839@gmail.com)
- Jelaica Mae J. Mayores (jelaicamaemayores@gmail.com)
- Chrisnell Joy C. Limpiada (cjlimpiada03@gmail.com)
- Regina G. Gonzales (reginakim@gmail.com)

For inquiries or technical concerns, you may contact the researchers via the emails provided above.

Source Code

(LOCKER KIOSK)



C:\Users\jelai\OneDrive\Desktop\thesis-final\arduino-raspi\integrated_gui.py

```
import tkinter as tk
from PIL import Image, ImageTk
from pages.welcome import WelcomePage
from pages.menu import MenuPage
from pages.verification_code import VerificationCode
from pages.retrieve_general_category import RetrieveGeneralCategoryForm
from pages.retrieval_compartment import RetrievalCompartmentForm
from pages.confirm_retrieval import ConfirmRetrieval
from pages.drop_off_input_details import DropOffForm
from pages.drop_off_general_category import DropoffGeneralCategoryForm
from pages.drop_off_compartment import DropOffCompartmentForm
from pages.drop_off_detection import ProceedDropOff
from pages.drop_off_finished import DropOffFinished
from pages.lostfound_menu import LostFoundMenuPage
from pages.lostfound_input_details import LostFoundForm
from pages.lostfound_dropoff_categories import FoundDropoffForm
from pages.lostfound_retrieve_categories import LostRetrieveForm
from pages.lostfound_dropoff_compartment import LostFoundDropOffCompartment
from pages.lostfound_dropoff_detection import LostFoundDropOffDetection
from pages.lostfound_dropoff_finished import LostFoundDropOffFinished
from pages.lostfound_retrieval_compartment import
LostFoundRetrievalCompartment
from pages.lostfound_confirm_retrieval import LostFoundConfirmRetrieval

from machine import Machine

class Root(tk.Tk):


    # Initialize the GUI
    current_language = 'English'

    def __init__(self, machine: Machine, debug: bool = False):
```

```
        super().__init__()
        self.machine = machine
        self.debug = debug
        self.title('URS')
        self.geometry('800x480')
        self.resizable(False, False)


        # Initialize assets
        self.bg_image =
ImageTk.PhotoImage(Image.open('assets/background.png').resize((800, 480)))
        self.tapscreen_icon = ImageTk.PhotoImage(Image.open('assets/tap-
screen.png').resize((154, 125)))
        self.dropoff_icon =
ImageTk.PhotoImage(Image.open('assets/drop_off_icon.png').resize((90, 50)))
        self.retrieve_icon =
ImageTk.PhotoImage(Image.open('assets/retrieve_icon.png').resize((50, 40)))
        self.retrieve_icon_sm =
ImageTk.PhotoImage(Image.open('assets/retrieve_icon.png').resize((30, 30)))
        self.lost_found_icon =
ImageTk.PhotoImage(Image.open('assets/lost_found_icon.png').resize((50, 50)))
        self.academic_icon = ImageTk.PhotoImage(Image.open('assets/academic-
icon.png').resize((30, 30)))
        self.packages_icon = ImageTk.PhotoImage(Image.open('assets/packages-
icon.png').resize((30, 30)))
        self.personal_belongings_icon =
ImageTk.PhotoImage(Image.open('assets/personal-belongings-
icon.png').resize((30, 30)))
        self.electronic_devices_icon =
ImageTk.PhotoImage(Image.open('assets/devices-icon.png').resize((30, 30)))
        self.others_icon = ImageTk.PhotoImage(Image.open('assets/others-
icon.png').resize((30, 30)))
        self.confirm_retrieve_image =
ImageTk.PhotoImage(Image.open('assets/confirm-retrieve.png').resize((200,
180)))
        self.proceed_drop_off =
ImageTk.PhotoImage(Image.open('assets/drop_off_proceed.png').resize((200,
180)))
        self.finish_drop_off=
ImageTk.PhotoImage(Image.open('assets/drop_off_finished.png').resize((200,
180)))
        self.back_button=
ImageTk.PhotoImage(Image.open('assets/back_arrow.png').resize((30, 30)))
        self.back_button_id=
ImageTk.PhotoImage(Image.open('assets/back_arrow.png').resize((30, 30)))


        for compartment_key in self.machine.compartments.keys():
            self.machine.compartments[compartment_key].turn_off_relay()
            status = self.machine.get_compartment_status(compartment_key)


            if status == "available":
                self.machine.compartments[compartment_key].set_color_green()
```

```
            else:
                self.machine.compartments[compartment_key].set_color_red()

        # Initialize root memmap
        self.initial_memory = {
            'dropoff': {
                'compartment': '',
                'sender': '',
                'sender_contact': '',
                'receiver': '',
                'receiver_contact': '',
                'item': 'Placeholder',
                'item_category': '',
            },
            'retrieve': {
                'compartment': '',
                'otp': ''

            },
            'lost_and_found': {
                'name': '',
                'contact': '',
            },
            'lost_and_found_dropoff': {
                'compartment': '',
                'sender': '',
                'sender_contact': '',
                'item': 'Placeholder',
                'item_category': '',
                'item_subcategory': '',
                'item_detail': '',
            },
            'lost_and_found_retrieve': {
                'compartment': '',
                'receiver': '',
                'receiver_contact': '',
                'item': 'Placeholder',
                'item_category': '',
                'item_subcategory': '',
                'item_detail': '',
            },
        }
        self.memory = self.initial_memory

        self.show_welcome_page()
        self.mainloop()

    def reset_memory(self):
        self.memory = self.initial_memory

    def show_welcome_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = WelcomePage(self)
```

```
        homepage.pack()

    def show_menu_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = MenuPage(self)
        homepage.pack()

    def show_verification_code_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = VerificationCode(self)
        homepage.pack()

    def show_retrieve_general_category_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = RetrieveGeneralCategoryForm(self)
        homepage.pack()

    def show_retrieval_compartment_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = RetrievalCompartmentForm(self)
        homepage.pack()

    def show_confirm_retrieval_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = ConfirmRetrieval(self)
        homepage.pack()

    def show_drop_off_input_details_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = DropOffForm(self)
        homepage.pack()

    def show_drop_off_general_category_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = DropoffGeneralCategoryForm(self)
        homepage.pack()

    def show_drop_off_compartment_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = DropOffCompartmentForm(self)
        homepage.pack()

    def show_drop_off_detection_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = ProceedDropOff(self)
```

```
        homepage.pack()

    def show_drop_off_finished_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = DropOffFinished(self)
        homepage.pack()

    def show_lostfound_menu_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = LostFoundMenuPage(self)
        homepage.pack()

    def show_lostfound_input_details_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = LostFoundForm(self)
        homepage.pack()

    def show_lostfound_dropoff_categories_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = FoundDropoffForm(self)
        homepage.pack()

