DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON

BY

FONG KARR KEET

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON

Academic Session: 2022/2023

I FONG KARR KEET

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
No.28, Lorong 14,
Taman Petani Jaya,
08000 Sungai Petani, Kedah.

Dr. Goh Hock Guan
Supervisor's name

Date: 25/4/2023

Date: 26/4/2023

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: __25/4/2023____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***Fong Karr Keet***___________________ (ID No: __***19ACB03625***__) has completed this final year project entitled "_DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON_" under the supervision of ___Dr. Goh Hock Guan___ (Supervisor) from the Department of ___Computer and Communication Technology__, Faculty of _Information and Communication Technology_____ .

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*FONG KARR KEET*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________

Name : ____FONG KARR KEET___

Date : ______25/4/2023__________

ACKNOWLEDGEMENTS

I would like to convey my heartful gratitude and appreciation to my supervisors, Ts Dr Goh Hock Guan, and my moderator, Puan Nor' Afifah Binti Sabri, for providing me with this beautiful opportunity to engage in an embedded networking design project. It is my first step toward establishing a profession in embedded networking design. A million thanks to you.

Aside from that, I also need to express my profound admiration and gratitude to Tunku Abdul Rahman University College, which provides this golden opportunity and all the facilities for me to carry out this development-based project. Furthermore, I would like to thank MDT Innovations Sdn Bhd for supplying the hardware and briefly presenting the current project's flow and requirements.

Furthermore, I would like to thank my course mates and friends for offering me valuable feedback and inspiration, which assisted me in completing my dissertation. Lastly, I must express my appreciation and thank my parents and my family for their love, support, and constant encouragement throughout the course.

ABSTRACT

In this project, I will propose and construct a gantry-based UTAR Hospital Linen Tracking through RFID using Python in order to minimize the workload of the hospital staff, boost the efficiency of the linen tracking system, and save money by reducing the loss of linen. The gantry-based RFID linen tracking system comprises a few components, like antenna, RFID tags, RFID gantry, and a back-end system to enable users to monitor the data uploaded to the cloud. The proposed system traces and monitors incoming and outsourcing linen from UTAR hospital and the laundry plant suppliers. Linen included are pillowcases, blankets, bedsheets, patient garments, surgical drapes, aprons, surgical caps, cleaning clothes, etc. As the linen products embedded with the RFID tag are scanned by the RFID reader or RFID gantry, unique identifications of the tags will be identified, and their location and allocation procedure will be stored in the cloud database. As a result, hospital staff may trace all the linens by accessing the cloud database record system on the industrial panel provided. I have used a dedicated Python integrated development environment (IDE), PostgreSQL, an open-source database, and Microsoft Power BI dashboard as the software for this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

UTAR	University Tunku Abdul Rahman
RFID	Radio-frequency Identification
ID	Identification
RF	Radiofrequency
IoT	Internet of Things
LMS	Linen Management System
IDE	Hardware Description Language
DB	Database
LOS	Line of Sight
TSDB	Time-Series Databases
SDLC	Software Development Life Cycle
UI	User Interface
SQL	Structured Query Language
TCM	Traditional Chinese Medicine
WM	Western Medicine

Chapter 1
Introduction

I will present the background and motivation for my progress on the gantry-based UTAR Hospital Linen Tracking through RFID using Python, my contributions to the field, and the outline of the project in this chapter.

1.1 Problem Statement and Motivation

Nowadays, several hospitals or healthcare facilities still use manual data entry or tracking methods on each valued asset, particularly for linens. The manual technique tends to raise the burden on the hospital's employees, immediately reducing asset efficiency and posing a hygiene risk to people. Physical recording or tracking methods may tend to disinformation, typographical errors, or even blunders. This error might probably lead to the loss of linen in hospitals, and hospitals will be compelled to restock and compensate for the losses. As a result, a linen tracking system that utilizes RFID technology as the solution for the problem or difficulties produced by the manual method is proposed for the new opening of UTAR hospital. This solution can enhance asset efficiency while minimizing operational loss. On the other hand, RFID tags enable management to trace all the linens via the data uploaded to the cloud. Therefore, the loss of linen will be reduced significantly due to the movement of linen visible in the RFID linen tracking system.

Taking on this project allows me to have the opportunity to learn more about the fascinating deep understanding of RFID technology. This project tends to enhance my academic knowledge and learning, especially in the embedded network devices related to the Internet of Things (IoT). In this project, I was tasked with helping in developing a linen tracking system that uses RFID technology. This project is the integration of classroom knowledge and theory with hands-on applications and abilities. As a newcomer to the profession of IoT, this project will offer me practical experience. It will be a rewarding experience in embedded system development related to the IoT, as well as an opportunity to hone embedded skills and gain rich knowledge about RFID technology. Other than that, this project would also help in the development of UTAR Hospital. I would be delighted to contribute something to UTAR with the expertise I gained at UTAR. During the Covid-19 pandemic season, the burden on hospital employees has drastically risen due to a massive number of infected patients being allocated to the hospital. Several hospitals without the RFID linen tracking system will manually assign the staff to manage and process the incoming clean and outgoing dirty linen. This action is hugely inconvenient and places a significant workload on the hospital staff. In order to address this issue, I proposed to develop a gantry-based linen tracking system using RFID to effectively track and monitor that linen while minimizing the burdensome on the hospital staff.

1.2 Project Objectives

1. This project aims to enhance the efficiency of the linen tracking system by at least 10% over the manual approach of data entry or data tracking with the newly developed gantry-based UTAR Hospital Linen Tracking through RFID using Python.

This first objective offers an improved version of the linen tracking system, which makes the entire process more efficient. The manual input or tracking of each valued asset, notably for linens in the hospital, is time-consuming and inconvenient for hospital staff. This RFID-Based linen tracking system automatically centralized all the precise data and uploaded it to the database. Consequently, the linen tracking system resulted in higher accuracy and error-free output. RFID solutions offer a more enhanced version of inventory management and accounting procedures that make the transition of linen smoother and more efficient. Besides that, the newly invented system also provides better transparency of the linen tracking system by displaying precise data on the user-friendly dashboard.

2. This project aims to save at least 20% of the restocking cost that compensates for linens' losses with the newly developed gantry-based UTAR Hospital Linen Tracking through RFID using Python.

The second objective boosts the linen tracking system's transparency to eliminate linens losses, reduce restocking costs, and compensate for the losses. The RFID-Based linen tracking system increases the clarity of the linen laundry process, which means there will be proper tracking and monitoring of each incoming and outgoing linen. Each RFID tag embedded with linens will store all the processes it goes through, the linen's present location, and the date of manufacture. All this data will be transmitted to RFID readers when the RFID tags receive RF signals from RFID readers **[8]**. Hence, the linen tracking system will trace all the linen, dramatically reducing linen loss. Consequently, the hospital does not need to spend too much on overstock linens or sheets yearly to satisfy fulfilment and compensate for losses.

3. This project aims to reduce 20% of the workload of the hospital staff with the newly created gantry-based UTAR Hospital Linen Tracking through RFID using Python.

The third objective is to minimize the workload of the hospital staff. By installing this gantry-based RFID linen tracking system, the hospital staff is relieved from the burden of manually recording or tracking the linen data one by one, which is troublesome. The collected linen data will be uploaded to the database every 5 seconds. Therefore, the hospital staff is not required to report to the laundry plant suppliers about the daily outsourcing of linen amount from the hospital. The laundry plant suppliers can check the daily outsourcing linen amount from the hospital via the user-friendly dashboard directly connected to the database.

1.3 Project Scope and Direction

The project's scope is to track and monitor the incoming and outsourcing linen from UTAR hospital more efficiently and productively. There are numerous issues included in the project that must be taken into consideration. Some of these critical concerns are project timeliness, process sequencing, process transparency, laundry production, linen distribution, rescheduling of distribution and the ways departments of the hospital and laundry services collaborate, end-user difficulties, and confirmation. Multiple topics excluded from this project must be considered: instruction or advice on managing the linen tracking system, type of linen products, and so on.

In addition, this project will necessitate the use of both hardware and software. The MDT Innovations Company has contributed to this project by sponsoring some RFID hardware, such as one set of RFID gantry, antennas, and RFID tags(microchips) for UTAR students currently taking this project. The company also shows an example of the LMS System, the back-end system, and explains the RFID linen tracking system's flow to the UTAR students involved in this project.

On the other hand, I have decided to use a python programming language in developing the RFID-based linen tracking system using RFID gantry. I used PyCharm, a dedicated Python integrated development environment (IDE), PostgreSQL, an open-source time series database, and Microsoft Power BI dashboard as the software for this project. The hardware involved in this project includes a computer that processes the database and monitors and tracks the linen status, an RFID gantry that transmits data with an RFID tag, antennas that detect the RFID tags, and an RFID tag that stores and records data and is embedded in the linen.

In conclusion, this project required a fundamental grasp of RFID technology, hardware configuration, and an understanding of the detailed flow of the linen tracking system. Python language, implementation of a database by using PostgreSQL, and the Microsoft Power BI dashboard for displaying the accurate linen data must be acknowledged to accomplish this project.

1.4 Impact, significance, and contribution

This project contributes to UTAR Hospital with an effective linen tracking system. The RFID-based linen tracking system is able to enhance the efficiency of the UTAR hospital's linen tracking system by replacing the manual data tracking technique. Besides that, the RFID-enabled linen tracking system also minimizes the labour of UTAR hospital employees by getting rid of the manual data tracking system. Last but not least, the linen tracking system that uses RFID technology eliminates linens losses in UTAR hospitals and reduces the cost of restocking and compensating for the losses by boosting the transparency of the linen tracking system.

1.5 Background information

Since January 2014, University Tunku Abdul Rahman (UTAR) has planned to build a non-profit specialized hospital in Kampar to supply healthcare services to populaces and the community, particularly Kampar citizens. It is expected that once the UTAR hospital is officially open, a vast quantity of linens will be incoming and outgoing from UTAR hospital. Linen management is a crucial aspect of hospital management **[5]**. Hospital linen can be referred to as any textile materials used solely in the hospital. Besides that, it must be sanitized in the laundry before reuse to guarantee the hygiene of the hospital environment. For instance, linens are pillowcases, blankets, bedsheets, patient garments, surgical drapes, aprons, surgical caps, cleaning cloths, etc. An effective linen tracking system is necessary to track and monitor the enormous numbers of UTAR hospital linens. Radio-frequency Identification (RFID) is a feasible technology that can identify, track, and interact with objects and people utilizing radio frequency from radio waves. In this project, I will develop a gantry-based Hospital Linen Tracking through RFID using Python to keep track of the incoming and outgoing linen for UTAR Hospital.

Radio-frequency Identification (RFID) is a viable technology that has recently become essential to everyone's daily life. It is a sort of monitoring system which use radio frequency of radio waves to identify, track and interact with objects and people. Nowadays, RFID implementation has become increasingly noticeable in various industries, including the healthcare industry. By employing RFID, we can boost the efficiency of the linen tracking system in a hospital.

Every RFID system consists of 3 vital components such as a scanning antenna, a transponder, and a transceiver **[8]**. The RFID reader, a combination of a scanning antenna and a transceiver, is a network-connected device that can be either mobile or permanently fixed. It emits signals that activate the RFID tags using radio waves. Once activated, the RFID tag transmits a wave back to the antenna, which is converted into data. In the hospital's linen tracking system, the RFID reader captures linen data and uploads it to the cloud. The way to capture linen data is by scanning RFID tags that serve as transponders in the RFID system using RFID readers.

