RFID INTEGRATED SMART SURVEILLANCE SYSTEM

NURUL AIMAN ASYIQIN BINTI NOR AZRIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

RFID INTEGRATED SMART SURVEILLANCE SYSTEM

NURUL AIMAN ASYIQIN BINTI NOR AZRIN

This report is submitted in partial fulfilment of the requirements for the

Bachelor of Computer Science (Computer Security)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

RFID INTEGRATED SMART SURVEILLANCE SYSTEM

Is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT: ________________________________ Date: 17/05/2021

(NURUL AIMAN ASYIQIN BINTI NOR AZRIN)

I hereby declare that I have read this project report and have found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Computer Security) With Honours.

DR. ZAHEERA ZAINAL ABIDIN
Pensyarah Kanan
Fakulti Teknologi Maklumat Dan Komunikasi (FTMK)
Universiti Teknikal Malaysia Melaka (UTeM)

SUPERVISOR: ________________________________ Date: 30/06/2021

(DR. ZAHEERA BINTI ZAINAL ABIDIN)

DEDICATION

To my dearest parents, Nor Azrin Bin Md Latip and Zubaidah Binti Abdullah, my siblings, Nurul Nabilah Najwa Binti Nor Azrin and Muhammad Nafiz Bin Nor Azrin and the rest of my adorable family, thank you for the patience, support and prayers.

To Aisyah Binti Mohamad Hafizul, thank you for everything.

To all of my friends who have always supported me in completing this project,

Thanks a lot.

Just as importantly, special thanks to my beautiful and supportive supervisor. Thank you for the guidance, the encouragement, the motivation and the praises you gave to me. They all have extremely helped in completing the project

May Allah bless us all. Aamin

ACKNOWLEDGEMENT

First of all, I am eternally grateful to ALLAH S.W.T for His mercy and guidance in giving me strength and answers to the problems I encountered countless of times in completing this project. I would also like to express my gratitude and appreciation to all of those who have made it possible for me to complete this project. Special thanks to Dr. Zaheera, who have guided me with detailed suggestions, encouragements and have extensively helped me to coordinate my project both in the project and its report.

Additionally, I would like to thank my parents for the encouragement, the prayers and the support that had my back throughout my whole studies and during the completion of this project . furthermore, I would like to extend my thanks to Universiti Teknikal Malaysia Melaka (UTeM) for providing me the opportunity to produce this project. Last but not least, I am exceptionally grateful to all of my friends to have motivated me in all aspects of life. Thank you very much to all of you. I am eternally grateful for each and every one of you.

ABSTRACT

RFID is an abbreviation that stands for radio-frequency identification. It uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID system consists of a tiny radio transponder, a radio receiver and transmitter. This technology has been actively used to track parcels and items in an inventory or even used to assist the toll payments in Malaysia, providing it as a service alongside manual payments and smart tags. In this project. RFID will be used as access control on each entrance and rooms in a university. This is to detect the movements of its users in the building. The RFID readers will be places at each entrance and the users are to tap their tags/card onto the reader to record their presence in the room. This project is aimed at creating an internal monitoring system to keep track of the movements of staffs and students within the university grounds.

ABSTRAK

RFID merupakan singkatan yang bererti identifikasi melalui radio frekuensi. RFID menggunakan medan elektromagnetik untuk mengenalpasti and mengesan tag yang dilekatkan pada objek secara automatik. Sistem RFID terdiri daripada transponder radio kecil, penerima radio dan pemancar. Teknologi ini telah digunakan secara meluas dalam mengesan bungkusan dan item di dalam inventori. Ia juga bahkan digunakan untuk menyelia pembayaran tol di Malaysia. Ia disediakannya sebagai perkhidmatan tambahan di samping pembayaran manual dan tag pintar yang sedia ada. Dalam projek ini, RFID akan digunakan sebagai mekanisma akses kawalan pada setiap pintu masuk dan bilik di universiti. Ini untuk mengesan pergerakan pengguna di dalam bangunan. Pembaca RFID akan berada di setiap pintu masuk dan pengguna harus mengetuk tanda / kad mereka pada pembaca RFID tersebut untuk merakam kehadiran mereka di dalam bilik. Projek ini bertujuan mewujudkan sistem pemantauan dalaman untuk mengawasi pergerakan staf dan pelajar di kawasan universiti.

Table of Contents

List of Tables Page

List of Figures Page

List of Attachments Page

CHAPTER 1: INTRODUCTION

1.1 Introduction

RFID (Radio Frequency Identification) is a widely used technology in the modern days of this era. Its seamless and almost immediate detection of the tags exposed to the range of its readers allows it to track the movement of the object attached to it quickly. The main idea of this project is inspired from a variety of sources. Firstly, the usage of RFID in inventory and parcel warehouses is the main source of inspiration for this project. The idea of scanning objects as it passes to through the readers gave an insight of how humans could also be detected while passing through the range of the reader while possessing the RFID-readable tags or cards.

Secondly, due to the ongoing pandemic, it has become a widely known procedure in Malaysia to scan the venues prior to entering it in order to create a log of where each citizen went. This is a good strategy to track the places that each individual have been to in case if they are tested positive of the Covid-19 disease. However, this has not been implemented in depth in a university's faculty compound. Faculties can register for the MySejahtera QR code to be implemented in their compound but this goes as far as detecting a person has entered the faculty building with the equipped technology.

The implementation of QR code scanning is also not as convenient because this meant that every student and staff are required to own a smartphone with a threshold of a certain Android or IOS version in order to be able to scan the QR codes. This will of course lead to additional cost on both parties (University and the users.) Thus, this project is aimed at creating an internal monitoring system to keep track of the movements of staffs and students within the university grounds using the technology of RFID

1.2 Problem Statement (PS)

This project embarks on the following problems that exist in the current environment of the university. Firstly, the recent Covid-19-related issue is synonymous with the concept of contact tracing. Contact tracing is essentially tracking the places a person went to as soon as they step out of their houses. This issue is related to the university in such manner, that the university lacks the surveillance on the personnel that enter or leave a facility/rooms within the university compound.

Secondly, a university consist of multiple locations that are deemed as confidential and only allows a certain group of people to enter. Thus, they have implemented CCTVs and the existing access control system that allows certain staffs to interact with these facility. However, in events of ‘tailgating' or ‘piggybacking,' the methods of detecting these individuals are unavailable.

Thirdly, there is a possibility that the university might have to encounter emergency situations that requires them to pinpoint the last known whereabouts of its lecturers and students. In this case, There is a scarcity of a centralised system that can detect the locality of students' and lecturers' whereabouts.

Table 1.1: Summary of Problem Statement

PS	Problem Statement
PS$_1$	Lack of mechanism in surveillance monitoring system to detect student and staff movement in the university.
PS$_2$	The method for detecting tailgating and piggybacking are not available.
PS$_3$	There is a scarcity of a centralised system that can detect the locality of students' and lecturers' whereabouts

1.3 Project Question (PQ)

Table 1.2: Summary of Project Question

PS	PQ	Project Question
PS$_1$	PQ$_1$	What is the mechanism that can be used to track the places students and staffs went to within the university?
PS$_2$	PQ$_2$	What is the method to detect the responsible personnel in events of tailgating/piggybacking?
PS$_3$	PQ$_3$	What is the ideal centralized method for detecting the locality of students and lecturers?

1.4 Project Objective (PO)

Table 1.3: Summary of Project Objectives

PS	PQ	PO	Project Objective
PS$_1$	PQ$_1$	PO$_1$	To study a mechanism to track the places students and staffs went within the university compound
PS$_2$	PQ$_2$	PO$_2$	To develop a method to detect the responsible personnel in events of tailgating/piggybacking.
PS$_3$	PQ$_3$	PO$_3$	To propose an ideal centralized method for detecting the locality of students and lecturers?

1.5 Project Scope

The scope of this project involves the students and staff in the university areas. This project constructs a monitoring system that integrates with RFID technology to track the user whereabouts. This project aims to produce a product prototype. The RFID readers may not be able to detect long ranges due to cost.

1.6 Project Contribution (PC)

Table 1.4: Summary of Project Contribution

PS	PQ	PO	PC	Project Contribution
PS_1	PQ_1	PO_1	PC_1	Prototype of a Smart Access Control System that incorporates RFID technology as an additional layer of security to the existing access control system.
PS_2	PQ_2	PO_2	PC_2	A new method using RFID readable card/tags to every university members as their matric card
PS_3	PQ_3	PO_3	PC_3	A centralised model for Smart Access Control System that assign RFID readers to rooms to indicate the locality of the student or lecturer when they enter that room.

1.7 Thesis Organization

Chapter 1: Introduction

This chapter introduce the readers to the general background, the problem statement, the objective, and scope of the project

Chapter 2: Literature Review

This chapter consists of the study in relation to the creation of the project. This includes the review of thesis, papers, journals, etc.

Chapter 3: Project Methodology

This chapter focuses on the materials and data gathered for the study. There will be elaboration on the data and information are collected from the previous studies.

Chapter 4: Design

This chapter discusses on how the data and information collected in the previous chapter are designed to form a plan of processes before implementation. Deliverables included in this chapter are charts and drawings to describe the planned processes

Chapter 5: Implementation

This chapter explains the implementation processes of the project based on the plan as explained in the previous chapter. This chapter includes pictures and steps of implementation.

Chapter 6: Testing And Analysis

This chapter consists of the testing and analysis executed on this project. All the data and information are collected and tested in this chapter.

Chapter 7: Project Conclusion

This chapter concludes the study, the project and explains some recommendations for the future improvements.

1.8 Conclusion

To conclude, this chapter has explained the backgrounds of the project, its problem statements, and objectives as well as the scope of the proposed project. The thesis's structure was also briefly discussed in this chapter. The next chapter will discuss an in-depth on past related works of the proposed project.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter looks at previous RFID-related initiatives that have been completed. This section discusses the RFID's numerous applications and why it was chosen as the technology for this project. Finally, the chapter will wrap up and summarise the information presented in this chapter.

2.2 Related Work/Previous Work

The first issue that any user who has used or come across a personnel/company that employs barcode technology is that barcodes are a technology that relies on the concept of line of sight. To put it another way, the scanner must be able to view the barcode before it can read it. In addition, certain other difficulties impacting barcode technology, such as environmental conditions, have surfaced. Scanning barcodes on a label is inefficient due to factors such as temperature, grime, and dangerous contamination.

RFID is not a new technology. It has been around for quite a while. It was initially employed by the US government in the 1940s to distinguish between hostile and friendly aircraft. Furthermore, it was not a technology that had lately been inhabited. RFID was used to track livestock and nuclear items during the 1970s.

The usage of radio tags was then commercialized in the 1980s and 1990s for sectors such as package delivery, luggage management, food tracking in supermarkets, and highway toll monitoring. Following that, in 1997, Mobil, a gas station firm, utilized RFID to record transactions and deduct the gas cost from the customer's credit card utilizing what is now known as the payWave concept.

As a result, to support my claim, RFID is not a new technology. It has been around for a long time and is still in use. RFID is frequently utilized in supply chain operations as well. Supply chain operation is an entire process of making and selling goods. This encompasses everything from getting suppliers to manufacturing the goods

This technology is employed because it is a cost-cutting strategy for a supply chain that is becoming increasingly expensive. This technology is implemented by attaching RFID tags to the goods, which are then detected by radio waves by the reader. Due to the discovery of various problems, some publications have said that RFID will replace the original barcode technology.

Finally, in attempt to lessen the risk of a Covid-19 pandemic, RFID was used to track and detect contact tracing. The staffs that are in same space as the Covid-19 patient can be detected from the tags that they wear and through the readers placed in the patient's rooms (Mehta, S., Grant, K., Atlin, C., & Ackery, A., 2020).

2.2.1 Project Domain

This domain for this project is access control system. Access control is concerned with limiting the activity of legitimate users. Access control consist of 4 models. (Gentry, 2021).

- Mandatory Access Control (MAC)
  MAC limits access control administration to the owner. This means that the end user has no control over any settings that grant anyone access. MAC is related with two security models: Biba and Bell-LaPadula.
- Role-Based Access Control (RBAC)
  RBAC approach provides access control based on an individual's role in an organisation. As a result, rather than granting permissions to a person as a security manager, the role of security manager already has permissions.
- Discretionary Access Control (DAC)
  When compared to the most restrictive MAC model, the DAC model is the least restrictive. DAC gives a person complete control over any objects they possess, as well as the programmes that go with them.

- Rule-Based Access Control (RBAC or RB-RBAC).
  RB-RBAC assigns responsibilities to users dynamically based on the system administrator's criteria. RBAC would be the tool of choice if someone was only given access to files during particular hours of the day. The additional RB-RBAC "rules" that must be implemented may need to be "coded" into the network in the form of code by system administrator.

In this project, the role-based access control model will be used where the two users of the system are users and admin.

2.2.2 Project Terms

2.2.2.1 RFID

RFID is a technology similar to barcodes. It uses radio frequency waves as its medium of data transmission.

Technicality of RFID

RFID technology consists of 3 components

1. The tags/transponder

   RFID tags contains hard copper coils connected to an integrated circuit (IC) that has an antenna attached to it. These components are then attached to a house which can be a form of a card of a tag. RFID tags can be as small as a grain of rice or one third of a millimetre. Information are stored in the IC and the antenna is used by the reader to capture these information through radio frequency. There are 2 types of tags (Attaran, 2007).

   i. Active tags
      Active tags are self powered by batteries. They are the expensive version of passive tags and they have limited lifespan, unless recharged. They act as a mini computer and their prices ranged from $4 to $20 per tag.

   ii. Passive tags
      Passive tags are not battery powered. They are detected using the radiated energy from readers to transmit their data. They are cheap and

they have unlimited lifespan. Their price ranges from 5 cents to 25 cents per tag.

RFID tags are read-only. RFID tags have memory spaces in them. Some of these memory spaces contain data that are reserved, permanent and not changeable. However, there are other spaces in the memory that are empty and can be filled with information of choice. One of the difference between a barcode and an RFID tags is the amount of information it can hold. Barcodes can hold 12 to 15 info characters consisting of digits from 0 to 9. RFID tags uses a 94 character technology to hold their information.

2. The reader

   RFID reads contains a radio frequency transmitters and receivers that is controlled by a microprocessor or digital signal processor that communicates with the tags.

3. Computer

   The RFID reader collect the information form the tag's antenna before sending it to the connected computer for information processing.

2.2.2.2 Arduino

Arduino is a free and open-source platform for building and programming electronic devices. It can receive and transfer data to most devices, as well as transmit commands to specific electronic gadgets over the internet. It programmes the board using an Arduino Uno circuit board and a software programme using simplified C/C++

2.3 Critical Review Of Current Problem And Justification

The current situation of the faculty does not apply any access control mechanism to monitor the movements of its visitors. The labs in the faculty are equipped with access control systems but they are not able to detect the exact personnel (i.e. students) that exist in the current room. The current access control system are also not applied in every single room in the faculty. Thus, if a person is not in the lab, or within the sight of the CCTV cameras, there is no way to detect the location of this person. This problem is significantly important as it presents risk to the security of the faculty. In addition to that, in the midst of a Covid-19 pandemic, the current access control system does not provide a way to monitor contact tracing between the people in the faculty.

2.3.1 Project Methodologies

2.3.1.1 Access Control With Biometric Scanning

A biometric access control system is, in essence, a pattern recognition unit that collects a specific type of biometric data from a person, focuses on a relevant aspect of that data, compares that feature to a predefined collection of attributes in its database, and then takes action based on the accuracy of the comparison. The problem with this method is that biometric authentication methods raise privacy and security problems, as there is no way to restore the harm once biometric data has been hacked. You can reset a compromised password, but you cannot alter a hacked biometric like a fingerprint, ear picture, or iris scan (Fauscette, 2018).

2.3.1.2 Remote Access Control

Remote Access or Smart Locks allow the user to operate the lock without having to be in close contact to it. Rather than being physically present, the locks are controlled by software communicated via Bluetooth, Wi-Fi, or Z-wave technologies. The user can lock or unlock doors or drawers from a remote place via remote access through applications (SENSEON, 2020). Remote access control posed threats of information eavesdropping and data manipulation as it travels through the internet.

2.3.1.3 Access Control With Keypad

Electronic locking systems using keypads are the most basic. The equipment uses a keypad to replace the key to a lock or door. Most systems contain a physical backup key. The user enters a preprogramed code using a basic reader. The system unlocks the door if the code matches. Multiple codes and temporary codes can be entered on the keypad. Electronic keypad locks can be hacked and there is no failsafe system in place. Some locks are only compatible with deadbolts. Others will only work if the door is entirely closed.

2.3.2 Project Techniques

2.3.2.1 RFID with Facial Recognition

As mentioned by (Farooq, U., ul Hasan, M., Amar, M., Hanif, A., & Asad, M. U., 2014), in the journal RFID Based Security and Access Control System, RFID technology was used along with facial recognition techniques in order to provide a security access control system for a hostel.

The RFID tag is issued alongside a face capture before feeding it to the neural network during registration phase. Then, during the recognition phase the RFID tag is detected along side a face capture and sends it to the neural network. If the these data existed then access is granted.

2.3.2.2 RFID with Inductive Loop

In (Ostojic, G., Stankovski, S., Lazarevic, M., & Jovanovic, V. ,2007), RFID is used with inductive loop to detect the presence of a vehicle. When things pass across inductive loops, they measure the change in the field, similar to a metal detector. The loop field changes when a vehicle drives over a loop sensor, allowing the detecting device to detect the presence of an object (mainly a vehicle).