    def show_lostfound_retrieve_categories_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = LostRetrieveForm(self)
        homepage.pack()

    def show_lostfound_dropoff_compartment_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = LostFoundDropOffCompartment(self)
        homepage.pack()

    def show_lostfound_dropoff_detection_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = LostFoundDropOffDetection(self)
        homepage.pack()

    def show_lostfound_dropoff_finished_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = LostFoundDropOffFinished(self)
        homepage.pack()

    def show_lostfound_retrieval_compartment_page(self):
        for child in self.winfo_children():
            child.destroy()
        homepage = LostFoundRetrievalCompartment(self)
```

```
            homepage.pack()

        def show_lostfound_confirm_retrieval_page(self):
            for child in self.winfo_children():
                child.destroy()
            homepage = LostFoundConfirmRetrieval(self)
            homepage.pack()

    if __name__ == '__main__':
        machine = Machine(port="/dev/ttyACM0", gsm_port="/dev/ttyS0")
        root = Root(machine=machine)
```

C:\Users\jelai\OneDrive\Desktop\thesis-final\arduino-raspi\machine.py

```
import firebase_admin
import json
import random
import serial
import sys
import time
import threading

from datetime import datetime
from firebase_admin import credentials, firestore
from google.cloud.firestore_v1.base_query import FieldFilter
from loguru import logger
from utils import (
    Compartment,
    CompartmentStatus,
    Transaction,
    TransactionStatus,
    dateutil
)
from sim808 import Sim808

log_base_format = '<green>{time}</green> | <level>{level}</level> |
<cyan>{name}</cyan>:<cyan>{line}</cyan> | <level>{message}</level> {extra}\n'

def format_extra(extra):
    extra_str = ''
    if isinstance(extra, dict):
        extra_str = ' '.join(f'{key}={value}' for key, value in extra.items())
    elif isinstance(extra, str):
        extra_str = extra
    return extra_str

def log_format(record):
    record['extra'] = format_extra(record.get('extra', {}))
    return log_base_format.format(**record)

class Machine:
    def __init__(self, port: str = None, gsm_port: str = None, debug: bool =
False) -> None:
```

```
        import compartment
        self.debug = debug
        self.logger = logger.bind()
        logger.remove()
        self.logger.add(
            f'logs/machine.log',
            level="INFO",
            format=log_format
        )
        logger.add(
            sys.stdout,
            format=log_format,
            colorize=True
        )

        self.logger.info('Intializing machine')
        self.available_commands = [0,1,2,3,4,5,
                                   6,7,8,9,10,11,
                                   12,13,14,15,16,17,
                                   18,19,20,21,22,23,
                                   24,25,26,27,28,29,
                                   30,31,32,33,34,35,
                                   36,37,38,39,40,41,
                                   42,43,44,45,46,47,
                                   48,49,50,51,52,53]
        self.arduino = serial.Serial(port, 9600, timeout = 9)
        if port:
            self.arduino.reset_input_buffer()
            self.arduino.reset_output_buffer()
        self.logger.info(f'Arduino initialized', port=port)

        cred = credentials.Certificate('service.json')
        firebase_admin.initialize_app(cred)
        self.database = firestore.client()
        self.settings_reference =
self.database.collection('settings').document('current')
        self.settings_reference.on_snapshot(self._on_settings_change)
        self.logger.info(f'Firestore initialized')

        self.reminder_time = None
        self.last_reminder_date = None
        self.reminder_thread = threading.Thread(target=self.run_reminder)
        self.reminder_thread.start()
        self.logger.info(f'Reminder thread started')


        if not debug:
            self.sim808 = Sim808(gsm_port)
        self.logger.info(f'Sim808 initialized')

        self.compartments: dict[str, compartment.Compartment] = {
            '1': compartment.Compartment(
                machine=self,
```

```
            commands = {
                'turn_off_relay': 0,
                'turn_on_relay': 1,
                'set_color_red': 2,
                'set_color_green': 3,
                'item_detection': 4,
                'turn_off_LED': 5
            }
        ),
        '2': compartment.Compartment(
            machine=self,
            commands = {
                'turn_off_relay': 6,
                'turn_on_relay': 7,
                'set_color_red': 8,
                'set_color_green': 9,
                'item_detection': 10,
                'turn_off_LED': 11
            }
        ),
        '3': compartment.Compartment(
            machine=self,
            commands = {
                'turn_off_relay': 12,
                'turn_on_relay': 13,
                'set_color_red': 14,
                'set_color_green': 15,
                'item_detection': 16,
                'turn_off_LED': 17
            }
        ),
        '4': compartment.Compartment(
            machine=self,
            commands = {
                'turn_off_relay': 18,
                'turn_on_relay': 19,
                'set_color_red': 20,
                'set_color_green': 21,
                'item_detection': 22,
                'turn_off_LED': 23
            }
        ),
        '5': compartment.Compartment(
            machine=self,
            commands = {
                'turn_off_relay': 24,
                'turn_on_relay': 25,
                'set_color_red': 26,
                'set_color_green': 27,
                'item_detection': 28,
                'turn_off_LED': 29
            }
        ),
        '6': compartment.Compartment(
```

```
                machine=self,
                commands = {
                    'turn_off_relay': 30,
                    'turn_on_relay': 31,
                    'set_color_red': 32,
                    'set_color_green': 33,
                    'item_detection': 34,
                    'turn_off_LED': 35
                }
            ),
            '7': compartment.Compartment(
                machine=self,
                commands = {
                    'turn_off_relay': 36,
                    'turn_on_relay': 37,
                    'set_color_red': 38,
                    'set_color_green': 39,
                    'item_detection': 40,
                    'turn_off_LED': 41
                }
            ),
            '8': compartment.Compartment(
                machine=self,
                commands = {
                    'turn_off_relay': 42,
                    'turn_on_relay': 43,
                    'set_color_red': 44,
                    'set_color_green': 45,
                    'item_detection': 46,
                    'turn_off_LED': 47
                }
            ),
            '9': compartment.Compartment(
                machine=self,
                commands = {
                    'turn_off_relay': 48,
                    'turn_on_relay': 49,
                    'set_color_red': 50,
                    'set_color_green': 51,
                    'item_detection': 52,
                    'turn_off_LED': 53
                }
            ),
        }
        self.logger.info(f'Compartments initialized')

    def send_command(self, command: int):
        """
        Send command to arduino.