RFID tags consist of tiny microchips that include an integrated circuit (IC), a substrate, and an antenna. RFID tags receive the radio waves sent from the RFID reader and transmit the feedback signal back to the RFID reader. The RFID tag is used to identify items uniquely. Therefore, each RFID tag has a unique identification (ID). This implementation allows the mobile tracking of the object with a specific RFID tag. Typically, most RFID tags currently contain roughly 256 bits of storage to hold the data or information that can be represented by numbers **[10]**. The information stored on the RFID tag chip can be read remotely without physical contact by utilizing the Radiofrequency (RF) energy emitted by the RFID reader antenna. The RFID tag is attached to every single linen in the hospital for the linen tracking system.

1.6 Report Organization

This report is broken into five chapters, each describing a distinct aspect of the project. Chapter 1 covered the problem statements and motivation, project objective, project scope, and contribution. Previous works on deep learning and RFID's limitation are described in Chapter 2, the literature review. A comparison between RFID and barcodes is made. Lastly, the solution for each limitation is proposed. Chapter 3 will cover the system requirement, which is the hardware and software needed for this system. After that, it will list the implementation issue or challenges and the project's timeline. Chapter 4 is the preliminary work section covering the software setting up process, preliminary work results, and comments, highlighting the proposed method's feasibility. Chapter 5 describes the setting and configuration of the Gantry-based linen tracking system using RFID, system operation and extra system functionality. While for Chapter 6, it is all about the system performance evaluation and project challenges. Chapter 7 will conclude the project's overall status and will be summarized in conclusion and provide future work and recommendations.

Chapter 2

Literature Review

2.1 Previous works on Deep Learning

In this chapter, the literature review will be carried out by referring to current RFID linen tracking systems in the industry, described and posted by previous researchers in internet articles, newspapers, or journals. RFID-enabled linen tracking systems are widely employed in various sectors, especially the hospitality and healthcare industries. The RFID-based linen tracking system delivers precise asset information while minimizing asset loss **[1]**. When the linen tracking system tracks and stores all the crucial details on the linen, such as its current state and location, the chances of theft or loss can be considerably reduced.

ProLinenCare (M) Sdn Bhd is a Malaysian linen and laundry company. It has specialized in healthcare linen and laundry services since its inception in 2004. ProLinenCare discovered that several non-conformances have been detected and flagged by external auditors, primarily from the Linen Department, indicating inadequate inventory management, which is managed by employees manually **[9]**. Inventory inconsistencies and linen loss are two of ProLinenCare's most pressing business concerns, which have enormous consequences for healthcare linen management. ProLinenCare discovered that inventory shrinkage due to poor record-keeping is the primary source of inventory discrepancies. ProLinenCare detected the four potentialities of linen loss. There are deliberate theft, inadvertent theft, improper disposal, and abuse or improper use. However, linen loss can be avoided with a better inventory tracking and management system.

After ProLinenCare learns more about well-known technologies used in inventory management, namely RFID and Barcode, they attempt to enhance their linen monitoring and tracking system by utilizing those new technologies. They have made a comparison between the RFID and Barcode. In contrast, RFID has more pros than conventional technologies. Therefore, RFID seems to be a wiser option for ProLinenCare to prevent linen discrepancy and loss. RFID provides complete visibility of inventory management thru the data transmission between the RFID readers, and RFID tags are sewn into each linen-typed item. With the adequate transparency of the RFID linen tracking system, it is easy to do a continuous check to see whether linens are missing or stolen. ProLinenCare uses RFID technology to seamlessly track and manage a hospital's incoming and outgoing linen.

2.1.2 Strength and Weakness of RFID

RFID is a sophisticated automated identification system. The strengths were a more comprehensive scanning range, enhanced capabilities, speed, and ease. An RFID reader can scan a tag if it is within frequency range as well as scan through obstruction in some circumstances. There are no line-of-sight restrictions. Other than that, RFID systems may scan numerous objects concurrently. For instance, you may review all the arriving linens in the linen bag at once, allowing you to inspect the contents without performing individual barcode scans on each item. RFID readers can automatically scan tags in milliseconds.

However, RFID also has several significant flaws. The most critical weakness is the massive costs of RFID implementation. RFID systems are still typically more costly to install and operate than barcode systems. Even though the RFID readers can scan through most non-metallic materials, metal and water will interfere with the RFID signal. The capability to scan numerous items within a range is advantageous, but it also introduces potential concerns that might lead to malfunctions, such as tag and reader collisions. RFID also introduces certain security risks. Unauthorized devices may be able to access and even modify data on tags without the owner's knowledge **[7]**.

2.1.3 Strength and Weakness of Barcode

A barcode is a representation of numbers and letters using bars and spaces that machines can read **[4]**. It enables the precise and quick collection of real-time data via barcode readers. Barcode, a mature and proven technology, allows for more efficient data processing procedures with minimal errors. Barcodes are cheaper, smaller, lighter, and easier to use than RFID tags.

However, barcode technology also has some weaknesses. Barcode scanning necessitates using a particular instrument known as a barcode scanner, which produces light and gathers reflected light to decode the barcode via optical line of sight (LOS) scanning. Barcodes do not provide read/write operations. It is time-consuming since each item must be scanned individually. Scratched or crumpled barcodes may create scanning issues.

2.1.4 Comparison of RFID and Barcode

RFID is a wireless device that employs electromagnetic frequencies to identify the linen items in a hospital uniquely. RFID technology does not necessitate a clear line of sight to be read. While for the barcode, the scanner needs to physically see the barcode to read the details stored inside the barcode. Expenses for RFID are relatively higher compared to barcodes **[2]**.

Besides that, RFID readers can identify thousands of RFID items at once and scan distances up to several feet, depending on the RFID readers' strength. But barcode tags can only be read manually, one at a time. RFID technology can read, write, modify, and update the information stored inside the chips, but barcode technology can only read data.

Asides from that, RFID could be wholly automated once up and running. Unfortunately, barcode technology requires labourers to scan each tag to proceed to read data stored in the barcode. The overall comparison is shown in Table 2.3.1 below. Due to all the comparisons, RFID is more convenient for developing the linen tracking system for the hospital.

Table 2.4.1 Comparison between RFID and Barcode

Criteria \ System	RFID	Barcode
Cost of implementing the system	High	Low
Time consumption	Low	High
Ability towards information	Read, Write, Modify, Update	Write
The level of Manpower needed	Low	Medium
Modifiable	Yes	No

2.1.5 Limitations of RFID

Even though RFID technology is preferable for a hospital's linen tracking system, there are various problems or constraints to its implementation, including economic and technological concerns. Apollo Hospitals, the leading multispecialty healthcare unit in India, indicated that RFID could not be adopted in India's hospitals due to the RFID tags being costly and pricey. It might not even withstand India's fluctuating washing regimen **[6]**.

While pricing an RFID system, consider costs over the system's lifetime rather than the initial cost. To establish the viability of RFID, developers must undertake an entire lifecycle costing due to constant technological evolution.

RFID can be less reliable because RFID can only work if adequate RF signal strength exists. Metal items will impact the RF field, so do not expect 100% reliability while conducting an RFID process near metallic objects [8]. As a result, the emitted RF signal is RFID's vulnerability and potentially also one of the limitations of RFID.

2.1.6 Proposed solutions

This project proposes an RFID-based Linen Tracking System to offer adequate visibility of accurate real-time information on the hospital's incoming and outgoing linen items. It will provide various system reports with a simplified and user-friendly interface, such as linens outsourcing, linen ageing, linen loss analysis, facility distribution, real-time stock level, etc.

On the contrary, active RFID tags can be substituted by passive RFID tags to address economic concerns. This substitution is needed because Passive RFID tags tend to be smaller and less costly than active tags because passive tags do not need a power source or transmitter **[3]**. The transmission of the radio signal will happen between the antenna of the tag chip and the RFID system. Therefore, the expense of implementing the RFID-based linen tracking system will be reduced by substituting the active tags with passive tags.

Besides, a more advanced RFID reader is necessary to solve the technological concerns of insufficient RF signal strength caused by environmental factors. By using a more advanced RFID reader, a more robust RF signal strength is transmitted between the RFID reader and RFID tags within the distance range of each suitable environment.

Chapter 3

Proposed System Model

I will explain the system requirements, which are the hardware and software required for project development, the method or approach utilized in this project, system design and architecture, implementation issues or challenges, and the project's timeline in this chapter.

3.1 System Requirement

3.1.1 Hardware

The hardware involved in this project is a computer that processes the database, monitors and tracks the linen status, and an RFID gantry that transmits data with an RFID tag and an RFID tag that stores and records data and is embedded in the linen.

Laptop

Figure 3.1.1.1 Asus TUF FX505G-EBQ535T.

Table 3.1.1.1 Specifications of laptop.

Description	**Specifications**
Model	Asus TUF FX505G-EBQ535T
Processor	Intel(R) Core(TM) i7-8750H CPU
Operating System	Windows 10
Graphic	NVIDIA GeForce GTX1050Ti GDDR5 4GB
Memory	8GB DDR4 RAM
Storage	PCIE NVME 512GB M.2 SSD

RFID Gantry

Figure 3.1.1.2 RFID Gantry with and without weighing scale.

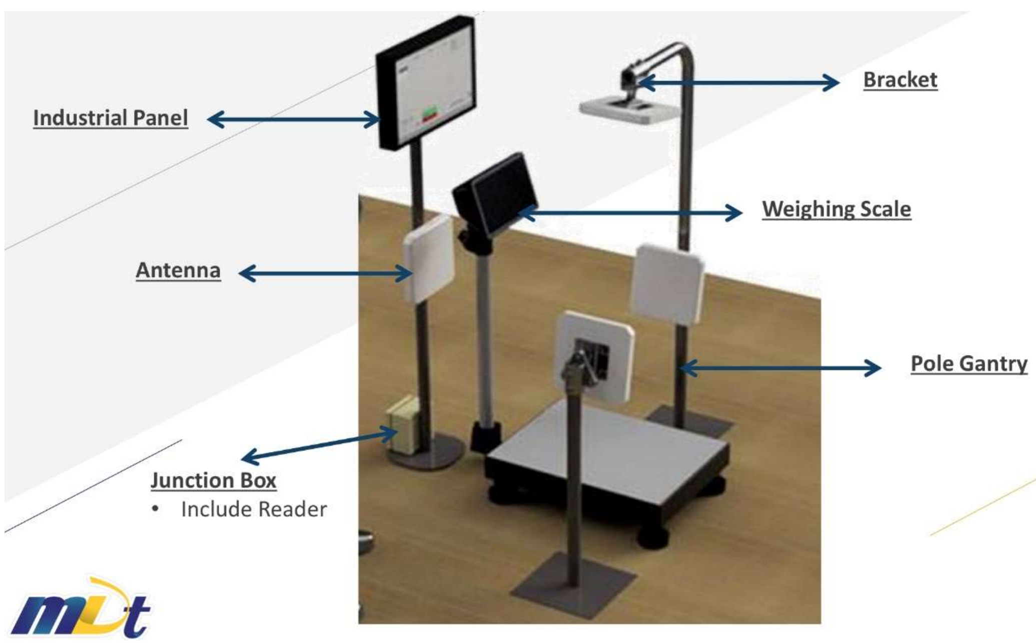


Figure 3.1.1.3 RFID Gantry's components.