A user stops their vehicle on an inductive loop (in front of the arm barrier) and presses a pushbutton on the control centre on the side with their hand. After then, the arm of the automatic barrier lifts, allowing access to the parking lot. A user also receives a printed parking ticket the moment they entered the parking lot.

There is a display with a light indicator (red/green) on the entry barrier. On the display, the number of available parking spaces is displayed. The arm of a barrier will not lift if there are no available parking spaces.

2.3.2.3 RFID with Deep learning and Sensor Fusion

As explained by (Johnston, 2018), in Amazon Go, RFID is implemented with additional techniques of deep learning algorithm and a fusion of sensors. Amazon Go did not expose the accuracy of its shopping system. However, from what can be observed by experiencing the system itself as a consumer, the shop has no checkout counters. Upon entering the mart, consumers will need to scan a QR code using the Amazon Go mobile application to detect their presence.

On the ceilings of the mart, there are hundreds of infra-red cameras mounted on it as well as electronic sensors to identify each customer. These cameras and sensors are also used to differentiate between items for sale or items with similar appearances such as flavours of the same canned drink. It is also used to detect consumers with a similar appearance.

2.3.3 Project Software and Project Hardware

2.3.3.1 Arduino

Arduino is a company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices using open-source hardware and software.

2.3.3.2 Node-RED

Node-RED is a visual programming tool that was originally developed by IBM for connecting hardware devices, APIs, and online services as part of the Internet of Things. Node-RED includes a flow editor that can be used to create JavaScript functions in a web browser.

2.3.3.3 SQL Server

Microsoft SQL Server is a relational database management system that Microsoft has created. It is a database server, which is a software product whose primary function is to store and retrieve data as requested by other software applications, which may run on the same computer or on a networked computer.

2.3.4 Conclusion

This chapter have explained the uses of RFID technology in different industries as well as its implementation. This chapter also briefed about the related access control techniques and methodologies. This chapter also explained the software and hardware used to execute the system.

2.4 Proposed Solution

In this project, the selected technique that I have chosen for this project is an access control system with RFID. This is because, the 3 methodologies that I have discussed earlier posed risks such as the inability to recover from biometric hacking, the risks of eavesdrop and data manipulation in remote access control, and the compatibility issues with keypad access control. An access control system with RFID is instant and easy.

RFID uses a unique identifier (UID) in its tags that exist in the database. Only the administrator (who assigned this tag) and the individual who possesses a separate RFID reader are aware of the UID. However, ruling out the possibility of an individual carrying a portable RFID reader, the next conceivable attempt will be focused towards the database in order to get access to the UID. Nonetheless, if the UID is encrypted, the problem will be solved.

2.5 Conclusion

In summary, this chapter has explained the literature review of this project which consist of the related research’s previous work. This chapter also briefed on the existing methodologies and techniques used previously in other projects. In the next chapter will be the methodology of this project. This include the project milestones and the activities that will be done in each stages.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

This chapter discuss on the methodologies used in this project. Following that, this chapter provides a full overview of the study approach used and the data collection process. In addition, the Gantt chart for this project will be shown in this chapter. The Gantt chart depicts how the work is organised and if the project is on schedule or not. The Gantt chart's purpose is to direct the project's progress.

3.2 Methodology

There are a variety of software development models from which to choose depending on the needs of the project. The right approach for the job can result in a high-quality product. The Rapid Application Development (RAD) model is used in this project. A type of incremental model is the RAD model. The components or functions are developed in parallel in the RAD model, as if they were mini projects. The projects are timed, delivered, and then put together into a working prototype. The strategy is working within the mile stones and is being presented on time. (Renger, 2021)

The RAD Model Design (Lucid Content Team, 2018)

Rapid application development is an agile software development methodology. In contrast to Waterfall methodologies, RAD prioritises working software and feedback over meticulous planning and requirements documentation. RAD consist of four main phases.

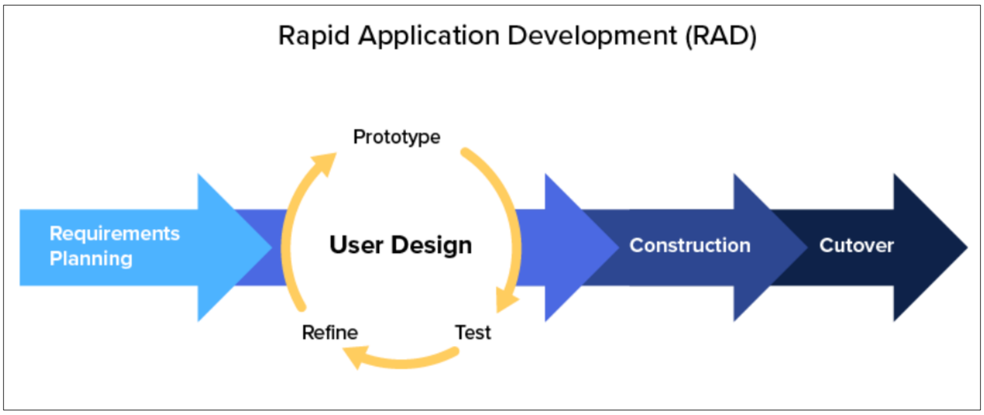


Figure 2.1: Rapid Application Development Methodology

(https://blog.yeeflow.com/rapid-application-development-rad-tools/)

Phase 1: Establishing Requirements

The planning phase is crucial to the project's eventual success. Developers, clients (software users), and team members communicate throughout this stage to determine the project's goals and expectations, as well as any current or anticipated issues that need to be handled. This phase includes the following stages:

1. Investigating the current issue
2. Defining the project's needs
3. Finalizing the requirements.

This phase is applied in the project before the semester started, when students are instructed to come up with a project title and are asked to propose them to their selected supervisor.

Phase 2: User design

Clients and developers collaborate throughout this phase to construct a customizable software that allows consumers to test each prototype of the product at each stage to guarantee it fits their needs. This strategy allows developers to make changes to the model as they go until they achieve a suitable result. This phase of the project is where students develop their initial prototype of each module and present it to the supervisor at the following meeting. Based on the comments obtained and discussed during each session, the prototype is then modified and updated.

Phase 3: Rapid construction

Phase 3 transforms the prototypes created during the design phase into a working model. Because the majority of the difficulties and adjustments were handled during the rigorous iterative design phase, developers then built the final working model. The following are the components of the phase:

1. Preparation for a rapid construction
2. Development of programmes and applications
3. The coding
4. System, unit, and integration testing

This third step is crucial since the client can still provide feedback at any point during the process. They can make suggestions for improvements, revisions, or even new ideas to solve problems as they arise. Students have constructed a working system based on the prototype in this level. The system should be completely connected to the database's data. The supervisor may offer ideas to improve the system

Phase 4: Cutover

This is the stage when the finalised work is ready to be released. It entails data conversion, system testing, and maintenance, as well as user training. While the coders and clients continue to seek for flaws in the system, all final improvements are made. Following the end of the system development, the last step is implemented. Students create test case scripts in this phase to ensure that the system can handle any type of error.

3.3 Project Milestone

3.3.1 Project Milestone

Project milestone refers to a specific point in a project's timeline. These points can be used to identify the beginning and end of a project, as well as the conclusion of a key phase of work. The project milestone is as shown in Table 3.1.

Table 3.1: Project Milestone

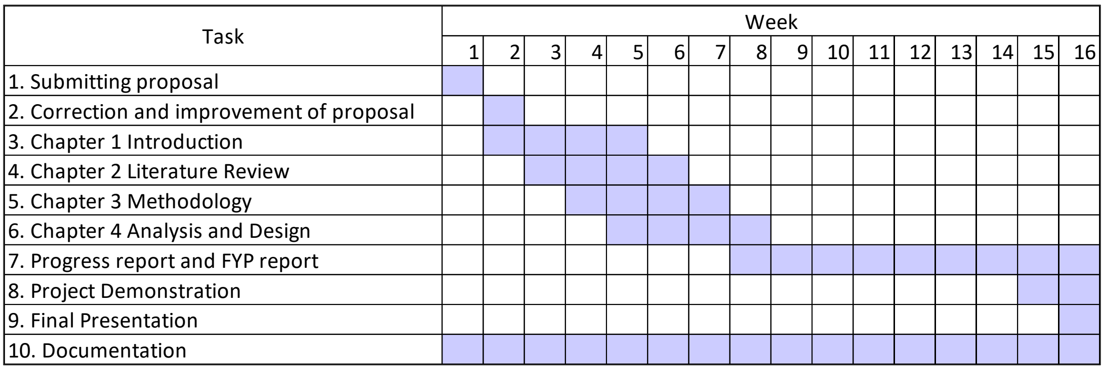
WEEK	ACTIVITY	NOTE / ACTION
<W0 (<21/3)	Select a suitable project topic and potential Supervisor	Action – Student
W1 (15/3 > 21/3) Meeting 1	Proposal PSM: Discussion with Supervisor	Deliverable – Proposal Action – Student
	Proposal assessment & verification	Action – Supervisor
W2 (29/3 > 4/4)	Proposal Correction/Improvement	Action – Student
	Proposal submission to Committee via email	
	Proposal Approval	Action – PSM/PD Committee
	List of Supervisor/Title	
W3 (29/3 > 4/4) Meeting 2	Proposal Presentation & Submission via PSM ULearn	Deliverable-Proposal Presentation (PP) and Completed Proposal Form Action – Student
	Chapter 1 (System Development Begins)	Action – Student
W4 (5/4 > 11/4)	Chapter 1	Deliverable – Chapter Action – Student, Supervisor
W5 (12/4 > 18/4)	Chapter 2	Action – Student
W6 (19/4 > 25/4) Meeting 3	Chapter 2	Deliverable – Chapter 2
	Project Progress	Progress Presentation 1 (PK1)

		Action – Student, Supervisor
	Student Status	Warning Letter 1 Action – Supervisor, PSM/PD Committee
W7 (26/4 > 2/5)	Chapter 3	Action – Student
W8 (3/5 > 9/5)	Chapter 3	Deliverable: Chapter 3 Action – Student, Supervisor
W9 (10/5 > 16/5)	MID SEMESTER BREAK	
W10 (17/5 >23/5) Meeting 4	Chapter 4	Action – Student
	Project Progress	Progress Presentation 2 (PK 2) Action – Student, Supervisor
	Student Status	Warning Letter 2 Action – Supervisor, PSM/PD Committee
W11 (24/5 >30/5) Demonstration	Project Demo	Action – Student, Supervisor
	Determination of student status (Continue/Withdraw)	Submit student status to PSM/PD Committee Action – Supervisor, PSM/PD Committee
W12 (31/5 > 6/6)	Project Demo PSM1 Report	Action – Student, Supervisor
W10 (7/6 >13/6) Meeting 5	Project Demo PSM1 Report Schedule the Presentation	Action – Student, Supervisor Action – PSM/PD Committee Presentation Schedule

W14 (14/6 > 20/6)	Project Demo	Deliverable – Complete PSM1 Draft Report Action – Student, Supervisor
W15 (21/6 >27/6) Final Presentation	FINAL PRESENTATION Submission of the PSM1 Report onto the PSM ULearn.	Action – Student, Supervisor, Evaluator, PSM/PD Committee
W16 (28/6 > 4/7)	REVISION WEEK Correction on the draft report based on the Supervisor and Evaluator's comments during the final presentation session. Submit PSM1 Logbooks to PSM ULearn. Submit an EoS Survey form.	Deliverable – Complete PSM1 Logbooks Action – Student, Supervisor
		EoS Survey Action - Student
	Submission of overall marks to PSM/PD committee	Deliverable: Overall PSM1 score sheet Action – Supervisor, Evaluator, PSM/PD Committee
W17 & W18 (5/7 > 18/7)	FINAL EXAMINATION WEEKS	

3.3.2 Project Gantt Chart

A Gantt chart is a form of bar chart that shows the progress of a project. The project's Gantt Chart is as shown below.



Task	Week															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1. Submitting proposal	■															
2. Correction and improvement of proposal		■														
3. Chapter 1 Introduction		■	■	■	■											
4. Chapter 2 Literature Review			■	■	■	■										
5. Chapter 3 Methodology				■	■	■	■									
6. Chapter 4 Analysis and Design					■	■	■	■								
7. Progress report and FYP report								■	■	■	■	■	■	■	■	■
8. Project Demonstration															■	■
9. Final Presentation																■
10. Documentation	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Figure 2.2: Project Gantt Chart

3.4 Conclusion

This chapter has explained the methodology used in creating this project, In addition to that, this chapter have also elaborated on the phases of the methodology model in relation to this project and how it is applied. Lastly, the detailed project milestones and project Gantt chart is also shown in this chapter to represent the flow of schedule that is used in this project. In the next chapter, this report will discuss on the design of the project, which elaborates on the architecture, requirements and component of the project.

CHAPTER 4: ANALYSIS AND DESIGN

4.1 Introduction

This chapter includes an analysis on both project problems and requirement. The requirement analysis will be on the data, functional and non-functional and other requirements. Additionally, this chapter will explain the system design. System design refers to the system architecture, user interface design and database design. Database design will consist on the conceptual and logical design. Lastly, this chapter will explain the software and physical database design.

4.2 Problem Analysis

The developed system has been adopted from the existing access control system using card scanner with fingerprint reader and keypads. The existing system works by either tapping the matric card to the scanner or using fingerprint or accessing using the keypad provided. The scanner then compares the value of the input (consisting a hybrid of a fingerprint scanner, card scanner and keypad lock) to its list of correct password. If the values match then access is granted. However this system is not available for everybody. It is reserved for staffs only. Taking an example of a lab with this access control system. Students who are in the class are not identified using the access control system. The existing system only notes down the lecturer that enters the room. The existing system is illustrated in the DFD as drawn below.

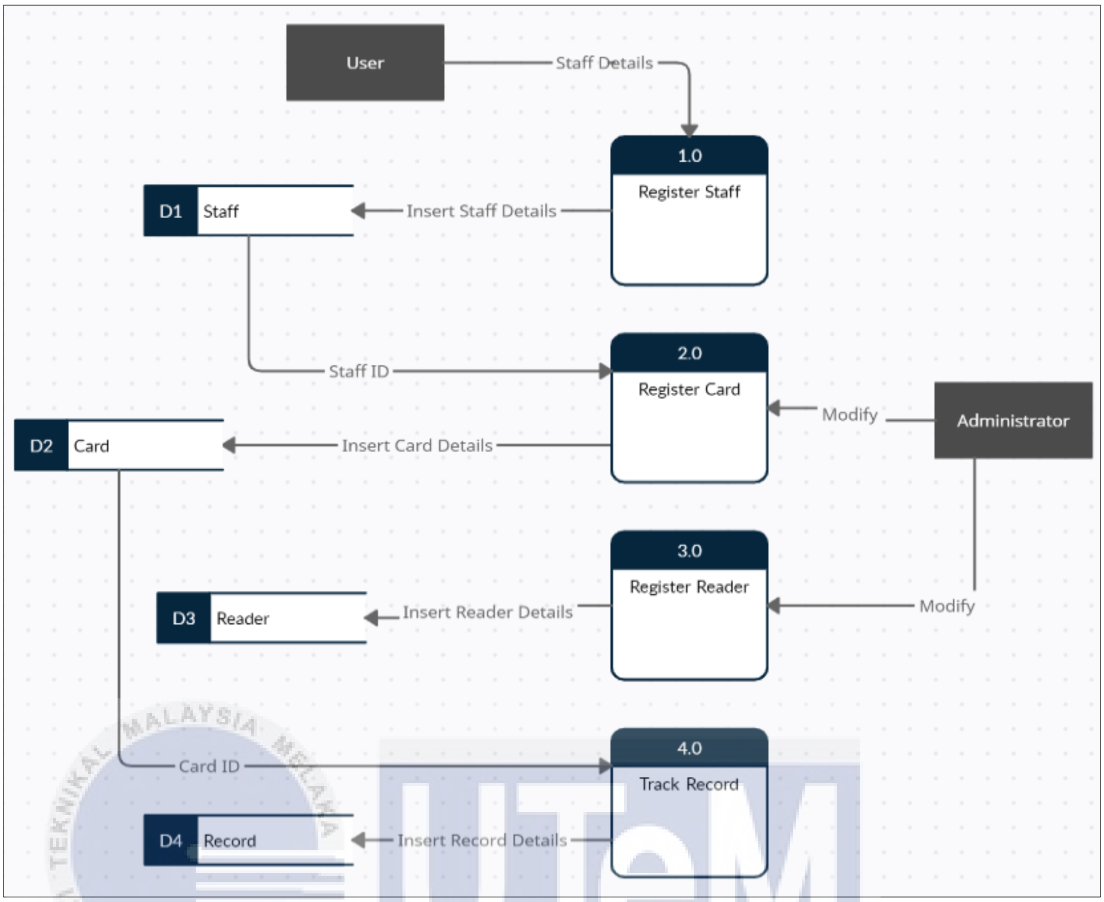


Figure 4.1: DFD of Existing System

4.3 Requirement Analysis

4.3.1 Data Requirement

Data requirement in this project can be categorize into two. The input and output data. Generally, the user's RFID-readable card/tag is the input data in this project whereas the record indicating that the card/tag has been read is the output of this project. Since this project is aimed at creating a simple system, therefore, different inputs and outputs is received/produce respectively based on the database tables. This can be understood better based on the data dictionary of each tables in the database. These data dictionary as shown below.