        Can be used to explicitly invoke Arduino operation without calling
specific functions \n

        Parameters:
        command (int) : Command to send
```

```
        """
        self.logger.debug(f'Send Command to Arduino: {command}')
        if(command in self.available_commands):
            while True:
                self.arduino.write(bytes(str(command)+'\n','utf-8'))
                time.sleep(0.5)
                response = self.get_arduino_response()
                if(response == 'ok'):
                    break
        else:
            raise Exception('Unknown command')

    def get_arduino_response(self) -> str:
        """
        Get arduino serial response

        Returns:
        response (str) : Arduino response
        """
        try:
            response = self.arduino.readline().decode('utf-8').rstrip()
        except UnicodeDecodeError:
            response = self.arduino.readline().decode('utf-8').rstrip()
        self.logger.debug(f'Got response from arduino: {response}')
        return response

    def send_message(self, to: str, msg: str):
        """
        Send a message using twilio service. Phone or msg_service must be
given

        :param to: Phone number to send to
        :param msg: Message body
        """
        if not self.debug:
            self.sim808.send_sms(to, msg)
        self.logger.info(f'Sent message to {to}')
        self.logger.info(f'Message: {msg}')

    def _generate_otp(self) -> str:
        """
        Generate random 4 digit OTP
        """
        otp = random.randint(1000, 9999)
        return str(otp)

    def validate_otp(self, compartment_id: str, otp: str) -> bool:
        """
        Validate if otp is valid for a compartment.
        OTP is validated against compartment's pending transaction

        :param compartment_id: Compartment number (must be a key of
`self.compartments`)
        :param otp: One time password
```

```
        :return: Valid
        """
        transaction_collection = self.database.collection('transactions')

        pending_transaction = transaction_collection \
            .where(filter=FieldFilter('compartment_id', '==', compartment_id))
\
            .where(filter=FieldFilter('status', '==',
TransactionStatus.pending)) \
            .limit(1) \
            .get()

        if not pending_transaction:
            return False

        pending_transaction = pending_transaction[0]
        transaction = Transaction(**pending_transaction.to_dict())

        if otp != transaction.otp:
            return False

        return True

    def get_compartment_status(self, compartment_id: str) -> str:
        """
        Get currnt status of given comparment ID

        :param compartment_id: Compartment number (must be a key of
`self.compartments`)
        :return: Compartment status
        """
        compartment_document =
self.database.collection('compartments').document(compartment_id)
        compartment = compartment_document.get()
        if not compartment.exists:
            self.logger.warning(f'Compartment {compartment_id} does not
exists')
            raise Exception(f'Compartment {compartment_id} does not exists')

        compartment = Compartment(**compartment.to_dict())
        return compartment.status

    def dropoff_item(self, compartment_id: str, details: dict) -> str:
        """
        Emulate a drop-off operation on specific compartment

        :param compartment_id: Compartment number (must be a key of
`self.compartments`)
        :param details: Drop-off detail, see `utils.Transaction` for details
        :return: Transaction otp
        """
        compartment_document =
self.database.collection('compartments').document(compartment_id)
        transaction_collection = self.database.collection('transactions')
```

```
        #transaction_collection = self.database.orderBy("", "asc")

        datetime_now = dateutil.get_datetime_gmt()

        compartment = compartment_document.get()
        if not compartment.exists:
            self.logger.warning(f'Compartment {compartment_id} does not
exists')
            raise Exception(f'Compartment {compartment_id} does not exists')

        # Get compartment status
        compartment = Compartment(**compartment.to_dict())
        if compartment.status != CompartmentStatus.available:
            self.logger.warning(f'Compartment {compartment_id} is not
available', status=compartment.status)
            raise Exception(f'Compartment {compartment_id} is not available')

        # Get pending transaction
        pending_transaction = transaction_collection \
            .where(filter=FieldFilter('compartment_id', '==', compartment_id))
\
            .where(filter=FieldFilter('status', '==',
TransactionStatus.pending)) \
            .limit(1) \
            .get()

        if pending_transaction:
            self.logger.warning(f'Compartment {compartment_id} has pending
transaction')
            raise Exception(f'Compartment {compartment_id} has pending
transaction')

        otp = self._generate_otp()
        transaction = Transaction(
            compartment_id=compartment.id,
            transaction_type="regular",
            status=TransactionStatus.pending,
            otp=otp,
            dropoff_at=datetime_now,
            **details
        )
        _, transaction_ref =
transaction_collection.add(transaction.model_dump())
        self.logger.info(f'Transaction added with id: {transaction_ref.id}')

        compartment.status = CompartmentStatus.unavailable
        compartment.updated_at = datetime_now
        compartment_document.set(compartment.model_dump(), merge=True)
        self.logger.info(f'Compartment updated with id: {compartment_id}')

        # Send message notification to sender and receiver contact
        return otp

    def release_item(self, compartment_id: str, otp: str) -> bool:
```

```
        """
        Emulate a release-off operation on specific compartment

        :param compartment_id: Compartment number (must be a key of
`self.compartments`)
        :param otp: One time password
        :return: Release success
        """
        compartment_document =
self.database.collection('compartments').document(compartment_id)