There are two types of RFID gantry, one is equipped with a weighing scale, and another is without a weighing scale. Small and big RFID gantries are used to capture linen data and submit it to the cloud and will be set up in the hospital and laundry plant. A small RFID gantry is used to weigh the linen bag directly. In contrast, the giant RFID gantry with a significant platform weighing scale allows the trolley with the linen bag to weigh on it. RFID gantry will automatically calculate the weight of linen sacks by deducting the total weight from the trolley weight, which can be obtained from the RFID tag of the trolley. The junction box, which is the component of the RFID gantry, will receive data from the antenna and upload the data to the cloud. The staff can check the data thru the industrial panel, which is quite robust.

RFID tag

Figure 3.1.1.4 RFID Tag.

Table 3.1.1.2 Specifications of RFID Tag.

Description	Specifications
Model	LinTag™ 200
Operating Frequency	860-960 MHz
Chip Type	MONZA M5
Reading distance	7 meters
Memory	128bits EPC
Anti-Collision	Yes
Storage	32GB/64GB Supports up to 128 GB Micro SD card
Reading Distance	Up to 23 ft (7 m)

LinTag is an RFID tag embedded with the linen, and each of them is set up with unique identification to allow an RFID scanner to identify each linen. This feature will ensure that all the linen data can be monitored and tracked easily. The RFID tag is waterproof, highly resistant to harsh liquid and fluctuating temperatures, and performs consistently.

3.1.2 Software

The software involved in this project is PyCharm, a dedicated Python integrated development environment (IDE), PostgreSQL, an open-source database, and Microsoft Power BI dashboard as the software for this project.

PyCharm Community Version

Figure 3.1.2.1 PyCharm

Table 3.1.2.1 System requirement of PyCharm Community Version

Requirement	Minimum	Recommended
RAM	4 GB of free RAM	8 GB of total system RAM
CPU	Any modern CPU	Multi-core CPU
Disk space	2.5 GB and another 1 GB for caches	SSD drive with at least 5 GB of free space
Monitor resolution	1024×768	1920×1080
Operating system	64-bit versions of Windows 8 or later, macOS 10.14 or later, Linux	The latest 64-bit version of Windows, macOS, or Linux

PyCharm is a Python-specific Integrated Development Environment (IDE) that provides various necessary tools for Python developers. These tools are closely integrated to offer a pleasant environment for productive Python, web, and data science development. I chose the PyCharm community version as IDE because it offers all the essential features a decent IDE should provide, including code completion, code inspections, debugging, error highlights and remedies, etc.

PostgreSQL

Figure 3.1.2.2 PostgreSQL

Table 3.1.2.2 System requirement of PostgreSQL

Requirement	Minimum
Processor	1 GHz
RAM	2 GB
HDD	512MB

PostgreSQL is a sturdy open-source object-relational database management system with a solid reputation for stability, feature robustness, and performance over 30+ years of continuous development. Besides, I picked PostgreSQL because it is equipped with the PyCharm Database Navigation, which is user-friendly.

Microsoft Power Bi

Figure 3.1.2.3 Microsoft Power BI

Microsoft Power BI is a safe and reliable Microsoft cloud service that offers users access to Power BI dashboards, reports, and applications. I chose Microsoft Power BI as a dashboard because it is user-friendly, equipped with many features and easy to use.

3.2 Design Specifications

I am using the Agile software development life cycle (SDLC) methodology as my project's methodology because this methodology offers flexibility and opportunity for frequent changes or updates in the development project. The iteration loop of the agile methodology leads me to focus on high-quality development by undergoing multiple updates, testing, and reviews during each iteration. There are five phases in SDLC methodologies: planning, designing, developing, testing and maintenance.

In the planning phase, I will set up the hardware, like an RFID gantry, antenna, and RFID tags. Other than that, I also configure the software I will use in this project, such as PyCharm, PostgreSQL, and Microsoft Power BI dashboard. I will briefly design the project flow, decide the type of data that should be included, and the ways to pass data to the database and publish it to the dashboard in the design phase. While for the developing stage, I will start coding and implementing the database and dashboard. I will also try to relate or link the hardware to my programming code. In the testing and maintenance phase, I will test multiple times on each function created to ensure that the system's performance achieves the objective and 100% accurate data.

Planning Phase:

I will try to connect the hardware, like the RFID gantry, to my laptop and start to set up Impinj. I will also set up the software needed, like PyCharm, PostgreSQL, and Microsoft Power BI dashboard, and install some of the Python packages required for the program. I will review my objective and planning progress again to ensure that my project is related to the project's aim.

Designing Phase:

I will design the flow of the actual linen tracking system and decide on data that should be displayed for the users, like the RFID tag id and its label, timestamp, the linen type, and origin location. I have chosen PostgreSQL as my project's database and Microsoft Power BI dashboard as my project's dashboard. I have found ways to pass data to the database and publish it to the dashboard.

Implementation Phase:

I started to code the Python programming in the PyCharm IDE and tried to receive the scanned tag data. I found out that there were many duplications of scanned RFID tag information. I must eliminate those duplication data to avoid errors by filtering out the duplication tag using Python code. I will try to push the data to the database and display the data in the dashboard.

Testing and maintenance phase:

In the last phase, I underwent multiple testing on the program developed to ensure that it works and displays 100% correct data. I must ensure that the project is always related to the project's goals. For future work, I will try to develop more functions for the user to access the linen tracking system.

3.3 System Design/ Overview

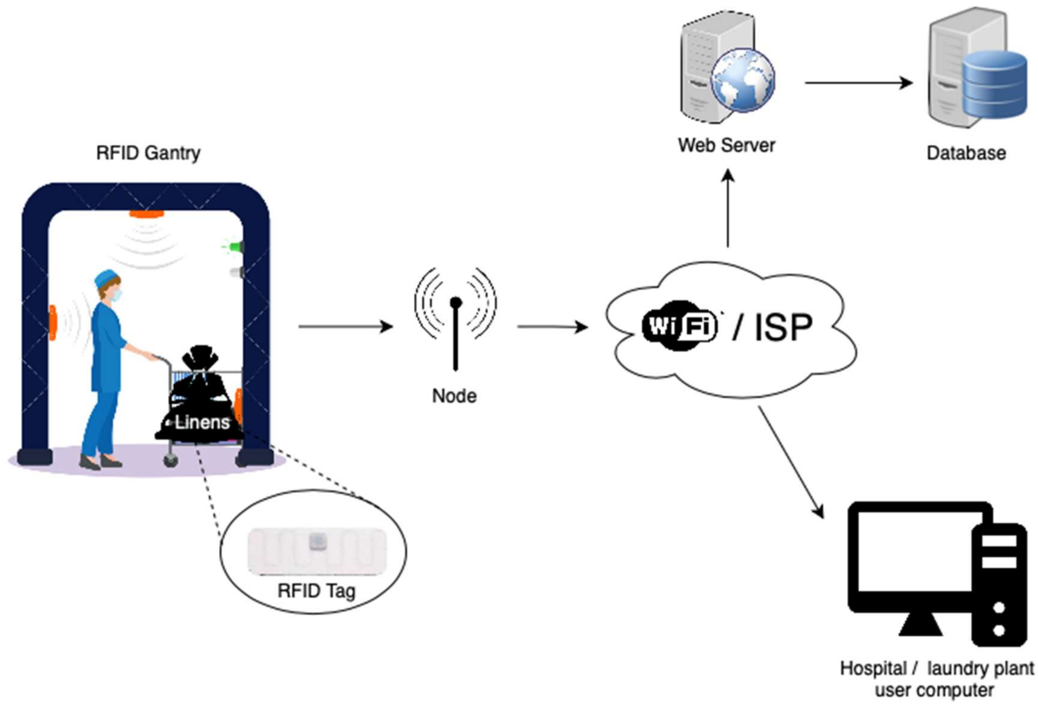


Figure 3.3.1 System Overview

Figure System Overview above shows the overall flow of the RFID-based linen tracking system. Firstly, each embedded with RFID tag linens is put in a linen bag. The RFID gantry scanned incoming and outgoing linen using the antenna, and the system removed the duplication scanned tag. The data received will direct to a node and access the database or dashboard through the internet. Users from hospitals or laundry plants can also check and record data through the Internet.

3.3 Project Timeline

Task Name	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12
Planning and Meeting with Supervisor												
Define Problem Statement and Objective												
Setup RFID hardware and test run												
Start coding and developing the monitor system												
Setup database and push data to database												
Create a linen tracking dashboard												
Finalize report												

Figure 3.6.1 Project Timeline

Chapter 4
Preliminary Work

In this chapter, I will briefly discuss the task required to be done and prepared to complete this final year project. In the preliminary work, I was required to set up the software and hardware and implement a function for the project by knowing the system's flow.

4.1 Setting up

4.1.1 Software Setup

Before starting to develop the gantry-based UTAR Hospital Linen Tracking System through RFID using Python, three software programs needed to be installed and downloaded on my laptop:

1. PyCharm Community Version
2. PostgreSQL
3. Microsoft Power BI

I was required to create a new project file in the PyCharm Community Version to code a program for the system. Other than that, I have created a PostgreSQL database for storing linen data. After creating a database, I connected Microsoft Power Bi with the PostgreSQL database to implement a dashboard visualizing the linen data.

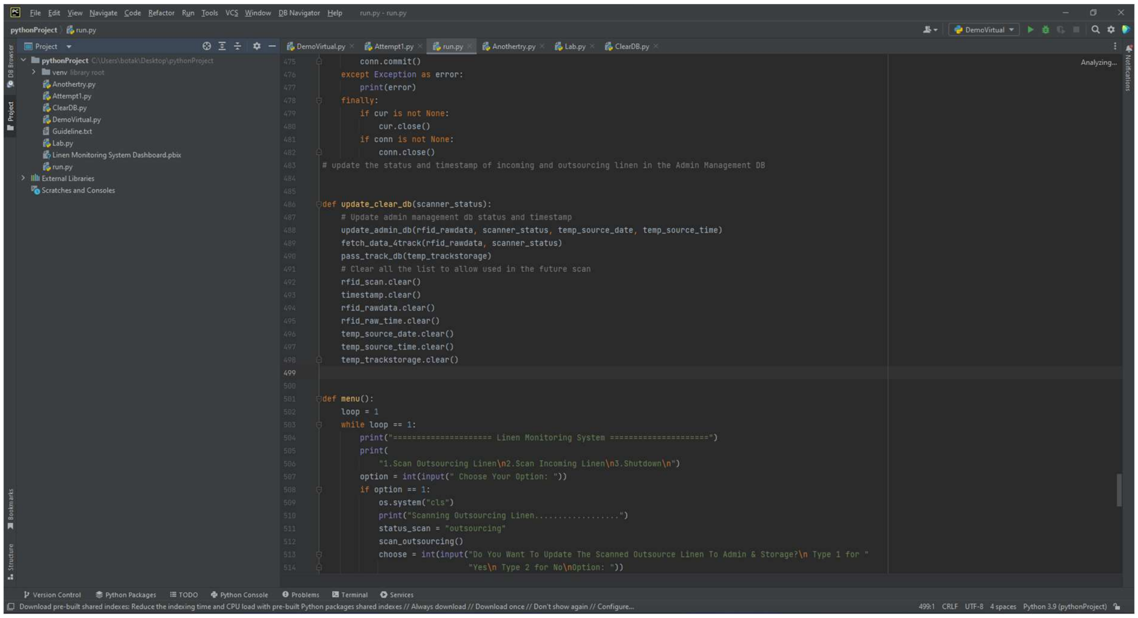


Figure 4.1.1.1 PyCharm Community Version Environment.