4.3.1.1 Table: Users

Table 4.1: Data Dictionary for Table Users

No.	Name	Data Type	Length	Primary Key	Description	Mandatory (Yes/No)	Remarks
1.	u_id	numeric(10,0)	10	Yes	A unique user ID	Yes	Should be left justified
2.	matric_num	varchar(10)	10		User's matric number	Yes	
3.	name	varchar(50)	50		User's name	Yes	
4.	password	varbinary(50)	50		User's account password	Yes	
5.	privilege	numeric(1,0)	1		User's account privilege	Yes	
6.	date_created	datetime			Date of data added to the database	Yes	
7.	status	numeric(1,0)	1		Account availability	No	1 - Admin 0 - User
8.	email	varchar(50)	50		User's email	Yes	

4.3.1.2 Table: Card

Table 4.2: Data Dictionary for Table Card

No.	Name	Data Type	Length	Primary Key	Description	Mandatory (Yes/No)	Remarks
1.	c_id	numeric(10,0)	10	Yes	A unique card ID	Yes	Should be left justified
2.	u_id	numeric(10,0)	10		A unique user ID	Yes	
3.	guid	varchar(8)	8		Card unique ID	Yes	
4.	date_created	datetime			Date of data added to the database	Yes	
5.	status	numeric(1,0)	1		Card availability	No	1 - Active 0 - Inactive

4.3.1.3 Table: Building

Table 4.3: Data Dictionary for Table Building

No.	Name	Data Type	Length	Primary Key	Description	Mandatory (Yes/No)	Remarks
1.	b_id	numeric(10,0)	10	Yes	A unique building ID	Yes	Should be left justified
2.	u_id	numeric(10,0)	10		A unique user ID	Yes	
3.	name	varchar(50)	8		Building name	Yes	
5.	status	numeric(1,0)	1		Building availability	No	1 - Active 0 - Inactive

4.3.1.4 Table: Reader

Table 4.4: Data Dictionary for Table Reader

No.	Name	Data Type	Length	Primary Key	Description	Mandatory (Yes/No)	Remarks
1.	r_id	numeric(10,0)	10	Yes	A unique reader ID	Yes	Should be left justified
2.	u_id	numeric(10,0)	10		A unique user ID	Yes	
3.	b_id	numeric(10,0)	10		A unique building ID	Yes	
4.	date_created	datetime			Date of data added to the database	Yes	
5.	status	numeric(1,0)	1		Reader availability	No	1 - Active 0 - Inactive

4.3.1.5 Table: Record

Table 4.5: Data Dictionary for Table Record

No.	Name	Data Type	Length	Primary Key	Description	Mandatory (Yes/No)	Remarks
1.	rec_id	numeric(10,0)	10	Yes	A unique record ID	Yes	Should be left justified
2.	c_id	numeric(10,0)	10		A unique card ID	Yes	
3.	r_id	numeric(10,0)	10		A unique reader ID	Yes	
4.	date_in	datetime			Date of card tap in	Yes	
5.	date_out	datetime			Date of card tap out	No	Update to add date

4.3.1.6 Table: Logs

Table 4.6: Data Dictionary for Table Logs

No.	Name	Data Type	Length	Primary Key	Description	Mandatory (Yes/No)	Remarks
	l_id	numeric(10,0)	10	Yes	A unique log ID	Yes	Should be left justified
1.	rec_id	numeric(10,0)	10		A record ID	Yes	
2.	c_id	numeric(10,0)	10		A card ID	Yes	
3.	r_id	numeric(10,0)	10		A reader ID	Yes	
4.	date_in	datetime			Date of data added	Yes	
5.	status	numeric(1,0)	1		Log status	No	1 - IN 0 - OUT

4.3.2 Functional Requirement

The system general functional flow is that the user taps on the reader of the room it wishes to enter before entering. Then, the reader records the unique id of the card and indicate if it existed in the database. If the card is a registered card and it existed in the database then it keeps the data it reads into the database. Else, it ignores the data. When the data is entered in the database, then the system will recognize the user is inside the room.

When the user wishes to exit the room, they need to tap the card again at the reader before exiting. The reader detects the card id and surfs through the database to find the existing record of the card that indicates that the user was inside the room. Then it updates that record to indicate that the user has now exited the room. This flow is illustrated through the DFD level 1 as shown below in Figure 4.2.

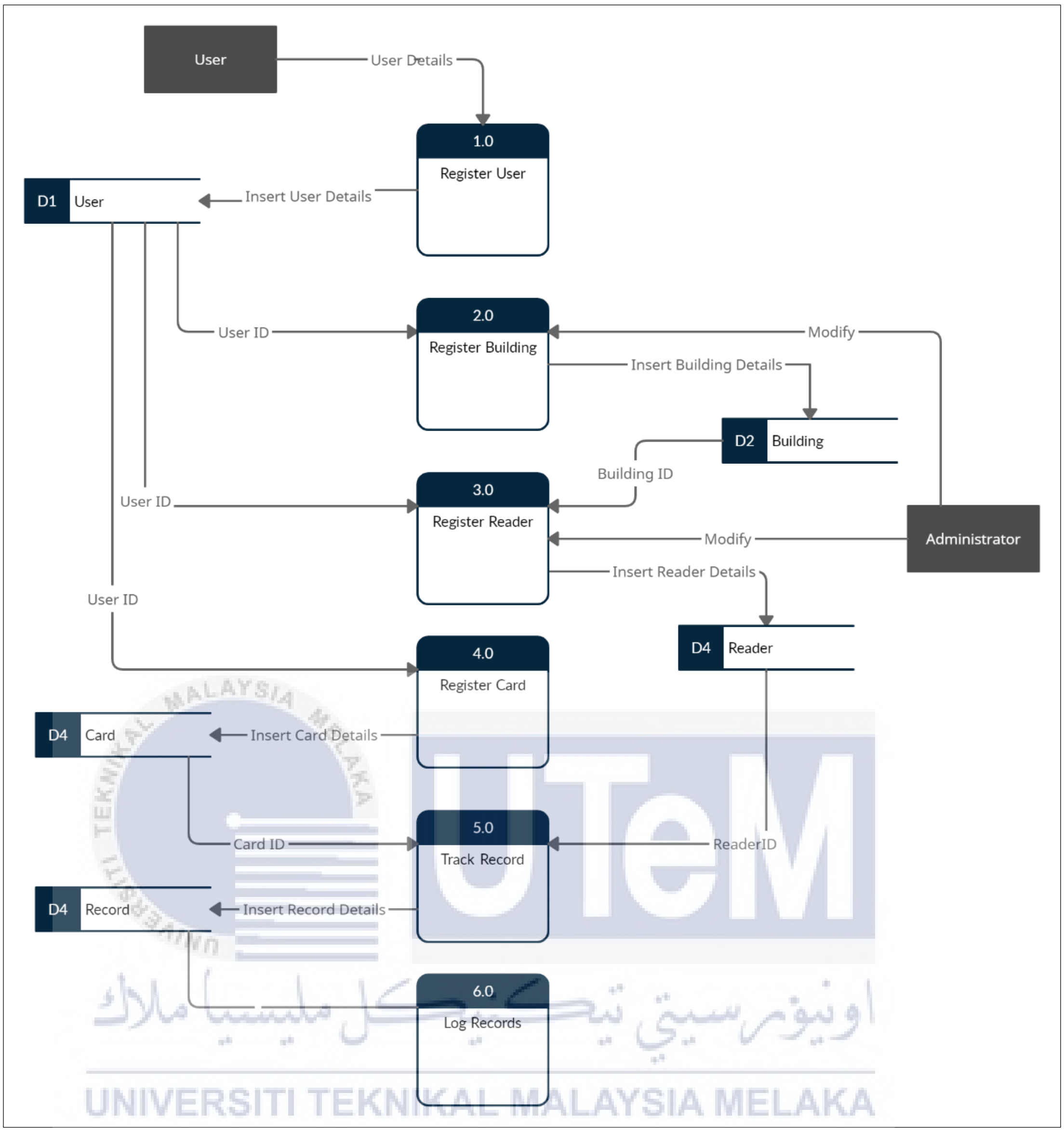


Figure 4.2: The Project's DFD

4.3.3 Non-functional Requirement

This project is aimed to track the records of each cards tapped onto the RFID reader and saves it in a database. This system will also be able to add users and assign cards to them before saving a record of the registered cards. Additionally, each records will be tracked with a status that indicate if the user is in or out of the room through a log table in the database.

The resources needed in this is a single computer to hold the database of the system and the system itself. Other than that, it also needs a minimum of one and a maximum number of 5 RFID tags connected to a single Arduino UNO board. The accuracy of this system depends on the connectivity of the wires between the RFID readers and the Arduino board as well as the range of radio frequency between the card and the RFID reader.

For the reader that this project uses, the maximum range of where a card is detected from the reader is 5 cm. The amount of data a single database can hold depends on the storage that the computer can allocate to the system. Each computer differs based on its storage size.

4.3.4 Others Requirement

4.3.4.1 Software Requirement

i. **Microsoft SQL Server 2018**

Microsoft SQL Server is a database management system that was created by Microsoft. In this project, the SQL Server is the project's database.

ii. **Arduino IDE**

The Arduino Software (IDE) includes a text editor for writing code, a message area, a text console, a toolbar with buttons for basic functions, and a series of menus. It links to Arduino a hardware in order to upload and communicate with programmes. In this project the IDE is used initially to compile and upload codes to the Arduino board. Occasionally, it is also used to debug the output of the codes through the built in serial monitor provided in the IDE itself.

iii. **Node-RED**

Node-RED is a flow-based text editor that runs on the node.js module. In this project, this editor is used to integrate the input and outputs of the Arduino board, the RFID reader, the database and the system itself.

4.3.4.2 Hardware Requirement

i. Arduino Uno Rev 3 Board

Arduino Uno is a microcontroller board based on the ATmega328P. This board is used to control and debug the MFRC522 RFID reader.

ii. MFRC522 RFID Reader

The MFRC522 RFID reader can both read and write data into the memory elements of an RFID-readable Card/Tags. The reader can read data only from passive tags that operate on 13.56MHz. In this project, the reader is used to detect the RFID cards.

iii. RFID-readable Card/Tags

RFID tags/card contain an antenna and an integrated circuit, which are used to transmit data to the RFID reader. The reader then converts the radio waves to a more usable form of data.

iv. Male to Female Jumper Wires

The jumper wires are used to connect between the RFID reader and the Arduino Uno board.

v. USB cable type A/B.

The USB cable is used to connect between the Arduino Uno board and the computer.

4.4 High-Level Design

The high level design consist of the system flow that shows the types of data taken to be processed from the dashboard interface and the RFID reader. It also shows how the contents of each module is implemented in the system. The design can be better understood based on the flowchart shown in the figure below.

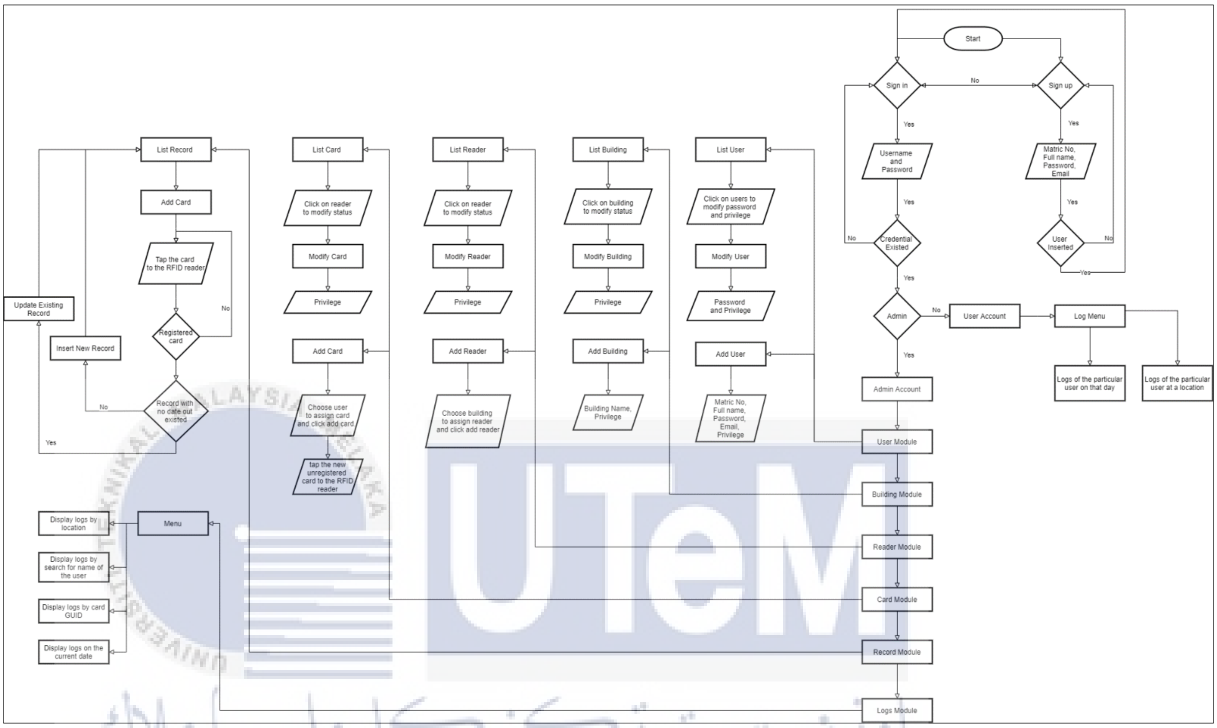

Start
Sign in
No
Sign up
Yes
Yes
Username and Password
Matric No, Full name, Password, Email
Yes
Yes
No
Credential Existed
User Inserted
No
Yes
Yes
Admin
No
User Account
Log Menu
Yes
Admin Account
Logs of the particular user on that day
Logs of the particular user at a location
User Module
Building Module
Reader Module
Card Module
Record Module
Logs Module
List Record
Add Card
Tap the card to the RFID reader
No
Registered card
Update Existing Record
Insert New Record
No
Record with no date out existed
Yes
List Card
Click on reader to modify status
Modify Card
Privilege
Add Card
Choose user to assign card and click add card
tap the new unregistered card to the RFID reader
List Reader
Click on reader to modify status
Modify Reader
Privilege
Add Reader
Choose building to assign reader and click add reader
List Building
Click on building to modify status
Modify Building
Privilege
Add Building
Building Name, Privilege
List User
Click on users to modify password and privilege
Modify User
Password and Privilege
Add User
Matric No, Full name, Password, Email, Privilege
Menu
Display logs by location
Display logs by search for name of the user
Display logs by card GUID
Display logs on the current date

Figure 4.3: The Project's Flow Chart

4.4.1 System Architecture

The system architecture consist of three main components. The first component is the MFRC522 RFID reader that is connected to the Arduino Uno microcontroller. The Arduino Uno is the brain of the RFID reader. The RFID reader is the card/tag scanner. The RFID reader cannot operate on its own.

The second component is the SQL Server. The SQL Server is the system's database. Data that does not exist in the database means that it is not registered. The SQL Server resides locally in the computer and is not hosted.

The third component of the system is the Node-RED application. The Node-Red application is a platform allows the integration of hardware, software and database. Some of the programming languages it can support are HTML, CSS, JSON, SQL, DOS batch language, JavaScript and its related framework AngularJS. All of these languages are implemented in the creation of this system. The system architecture as shown in the figure below.