        transaction_collection = self.database.collection('transactions')
        datetime_now = dateutil.get_datetime_gmt()

        compartment = compartment_document.get()
        if not compartment.exists:
            self.logger.warning(f'Compartment {compartment_id} does not
exists')
            raise Exception(f'Compartment {compartment_id} does not exists')

        # Get compartment status
        compartment = Compartment(**compartment.to_dict())
        if compartment.status != CompartmentStatus.unavailable:
            self.logger.warning(f'Compartment {compartment_id} is not
unavailable', status=compartment.status)
            raise Exception(f'Compartment {compartment_id} is not
unavailable')

        # Get pending transaction
        pending_transaction = transaction_collection \
            .where(filter=FieldFilter('compartment_id', '==', compartment_id))
\
            .where(filter=FieldFilter('status', '==',
TransactionStatus.pending)) \
            .limit(1) \
            .get()

        if not pending_transaction:
            self.logger.warning(f'Compartment {compartment_id} has no pending
transaction')
            raise Exception(f'Compartment {compartment_id} has no pending
transaction')

        pending_transaction = pending_transaction[0]
        transaction = Transaction(**pending_transaction.to_dict())

        if transaction.transaction_type != 'regular':
            self.logger.warning(f'Transaction is not for regular')
            raise Exception(f'Transaction is not for regular')

        if otp != transaction.otp:
            self.logger.warning('OTP does not match')
            raise Exception('OTP does not match')

        transaction.received_at = datetime_now
```

```
        transaction.status = TransactionStatus.completed
        transaction_collection.document(pending_transaction.id).set(transactio
n.model_dump())
        self.logger.info(f'Transaction updated with id:
{pending_transaction.id}')

        compartment.status = CompartmentStatus.available
        compartment.updated_at = datetime_now
        compartment_document.set(compartment.model_dump(), merge=True)
        self.logger.info(f'Compartment updated with id: {compartment_id}')

        return True

    def dropoff_lost_item(self, compartment_id: str, details: dict) -> str:
        """
        Emulate a drop-off lost item operation on specific compartment

        :param compartment_id: Compartment number (must be a key of
`self.compartments`)
        :param details: Drop-off detail, see `utils.Transaction` for details
        """
        compartment_document =
self.database.collection('compartments').document(compartment_id)
        transaction_collection = self.database.collection('transactions')
        #transaction_collection = self.database.orderBy("", "asc")

        datetime_now = dateutil.get_datetime_gmt()

        compartment = compartment_document.get()
        if not compartment.exists:
            self.logger.warning(f'Compartment {compartment_id} does not
exists')
            raise Exception(f'Compartment {compartment_id} does not exists')

        # Get compartment status
        compartment = Compartment(**compartment.to_dict())
        if compartment.status != CompartmentStatus.available:
            self.logger.warning(f'Compartment {compartment_id} is not
available', status=compartment.status)
            raise Exception(f'Compartment {compartment_id} is not available')

        # Get pending transaction
        pending_transaction = transaction_collection \
            .where(filter=FieldFilter('compartment_id', '==', compartment_id))
\
            .where(filter=FieldFilter('status', '==',
TransactionStatus.pending)) \
            .limit(1) \
            .get()

        if pending_transaction:
            self.logger.warning(f'Compartment {compartment_id} has pending
transaction')
```

```
            raise Exception(f'Compartment {compartment_id} has pending
transaction')

        transaction = Transaction(
            compartment_id=compartment.id,
            transaction_type="lost_and_found",
            status=TransactionStatus.pending,
            dropoff_at=datetime_now,
            **details
        )
        _, transaction_ref =
transaction_collection.add(transaction.model_dump())
        self.logger.info(f'Transaction added with id: {transaction_ref.id}')

        compartment.status = CompartmentStatus.unavailable
        compartment.updated_at = datetime_now
        compartment_document.set(compartment.model_dump(), merge=True)
        self.logger.info(f'Compartment updated with id: {compartment_id}')

    def retrieve_found_item(self, compartment_id: str, details: dict) -> bool:
        """
        Emulate a release-off operation on specific compartment

        :param compartment_id: Compartment number (must be a key of
`self.compartments`)
        :param details: Item details, required keys (item_category,
item_subcategory, item_detail, receiver, receiver_contact)
        """
        compartment_document =
self.database.collection('compartments').document(compartment_id)
        transaction_collection = self.database.collection('transactions')
        datetime_now = dateutil.get_datetime_gmt()

        compartment = compartment_document.get()
        if not compartment.exists:
            self.logger.warning(f'Compartment {compartment_id} does not
exists')
            raise Exception(f'Compartment {compartment_id} does not exists')

        # Get compartment status
        compartment = Compartment(**compartment.to_dict())
        if compartment.status != CompartmentStatus.unavailable:
            self.logger.warning(f'Compartment {compartment_id} is not
unavailable', status=compartment.status)
            #raise Exception(f'Compartment {compartment_id} is not
unavailable')

        # Get pending transaction
        pending_transaction = transaction_collection \
            .where(filter=FieldFilter('compartment_id', '==', compartment_id))
\
            .where(filter=FieldFilter('status', '==',
TransactionStatus.pending)) \
            .limit(1) \
```

```
            .get()

        if not pending_transaction:
            self.logger.warning(f'Compartment {compartment_id} has no pending
transaction')