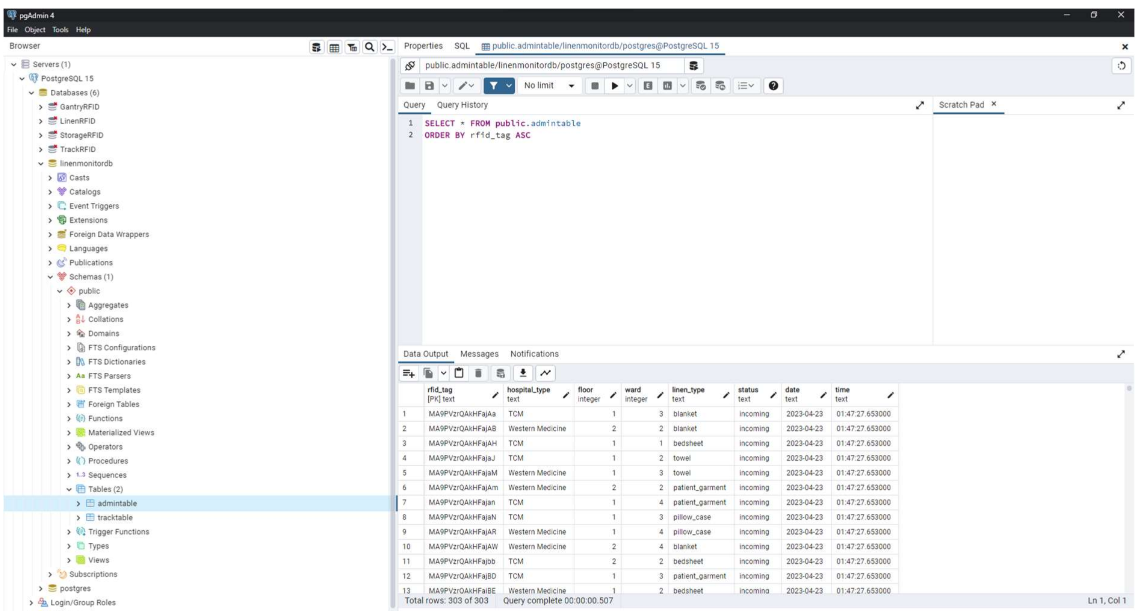


Figure 4.1.1.2 PostgreSQL Environment.

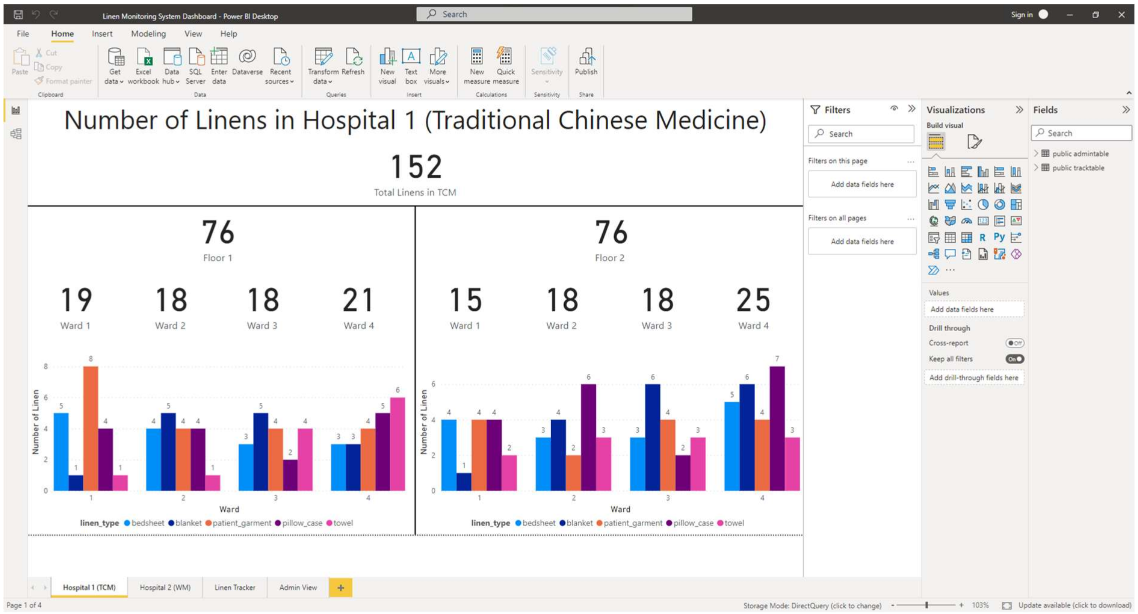


Figure 4.1.1.3 Microsoft Power BI Environment.

4.1.2 Hardware Setup

RFID hardware involves an RFID reader, antenna, and RFID tags. Below is the image of the RFID hardware setup:

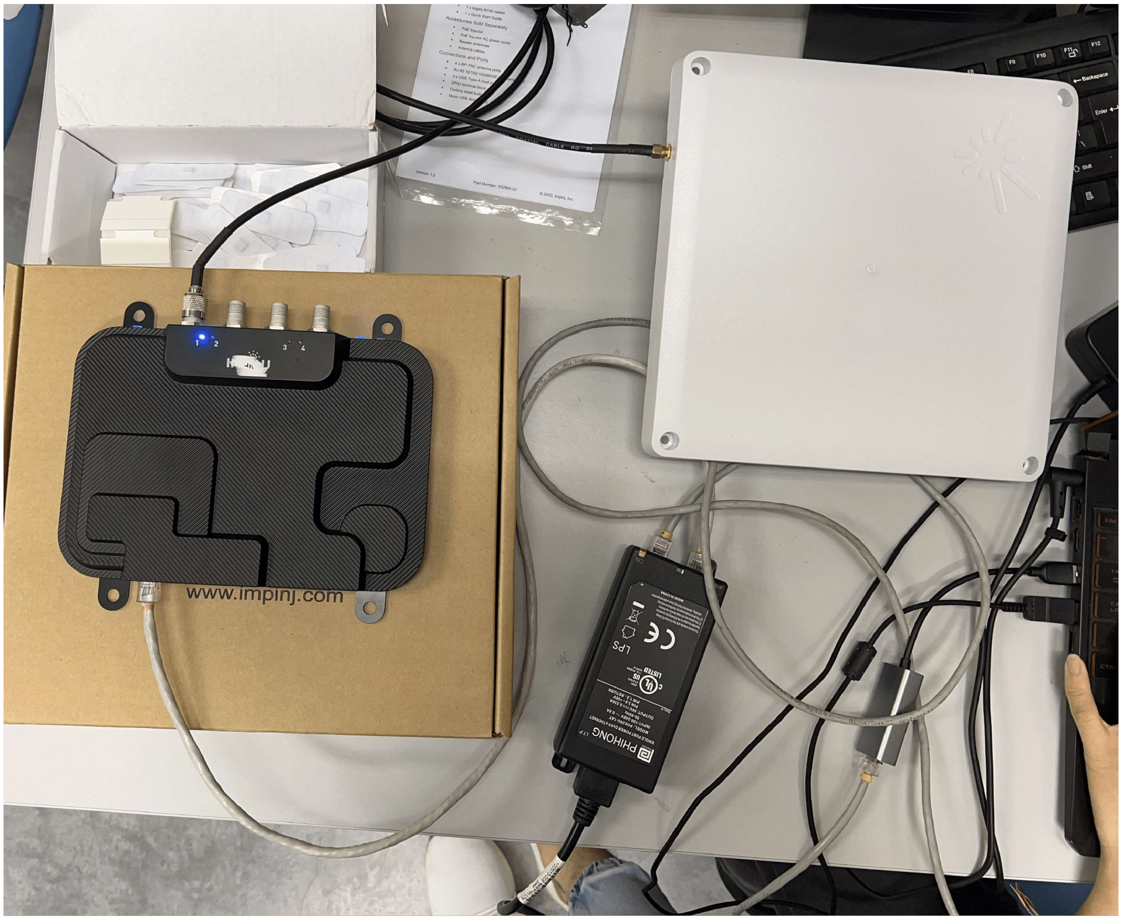


Figure 4.1.2.1 RFID hardware setup.

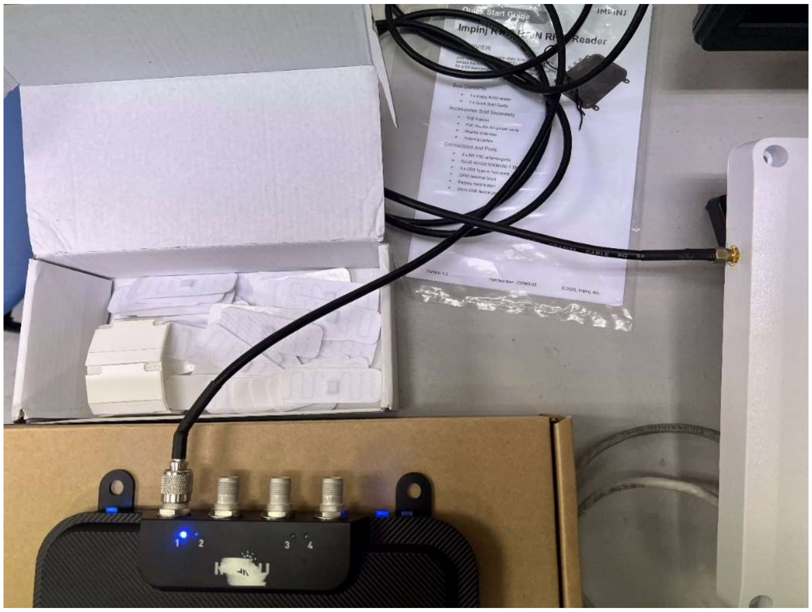


Figure 4.1.2.2 RFID Reader Port 1 to the antenna.

Figure 4.1.2.3 Battery Power Brick.

According to the proper guideline, I connected the RFID reader port 1 with the antenna and used an ethernet wire to connect the RFID reader to the laptop. After that, I need to ensure that the battery power brick's green light indicator is on to prove a power supply to the RFID reader. Other than that, I need to make sure the RFID Reader antenna port 1's blue light indicator is on to ensure the reader works fine.

Figure 4.1.2.4 OUT port is connected to the RFID reader.

Figure 4.1.2.5 IN port is connected to the laptop via the USB-ethernet converter.

Besides, I have connected my laptop to the IN port of the battery power brick via the USB-ethernet converter due to my laptop's ethernet port malfunction. The OUT port is connected to the RFID reader through an ethernet cable.