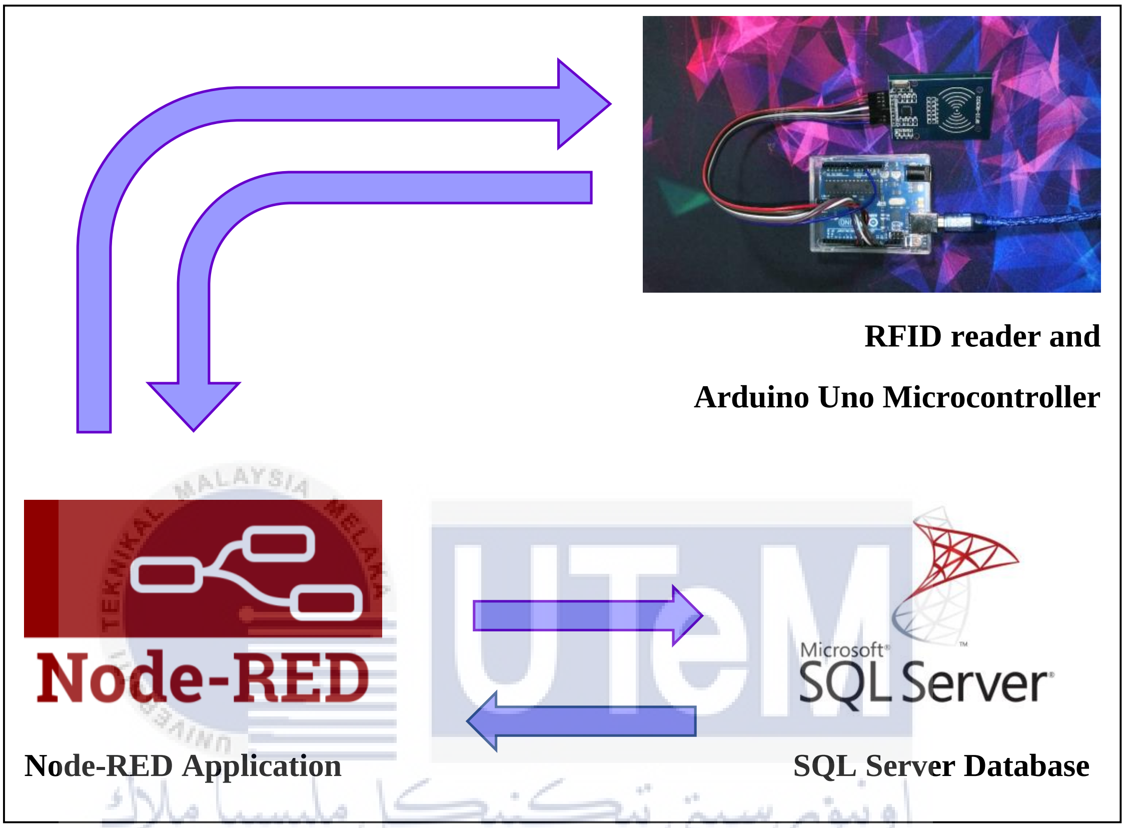


Figure 4.4: The Architecture of Monitoring and Access Control System

4.4.2 User Interface Design

The user interface design can be breakdown into 3 parts. The navigation design, the input design and the output design. The navigation designs refer to design of navigation bars or navigation buttons implemented in the proposed system. The input designs refers to the design for input fields like number, text, drop down, etc.. The input design also include the validation rule of each input field. The output design refers to the design of the proposed system with reports and graphs that can be classified in terms of periodically or ad-hoc basis. The designs for each part are as shown below:

a) Navigation Design

i. Navigation side bar. The navigation side bar is present throughout all the screens and it contains a single navigation property which is to the Homepage.

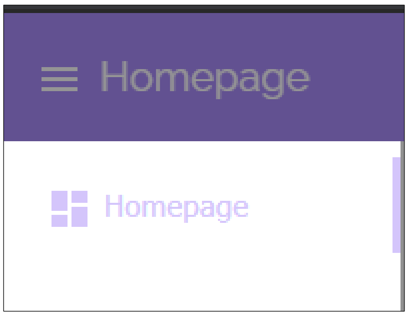


Figure 4.5: Navigation Design: Navigation side bar

ii. Admin navigation tiles. The admin navigation tiles consist of 6 navigation buttons which redirect to 6 different modules that the system offers for the admin account.

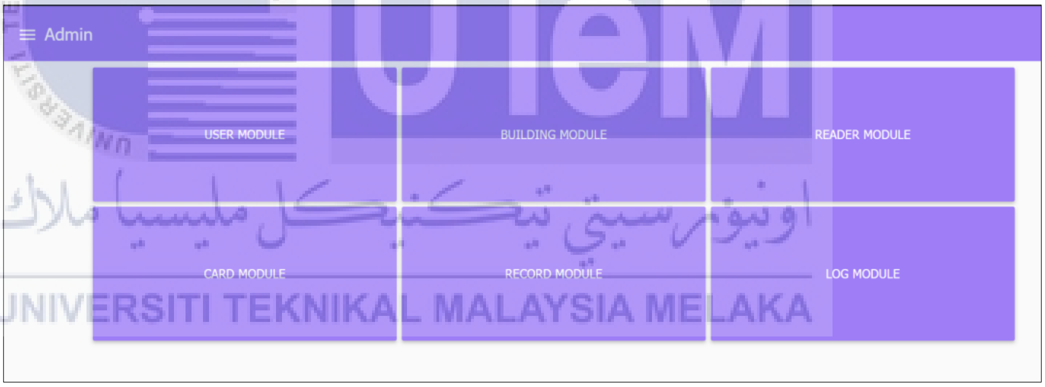
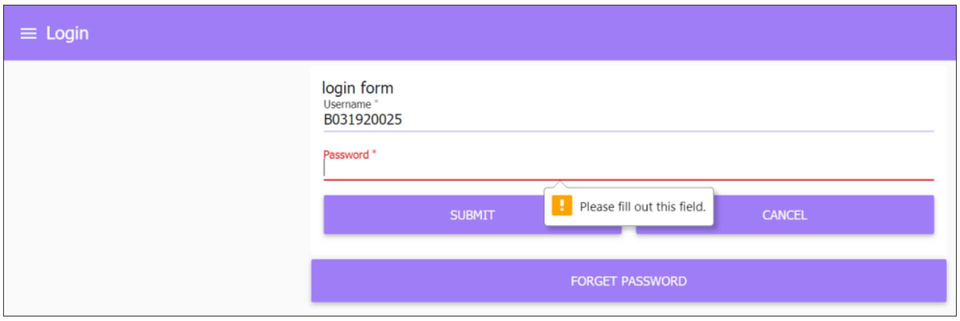


Figure 4.6: Navigation Design: Admin navigation tiles

b) Input Design

i) Sign in input. The sign in input design consists of two fields which are the username field and the password field. The validation for these fields are that they all are required fields thus failure of providing data to these field will prevent the browser form sending the data.



login form

Username *

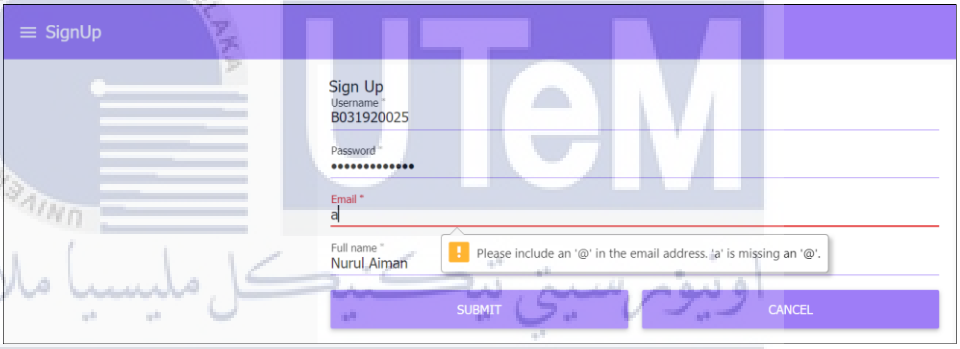
B031920025

Password *

Please fill out this field.

Figure 4.7: Input Design: Sign in input

ii) Sign up input. The sign up input design consist of four fields which are the username field, password field, email field and a full name field. The validation for these fields are that they all are required fields. The email field checks the data for an email format. Failure of providing the correct data to this field will prevent the browser from sending the data.



Sign Up

Username *

B031920025

Password *

•••••••••••••

Email *

a

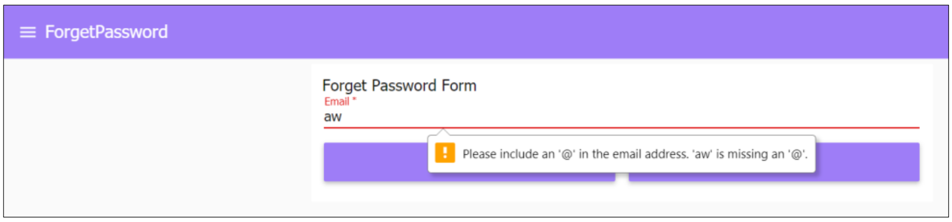
Full name *

Nurul Aiman

Please include an '@' in the email address. 'a' is missing an '@'.

Figure 4.8: Input Design: Sign up input

iii) Forget password input. The forget password input design consist of a single field which is the email field. The email field is a required field that checks the data for an email format. Failure of providing data to this field will prevent the browser from sending the data.



Forget Password Form

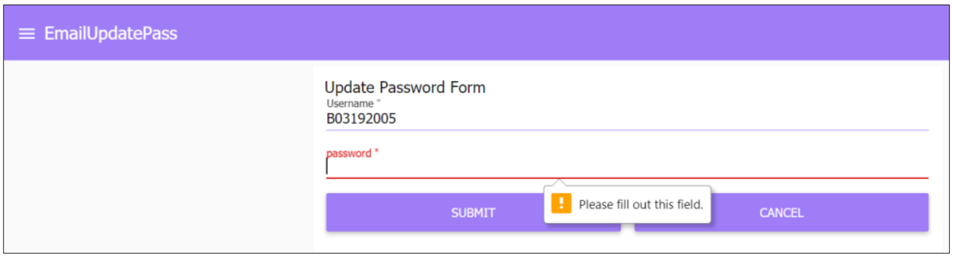
Email *

aw

Please include an '@' in the email address. 'aw' is missing an '@'.

Figure 4.9: Input Design: Forget password input

iv) Update password input. The update password input design consist of two fields which are the username field and password field The validation for these fields are that they all are required fields thus failure of providing data to these field will prevent the browser form sending the data.


EmailUpdatePass

Update Password Form

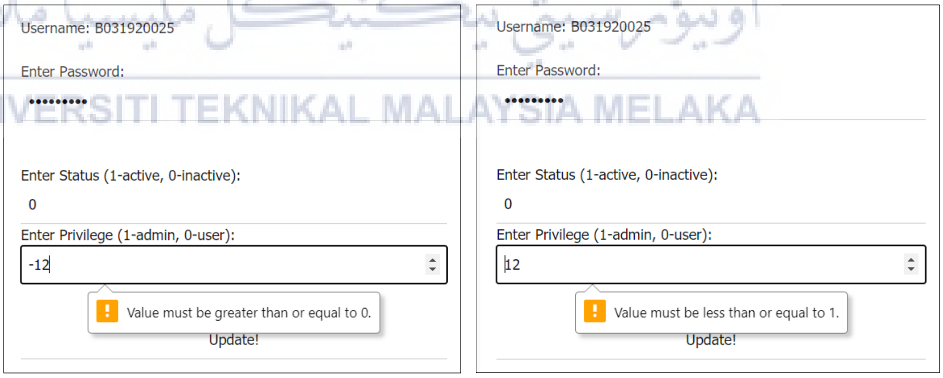
Username *
B03192005

password *

Please fill out this field.

Figure 4.10: Input Design: Update password input

v) Modify user input. The modify user input design consist of three fields which are the password field, status field and privilege field The validation for these fields are that they all are required. The status and privilege field is a number field with a minimum range of 0 and maximum of 1. Failure of providing the correct data to these fields will prevent the browser from sending the data.


Username: B031920025

Enter Password:
•••••••••

Enter Status (1-active, 0-inactive):
0

Enter Privilege (1-admin, 0-user):
-12

Value must be greater than or equal to 0.

Username: B031920025

Enter Password:
•••••••••

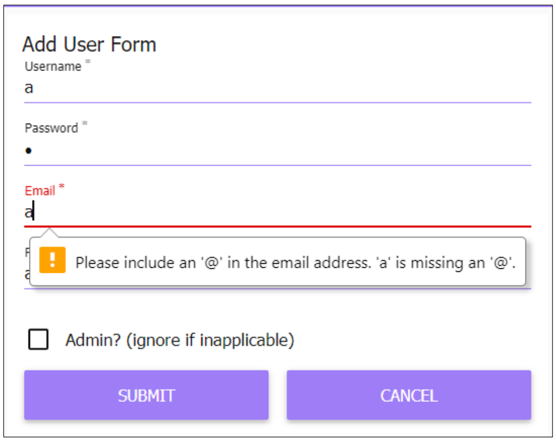
Enter Status (1-active, 0-inactive):
0

Enter Privilege (1-admin, 0-user):
12

Value must be less than or equal to 1.

Figure 4.11: Input Design: Modify user input

vi) Add user input. The add user input design consist of 4 fields and a checkbox which are the username field, password field, email field, full name field and privilege checkbox. The validation for four of these fields are that they are required fields except the checkbox. The email field checks the data for an email format. Failure of providing the correct data to these fields will prevent the browser from sending the data.



Add User Form

Username *
a

Password *
•

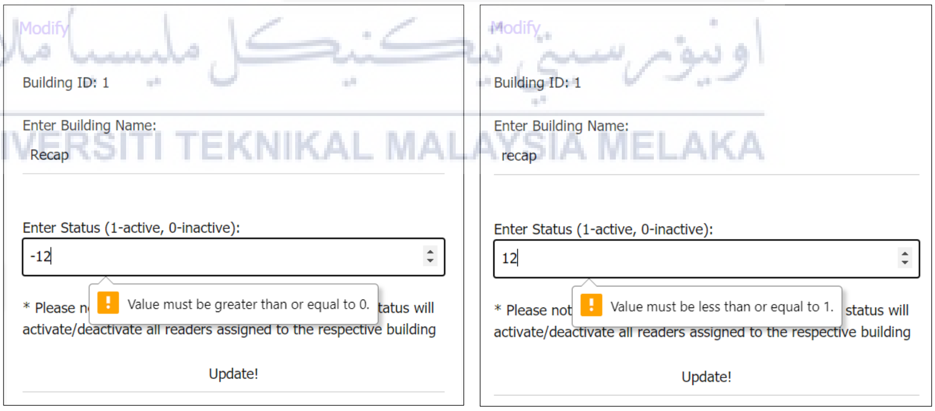
Email *
a

Please include an '@' in the email address. 'a' is missing an '@'.

☐ Admin? (ignore if inapplicable)

Figure 4.12: Input Design: Add user input

vii) Modify building input. The modify building input design consist of two fields which are the building name field and status field. The validation for these fields are that they all are required. The status field is a number field with a minimum range of 0 and maximum of 1. Failure of providing the correct data to these fields will prevent the browser from sending the data.



Modify

Building ID: 1

Enter Building Name:
Recap

Enter Status (1-active, 0-inactive):
-12

Value must be greater than or equal to 0.

* Please n tatus will activate/deactivate all readers assigned to the respective building

Modify

Building ID: 1

Enter Building Name:
recap

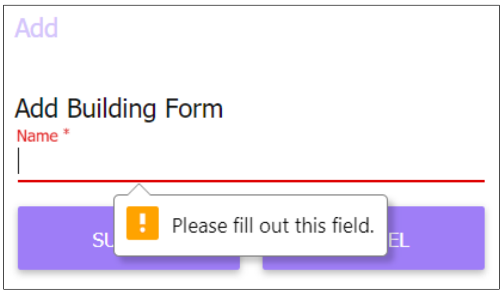
Enter Status (1-active, 0-inactive):
12

Value must be less than or equal to 1.

* Please not status will activate/deactivate all readers assigned to the respective building

Figure 4.13: Input Design: Modify building input

viii) Add building input design. The add building input design consist of a single field which is the name field The validation this field is that it is a required field. Failure of providing the data to this field will prevent the browser from sending the data.



Add

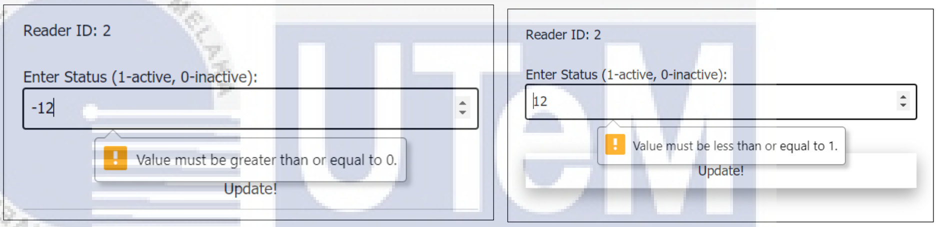
Add Building Form

Name *

Please fill out this field.

Figure 4.14: Input Design: Add building input

ix) Modify reader input. The modify reader input design of a single field which is the status field. The validation for the status field is that it is a required field. The status field is also a number field with a minimum range of 0 and maximum of 1. Failure of providing the correct data to this field will prevent the browser from sending the data.



Reader ID: 2

Enter Status (1-active, 0-inactive): -12

Value must be greater than or equal to 0.

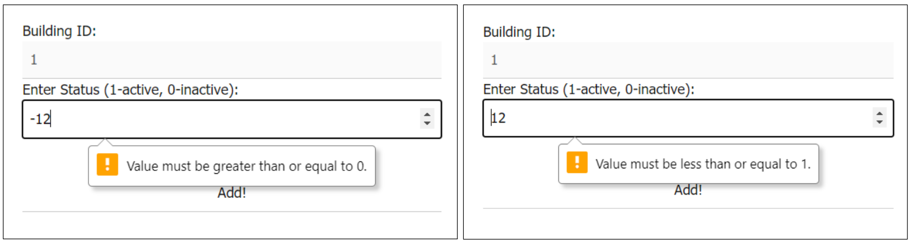
Reader ID: 2

Enter Status (1-active, 0-inactive): 12

Value must be less than or equal to 1.

Figure 4.15: Input Design: Modify reader input

x) Add reader input. The add reader input design consist of a single field which is the status field The validation for the status field is that it is a required field. The status field is also a number field with a minimum range of 0 and maximum of 1. Failure of providing the correct data to this field will prevent the browser from sending the data.



Building ID: 1

Enter Status (1-active, 0-inactive): -12

Value must be greater than or equal to 0.