            raise Exception(f'Compartment {compartment_id} has no pending
transaction')

        pending_transaction = pending_transaction[0]
        transaction = Transaction(**pending_transaction.to_dict())

        if transaction.transaction_type != "lost_and_found":
            self.logger.warning(f'Transaction is not for lost and found')
            raise Exception(f'Transaction is not for lost and found')

        required_keys = ['item_category', 'item_subcategory', 'item_detail',
'receiver', 'receiver_contact']
        for key in required_keys:
            if key not in details.keys():
                raise Exception(f'Key {key} is required')

        if details.get('item_category') != transaction.item_category \
            or details.get('item_subcategory') != transaction.item_subcategory
\
            or details.get('item_detail') != transaction.item_detail:
            self.logger.warning(f'Transaction details for lost and found does
not match')
            raise Exception(f'Transaction details for lost and found does not
match')

        transaction.receiver = details.get('receiver')
        transaction.receiver_contact = details.get('receiver_contact')
        transaction.received_at = datetime_now
        transaction.status = TransactionStatus.completed
        transaction_collection.document(pending_transaction.id).set(transactio
n.model_dump())
        self.logger.info(f'Transaction updated with id:
{pending_transaction.id}')

        compartment.status = CompartmentStatus.available
        compartment.updated_at = datetime_now
        compartment_document.set(compartment.model_dump(), merge=True)
        self.logger.info(f'Compartment updated with id: {compartment_id}')

    def _on_settings_change(self, doc_snapshot, changes, read_time):
        '''
        Callback function for detected changes in
        setting coming from Firebase
        '''
        data = doc_snapshot[-1].to_dict()
        self.reminder_time = datetime.strptime(data['reminder_time'], '%I:%M
%p').time()
        self.last_reminder_date = datetime.strptime(data['last_reminder'],
'%Y-%m-%d').date()
```

```
        #self.last_reminder_date = datetime.strptime(data['last_reminder'],
'%Y-%m-%d').date()
        self.logger.info(f'Reminder time has been update to:
{self.reminder_time}')

    def run_reminder(self):
        settings_document =
self.database.collection('settings').document('current')
        transaction_collection = self.database.collection('transactions')
        while True:

            # print(self.reminder_time)sssssssssssssssssss
            # print(self.last_reminder_time)

            if self.reminder_time is None or self.last_reminder_date is None:
                continue

            if datetime.now().time() >= self.reminder_time and \
               datetime.now().date() > self.last_reminder_date:
                self.logger.info('Triggering daily reminder')
                self.last_reminder_date = datetime.now().date()
                settings_document.update({'last_reminder':
datetime.now().strftime('%Y-%m-%d')})

                pending_transactions =
transaction_collection.where(filter=FieldFilter('status', '==', 'pending')) \
                                                        .where(filter=Fie
ldFilter('transaction_type', '==', 'regular')) \
                                                        .get()
                for transaction_data in pending_transactions:
                    transaction = Transaction(**transaction_data.to_dict())
                    message = f'Hello, this is from UniLOCK Team. You have a
pending item on compartment {transaction.compartment_id}'
                    self.send_message(transaction.receiver_contact, message)
                self.logger.info('Daily reminder fired')
```

C:\Users\jelai\OneDrive\Desktop\thesis-final\arduino-raspi\sim808.py

```
import serial
import time
import datetime
import re

class Sim808:
    '''
    Initialize a Sim808 object for communicating with a SIM808 module

    Parameters:
    port (str) : Serial port of SIM808 module
    '''

    def __init__(self, port):
        '''
        Initialize a Sim808 object for communicating with a SIM808 module
```

```
        Parameters:
        port (str) : Serial port of SIM808 module
        '''

        self.sim808 = serial.Serial(port, 115200, timeout=5)
        self.initialize()

    def initialize(self):
        '''
        Check if SIM808 exists and functioning
        '''
        self.sim808.reset_input_buffer()
        self.sim808.reset_output_buffer()
        self.send_command('AT\r\n')
        response = self.read_response()
        time.sleep(5)
        print(response)
        if('OK' not in response):
            raise Exception('Error starting sim808')
        self.send_command('AT+CMGF=1\r\n')
        self.read_response()

    def read_response(self):
        '''
        Get the response from SIM808 Serial COM

        Returns:
        str : SIM808 response
        '''
        response = b''
        while(self.sim808.inWaiting()):
            bit = self.sim808.read()
            response += bit

        # Try decoding with UTF-8, ignoring errors for non-decodable bytes
        try:
            return response.decode('utf-8')
        except UnicodeDecodeError:
            return response.decode('utf-8', errors='ignore')  # Skip invalid
characters

    def send_command(self, command: str, timeout: float = 1):
        '''
        Send a command to SIM808 Serial COM

        Parameters:
        command (str) : Command to send
        timeout (float) : Timeout allows module to receive command in full
        '''
        self.sim808.write(command.encode())
        time.sleep(timeout)

    def send_sms(self, number: str, message: str):
```

```
        '''
        Send a SMS message

        Parameters:
        number (str) : Number to send message to. Should contain country code
        message (str) : Message to send
        '''
        self.send_command('AT+CMGS="' + number + '"\r')
        time.sleep(0.1)
        self.send_command(message + '\x1A\n')
        print(f'Sleeping for {0.1*(len(message)/5)}s')
        time.sleep(0.1*(len(message)/5))
        response = self.read_response()
        while 'OK' not in response:
            time.sleep(0.1)
            response = self.read_response()
        print(response)

    def read_unread_sms(self):
        '''
        Get unread sms

        Returns:
        sms (str): unread sms
        '''
        self.send_command('AT+CMGL=\"REC UNREAD\"\r\n')
        time.sleep(1)
        response = self.read_response()
        return response

    def get_time(self):
        '''
        Get network date and time

        Returns:
        datetime (str) : Network date and time
        '''
        self.send_sms("+639155882825", "CLOCK COMMAND")
        start_time = datetime.datetime.now()
        time.sleep(3)
        messages = self.read_unread_sms()
        end_time = datetime.datetime.now()
        datetime_pattern = r'\d{2}/\d{2}/\d{2},\d{2}:\d{2}:\d{2}\+32'
        match = re.findall(datetime_pattern, messages)
        if match:
            returned_datetime = datetime.datetime.strptime(match[-
1].replace('+32',''),'%y/%m/%d,%H:%M:%S')
            returned_datetime += (end_time - start_time)
        return returned_datetime

    def delete_all_sms(self):
        '''
        Delete all stored sms (inbox and sent)
        '''
```

```
        self.send_command('AT+CMGD=1,4\r\n')
        time.sleep(5)
```

C:\Users\jelai\OneDrive\Desktop\thesis-final\arduino-raspi\compartment.py

```
from machine import Machine

class Compartment:
    def __init__(self, machine: Machine, commands: dict) -> None:
        """
        Initialize a new compartment object

        :param machine: Machine instance
        :param commands: List of commands to use in arduino

        .. code-block:: python
        commands = {
            'turn_off_relay': 0,
            'turn_on_relay': 1,
            'set_color_red': 2,
            'set_color_green': 3,
            'item_detection': 4,
            'turn_off_LED': 5,
        }
        """
        self.machine = machine
        self.turn_off_relay_cmd = commands.pop('turn_off_relay')
        self.turn_on_relay_cmd = commands.pop('turn_on_relay')
        self.set_color_red_cmd = commands.pop('set_color_red')
        self.set_color_green_cmd = commands.pop('set_color_green')
        self.item_detection_cmd = commands.pop('item_detection')
        self.turn_off_LED_cmd = commands.pop('turn_off_LED')

    def turn_off_relay(self):
        """
        Turn off relay
        """
        self.machine.send_command(self.turn_off_relay_cmd)

    def turn_on_relay(self):
        """
        Turn on relay
        """
        self.machine.send_command(self.turn_on_relay_cmd)

    def set_color_red(self):
        """
        Set LED to red
        """
        self.machine.send_command(self.set_color_red_cmd)

    def set_color_green(self):
```

```
        """
        Set LED to green
        """
        self.machine.send_command(self.set_color_green_cmd)

    def item_detection(self):
        """
        Detect item using ir proximity sensor
        """
        self.machine.send_command(self.item_detection_cmd)
        response = self.machine.get_arduino_response()
        while not response:
            response = self.machine.get_arduino_response()
        return response

    def turn_off_LED(self):
        """
        Turn off red and green LED
        """
        self.machine.send_command(self.turn_off_LED_cmd)
```

C:\Users\jelai\OneDrive\Desktop\thesis-final\arduino-raspi\arduino.ino

```
const int relayPin1 = 41;
const int relayPin2 = 25;
const int relayPin3 = 27;
const int relayPin4 = 29;
const int relayPin5 = 39;
const int relayPin6 = 31;
const int relayPin7 = 37;
const int relayPin8 = 35;
const int relayPin9 = 33;

const int irPin1 = 4;
const int irPin2 = 8;
const int irPin3 = 9;
const int irPin4 = 10;
const int irPin5 = 5;
const int irPin6 = 6;
const int irPin7 = 11;
const int irPin8 = 12;
const int irPin9 = 7;

const int redPin1   = 52;
const int greenPin1 = 53;

const int redPin2   = 14;
const int greenPin2 = 15;

const int redPin3   = 16;
const int greenPin3 = 17;
```

```
const int redPin4   = 18;
const int greenPin4 = 19;

const int redPin5   = 48;
const int greenPin5 = 50;

const int redPin6   = 20;
const int greenPin6 = 21;

const int redPin7   = 44;
const int greenPin7 = 46;

const int redPin8   = 22;
const int greenPin8 = 23;

const int redPin9   = 40;
const int greenPin9 = 42;

int currentCommand = -1;

void setup() {
  Serial.begin(9600);
  pinMode(redPin1,   OUTPUT);
  pinMode(greenPin1, OUTPUT);
  pinMode(irPin1, INPUT);
  pinMode(relayPin1, OUTPUT);

  pinMode(redPin2,   OUTPUT);
  pinMode(greenPin2, OUTPUT);
  pinMode(irPin2, INPUT);
  pinMode(relayPin2, OUTPUT);

  pinMode(redPin3,   OUTPUT);
  pinMode(greenPin3, OUTPUT);
  pinMode(irPin3, INPUT);
  pinMode(relayPin3, OUTPUT);

  pinMode(redPin4,   OUTPUT);
  pinMode(greenPin4, OUTPUT);
  pinMode(irPin4, INPUT);
  pinMode(relayPin4, OUTPUT);

  pinMode(redPin5,   OUTPUT);
  pinMode(greenPin5, OUTPUT);
  pinMode(irPin5, INPUT);
  pinMode(relayPin5, OUTPUT);

  pinMode(redPin6,   OUTPUT);
  pinMode(greenPin6, OUTPUT);
  pinMode(irPin6, INPUT);
  pinMode(relayPin6, OUTPUT);

  pinMode(redPin7,   OUTPUT);
  pinMode(greenPin7, OUTPUT);
```

```
  pinMode(irPin7, INPUT);
  pinMode(relayPin7, OUTPUT);

  pinMode(redPin8,   OUTPUT);
  pinMode(greenPin8, OUTPUT);
  pinMode(irPin8, INPUT);
  pinMode(relayPin8, OUTPUT);

  pinMode(redPin9,   OUTPUT);
  pinMode(greenPin9, OUTPUT);
  pinMode(irPin9, INPUT);
  pinMode(relayPin9, OUTPUT);
}

void loop() {
  if(currentCommand == -1) {
    receiveCommand();
  }

  // Compartment 1 Commands
  else if(currentCommand == 0) {
    digitalWrite(relayPin1, HIGH);
    currentCommand = -1;
  }
  else if(currentCommand == 1) {
    digitalWrite(relayPin1, LOW);
    currentCommand = -1;
  }
  else if(currentCommand == 2) {
    setColor1(255, 0, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 3) {
    setColor1(0, 255, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 4) {
    detectItem1();
    currentCommand = -1;
  }
  else if(currentCommand == 5){
    offLED1();
    currentCommand = -1;
  }

  // Compartment 2 commands
  else if(currentCommand == 6) {
    digitalWrite(relayPin2, HIGH);
    currentCommand = -1;
  }
  else if(currentCommand == 7) {
    digitalWrite(relayPin2, LOW);
    currentCommand = -1;