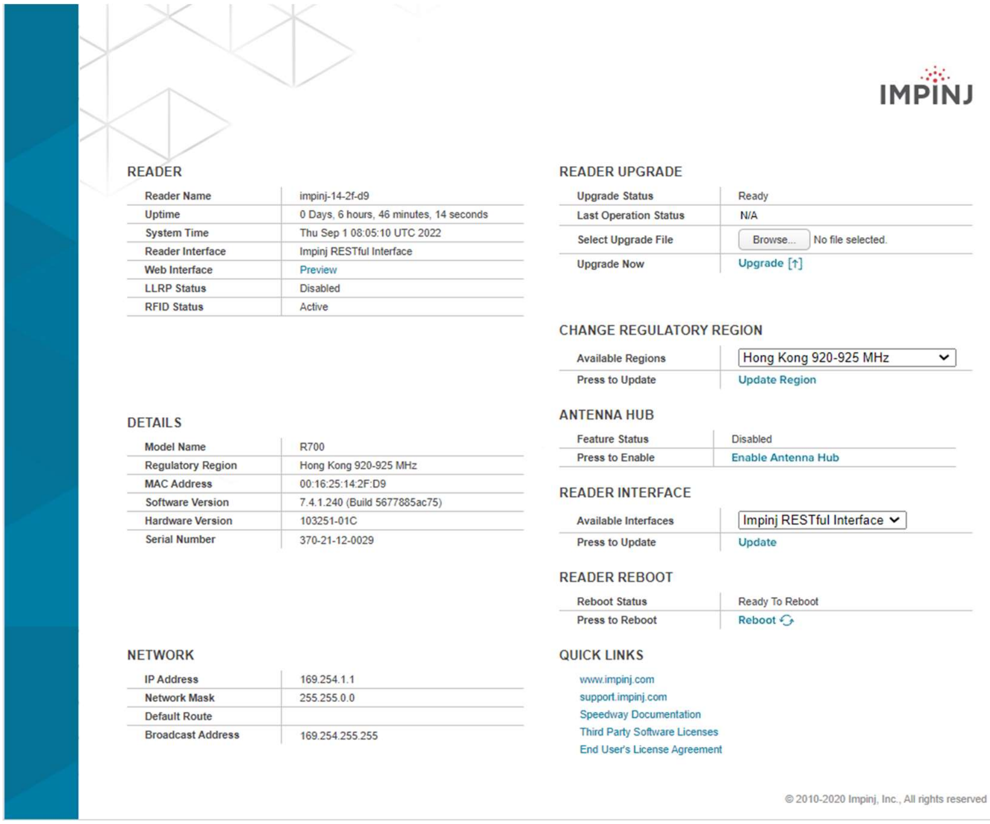


Figure 4.1.2.6 Impinj setup.

After finishing all the hardware setup, I need to ensure a connection between the RFID reader and my laptop via the Impinj website by accessing with the hostname, Impinj-14-2f-d9. After login into the Impinj website, I must ensure that the Available Region is set to Hong Kong 920-925 MHz and Available Interface is set to Impinj RESTful Interface.

Figure 4.1.2.7 RFID tags that are ready to be scanned.

After all the settings in the Impinj website, the RFID tags are ready to scan by running the Python code to stream all the responses into the command prompt.

4.2 Function Implementation

4.2.1 Getting Raw Data of RFID Tags

I used the run() function to scan the RFID tags data. I have extracted the RFID and timestamp from the scanning data and stored them in the array lists "rfid_scan" and "timestamp" for further utilizing. Below figures are the run() function codes that are used to scan RFID tags:

```
def run():
    # Handle Ctrl+C interrupt
    signal.signal(signal.SIGINT, signal_handler)

    arguments = get_command_line_arguments()

    hostname = 'http://{0}'.format(arguments.reader)

    try:
        requests.get(urljoin(hostname, '/api/v1/status')).raise_for_status()
    except (requests.ConnectionError, requests.exceptions.HTTPError):
        print('Error : Unable to connect to the Impinj Reader API on "{0}"'.format(hostname))
        if len(hostname.split(':')) == 2:
            print('       Have you provided the port number with your reader hostname?')
            print('       ex.  --reader <your-reader-hostname>:<api-port>')
        sys.exit(1)

    if arguments.stop:
        stop_request_url = urljoin(hostname, 'api/v1/profiles/stop')
        print_request('POST', stop_request_url)
        check_response(requests.post(stop_request_url))

    if arguments.schema:
        schema_request_url = urljoin(hostname, '/api/v1/profiles/inventory/presets-schema')
        print_request('GET', schema_request_url)
        check_response(requests.get(schema_request_url))

    if arguments.status:
        status_request_url = urljoin(hostname, '/api/v1/status')
        print_request('GET', status_request_url)
        check_response(requests.get(status_request_url))

    if arguments.toDelete:
        delete_request_url = '{0}/api/v1/profiles/inventory/presets/{1}'.format(hostname, arguments.toDelete)
        print_request('DELETE', delete_request_url)
        check_response(requests.delete(delete_request_url))
```

Figure 4.2.1.1 Function used to scan RFID tags (Part 1).

```python
    if arguments.customPresetPath:
        with open(arguments.customPresetPath) as customPresetFile:
            # Add or update the provided preset
            try:
                custom_preset = json.load(customPresetFile)
                if arguments.presetId is None:
                    print('Error : An id must be provided for custom presets.')
                    sys.exit(1)
                preset_request_url = '{0}/api/v1/profiles/inventory/presets/{1}'.format(hostname, arguments.presetId)
                print_request('PUT', preset_request_url)
                check_response(requests.put(preset_request_url, data=json.dumps(custom_preset)))
            except ValueError:
                print('Error : the provided custom configuration contains invalid json')
                sys.exit(1)

    if arguments.listPresets:
        get_preset_list_request_url = urljoin(hostname, '/api/v1/profiles/inventory/presets')
        print_request('GET', get_preset_list_request_url)
        check_response(requests.get(get_preset_list_request_url))

    if arguments.details:
        get_preset_details_request_url = '{0}/api/v1/profiles/inventory/presets/{1}'.format(hostname, arguments.details)
        print_request('GET', get_preset_details_request_url)
        check_response(requests.get(get_preset_details_request_url))

    if arguments.toStart:
        start_request_url = '{0}/api/v1/profiles/inventory/presets/{1}/start'.format(hostname, arguments.toStart)
        print_request('POST', start_request_url)
        check_response(requests.post(start_request_url))
```

Figure 4.2.1.2 Function used to scan RFID tags (Part 2).

```python
    if arguments.stream:
        event_stream_request_url = urljoin(hostname, 'api/v1/data/stream')
        print_request('GET', event_stream_request_url)
        event_stream_response = requests.get(event_stream_request_url, stream=True)
        check_response(event_stream_response, print_body=False)
        t_end = time.time() + 5.0
        while True:
            mystring = event_stream_response.raw.readline().decode("utf-8")
            print(str(t_end) + " " + str(time.time()))
            if time.time() > t_end:
                break
            try:
                r_timestamp = re.search('"timestamp":"(.+?)","tagInventoryEvent', mystring).group(1)
                rfid = re.search('"epc":"(.+?)","antennaPort', mystring).group(1)
                print("Time now: " + r_timestamp + ",   RFID: " + rfid)
                rfid_scan.append(rfid)
                timestamp.append(r_timestamp)
            except AttributeError:
                pass

    if arguments.listProfiles:
        list_profiles_request_url = urljoin(hostname, 'api/v1/profiles')
        print_request('GET', list_profiles_request_url)
        check_response(requests.get(list_profiles_request_url))
# Main Scanning Function
```

Figure 4.2.1.3 Function used to scan RFID tags (Part 3).

4.2.2 Filtering Out the Duplicated Data

I have noticed there are multiple duplications of the RFID tags data. So, I created a check duplication function to eliminate the duplicated data and store the processed data in an array list called "rfid_rawdata". I also saved all the timestamps for each RFID tag data in an array list called "rfid_raw_time". It will update the timestamp for each particular RFID tag data when it finds a duplication. By using the function, I was able to ensure that all the RFID tags' IDs stored in the newly created array list are unique and without any duplication.

```
def check_dup(rfid_list, r_timestamp_list):
    for i in rfid_list:
        if i not in rfid_rawdata:
            rfid_rawdata.append(i)
            rfid_raw_time.append(r_timestamp_list[rfid_list.index(i)])
        else:
            rfid_raw_time[rfid_rawdata.index(i)] = r_timestamp_list[rfid_list.index(i)]
```

Figure 4.2.2.1 Check duplication function.

4.2.3 Adding Extra Information to RFID Tags Data

After eliminating all the duplicated data, I have formed a new 2D array list to store the RFID tag from the array list called "rfid_rawdata" and timestamp from the array list called "rfid_raw_time" with some extra information like label name, antenna port number, the linen type, and origin location. I am just making an assumption about the linen type and location information using the Python random module. The antenna port is always set as one since we only have one antenna port. The module will randomize and choose for me the linen type and location for assigning to each RFID tag id.

```
x = 1
antenna_port = 1
linen_type = ["blanket", "towel", "bedsheet", "patient_garment", "pillow_case"]
location = ["ICU_dep", "Surgeon_dep", "Dobi", "Infect_dep"]

for i in rfid_rawdata:
    rfid_lib.append(("linen" + str(x), i, antenna_port, random.choice(linen_type), random.choice(location),
                     rfid_raw_time[rfid_rawdata.index(i)]))
    x += 1
```

Figure 4.2.3.1 Adding information code.

4.2.4 Creating a Database Table and Pushing Data into the Database

After combining all those information into a 2D array list, I created a table named "admintable" in the database, "linenmonitordb", with the Python function "create_db" to store processed data. I pushed all the processed data in the 2D array list into my PostgreSQL database table using the function "pass_admin_db". This function will upload the information of label name, tag ID, timestamp, origin location, linen types, and antenna port.

```
def create_db():
    hostname = "localhost"
    database = "linenmonitordb"
    username = "postgres"
    pwd = "admin"
    port_id = 5432
    conn = None
    cur = None
    try:
        conn = psycopg2.connect(
            host=hostname,
            dbname=database,
            user=username,
            password=pwd,
            port=port_id
        )
        cur = conn.cursor()
        # create Admin Management DB (which contain all the RFID tags information)
        create_administrator_table = ''' CREATE TABLE IF NOT EXISTS admintable (
                                    label_name          text,
                                    rfid_tag            text PRIMARY KEY,
                                    antenna_port_no     text,
                                    linen_type          text,
                                    origin_location     text,
                                    timestamp           text
        )'''

        cur.execute(create_administrator_table)
        conn.commit()
    except Exception as error:
        print(error)
    finally:
        if cur is not None:
            cur.close()
        if conn is not None:
            conn.close()
```

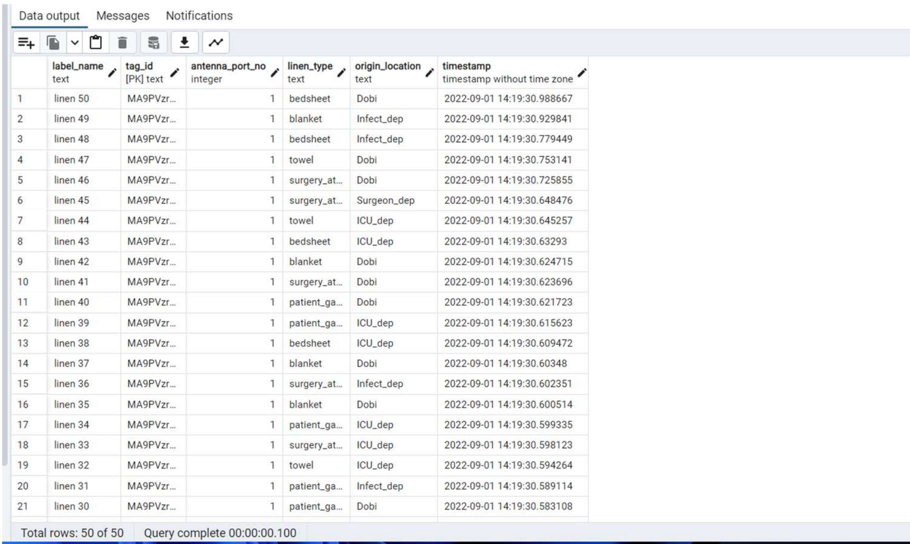
Figure 4.2.4.1 Python Coding for creating PostgreSQL database table.

```python
def pass_admin_db(arr):
    hostname = "localhost"
    database = "linenmonitordb"
    username = "postgres"
    pwd = "admin"
    port_id = 5432
    conn = None
    cur = None
    try:
        conn = psycopg2.connect(
            host=hostname,
            dbname=database,
            user=username,
            password=pwd,
            port=port_id
        )
        cur = conn.cursor()
        insert_data = 'INSERT INTO admintable ( label_name, tag_id, antenna_port_no, linen_type, origin_location, ' \
                      'timestamp) VALUES (%s, %s, %s, %s, %s, %s) '
        for i in arr:
            cur.execute(insert_data, i)
        conn.commit()
    except Exception as error:
        print(error)
    finally:
        if cur is not None:
            cur.close()
        if conn is not None:
            conn.close()
```

Figure 4.2.4.2 Python Coding for passing data to PostgreSQL database.