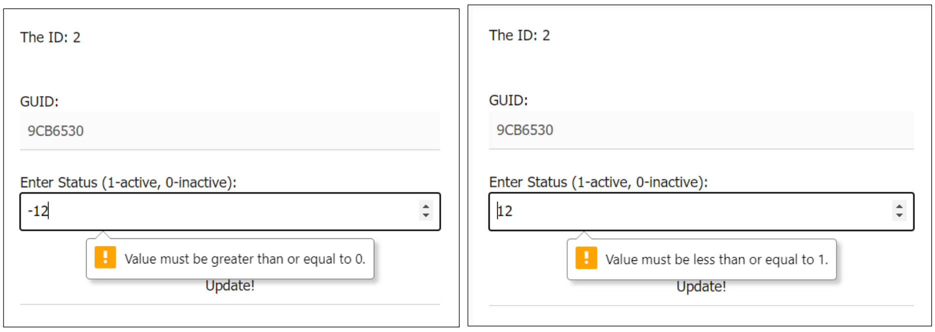
Building ID: 1

Enter Status (1-active, 0-inactive): 12

Value must be less than or equal to 1.

Figure 4.16: Input Design: Add reader input

xi) Modify card input. The modify card input design of a single field which is the status field. The validation for the status field is that it is a required field. The status field is also a number field with a minimum range of 0 and maximum of 1. Failure of providing the correct data to this field will prevent the browser from sending the data.



The ID: 2

GUID:
9CB6530

Enter Status (1-active, 0-inactive):
-12

Value must be greater than or equal to 0.

Update!

The ID: 2

GUID:
9CB6530

Enter Status (1-active, 0-inactive):
12

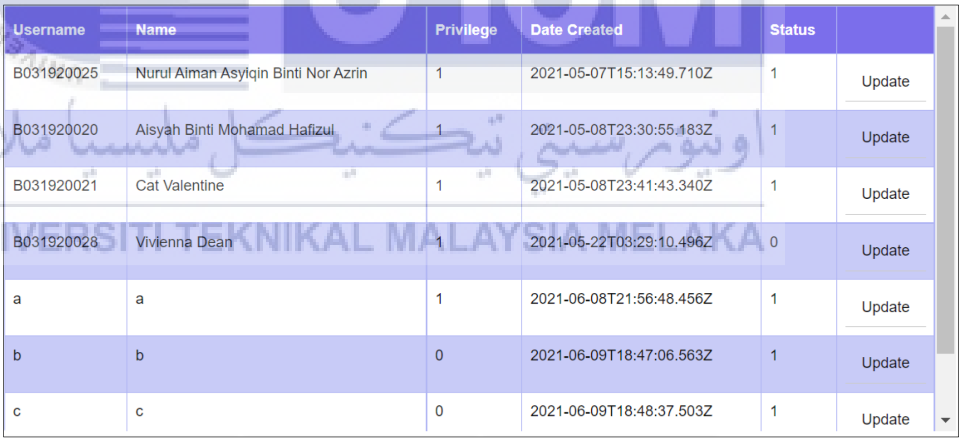
Value must be less than or equal to 1.

Update!

Figure 4.17: Input Design: Modify card input

c) Output Design

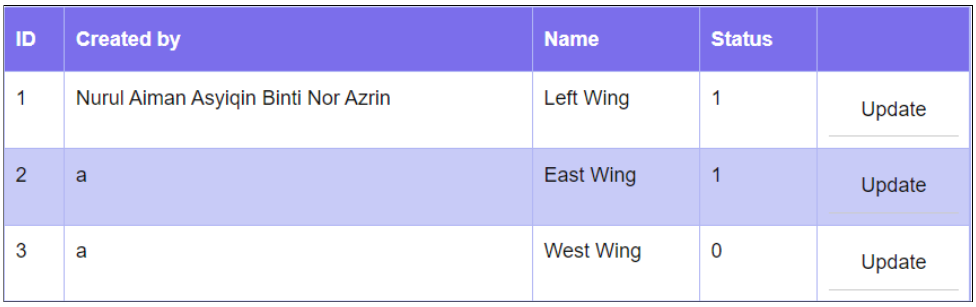
i) User list output. The user list output design displays the list of registered users in the form of a table.



Username	Name	Privilege	Date Created	Status	
B031920025	Nurul Aiman Asyiqin Binti Nor Azrin	1	2021-05-07T15:13:49.710Z	1	Update
B031920020	Aisyah Binti Mohamad Hafizul	1	2021-05-08T23:30:55.183Z	1	Update
B031920021	Cat Valentine	1	2021-05-08T23:41:43.340Z	1	Update
B031920028	Vivienna Dean	1	2021-05-22T03:29:10.496Z	0	Update
a	a	1	2021-06-08T21:56:48.456Z	1	Update
b	b	0	2021-06-09T18:47:06.563Z	1	Update
c	c	0	2021-06-09T18:48:37.503Z	1	Update

Figure 4.18: Output Design: User list output

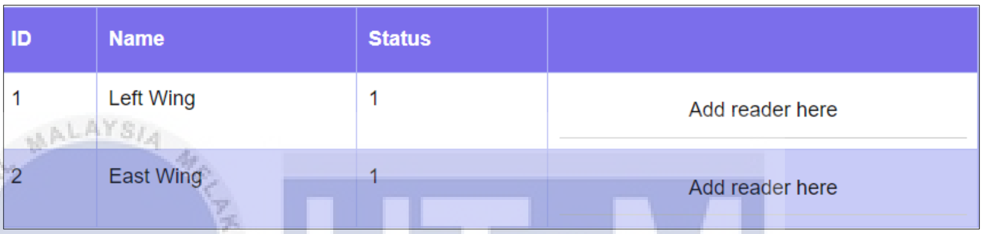
ii) Building list output. The building list output design displays the list of registered building in the form of a table



ID	Created by	Name	Status	
1	Nurul Aiman Asyiqin Binti Nor Azrin	Left Wing	1	Update
2	a	East Wing	1	Update
3	a	West Wing	0	Update

Figure 4.19: Output Design: Building list output

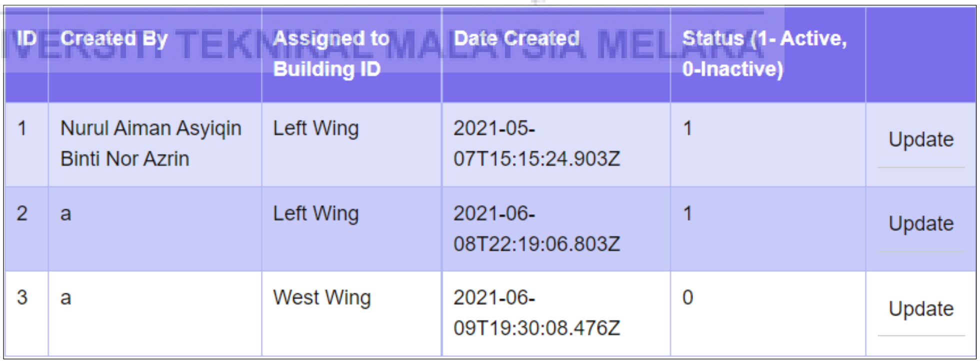
iii) Available building list output. The available building list output design displays the list of registered building that are active in the form of a table.



ID	Name	Status	
1	Left Wing	1	Add reader here
2	East Wing	1	Add reader here

Figure 4.20: Output Design: Available building list output

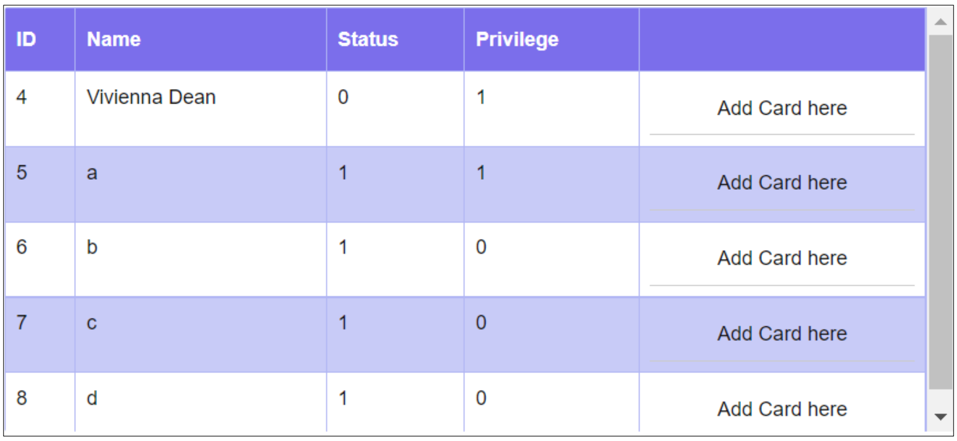
iv) Reader list output. The reader list output design displays the list of registered RFID readers in the form of a table.



ID	Created By	Assigned to Building ID	Date Created	Status (1- Active, 0-Inactive)	
1	Nurul Aiman Asyiqin Binti Nor Azrin	Left Wing	2021-05-07T15:15:24.903Z	1	Update
2	a	Left Wing	2021-06-08T22:19:06.803Z	1	Update
3	a	West Wing	2021-06-09T19:30:08.476Z	0	Update

Figure 4.21: Output Design: Reader list output

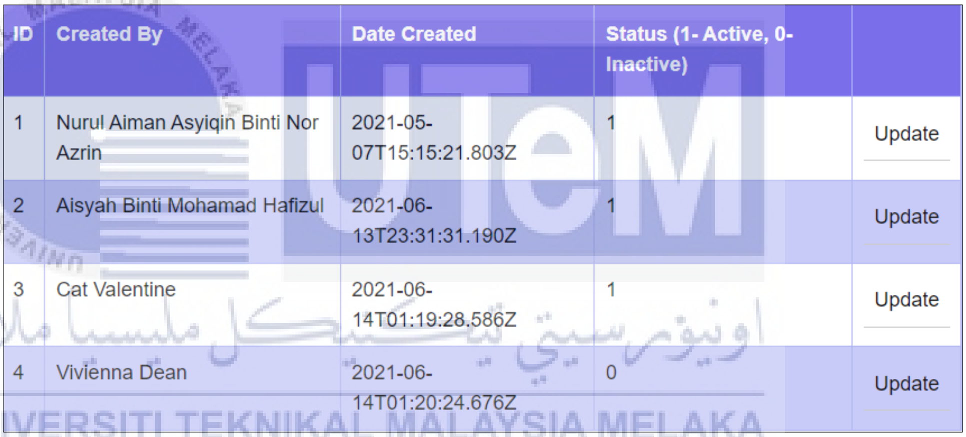
v) Unassigned user list output. The unassigned user list output design displays the list of user that have not been assigned to an RFID card in the form of a table.



ID	Name	Status	Privilege	
4	Vivienna Dean	0	1	Add Card here
5	a	1	1	Add Card here
6	b	1	0	Add Card here
7	c	1	0	Add Card here
8	d	1	0	Add Card here

Figure 4.22: Output Design: Unassigned user list output

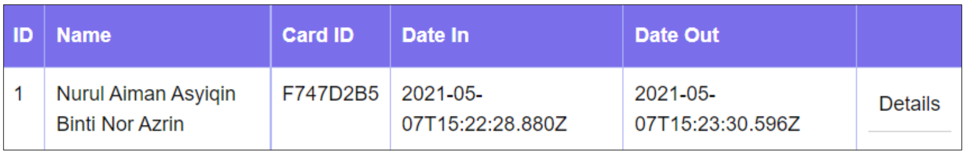
vi) Card list output. The card list output design displays the list of registered cards in the form of a table.



ID	Created By	Date Created	Status (1- Active, 0-Inactive)	
1	Nurul Aiman Asyiqin Binti Nor Azrin	2021-05-07T15:15:21.803Z	1	Update
2	Aisyah Binti Mohamad Hafizul	2021-06-13T23:31:31.190Z	1	Update
3	Cat Valentine	2021-06-14T01:19:28.586Z	1	Update
4	Vivienna Dean	2021-06-14T01:20:24.676Z	0	Update

Figure 4.23: Output Design: Card list output

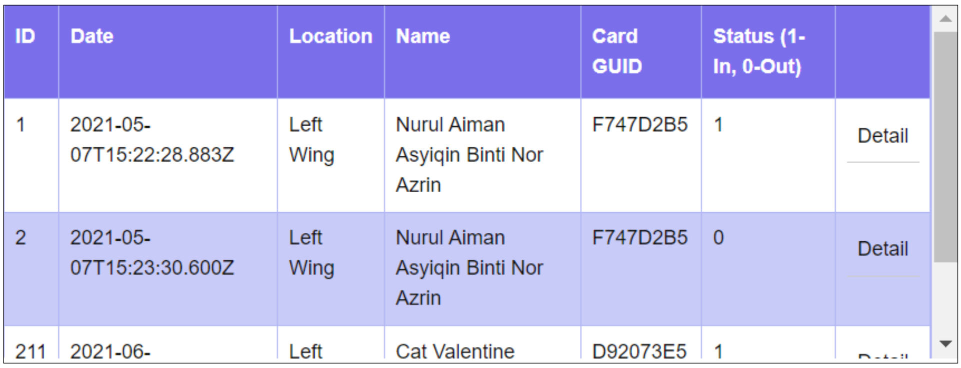
vii) Record list output. The record list output design displays the list of record in the form of a table.



ID	Name	Card ID	Date In	Date Out	
1	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	2021-05-07T15:22:28.880Z	2021-05-07T15:23:30.596Z	Details

Figure 4.24: Output Design: Record list output

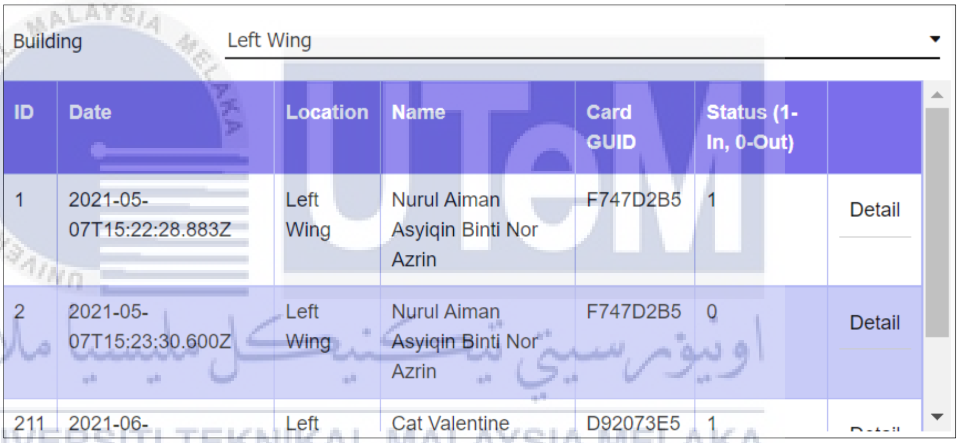
viii) Logs list output. The logs list output design displays the list of all record logs in the form of a table.



ID	Date	Location	Name	Card GUID	Status (1-In, 0-Out)	
1	2021-05-07T15:22:28.883Z	Left Wing	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	1	Detail
2	2021-05-07T15:23:30.600Z	Left Wing	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	0	Detail
211	2021-06-	Left	Cat Valentine	D92073E5	1	Detail

Figure 4.25: Output Design: Logs list output

ix) Log list by building output. The log list by building output design displays the list of record logs based on the searched building name in the form of a table.

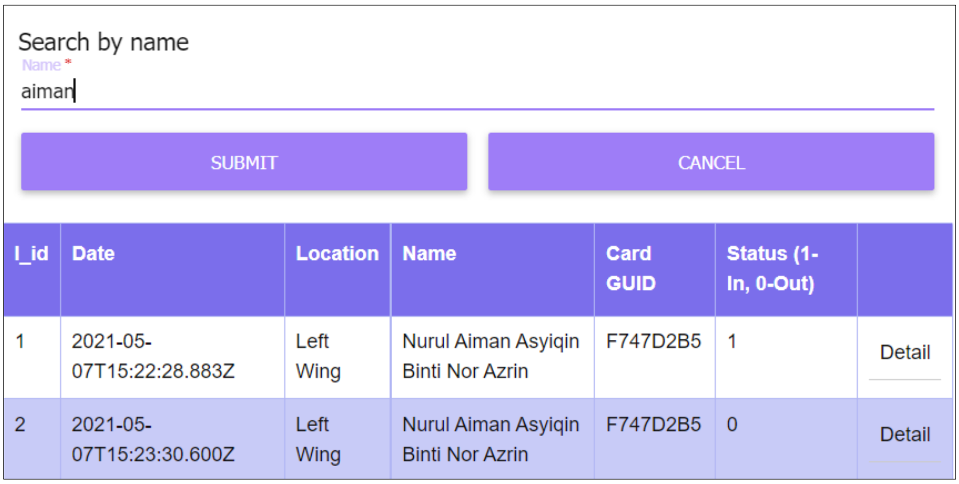


Building: Left Wing

ID	Date	Location	Name	Card GUID	Status (1-In, 0-Out)	
1	2021-05-07T15:22:28.883Z	Left Wing	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	1	Detail
2	2021-05-07T15:23:30.600Z	Left Wing	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	0	Detail
211	2021-06-	Left	Cat Valentine	D92073E5	1	Detail

Figure 4.26: Output Design: Log list by building output

x) Log list by name output. The log list by building output design displays the list of record logs based on the searched name of the registered card owner in the form of a table.