  }
```

```
  else if(currentCommand == 8) {
    setColor2(255, 0, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 9) {
    setColor2(0, 255, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 10) {
    detectItem2();
    currentCommand = -1;
  }
  else if(currentCommand == 11){
    offLED2();
    currentCommand = -1;
  }

  // Compartment 3 commands
  else if(currentCommand == 12) {
    digitalWrite(relayPin3, HIGH);
    currentCommand = -1;
  }
  else if(currentCommand == 13) {
    digitalWrite(relayPin3, LOW);
    currentCommand = -1;
  }
  else if(currentCommand == 14) {
    setColor3(255, 0, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 15) {
    setColor3(0, 255, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 16) {
    detectItem3();
    currentCommand = -1;
  }
  else if(currentCommand == 17){
    offLED3();
    currentCommand = -1;
  }
  // Compartment 4 commands
  else if(currentCommand == 18) {
    digitalWrite(relayPin4, HIGH);
    currentCommand = -1;
  }
  else if(currentCommand == 19) {
    digitalWrite(relayPin4, LOW);
    currentCommand = -1;
  }
  else if(currentCommand == 20) {
    setColor4(255, 0, 0);
    currentCommand = -1;
```

```
}
else if(currentCommand == 21) {
  setColor4(0, 255, 0);
  currentCommand = -1;
}
else if(currentCommand == 22) {
  detectItem4();
  currentCommand = -1;
}
else if(currentCommand == 23){
  offLED4();
  currentCommand = -1;
}

// Compartment 5 commands
else if(currentCommand == 24) {
  digitalWrite(relayPin5, HIGH);
  currentCommand = -1;
}
else if(currentCommand == 25) {
  digitalWrite(relayPin5, LOW);
  currentCommand = -1;
}
else if(currentCommand == 26) {
  setColor5(255, 0, 0);
  currentCommand = -1;
}
else if(currentCommand == 27) {
  setColor5(0, 255, 0);
  currentCommand = -1;
}
else if(currentCommand == 28) {
  detectItem5();
  currentCommand = -1;
}
  else if(currentCommand == 29){
  offLED5();
  currentCommand = -1;
}

// Compartment 6 commands
else if(currentCommand == 30) {
  digitalWrite(relayPin6, HIGH);
  currentCommand = -1;
}
else if(currentCommand == 31) {
  digitalWrite(relayPin6, LOW);
  currentCommand = -1;
}
else if(currentCommand == 32) {
  setColor6(255, 0, 0);
  currentCommand = -1;
}
else if(currentCommand == 33) {
```

```
    setColor6(0, 255, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 34) {
    detectItem6();
    currentCommand = -1;
  }
    else if(currentCommand == 35){
    offLED6();
    currentCommand = -1;
  }

  // Compartment 7 commands
  else if(currentCommand == 36) {
    digitalWrite(relayPin7, HIGH);
    currentCommand = -1;
  }
  else if(currentCommand == 37) {
    digitalWrite(relayPin7, LOW);
    currentCommand = -1;
  }
  else if(currentCommand == 38) {
    setColor7(255, 0, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 39) {
    setColor7(0, 255, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 40) {
    detectItem7();
    currentCommand = -1;
  }
  else if(currentCommand == 41){
    offLED7();
    currentCommand = -1;
  }

  // Compartment 8 commands
  else if(currentCommand == 42) {
    digitalWrite(relayPin8, HIGH);
    currentCommand = -1;
  }
  else if(currentCommand == 43) {
    digitalWrite(relayPin8, LOW);
    currentCommand = -1;
  }
  else if(currentCommand == 44) {
    setColor8(255, 0, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 45) {
    setColor8(0, 255, 0);
    currentCommand = -1;
```

```
  }
  else if(currentCommand == 46) {
    detectItem8();
    currentCommand = -1;
  }
    else if(currentCommand == 47){
    offLED8();
    currentCommand = -1;
  }


  // Compartment 9 commands
  else if(currentCommand == 48) {
    digitalWrite(relayPin9, HIGH);
    currentCommand = -1;
  }
  else if(currentCommand == 49) {
    digitalWrite(relayPin9, LOW);
    currentCommand = -1;
  }
  else if(currentCommand == 50) {
    setColor9(255, 0, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 51) {
    setColor9(0, 255, 0);
    currentCommand = -1;
  }
  else if(currentCommand == 52) {
    detectItem9();
    currentCommand = -1;
  }
  else if(currentCommand == 53){
    offLED9();
    currentCommand = -1;
  }
}

void receiveCommand() {
  // Get and return command from Raspberry Pi
  if(Serial.available()) {
    int sent = Serial.readStringUntil('\n').toInt();
    Serial.println("ok");
    currentCommand = sent;
  }
}

void sendResponse(String response) {
  // Send response to the Raspberry Pi
  Serial.println(response);
}

void setColor1(int R, int G, int B) {
  analogWrite(redPin1,   R);
```

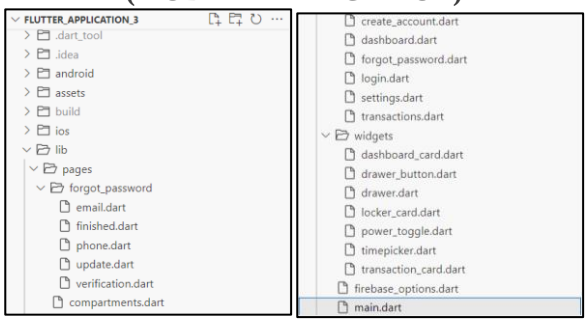
```
  analogWrite(greenPin1, G);
}

void setColor2(int R, int G, int B) {
  analogWrite(redPin2,   R);
  analogWrite(greenPin2, G);
}

void setColor3(int R, int G, int B) {
  analogWrite(redPin3,   R);
  analogWrite(greenPin3, G);
}

void setColor4(int R, int G, int B) {
  analogWrite(redPin4,   R);
  analogWrite(greenPin4, G);
}

void setColor5(int R, int G, int B) {
  analogWrite(redPin5,   R);
  analogWrite(greenPin5, G);
}

void setColor6(int R, int G, int B) {
  analogWrite(redPin6,   R);
  analogWrite(greenPin6, G);
}

void setColor7(int R, int G, int B) {
  analogWrite(redPin7,   R);
  analogWrite(greenPin7, G);
}

void setColor8(int R, int G, int B) {
  analogWrite(redPin8,   R);
  analogWrite(greenPin8, G);
}

void setColor9(int R, int G, int B) {
  analogWrite(redPin9,   R);
  analogWrite(greenPin9, G);
}

void detectItem1() {
  int sensorOut = digitalRead(irPin1);
  if (sensorOut == LOW){
    Serial.println("1");
  }
  else{
    Serial.println("0");
  }
}

void detectItem2() {
```

```
  int sensorOut = digitalRead(irPin2);
  if (sensorOut == LOW){
    Serial.println("1");
  }
  else{
    Serial.println("0");
  }
}

void detectItem3() {
  int sensorOut = digitalRead(irPin3);
  if (sensorOut == LOW){
    Serial.println("1");
  }
  else{
    Serial.println("0");
  }
}

void detectItem4() {
  int sensorOut = digitalRead(irPin4);
  if (sensorOut == LOW){
    Serial.println("1");
  }
  else{
    Serial.println("0");
  }
}

void detectItem5() {
  int sensorOut = digitalRead(irPin5);
  if (sensorOut == LOW){
    Serial.println("1");
  }
  else{
    Serial.println("0");
  }
}

void detectItem6() {
  int sensorOut = digitalRead(irPin6);