4.3 Preliminary Work Results

After completing the function implementation, all the data is uploaded to the database and dashboard. Below is the picture showing the database output storing all this information like label name, tag ID, timestamp, origin location, linen types, and antenna port. The dashboard will visualize all the information about each linen embedded with the tag, as shown in Figure 4.3.2.

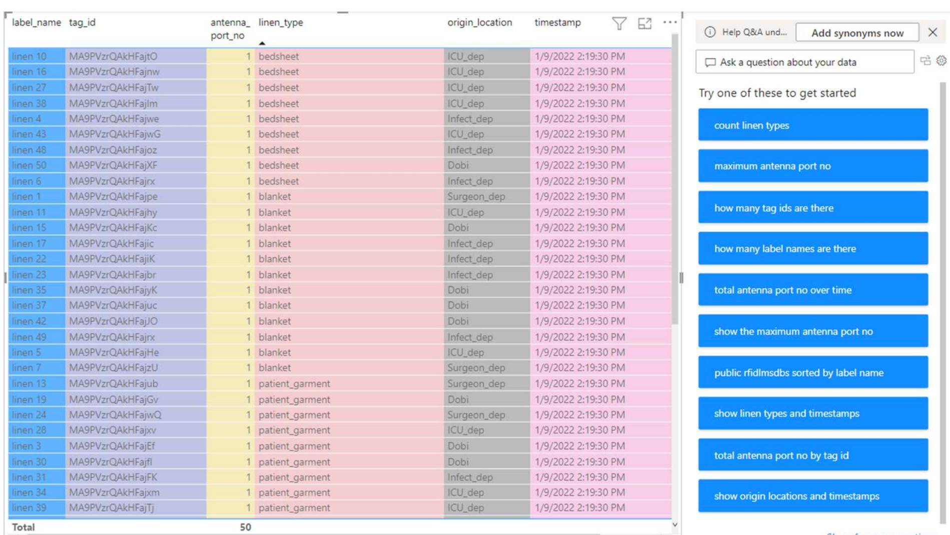


Data output Messages Notifications

	label_name text	tag_id [PK] text	antenna_port_no integer	linen_type text	origin_location text	timestamp timestamp without time zone
1	linen 50	MA9PVzr...	1	bedsheet	Dobi	2022-09-01 14:19:30.988667
2	linen 49	MA9PVzr...	1	blanket	Infect_dep	2022-09-01 14:19:30.929841
3	linen 48	MA9PVzr...	1	bedsheet	Infect_dep	2022-09-01 14:19:30.779449
4	linen 47	MA9PVzr...	1	towel	Dobi	2022-09-01 14:19:30.753141
5	linen 46	MA9PVzr...	1	surgery_at...	Dobi	2022-09-01 14:19:30.725855
6	linen 45	MA9PVzr...	1	surgery_at...	Surgeon_dep	2022-09-01 14:19:30.648476
7	linen 44	MA9PVzr...	1	towel	ICU_dep	2022-09-01 14:19:30.645257
8	linen 43	MA9PVzr...	1	bedsheet	ICU_dep	2022-09-01 14:19:30.63293
9	linen 42	MA9PVzr...	1	blanket	Dobi	2022-09-01 14:19:30.624715
10	linen 41	MA9PVzr...	1	surgery_at...	Dobi	2022-09-01 14:19:30.623696
11	linen 40	MA9PVzr...	1	patient_ga...	Dobi	2022-09-01 14:19:30.621723
12	linen 39	MA9PVzr...	1	patient_ga...	ICU_dep	2022-09-01 14:19:30.615623
13	linen 38	MA9PVzr...	1	bedsheet	ICU_dep	2022-09-01 14:19:30.609472
14	linen 37	MA9PVzr...	1	blanket	Dobi	2022-09-01 14:19:30.60348
15	linen 36	MA9PVzr...	1	surgery_at...	Infect_dep	2022-09-01 14:19:30.602351
16	linen 35	MA9PVzr...	1	blanket	Dobi	2022-09-01 14:19:30.600514
17	linen 34	MA9PVzr...	1	patient_ga...	ICU_dep	2022-09-01 14:19:30.599335
18	linen 33	MA9PVzr...	1	surgery_at...	ICU_dep	2022-09-01 14:19:30.598123
19	linen 32	MA9PVzr...	1	towel	ICU_dep	2022-09-01 14:19:30.594264
20	linen 31	MA9PVzr...	1	patient_ga...	Infect_dep	2022-09-01 14:19:30.589114
21	linen 30	MA9PVzr...	1	patient_ga...	Dobi	2022-09-01 14:19:30.583108

Total rows: 50 of 50 Query complete 00:00:00.100

Figure 4.3.1 PostgreSQL database result.



label_name	tag_id	antenna_port_no	linen_type	origin_location	timestamp
linen 10	MA9PVzrQAkHFajtO	1	bedsheet	ICU_dep	1/9/2022 2:19:30 PM
linen 16	MA9PVzrQAkHFajnw	1	bedsheet	ICU_dep	1/9/2022 2:19:30 PM
linen 27	MA9PVzrQAkHFajTw	1	bedsheet	ICU_dep	1/9/2022 2:19:30 PM
linen 38	MA9PVzrQAkHFajlm	1	bedsheet	ICU_dep	1/9/2022 2:19:30 PM
linen 4	MA9PVzrQAkHFajwe	1	bedsheet	Infect_dep	1/9/2022 2:19:30 PM
linen 43	MA9PVzrQAkHFajwG	1	bedsheet	ICU_dep	1/9/2022 2:19:30 PM
linen 48	MA9PVzrQAkHFajoz	1	bedsheet	Infect_dep	1/9/2022 2:19:30 PM
linen 50	MA9PVzrQAkHFajXF	1	bedsheet	Dobi	1/9/2022 2:19:30 PM
linen 6	MA9PVzrQAkHFajrx	1	bedsheet	Infect_dep	1/9/2022 2:19:30 PM
linen 1	MA9PVzrQAkHFajpe	1	blanket	Surgeon_dep	1/9/2022 2:19:30 PM
linen 11	MA9PVzrQAkHFajhy	1	blanket	ICU_dep	1/9/2022 2:19:30 PM
linen 15	MA9PVzrQAkHFajKc	1	blanket	Dobi	1/9/2022 2:19:30 PM
linen 17	MA9PVzrQAkHFajic	1	blanket	Infect_dep	1/9/2022 2:19:30 PM
linen 22	MA9PVzrQAkHFajiK	1	blanket	Infect_dep	1/9/2022 2:19:30 PM
linen 23	MA9PVzrQAkHFajbr	1	blanket	Infect_dep	1/9/2022 2:19:30 PM
linen 35	MA9PVzrQAkHFajyK	1	blanket	Dobi	1/9/2022 2:19:30 PM
linen 37	MA9PVzrQAkHFajuc	1	blanket	Dobi	1/9/2022 2:19:30 PM
linen 42	MA9PVzrQAkHFajJO	1	blanket	Dobi	1/9/2022 2:19:30 PM
linen 49	MA9PVzrQAkHFajrx	1	blanket	Infect_dep	1/9/2022 2:19:30 PM
linen 5	MA9PVzrQAkHFajHe	1	blanket	ICU_dep	1/9/2022 2:19:30 PM
linen 7	MA9PVzrQAkHFajzU	1	blanket	Surgeon_dep	1/9/2022 2:19:30 PM
linen 13	MA9PVzrQAkHFajub	1	patient_garment	Surgeon_dep	1/9/2022 2:19:30 PM
linen 19	MA9PVzrQAkHFajGv	1	patient_garment	Dobi	1/9/2022 2:19:30 PM
linen 24	MA9PVzrQAkHFajwQ	1	patient_garment	Surgeon_dep	1/9/2022 2:19:30 PM
linen 28	MA9PVzrQAkHFajxv	1	patient_garment	ICU_dep	1/9/2022 2:19:30 PM
linen 3	MA9PVzrQAkHFajEf	1	patient_garment	Dobi	1/9/2022 2:19:30 PM
linen 30	MA9PVzrQAkHFajfl	1	patient_garment	Dobi	1/9/2022 2:19:30 PM
linen 31	MA9PVzrQAkHFajFK	1	patient_garment	Infect_dep	1/9/2022 2:19:30 PM
linen 34	MA9PVzrQAkHFajxm	1	patient_garment	ICU_dep	1/9/2022 2:19:30 PM
linen 39	MA9PVzrQAkHFajTj	1	patient_garment	ICU_dep	1/9/2022 2:19:30 PM
Total		50			

Figure 4.3.2 Microsoft Power Bi Dashboard.

Chapter 5

System Implementation

In this section, I will describe and define how the proposed system is built and operated. Besides, I will also describe the process of deployment, system installation and actual development. The difficulties encountered during implementation will also be explained in detail.

5.1 Setting and Configuration

The figure below shows an overview of the hardware: RFID setup and configuration with wiring. RFID tags will be attached to bed sheets, towels, pillows and other linens. Once all the setup has been completed and the RFID tags have been placed accordingly, the antenna will scan the incoming linens and transmit the data to the RFID reader. The reader will read the scanned tag ID and the tag information will be displayed on the PC or laptop. When the IN port is successfully connected to the laptop's Ethernet port, the laptop will show that the Ethernet has been connected, or the Impinj website will be successfully loaded. Besides, Yellow light will be shown from the reader and the battery IN port. For the whole procedure, connection with the Antenna Port 1 had been tested in my circumstance.