Search by name

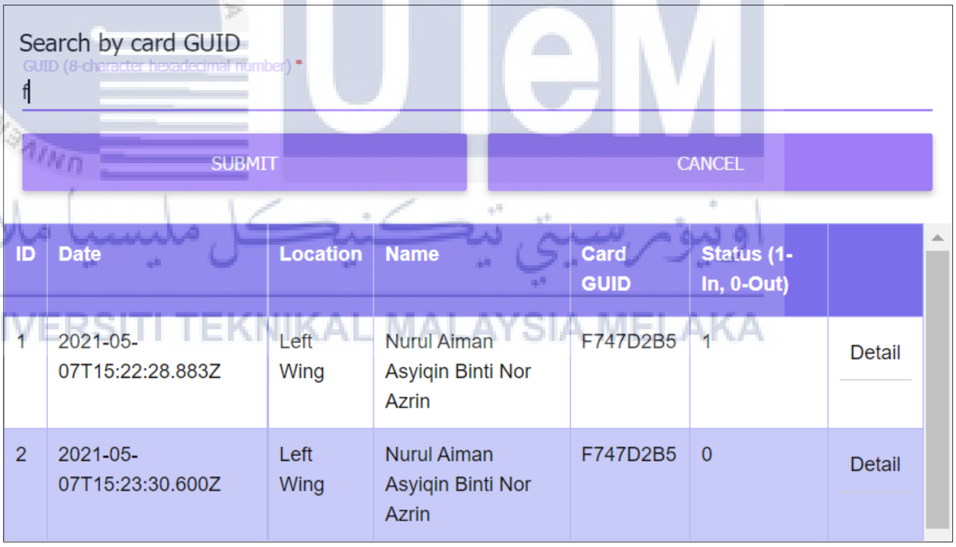
Name *

aiman

I_id	Date	Location	Name	Card GUID	Status (1- In, 0-Out)	
1	2021-05-07T15:22:28.883Z	Left Wing	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	1	Detail
2	2021-05-07T15:23:30.600Z	Left Wing	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	0	Detail

Figure 4.27: Output Design: Log list by name output

xi) Log list by card GUID output. The logs list by card GUID output design displays the list of record logs based on the searched card GUID in the form of a table.



Search by card GUID

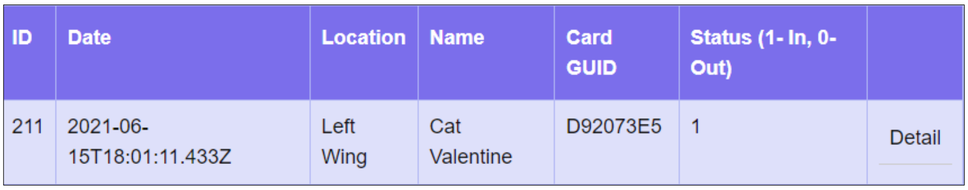
GUID (8-character hexadecimal number) *

f

ID	Date	Location	Name	Card GUID	Status (1- In, 0-Out)	
1	2021-05-07T15:22:28.883Z	Left Wing	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	1	Detail
2	2021-05-07T15:23:30.600Z	Left Wing	Nurul Aiman Asyiqin Binti Nor Azrin	F747D2B5	0	Detail

Figure 4.28: Output Design: Log list by card GUID output

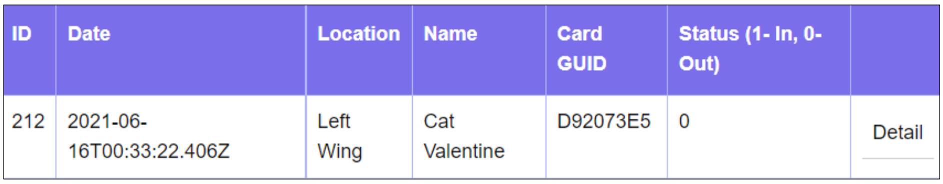
xii) Current-dated log list output. The current-dated log list output design displays the list of record logs based on the current date it is accessed.



ID	Date	Location	Name	Card GUID	Status (1- In, 0-Out)	
211	2021-06-15T18:01:11.433Z	Left Wing	Cat Valentine	D92073E5	1	Detail

Figure 4.29: Output Design: Current-dated log list output

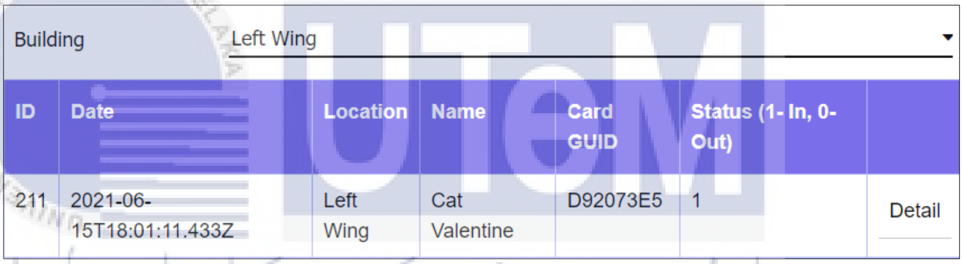
xiii) User current-dated log list output. The user current-dated log list output design displays the list of record logs belong to a specific user based on the current date it is accessed.



ID	Date	Location	Name	Card GUID	Status (1- In, 0- Out)	
212	2021-06-16T00:33:22.406Z	Left Wing	Cat Valentine	D92073E5	0	Detail

Figure 4.30: Output Design: User current-dated log list output

xiv) User log list by building output. The user log list by building output design displays the list of record logs belong to a specific user based on the searched building name in the form of a table.

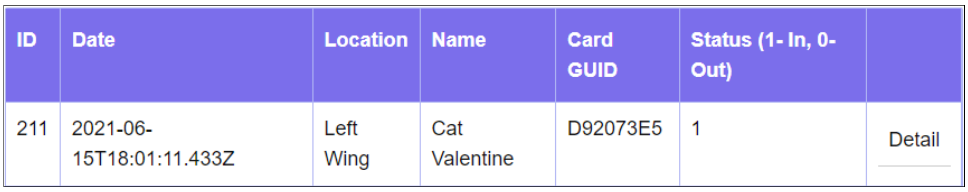


Building Left Wing

ID	Date	Location	Name	Card GUID	Status (1- In, 0- Out)	
211	2021-06-15T18:01:11.433Z	Left Wing	Cat Valentine	D92073E5	1	Detail

Figure 4.31: Output Design: User log list by building output

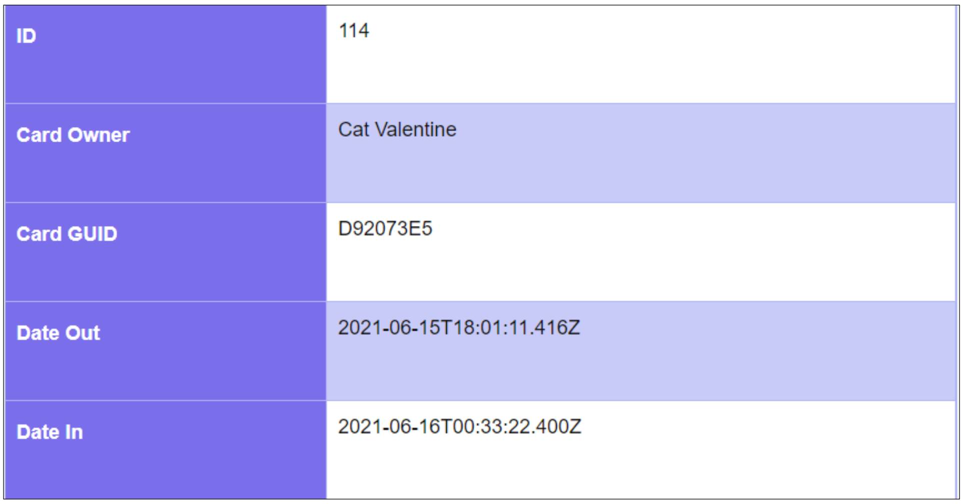
xv) User log list output. The user log list output design displays the list of all record logs belong to a specific user in the form of a table.



ID	Date	Location	Name	Card GUID	Status (1- In, 0- Out)	
211	2021-06-15T18:01:11.433Z	Left Wing	Cat Valentine	D92073E5	1	Detail

Figure 4.32: Output Design: User log list by building output

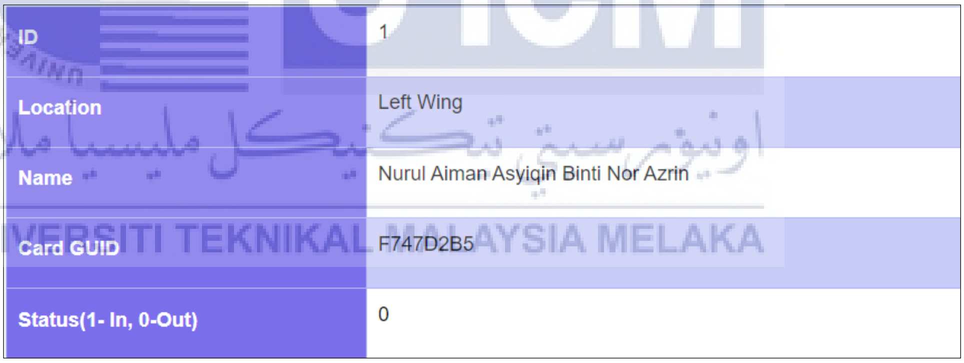
xvi) Record details output. The record details output design displays the details of a record in the form of a table.



ID	114
Card Owner	Cat Valentine
Card GUID	D92073E5
Date Out	2021-06-15T18:01:11.416Z
Date In	2021-06-16T00:33:22.400Z

Figure 4.33: Output Design: Record details output

xvii) Log details output. The Log details output design displays the details of a log in the form of a table.



ID	1
Location	Left Wing
Name	Nurul Aiman Asyiqin Binti Nor Azrin
Card GUID	F747D2B5
Status(1- In, 0-Out)	0

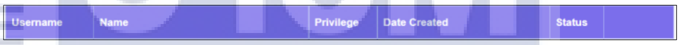
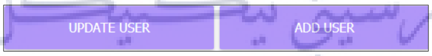
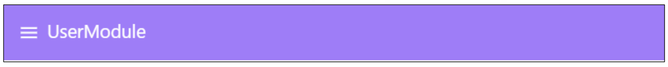
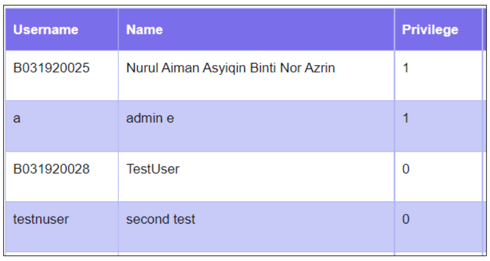
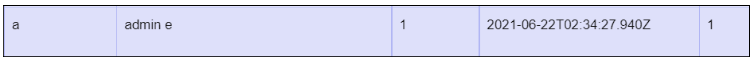
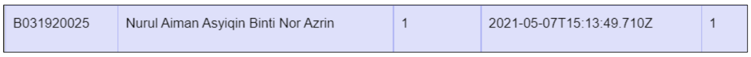
Figure 4.34: Output Design: Log details output

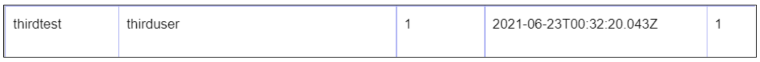
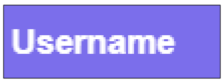
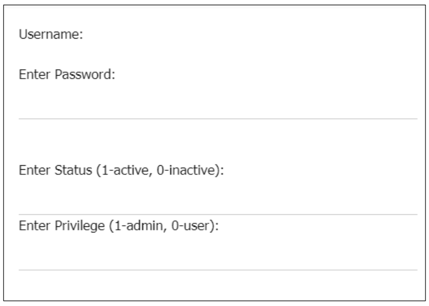
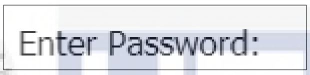
System Design

The proposed system consist of 2 parts which are hardware components and software application. The hardware components of the system are the RFID readers, the Arduino Uno board, USB cable type A/B and male to female jumper wires. The developed software application is a dashboard that consist of logs and modification features. The log features are available for both administrators and normal users. However, the modification features are reserved only for administrators.

The chosen theme for the proposed system is purple or specifically the colour scheme related to the code '#9e7dff' in hexadecimal digits. Hence the shorter name of the project is called the PurpleProject. These colours are used for the readability of the system. The following table shows the colour schemes used in this system.

Table 4.7: Proposed System Colour Scheme

Colour Codes	Colour	Examples
#7b6eeb	🟪	• Table headers  Username Name Privilege Date Created Status
#9e7dff	🟪	• Buttons  UPDATE USER ADD USER • Headers  ≡ UserModule
#b1b6f4	🟪	• Table borders  Username Name Privilege B031920025 Nurul Aiman Asyiqin Binti Nor Azrin 1 a admin e 1 B031920028 TestUser 0 testnuser second test 0
#c8cbf7	🟪	• Even number table rows (2, 4, 6, 8, …etc.)  a admin e 1 2021-06-22T02:34:27.940Z 1
#dee0fa	🟪	• Table rows when hovered over with a mouse  B031920025 Nurul Aiman Asyiqin Binti Nor Azrin 1 2021-05-07T15:13:49.710Z 1

#ffffff	□	- Odd number table rows (1, 3, 5, 7, …etc.)  thirdtest thirduser 1 2021-06-23T00:32:20.043Z 1 - Font colour  Username - Form background colour  Username: Enter Password: Enter Status (1-active, 0-inactive): Enter Privilege (1-admin, 0-user):
#111111	■	- Font colour  Enter Password:

Physical Design

The physical design of the prototype is take from several angles to illustrate and display the physical state of the product prototype. The design are as shown in these figures below.

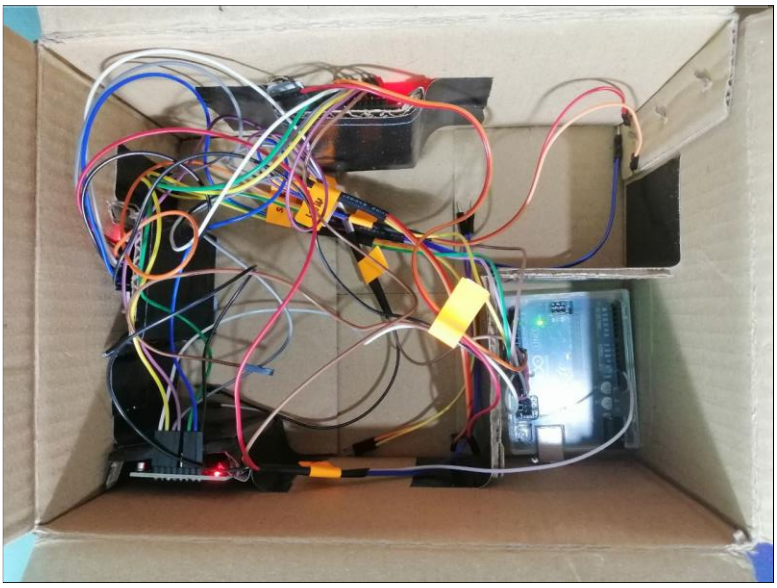


Figure 4.36 : Physical Design – Product contents

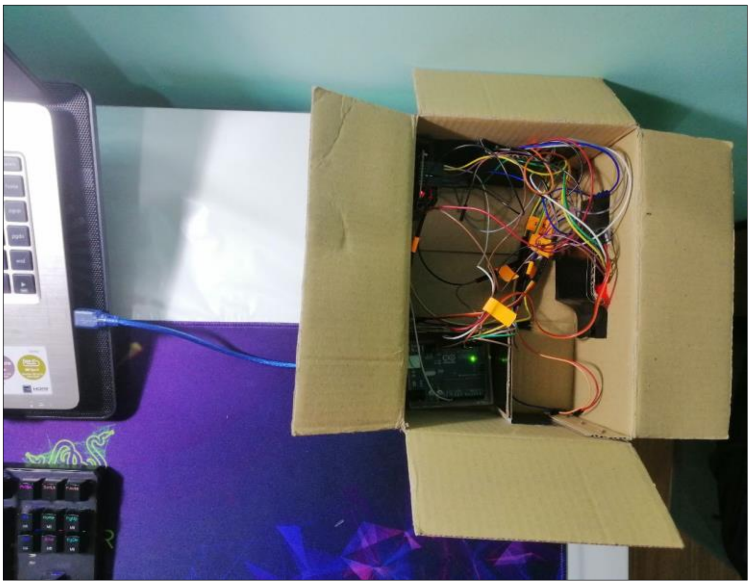


Figure 4.37 : Physical Design – Product top view (opened lid)

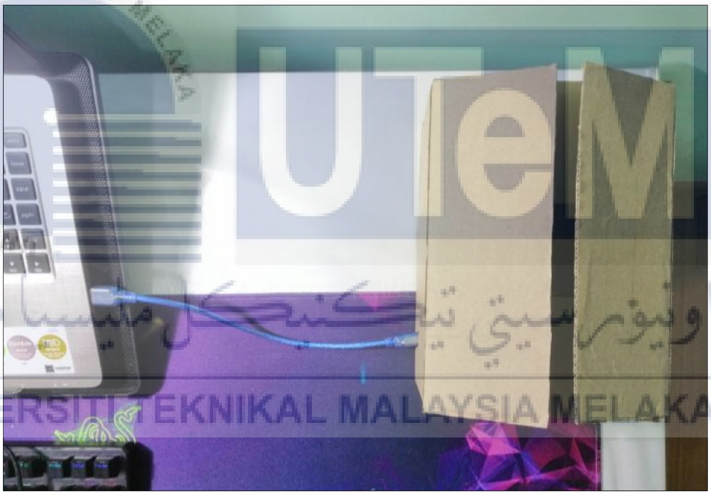


Figure 4.38 : Physical Design – Product top view (closed lid)

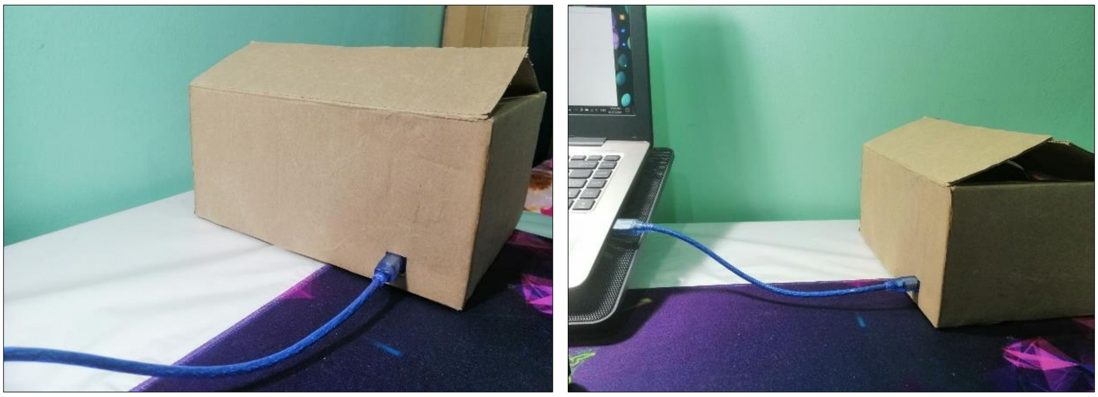


Figure 4.39 : Physical Design – Product side view

Experimental Environment

The proposed system was developed using mainly 3 different languages which are JavaScript, SQL and C. The system are arranged in flows connected to one another in order to perform processes. JavaScript is used along with AngularJS and JSON to retrieve, send and manipulate the data from and to the database. SQL queries was integrated within the JavaScript codes before it was sent to the database. C is a language used to communicate with the Arduino Uno board and the RFID reader. There are mainly two types of codes that is performed on the system.