  if (sensorOut == LOW){
    Serial.println("1");
  }
  else{
    Serial.println("0");
  }
}

void detectItem7() {
  int sensorOut = digitalRead(irPin7);
  if (sensorOut == LOW){
    Serial.println("1");
  }
```

```
  else{
    Serial.println("0");
  }
}

void detectItem8() {
  int sensorOut = digitalRead(irPin8);
  if (sensorOut == LOW){
    Serial.println("1");
  }
  else{
    Serial.println("0");
  }
}

void detectItem9() {
  int sensorOut = digitalRead(irPin9);
  if (sensorOut == LOW){
    Serial.println("1");
  }
  else{
    Serial.println("0");
  }
}

void offLED1(){
  digitalWrite(redPin1, LOW);
  digitalWrite(greenPin1, LOW);
}

void offLED2(){
  digitalWrite(redPin2, LOW);
  digitalWrite(greenPin2, LOW);
}

void offLED3(){
  digitalWrite(redPin3, LOW);
  digitalWrite(greenPin3, LOW);
}

void offLED4(){
  digitalWrite(redPin4, LOW);
  digitalWrite(greenPin4, LOW);
}

void offLED5(){
  digitalWrite(redPin5, LOW);
  digitalWrite(greenPin5, LOW);
}

void offLED6(){
  digitalWrite(redPin6, LOW);
  digitalWrite(greenPin6, LOW);
}
```

```
void offLED7(){
  digitalWrite(redPin7, LOW);
  digitalWrite(greenPin7, LOW);
}

void offLED8(){
  digitalWrite(redPin8, LOW);
  digitalWrite(greenPin8, LOW);
}

void offLED9(){
  digitalWrite(redPin9, LOW);
  digitalWrite(greenPin9, LOW);
}
```

(MOBILE APPLICATION)



C:\Users\HUAWEI\Documents\Thesis\flutter_application_3\lib\main.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'package:flutter_application_3/pages/login.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: Login(),
    );
  }
}
```

Appendix D
Pictures

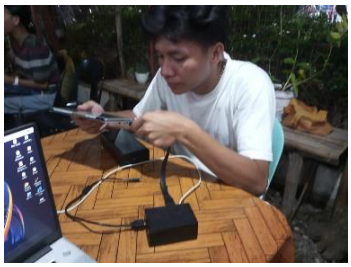
Needs Assessment

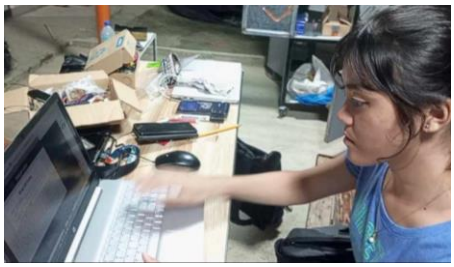


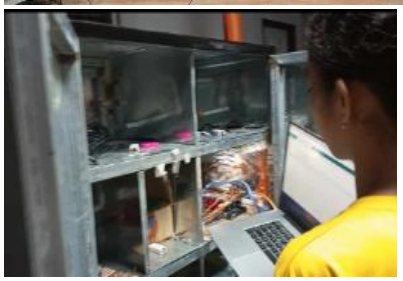

Topic Defense

Proposal Defense

Development of the Prototype











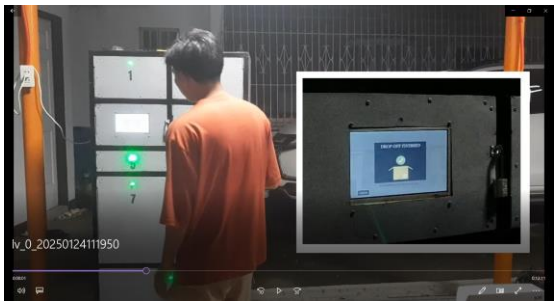


Prototype Defense

Data Gathering


SECURITY

Final Defense



Appendix E Curriculum Vitae

RAVEN BRYLE HEXTER O. RIEGO
Address: Labo, Sta Cruz, Marinduque
Mobile Number: +63 970 478 7090
Email Address: riegoraven839@gmail.com

EDUCATIONAL BACKGROUND

Elementary

School: Lucena East 8 ES. 2008-2009
Address: Brgy. Little Baguio II, Lucena City

School: Elvira Razon Aranilla ES 2009-2011
Address: Brgy. Ibabang Dupay, Lucena City

School:Lucena East 8 ES. 2011-2015
Address: Brgy. Little Baguio II, Lucena City

Junior High School

School: Lucena City NHS & Marcopper Schools Inc. 2015-2016
Address: Brgy. Ibabang Dupay Lucena City & Tapian, Sta.Cruz Marinduque

School: Lucena City NHS. 2016-2017
Address: Brgy. Ibabang Dupay, Lucena City

School: Marcopper Schools Inc. 2017-2018
Address: Brgy. Tapian, Sta.Cruz, Marinduque

School: Kilo-Kilo NHS 2018-2019
Address: Brgy. Kilo-Kilo, Sta.Cruz Marinduque

Senior High School

School: Landy National High School 2019-2021
Address: Landy, Sta Cruz., Marinduque

College

Bachelor of Science in Computer Engineering
School: Marinduque State University 2021-2025
Address: P. Manguera Sr. Rd., Brgy Tanza, Boac, Marinduque

PERSONAL INFORMATION

Date of Birth	:	July 8, 2002
Place of Birth	:	Sampaloc, Manila
Gender	:	Male
Civil Status	:	Single
Nationality	:	Filipino
Religion	:	Roman Catholic
Father's Name	:	Rodel Riego
Occupation	:	N/A
Mother's Name	:	Florizel Riego
Occupation	:	N/A

JELAICA MAE J. MAYORES
Address: Bagtingon, Buenavista, Marinduque
Mobile Number: +63 966 319 2450
Email Address: jelaicamaemayores@gmail.com

EDUCATIONAL BACKGROUND

Elementary
School: Bagtingon Elementary School 2008-2015
Address: Bagtingon, Buenavista, Marinduque

Junior High School
School: Bagtingon National High School 2015-2019
Address: Bagtingon, Buenavista, Marinduque

Senior High School
School: Buenavista National High School 2019-2021
Address: Poblacion, Brgy.Uno, Buenavista, Marinduque

College
Bachelor of Science in Computer Engineering
School: Marinduque State University 2021-2025
Address: P. Manguera Sr. Rd., Brgy Tanza, Boac, Marinduque

PERSONAL INFORMATION

Date of Birth	:	March 24, 2003
Place of Birth	:	Bagtingon, Buenavista, Marinduque
Gender	:	Female
Civil Status	:	Single
Nationality	:	Filipino
Religion	:	Roman Catholic
Father's Name	:	Gabriel M. Mayores Jr.
Occupation	:	Building Electricians, Foreman/Construction Worker
Mother's Name	:	Rosita J. Mayores
Occupation	:	Housewife

CHRISNELL JOY C. LIMPIADA
Address: Balanacan, Mogpog, Marinduque
Mobile Number: +63 916 777 3484
Email Address: cjlimpiada03@gmail.com

EDUCATIONAL BACKGROUND

Elementary
School: Bauang North Central School — 2008-2015
Address: Central East, Bauang, La Union

Junior High School
School: Don Eulogio de Guzman Memorial National High School — 2015-2018
Address: Calumbaya, Bauang, La Union
School: Balanacan National High School — 2018-2019
Address: Balanacan, Mogpog, Marinduque

Senior High School
School: Marinduque State University — 2019-2021
Address: Tanza, Boac, Marinduque

College
Bachelor of Science in Computer Engineering
School: Marinduque State University — 2021-2025
Address: P. Manguera Sr. Rd., Brgy Tanza, Boac, Marinduque

PERSONAL INFORMATION

Date of Birth	:	June 19, 2003
Place of Birth	:	San Fernando, La Union
Gender	:	Female
Civil Status	:	Single
Nationality	:	Filipino
Religion	:	Born Again Christian
Father's Name	:	Nelson J. Limpiada
Occupation	:	Tricycle Driver
Mother's Name	:	Christine C. Limpiada
Occupation	:	N/A

REGINA G. GONZALES
Address: Jolo, Santa Cruz, Marinduque
Mobile Number: 09707616615
Email Address: reginakim@gmail.com

EDUCATIONAL BACKGROUND

Elementary
School: Cecilio Apostol Elementary School — 2008-2015
Address: F.Yuseco St., Barangay 346, Zone 35, Santa Cruz, Manila

Junior High School
School: Makapuyat National High School — 2015-2019
Address: Napo, Santa Cruz, Marinduque

Senior High School
School: Makapuyat National High School — 2019-2021
Address: Napo, Santa Cruz, Marinduque

College
Bachelor of Science in Computer Engineering
School: Marinduque State University — 2021-2025
Address: P. Manguera Sr. Rd., Brgy Tanza, Boac, Marinduque

PERSONAL INFORMATION

Date of Birth	:	August 4, 2003
Place of Birth	:	Manila
Gender	:	Female
Civil Status	:	Single
Nationality	:	Filipino
Religion	:	Roman Catholic
Father's Name	:	Rene Rey
Occupation	:	Farmer
Mother's Name	:	Gene Gonzales
Occupation	:	Saleslady