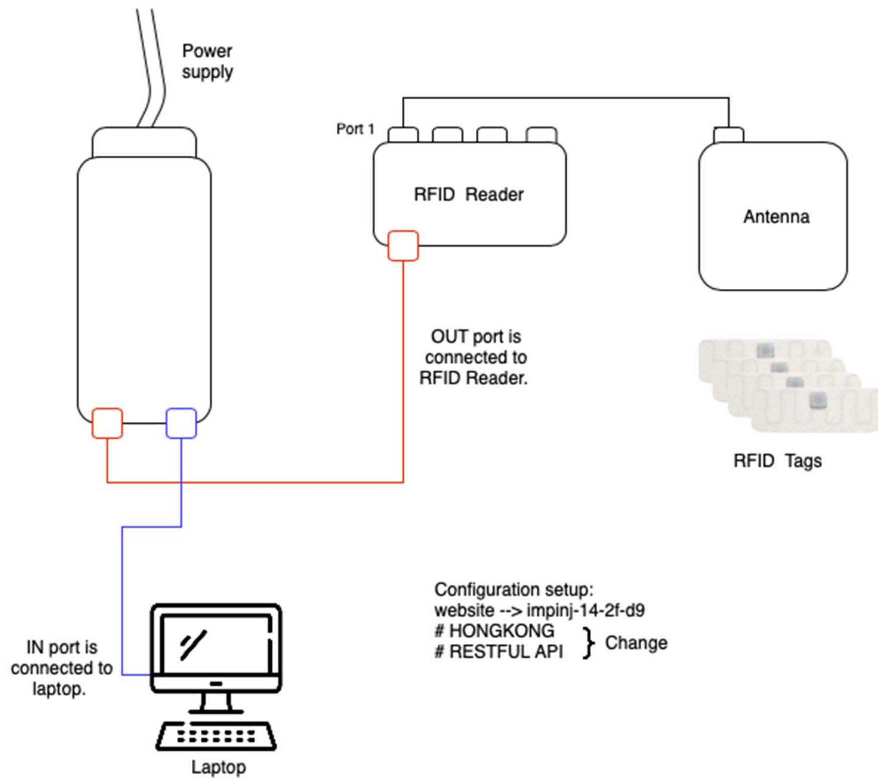


Figure 5.1.1 RFID Setup and Configuration.

5.2 System Operation

In this chapter, I will explain and describe the context of RFID linen tracking for hospital operations, including linen insourcing and outsourcing process, linen tracking, inventory management and data analysis with the utilization of RFID technology. Scanning RFID tags for check-in and checkout process with Antenna, aggregating and updating data like linen type, location, and status with real-time information to optimize inventory count are among the core system operations.

The system can be operated and performed in real-time to monitor and trace linens after completion of assembling the hardware side, as illustrated in Figure 5.1.1, and the software side. In the hospital, an RFID gantry will be placed at the back entrance to allow the antenna to scan the incoming and outgoing linens embedded with the RFID tags. After the RFID tag scanning and uploading process, the software will start processing the obtained data about the RFID tags and their respective linens and the timestamps. All processed data will be stored in the database and uploaded to a dashboard. The hospital staff will be able to visualize the data on the interactive dashboard and track the linen using a PC. Hospital staff is required to click on the shortcut to run the Python program for scanning the incoming or outsourcing linens. The scanning time is set to 5 seconds. After 5 seconds, the program will stop scanning, start data processing, and update the real-time data on the database. The dashboard will direct query the data stored in the database for further visualization. Four dashboards will be displayed to the hospital staff: Linen Assignation Dashboard for Traditional Chinese Medicine (TCM) and Western Medicine (WM), Linen Log Dashboard, and Linen Status Dashboard.

5.2.1 Linen Assignation Dashboard

Figures below shows the Linen Assignation Dashboard for Traditional Chinese Medicine (TCM) and Western Medicine (WM). Both dashboards record and show the linen assignment for each floor and each ward of two different hospitals. Total number of linens for two hospitals are shown at the top of each dashboard. Bar charts are designed to give a clearer and better view for the hospital staff, showing them the number of different types of linens at each ward.

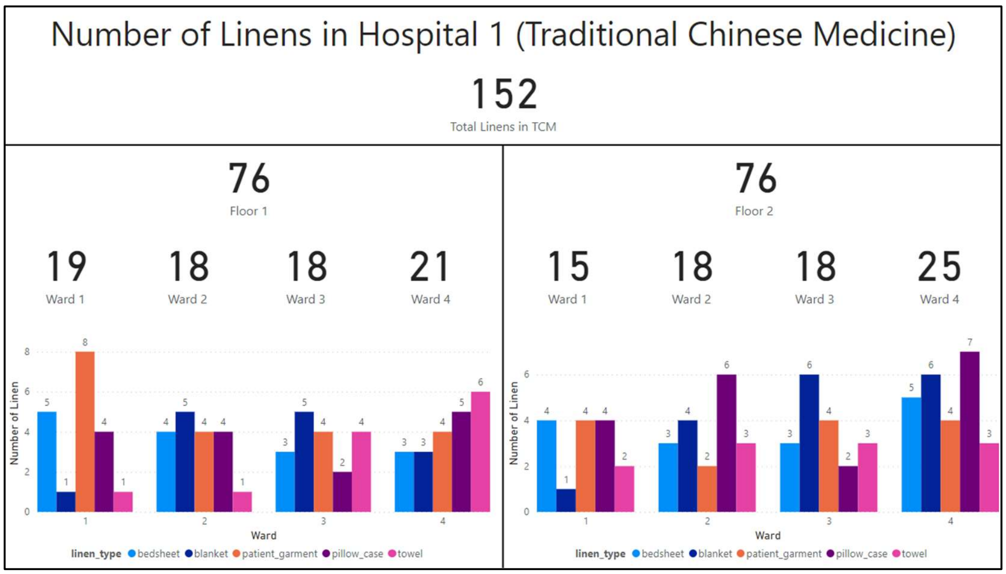


Figure 5.2.1.1 Linen Assignation Dashboard (TCM).

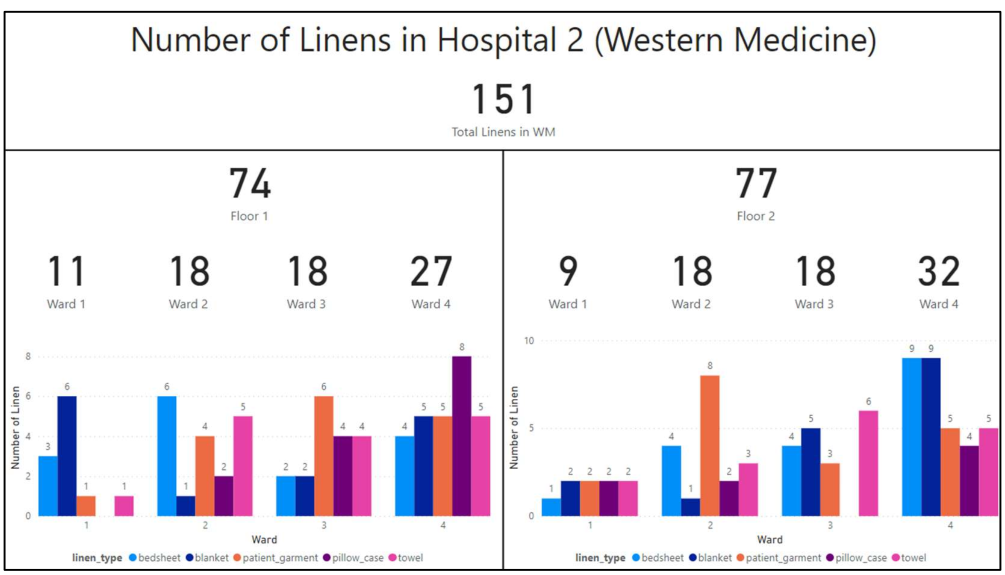


Figure 5.2.1.2 Linen Assignation Dashboard (WM).

5.2.2 Linen Log Dashboard

The figure below shows the Linen Log Dashboard for UTAR Hospital. I will separate the dashboard into two parts for the explanation.

For the left side, the chart designed gives an overview of incoming and outgoing linens in UTAR hospital in percentage. Beneath the chart, total number of linens in UTAR hospital, number of linens available in the hospital at the present time and the number of outsourcing linens for laundry will be recorded and displayed (from left to right).

For the right part, hospital staff is able to select the hospital type and date to trace the specific historical records. Total number of linens that has been scanned for check-in procedure is shown and then, followed by, the total number of linens that has been scanned for checkout process is displayed. A chart shows the percentage and number of different linen types available in the selected hospital for the selected date and time. The table below shows all the linen log record that is stored in the log database.

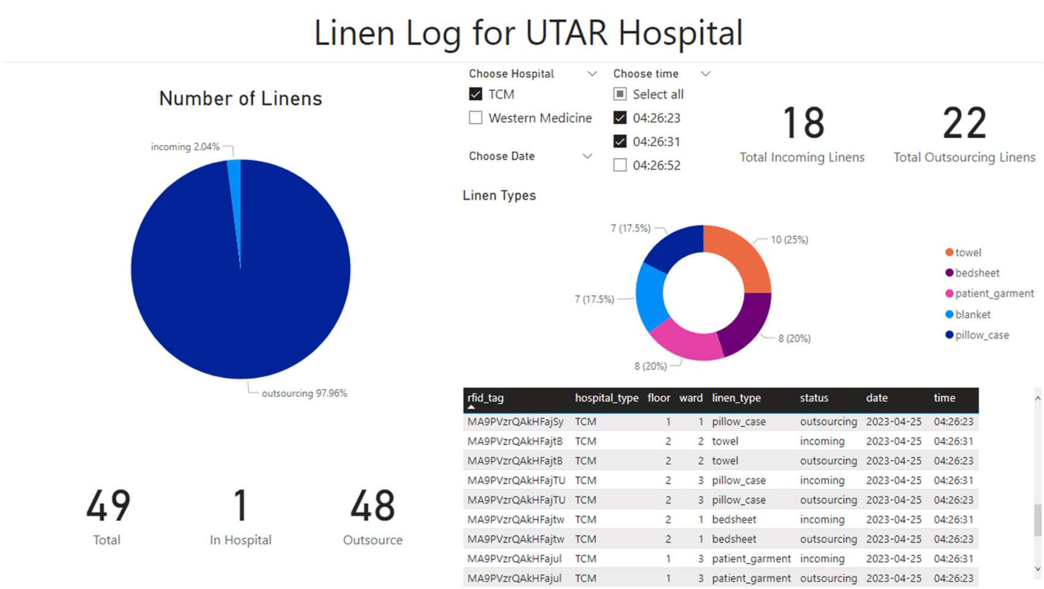


Figure 5.2.2.1 Linen Log Dashboard.

5.2.3 Linen Status Dashboard

The figure below shows the Linen Status Dashboard for UTAR hospital. For the dashboard, hospital staff is required to choose the hospital type, linen type, linen status, linen type or floor and ward that he or she would like to track. Once selection is done, the table will accurately record and show the tag information, tag ID, linen status, timestamp for the selected hospital type, linen type or floor and ward. Hospital staff is able to monitor inventory counts and variances such as matched, missing, misplaced, or excess items can be seen on the spot. This provides an effortless identification, location and tracking of linens.

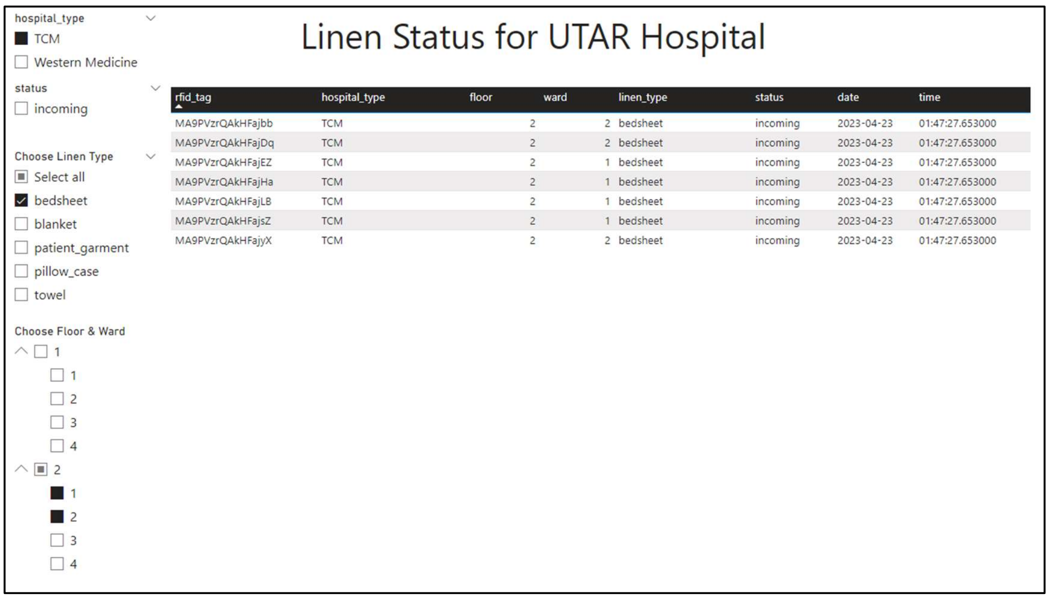


Figure 5.2.3.1 Linen Status Dashboard.