Displaying data

The logic behind displaying data to the dashboard differs based on the source of the data. Data can be retrieved from either a database or a JSON/CSV(Comma Separated Values) text file.

a. Data from database

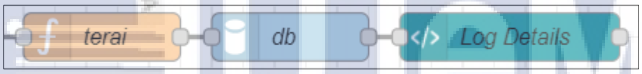


Figure 4.40 : Flow Snippets – Data from database

The orange function node named ‘terai’ consist of SQL queries that is written within the JavaScript language. The codes used are as shown below.

```
var b = msg.payload.b +1;

var pld;

pld = "select logs.l_id, logs.rec_id, logs.r_id, logs.c_id, logs.date_in,
logs.status, users.u_id, users.name as 'uname', building.b_id,
building.name as 'bname', card.guid"

pld = pld + " from users join card on users.u_id =  card.u_id"

pld = pld + " join logs on card.c_id = logs.c_id"

pld = pld + " join record on logs.rec_id  = record.rec_id"

pld = pld + " join reader on record.r_id  = reader.r_id"

pld = pld + " join building on reader.b_id = building.b_id"

pld = pld + " where l_id = (Select l_id From (Select Row_Number() Over
(Order By l_id) As RowNum, * From logs) t2"

pld = pld + " where RowNum= "+ b +")"

msg.payload = pld;

return msg;
```

The blue database node named ‘db’ will process the query and return the data in msg.payload as the output. The data is an array in the format of a JavaScript object. This object will be used to display into a table using the green template node named ‘Log Details’. The code inside of the template node is as shown below.

```
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Raleway">

<style>

#table {

  font-family: Arial, Helvetica, sans-serif;

  border-collapse: collapse;

  width: 100%;

}

#table td, #table th {

  border: 1px solid #b1b6f4;

  padding: 8px;

}

#table tr:nth-child(even){background-color: #c8cbf7;}

#table tr:hover {background-color: #dee0fa;}

#table th {

  padding-top: 12px;

  padding-bottom: 12px;

  text-align: left;

  background-color: #7b6eeb;

  color: white;

}

</style>

<table id="table" class ="w3-table" border="1"
style="width:100%;height:100%;" ng-repeat="row in msg.payload track by
$index" margin = "5px">

<tr><th>ID</th><td>{{row.rec_id}}</td></tr>

<tr><th>Location</th><td>{{row.bname}}</td></tr>

<tr><th>Name</th><td>{{row.uname}}</td></tr>

<tr><th>Card GUID</th><td>{{row.guid}}</td></tr>

<tr><th>Status(1- In, 0-Out)</th><td>{{row.status}}</td></tr>

</table>
```

b. Data from text files

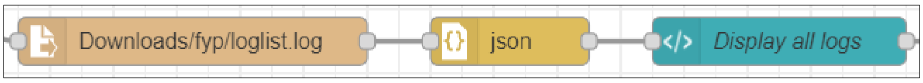


Figure 4.41 : Flow Snippets – Data from text files

The brown file node named ‘Downloads/fyp/loglist.log’ is links to an existing text file located in that path. The file contains data taken from database that are saved onto that file. The data in that file could also be taken from readings of the RFID reader. The data can either be in a CSV file format or a JSON String. In this case, the data in that file node is in the format of a JSON string. The dark yellow JSON node is a converter that converts data in the file node into a JavaScript object. This object will be used to display into a table using the green template node named ‘Display all logs’ similar to the earlier template node as explained before. The contents of the loglist.log file is as shown below.

```
[{"l_id":7,"rec_id":5,"r_id":1,"c_id":4,"date_in":"2021-06-
23T01:09:02.866Z","status":1,"u_id":4,"uname":"second
test","b_id":1,"bname":"Right Wing","guid":"9CB6530"}]
```

Inserting/Modifying Data

The logic behind inserting data differs to the type of storage that receive these data. There are 2 types of data storing platforms used in this project: a database and a text file (in the format of a JSON string or CSV). The concept of modifying data is to insert values to the database or files in order to update or modify the existing values.

a. Saving database output to a file

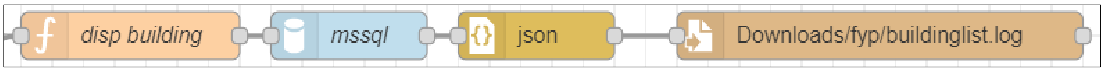


Figure 4.42 : Flow Snippets – Saving database output to a file

The orange function node named ‘disp building’ consist of an SQL query written in the format of a JavaScript string variable before sending it to the blue database node named ‘mssql’. The code inside of the function node is as shown below.

```
pld = "select b_id, users.name as 'admin', building.name as
'bname', building.status from building join users on
building.u_id = users.u_id";

msg.payload = pld

return msg;
```

The data retrieved from the database is the msg.payload which is in the form of a JavaScript object. The dark yellow JSON node is used to convert the data retrieved from the database node into a JSON string. It will then send the converted JSON string into the brown file node named ‘Downloads/fyp/buildinglist.log’. This node will find the file located in that path and inject the JSON string into it. The code inside of the text file is as shown below.

```
[{"b_id":1,"admin":"admin e","bname":"Right
Wing","status":1},{"b_id":2,"admin":"admin e","bname":"West
Wing","status":1},{"b_id":3,"admin":"admin
e","bname":"secondbuilding","status":1}]
```

b. Saving serial outputs to a file

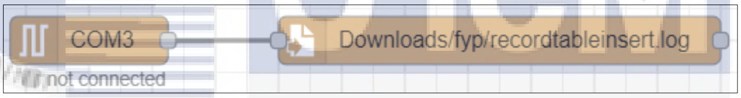


Figure 4.43 : Flow Snippets – Saving serial outputs to a file

In this project, the brown node named ‘COM3’ are a configured node to detect and read from the serial port of COM3. Serial outputs are data that has been read from the RFID reader. These data are sent to the computer using the serial ports that it is connected to. The data received from the ports can differ based on the output of the C programme that is uploaded to the Arduino Uno board. In this case, the C program will produce an output similar to a CSV file. This output will be saved in the brown file node named ‘’. The contents of the file is as shown below.

```
0,'F94917E5',9
```

4.4.3 Database Design

The system operates on six tables in the database which are the users table, the building table, the card table, the record table, the reader table and the logs table. The database design of this system will be illustrated using the entity relationship diagram (ERD) for better understanding of the system's flow.

4.4.3.1 Conceptual and Logical Database Design

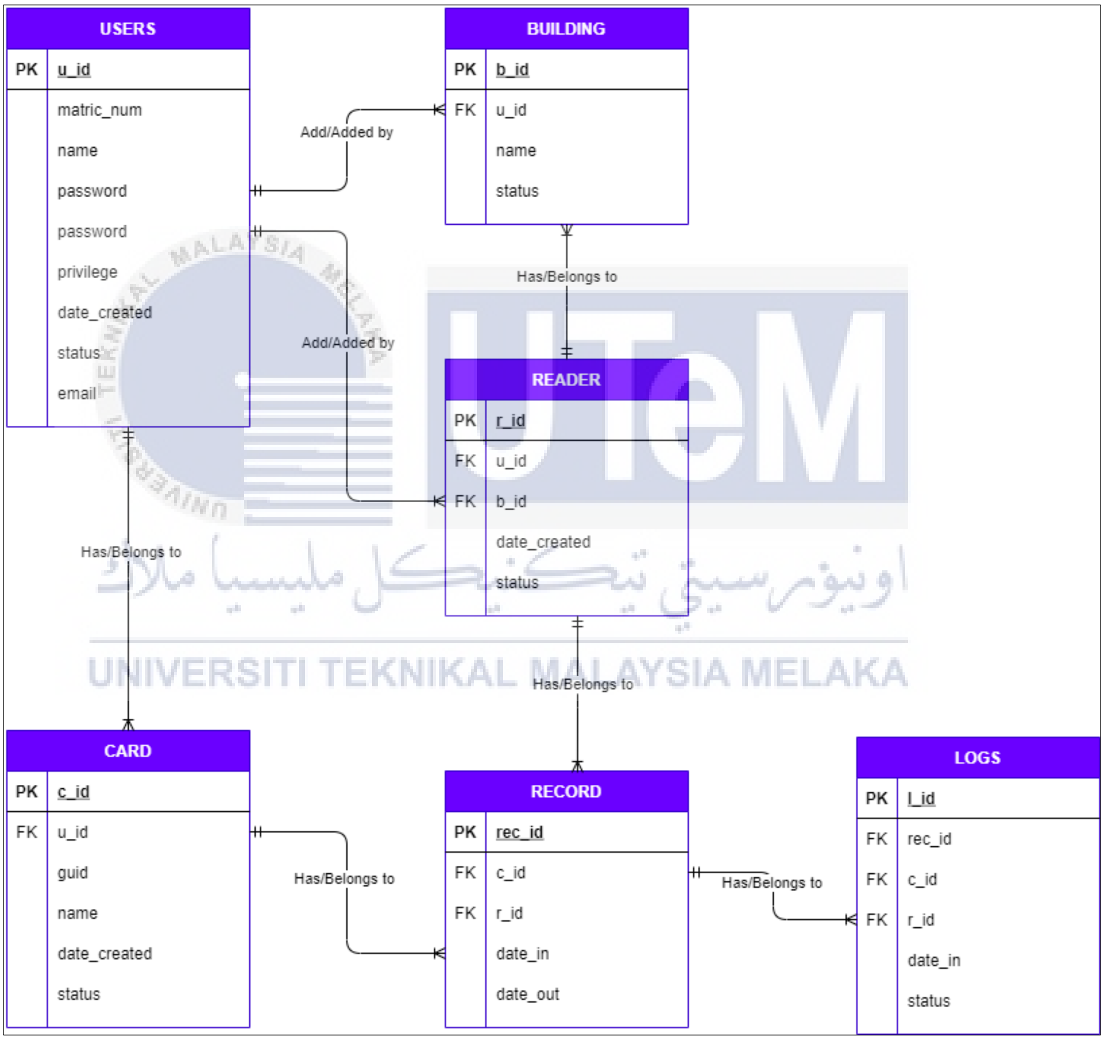


Figure 4.44: The ERD of The Monitoring and Access Control System

Based on the figure above, each tables are constructed and linked together based on a set of business rules. The business rules of the system is as explain below:

Business Rules

1. A user can have only one card at a time. A user may have more than a card if the status of their previous card are inactive (0).
2. A user can be either an admin or a normal user. Staffs and university students are considered normal users.
3. A building may have multiple readers but each reader must be assigned to a single building.
4. An unused building will automatically have unused readers.
5. A single card can communicate with many readers and a single reader can detect many cards.
6. A single record can have many logs but each log must belong to a single record

4.5 Detailed Design

The system consist of 2 main components which are the physical RFID reader and the dashboard. The dashboard consist of two types of users which are the administrators or admin and the normal users (staffs and students). Since, this is a system for the university, thus it is heavier on the admin side rather than the user side. The admin has a total of six features within this system while the users have only two.

The system will initially start by the admin registering users to the system. This feature can be further explained as the admin may create another admin or a normal user (staffs and students) to access the system. Then, the admin may register the buildings in the faculty and assign RFID readers located to that particular building in the system. The logic behind this is to detect the locality of a user.

Then, the admin may register a number of RFID-readable cards/tags to the system. Here, the admin will tap the cards on the RFID reader to detect the card's unique ID known as GUID so that this data can also be registered into the system. Afterwards, the admin will assign the registered cards to each user with a limit of one card per user.

In situations such as the RFID-readable card/tag that has already been assigned to a user is then faulty, lost or cannot be used for any other acceptable reason, the administrator must first disable the old card, before assigning a new card to the user.

Now the system can be used to detect the presence of a user in the university by tapping their RFID-readable card on the reader before entering the room/venue.

The process behind this is that, the reader will detect the card and logs it to the system and then sends it to the database. When the user leaves the room or venue, the user will need to tap again to announce their presence is leaving the room. Similarly, the reader will read that card GUID and sends it to the system before logging it into the database.

The logs of each card will be displayed in the dashboard indicating the person entering and leaving a certain room at a certain time.

4.5.1 Software Design

The proposed system consist of configuration and monitoring modules. The configuration module consists of user, building, reader, card and record modules. On the other hand, the monitoring module named log, performs real-time information in a form of timestamp log to detect presence of the user (either 'IN' or 'OUT') of from the building.

The total number of module for this proposed systems are six as mentioned above. In this section, the details of each module will be explain thoroughly to describe the process that runs within. The first module is the user module. This module is an independent module, which means that it can run on its own. However the rest of the module is dependant on this module thus, making it an important module to be configured first before configuring other modules.

This module features includes the two components. A login sub module and admin_user sub module. A login sub module refers to features like sign in, sign up and forget password. A sign in feature is when a user arrives at the system's home page and clicks on the 'Sign In' button. This will bring the user to a sign in form where they can input their username which is their matric number and their password.

These credentials must be registered by an admin or through the sign up form in the system prior to using them. The type of user is determined to be redirected into their account page based on their privilege. In the database, the privilege of '1' indicates the user is an admin. Otherwise, the privilege of '0' indicates that the user is either a normal staff or students. The sign up feature is when the user clicks the 'Sign Up' in the system's homepage.

This will redirect them to a page for them to sign up by entering their full name, their username (matric number), password and email. By default, insertion of data through this sign up form would indicate the user privilege as '0'. The forget password feature, enables the user to send an email so that they can change their password. It is located in the 'Sign In' page just below the sign in form.

The user will need to enter the email they use to register the account on the system in order to send an email. If the email is not registered then, the system will not automate the send of an email to change the account password. The admin_user sub module is only accessible after successful login as an admin account. This applies to the rest of the module that will be explained unless stated otherwise.

This submodule can be accessed by clicking on the 'User Module' tile in the admin page. This will the redirect the admin to a different page where there will be a list of registered users (both user and admin) in the form of a table. The first feature this submodule offers is user data modification. However, for the sake of system fluency and security, there are limitations to the kinds of data and modification they can see and make.

Firstly, in the list of registered users, the passwords are not shown. Even though in the database itself the passwords are hashed, the are no reason for the an admin to view that kind of data. This is because in real world implementation, a developer, an admin and a database admin are different people. Thus, there is no need for an admin to have the same level of security privilege as the rest.

The data shown in the system itself is enough for an admin to run the proposed system as per intended. The only data an admin can modify is by assigning a new password, a new status (to indicate the life of the account') or a new privilege (to upgrade or degrade an account.). To specify the account to modify, simply click on the 'Update User' button in the same row of the corresponding user and a form will be displayed on its side for the admin's perusal.

Click on the 'Update' button to update the data as per inserted in the form and wait for the list to re-update. To add a user, click on 'Add User' button and a different form will be displayed on the page. Enter the information needed such as full name, their username (matric number), password and email and an additional information that indicates the account privilege.

If the checkbox is checked then the account will be created as an admin account. Otherwise, if the checkbox is left unchecked then the account will be created as a normal user account. A 'Back' button at the bottom of the list is to return to the admin page that has tiles of other modules. The second module is a building module. This module indicates the location of which a scanner is added.

It is a simple module with features to add, modify or list out the registered buildings. This module is also located in the admin account page that can be access by clicking on the tiles with this module name on it. Similar to the user module, this module lists the building names and its status in a table form.

The status are to indicate whether the building is active or not. Its modification features are also similar to a user module where an admin can click on the 'Update Building' button to display a form to update the status of the building to either '1' if the building is active or '0', if the building is inactive. The admin can then click on the 'Update' button to submit the form and wait for the list to refresh.

Note that by updating the status will affect its dependant module. One of its dependant module is the reader module which will be explained a little later. The significance of mentioning this is because the building module is to represent locality. If the location does not exist or is inactive, naturally the reader assigned to that location is also not supposed to work.

Thus, by updating the building status to '0', which indicates the building as inactive will then cause the reader assigned to that building to be updated with the status of '0' to also indicate its inactivity. The building module also offers a feature to add a new building to the system. This can be done by clicking on the 'Add Building' button which will display an 'Add Building' form.

The admin will only need to enter the building name in the field given and click 'Add'. By default, the system will add the building and indicate it as an active building even without a reader assigned to it. A 'Back' button at the bottom of the list is to return to the admin page that has tiles of other modules. The third module is the reader module. This module is also another simple module located in the admin page.

This module has an add, modify and list features in regards to the RFID readers implemented in the system. By default, when an RFID reader is connected to the Arduino Uno that has been injected with the codes, the Arduino does not detect the presence of the reader by a unique ID of the reader. The RFID reader is detected based on which SDA or also called as SS pins on the RFID reader is connected to the Arduino board ports.

The ports that can be used to connect the reader ranges from 2 to 9. However, the codes uploaded to the Arduino Uno has been developed to add readers starting from port 9 to 2, where port 9 on the Arduino Uno indicate that its the first reader, port 8 as the second reader and port 7 as the third reader. The Figure below shows the logical representation of connection between the RFID reader SDA pin to Arduino Uno ports on the board.

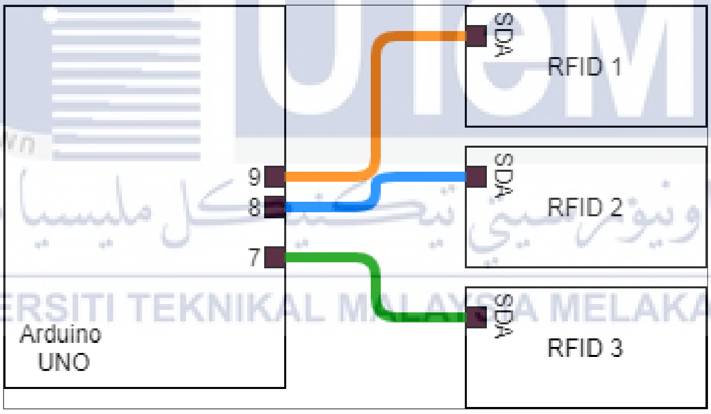


Figure 4.45: SDA-Port connection of RFID to Arduino

One Arduino board can support well up to 5 readers. However, the code currently uploaded on the Arduino Uno is able to support 3 readers which are specified on the ports as mentioned above. In the system, the reader module is for the admin to register the readers only when they exist. The admin can add the reader by first choosing the building it wants to assign to within the list that is shown when the admin clicks on ‘Add Reader’ button.

Simply choose the building by clicking ‘Add Reader Here’ and add status to the reader before clicking ‘Add’ button on the new display. The Reader is added once the list updates and shows the newly added reader. The admin can modify the reader’s status by clicking ‘Modify Reader to display a form. Make changes to reader and click ‘update’. A ‘Back’ button at the bottom of the list is to return to the admin page that has tiles of other modules.

The fourth module is the card module. The card module consist of features like listing registered card, adding cards and assigning them to a registered user (admin and normal user) and modifying the status of the card. The list of registered cards are displayed in table form. The status modification feature can be accessed by clicking the ‘Modify Card’ button and change the status of the card in the newly displayed form before clicking ‘Update’ to update the status.

The ‘Add Card’ button will redirect to a new page where there will be a list of users without cards assigned to them. Then, the admin can choose any user to assign the cards to and click on ‘Add Card Here’ before tapping the card to the RFID reader. This will add the card to the assigned user. Click ‘Back’ to go back to the page when it displays the list of cards and wait for the data to be added.

The fifth module is the record module. This module is the last configuration module. This module is to imitate the actual detection process of when a user taps their card onto the reader at the entrance of the room. This module has only has a single feature adding a record of registered cards that has been tapped onto the reader. The record is added based on existing record in the database.

If there is a record of the a card that has been added with the status of ‘IN’ or ‘1’ but has not tapped out, the next tap will log (with the current timestamp of when the record is added in the database) the card’s tap out which is indicated with the status of the card at the room/building/location as ‘0’. However if there are no un-tapped out records, a new line of record will be added. There are three types of scenarios based on the card/tag type that is catered in this module.

The first scenario is when the card/tag is registered and is active. If the card/tag meets these two condition, then the card details will be inserted into the database. The second scenario is when the card/tag is registered and is inactive. In this case, the system will notify the administrator that the card tapped is a bad card. The logic behind this is that an inactive card is either stolen/lost/damaged or the user that owns the card is no longer allowed to have access to the room/building. The third scenario is when the card/tag tapped is an unregistered card/tag. In this situation, the system will notify the admin that the card tapped is an unknown card.

The last module which is the only monitoring module is the log module. The log module consist of different types of log lists. After clicking the 'Log Module' tile in the admin page, the system will redirect to a new page which will display a menu of buttons. These buttons indicate the type of logs the system will produce. The kinds of logs can be generated based on the type below:

1. All – All available logs in the database.
2. Building - Logs based on the location of the reader that has been tapped onto
3. Name – Logs based on the card that belongs to a specific owner that has been tapped onto any reader
4. Card GUID – Logs of a specific card that has been tapped onto any reader
5. Today – Logs of card that has been tapped onto any reader on the current date the logs is accessed

Additionally, on every page of a log, the logs can be printed into a PDF file format for their perusal.

For a normal user account, the user can look at three types of logs that only belongs to them. After logging into the system using user credentials, the system will redirect to a new page which will display a menu of buttons. These buttons indicate the type of logs the system will produce. The kinds of logs can be generated based on the type below:

1. All – All available logs that belongs to the user in the database.
2. Building - Logs that belongs to the user based on the location of the reader that has been tapped onto
3. Today – Logs of card that belongs to the user that has been tapped onto any reader on the current date the logs is accessed

Additionally, on every page of a log, the logs can be printed into a PDF file format for their perusal.

4.5.2 Physical Database Design

The database is in local host-based database server that is access through the platform of Microsoft SQL Server Management Studio 18 installed on the host running on the specification shown in the table below.

Table 4.8: Host Specification

Attributes	Details
Device name	DESKTOP-CU2V553
Processor	Intel® Core™ i5-7200U CPU @ 2.50 GHz 2.71GHz
RAM	12.0GB
System type	64-bit Operating System, x64-based processor

The creation database of the system is by translating the Entity Relationship Diagram (ERD) and the data dictionary of the database into a queries using Data Definition Language(DDL) to create tables and also to apply constraints of foreign keys. The scripts of the DDL with primary key and constraints for foreign key creation and primary key for indexing purposes used in this system are as shown below:

```
create table users (
        u_id numeric(10,0) primary key not null,
        matric_num varchar(10) not null,
        name varchar(50) not null,
        password varbinary(50) not null,
        privilege numeric(1,0) not null,
        date_created datetime not null,
        status numeric(1,0) null,
        email varchar(50) not null
)


create table card
(
        c_id numeric(10,0) primary key not null,
        u_id numeric(10,0) not null,
        guid varchar(8) unique null,
        date_created datetime not null,
        status numeric(1,0) null,
        constraint FK_userincard FOREIGN KEY (u_id) REFERENCES 
users(u_id)
);


create table building
(
    b_id numeric(10,0) primary key not null,
    u_id numeric(10,0) not null,
        name varchar(50) not null,
        status numeric(1,0) null
        constraint FK_buildingincard FOREIGN KEY (u_id) 
REFERENCES users(u_id)
);
```

```
create table reader
(
        r_id numeric(10,0) primary key not null,
        u_id numeric(10,0) not null,
        b_id numeric(10,0) not null,
        date_created datetime not null,
        status numeric(1,0) null,
        constraint FK_userinreader FOREIGN KEY (u_id) REFERENCES
users(u_id),
        constraint FK_buildinginreader FOREIGN KEY (b_id)
REFERENCES building(b_id)
);


create table record
(
        rec_id numeric(10,0) primary key not null,
        c_id numeric(10,0) not null,
        r_id numeric(10,0) not null,
        date_in datetime not null,
        date_out datetime null,
        constraint FK_cardinrecord FOREIGN KEY (c_id) REFERENCES
card(c_id),
        constraint FK_readerinrecord FOREIGN KEY (r_id)
REFERENCES reader(r_id)
);

create table logs
(
        l_id numeric(10,0) primary key not null,
        rec_id numeric(10,0) not null,
        c_id numeric(10,0) not null,
        r_id numeric(10,0) not null,
        date_in datetime not null,
        status numeric(1,0) not null,
        constraint FK_recordinlog FOREIGN KEY (rec_id)
REFERENCES record(rec_id)
);
```

4.6 Conclusion

To conclude, RFID helps the process of monitoring the user’s movements to be easier due to effective and efficient data transmission between the RFID tag and the RFID reader. RFID technology is better than barcode as it does not need to be seen in order to be detected. As long as the RFID card are within the range that an RFID reader can detect. Compared to MySejahtera, this system also logs the current status of a person detected to be in or out of a building/room.

The similarity between the proposed system and the monitoring system developed by (Mehta, S., Grant, K., Atlin, C., & Ackery, A., 2020) is the purpose of the system itself. The system developed by (Mehta, S., Grant, K., Atlin, C., & Ackery, A., 2020) is applied to detect contact tracing of the hospital’s staff that tends in the ward filled with positive Covid-19 patients. The proposed system is to detect contact tracing of the personnel within the university. The only difference is that the proposed system is also used to monitor the movement of these staffs and student within the building not just in a single room. It is used to detect the exact person in a specified room.

References Page

1. Ahmad, S. M. S., Ali, B. M., & Adnan, W. A. W. (2012). Technical issues and challenges of biometric applications as access control tools of information security. *international journal of innovative computing, information and control*, *8*(11), 7983-7999.
2. Attaran, M. (2007). RFID: an enabler of supply chain operations. Supply Chain Management: An International Journal.
3. Farooq, U., ul Hasan, M., Amar, M., Hanif, A., & Asad, M. U. (2014). RFID based security and access control system. International Journal of Engineering and Technology, 6(4), 309.
4. Fauscette, M. (2018, April 17). *Biometrics Are Coming & So Are Security Concerns*. Dark Reading. https://www.darkreading.com/endpoint/biometrics-are-coming-and-so-are-security-concerns/a/d-id/1331536#:%7E:text=The%20central%20issue%20is%20that,stuck%20with%20the%20compromised%20biometric.
5. Gentry, S. (2021, May 13). *Access control: Models and methods [updated 2021]*. Infosec Resources. https://resources.infosecinstitute.com/certification/access-control-models-and-methods/#:%7E:text=Access%20control%20models%20have%20four,management%20of%20the%20access%20controls.
6. Johnston, C. (2018). Amazon opens a supermarket with no checkouts. Business Reporter, BBC News.
7. Lucid Content Team. (2018, August 10). *4 Phases of Rapid Application Development Methodology | Lucidchart Blog*. Lucidchart. https://www.lucidchart.com/blog/rapid-application-development-methodology
8. Mehta, S., Grant, K., Atlin, C., & Ackery, A. (2020). Mitigating staff risk in the workplace: the use of RFID technology during a COVID-19 pandemic and beyond. BMJ Health & Care Informatics, 27(3).

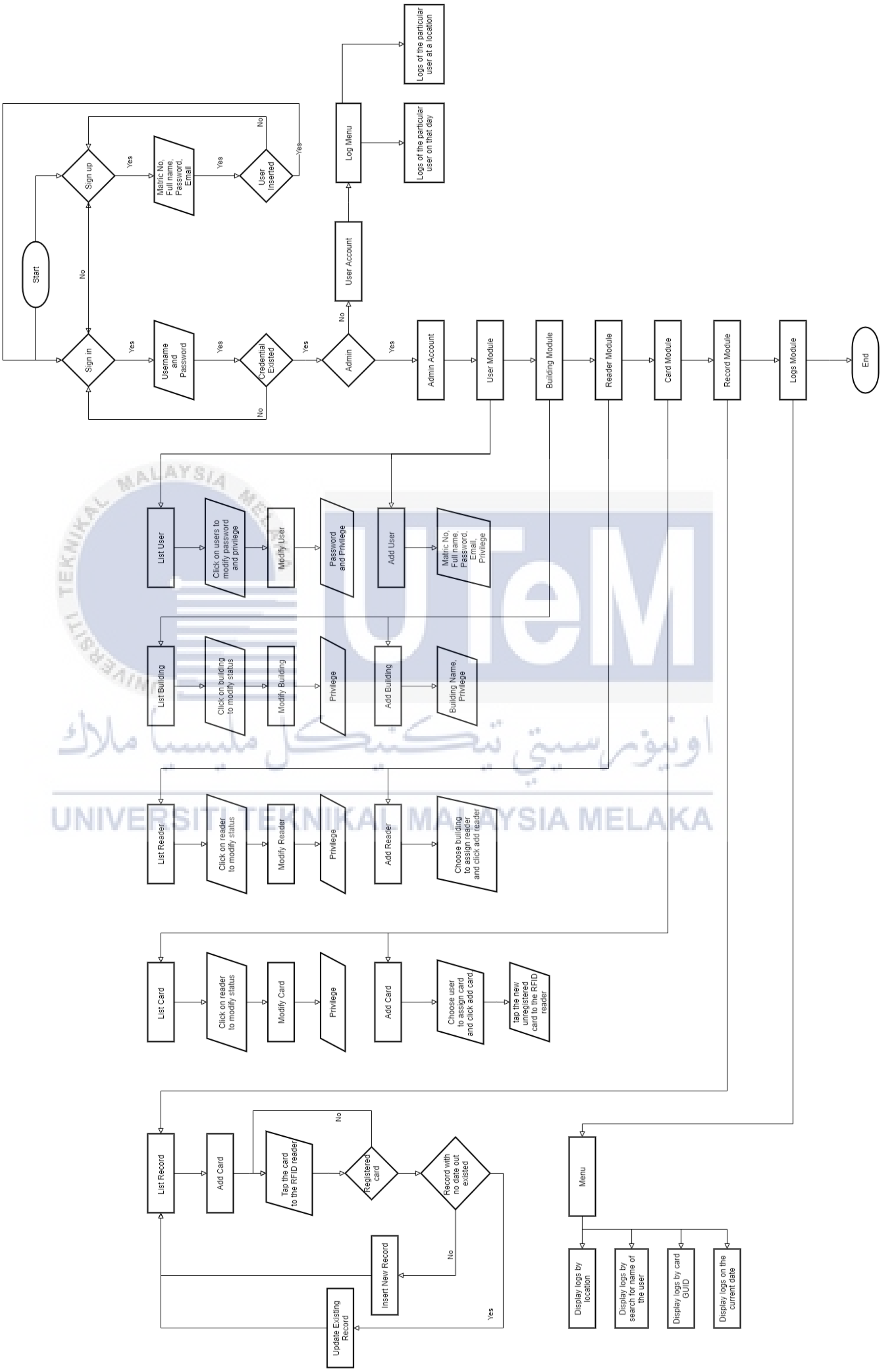
9. Ostojic, G., Stankovski, S., Lazarevic, M., & Jovanovic, V. (2007, September). Implementation of RFID technology in parking lot access control system. In *2007 1st Annual RFID Eurasia* (pp. 1-5). IEEE.

10. Renger. (2021, April 21). *Rapid application development | RAD tools*. Yeeflow. https://blog.yeeflow.com/rapid-application-development-rad-tools/

11. SENSEON. (2020, September 26). *Access Control*. https://www.senseon.com/access-control/

Attachments Page

Appendix A: Proposed system's flow chart


Start
Sign up
Yes
Matric No, Full name, Password, Email
Yes
User Inserted
No
Yes
No
Sign in
Yes
Username and Password
Yes
Credential Existed
No
Yes
Admin
No
User Account
Log Menu
Logs of the particular user on that day
Logs of the particular user at a location
Yes
Admin Account
User Module
Building Module
Reader Module
Card Module
Record Module
Logs Module
End
List User
Click on users to modify password and privilege
Modify User
Password and Privilege
Add User
Matric No, Full name, Password, Email Privilege
List Building
Click on building to modify status
Modify Building
Privilege
Add Building
Building Name, Privilege
List Reader
Click on reader to modify status
Modify Reader
Privilege
Add Reader
Choose building to assign reader and click add reader
List Card
Click on reader to modify status
Modify Card
Privilege
Add Card
Choose user to assign card and click add card
tap the new unregistered card to the RFID reader
List Record
Add Card
Tap the card to the RFID reader
No
Registered card
Record with no date out existed
No
Insert New Record
Yes
Update Existing Record
Menu
Display logs by location
Display logs by search for name of the user
Display logs by card GUID
Display logs on the current date