5.3 Expanding System Functionality

I will explain the new function or updates I implemented in this project to improve the system's functionality and practicality to fulfil the hospital's requirements. Several functions or updates were carried out on the system, including the threshold for the scanning period, updated real-life info and useful attributes, menu-driven interface, and linen log database.

5.3.1 The Threshold for the Scanning Period

Without a time limit for the scanning process, the user needs to eliminate the whole program to stop scanning. I have set 5 seconds as the threshold for the scanning period in the system. With the threshold, hospital staff just need to run the program to scan the RFID linen tags. Once 5 seconds countdown ends, it will automatically stop scanning and proceed with data processing.

```
if arguments.stream:
    event_stream_request_url = urljoin(hostname, 'api/v1/data/stream')
    print_request('GET', event_stream_request_url)
    event_stream_response = requests.get(event_stream_request_url, stream=True)
    check_response(event_stream_response, print_body=False)
    t_end = time.time() + 5.0
    while True:
        mystring = event_stream_response.raw.readline().decode("utf-8")
        print(str(t_end) + " " + str(time.time()))
        if time.time() > t_end:
            break
        try:
            r_timestamp = re.search('"timestamp":"(.+?)","tagInventoryEvent', mystring).group(1)
            rfid = re.search('"epc":"(.+?)","antennaPort', mystring).group(1)
            print("Time now: " + r_timestamp + ",   RFID: " + rfid)
            rfid_scan.append(rfid)
            timestamp.append(r_timestamp)
        except AttributeError:
            pass
```

Figure 5.3.1.1 Threshold for Scanning Period.

5.3.2 Update Real-life Info and Useful Attributes

Based on the previous Python function in Chapter 4, Function Implementation, the data has information about the RFID tag ID, label name, antenna port number, timestamp, the linen type, and origin location. Mostly all the information is just an assumption or randomized value. To make the system more practical and fitted into the hospital, I have created an algorithm that automatically assigned each linen to its hospital type, floor, and ward. In my case, there are only two hospitals, TCM and WM, two floors on each hospital and four wards on each floor. I also included the status of the linen as crucial information or attribute. This attribute allows me to track all the incoming and outsourcing linen from the hospital. Besides, I have separated timestamps into dates and times to have better accessibility on the program, such as tracking the incoming linen by a particular day or time.

```
def initial_pass_db(rfid, rtime):
    global ward
    status_s = "incoming"
    total_tags = len(rfid)  # 320
    h_count = total_tags // 2
    f_count = h_count // 2
    w_count = f_count // 4
    for i in rfid:
        if rfid.index(i) <= h_count:
            hospital = "TCM"  # <= 160
            if rfid.index(i) <= f_count:
                floor = 1  # <= 80
                if rfid.index(i) <= w_count:
                    ward = 1  # <= 20
                elif w_count < rfid.index(i) <= w_count * 2:
                    ward = 2  # 20 < x <= 40
                elif w_count * 2 < rfid.index(i) <= w_count * 3:
                    ward = 3  # 40 < x <= 60
                elif w_count * 3 < rfid.index(i) <= w_count * 4:
                    ward = 4  # 60 < x <= 80
            else:
                floor = 2  # >80
                if w_count * 4 < rfid.index(i) <= w_count * 5:
                    ward = 1  # 80 < x <= 100
                elif w_count * 5 < rfid.index(i) <= w_count * 6:
                    ward = 2  # 100 < x <= 120
                elif w_count * 6 < rfid.index(i) <= w_count * 7:
                    ward = 3  # 120 < x <= 140
                elif w_count * 7 < rfid.index(i) <= w_count * 8:
                    ward = 4  # 140 < x <= 160
        else:
            hospital = "Western Medicine"  # > 160
            if rfid.index(i) <= f_count * 3:
                floor = 1  # <= 240
                if w_count * 8 < rfid.index(i) <= w_count * 9:
                    ward = 1  # 160 < x <= 180
                elif w_count * 9 < rfid.index(i) <= w_count * 10:
                    ward = 2  # 180 < x <= 200
                elif w_count * 10 < rfid.index(i) <= w_count * 11:
                    ward = 3  # 200 < x <= 220
                elif w_count * 11 < rfid.index(i) <= w_count * 12:
                    ward = 4  # 220 < x <= 240
            else:
                floor = 2  # > 240
                if w_count * 12 < rfid.index(i) <= w_count * 13:
                    ward = 1  # 240 < x <= 260
                elif w_count * 13 < rfid.index(i) <= w_count * 14:
                    ward = 2  # 260 < x <= 280
                elif w_count * 14 < rfid.index(i) <= w_count * 15:
                    ward = 3  # 280 < x <= 300
                elif w_count * 15 < rfid.index(i) <= w_count * 16:
                    ward = 4  # 300 < x <= 320

        timestamp_str = rtime[rfid.index(i)]
        datetime_obj = datetime.datetime.strptime(timestamp_str, "%Y-%m-%d %H:%M:%S.%f")
        x_date = datetime_obj.date()
        x_time = datetime_obj.time()

        rfid_lib.append((i, hospital, floor, ward, random.choice(linen_type), status_s, x_date, x_time))


    pass_admin_db(rfid_lib)
```

Figure 5.3.2.1 Coding algorithm for attributes.

5.3.3 Menu-driven Interface

I have created a menu-driven interface for the program. It provides a user-friendly interface for hospital staff. Hospital staff can choose whether the linen is outsourcing or incoming, terminate the program, or whether they want to finalize and update the scanned linen data in the database. This feature offers the hospital staff a second chance to check or decide to update the scanned linen data to the admin database.

```python
def menu():
    loop = 1
    while loop == 1:
        print("==================== Linen Monitoring System ====================")
        print(
            "1.Scan Outsourcing Linen\n2.Scan Incoming Linen\n3.Shutdown\n")
        option = int(input(" Choose Your Option: "))
        if option == 1:
            os.system("cls")
            print("Scanning Outsourcing Linen..................")
            status_scan = "outsourcing"
            scan_outsourcing()
            choose = int(input("Do You Want To Update The Scanned Outsource Linen To Admin & Storage?\n Type 1 for "
                               "Yes\n Type 2 for No\nOption: "))
            if choose == 1:
                os.system("cls")
                print("Updating Scanned Linen to Admin & Storage, In progressing .......")
                # Function Update to Admin Database
                update_clear_db(status_scan)
                print("Scanned Outsourcing Linen has updated to Admin & Storage, Linen has been permitted to "
                      "outsource by admin.")
            else:
                os.system("cls")
                print("Scanned Outsourcing Linen has been denied to outsource by admin.")
                print("Back to menu, Loading .......")
                # Clear all the list to allow used in the future scan
                rfid_scan.clear()
                timestamp.clear()
                rfid_rawdata.clear()
                rfid_raw_time.clear()
                temp_source_date.clear()
                temp_source_time.clear()
                print("Previous Scanned RFID has been cleared, Can Proceed with another scan.")
                print("..................")
                os.system("cls")
```

Figure 5.3.3.1 Menu-driven Interface Function (Part 1).

```
    elif option == 2:
        os.system("cls")
        print("Scanning Incoming Linen.................")
        status_scan = "incoming"
        scan_incoming()
        choose = int(input("Do You Want To Update The Scanned Incoming Linen To Admin & Storage?\n Type 1 for "
                           "Yes\n Type 2 for No\nOption: "))
        if choose == 1:
            os.system("cls")
            print("Updating Scanned Linen to Admin & Storage, In progressing .......")
            # Function Update to Admin Database
            update_clear_db(status_scan)
            print("Scanned Incoming Linen has updated to Admin & Storage, Linen has been permitted to source into "
                  "hospital by admin.")
        else:
            os.system("cls")
            print("Scanned Incoming Linen has been rejected by admin.")
            print("Back to menu, Loading .......")
            # Clear all the list to allow used in the future scan
            rfid_scan.clear()
            timestamp.clear()
            rfid_rawdata.clear()
            rfid_raw_time.clear()
            temp_source_date.clear()
            temp_source_time.clear()
            print("Previous Scanned RFID has been cleared, Can Proceed with another scan.")
            print("..................")
            os.system("cls")

    elif option == 3:
        loop = 0

    else:
        os.system("cls")
        print("Invalid Option, Please Try Again")
        menu()

print("Program terminated manually!")
raise SystemExit
```

Figure 5.3.3.2 Menu-driven Interface Function (Part 2).

5.3.4 Linen Log Database Table

I have created a database table named "tracktable" to store all the linen log records. Once the scanned linen is triggered to update the status in the admin database table, it will automatically pass and store all the updated linen log records in the linen log database table. This "tracktable" is utilized to visualize all the log records for the hospital staff in the dashboard.

```
def create_db():
    hostname = "localhost"
    database = "linenmonitordb"
    username = "postgres"
    pwd = "admin"
    port_id = 5432
    conn = None
    cur = None
    try:
        conn = psycopg2.connect(
            host=hostname,
            dbname=database,
            user=username,
            password=pwd,
            port=port_id
        )
        cur = conn.cursor()
        # create Admin Management DB (which contain all the RFID tags information)
        create_administrator_table = ''' CREATE TABLE IF NOT EXISTS admintable (
                                    rfid_tag            text PRIMARY KEY,
                                    hospital_type       text,
                                    floor               int,
                                    ward                int,
                                    linen_type          text,
                                    status              text,
                                    date                text,
                                    time                text
        )'''
        create_track_table = ''' CREATE TABLE IF NOT EXISTS tracktable (
                                    rfid_tag            text,
                                    hospital_type       text,
                                    floor               int,
                                    ward                int,
                                    linen_type          text,
                                    status              text,
                                    date                text,
                                    time                text
        )'''
        cur.execute(create_administrator_table)
        cur.execute(create_track_table)
        conn.commit()
    except Exception as error:
        print(error)
    finally:
        if cur is not None:
            cur.close()
        if conn is not None:
            conn.close()
```

Figure 5.3.4.1 Create** a **Linen Log Database Table Function.

```python
def pass_track_db(arr):
    hostname = "localhost"
    database = "linenmonitordb"
    username = "postgres"
    pwd = "admin"
    port_id = 5432
    conn = None
    cur = None
    try:
        conn = psycopg2.connect(
            host=hostname,
            dbname=database,
            user=username,
            password=pwd,
            port=port_id
        )
        cur = conn.cursor()
        insert_data = 'INSERT INTO tracktable ( rfid_tag, hospital_type, floor, ward, linen_type, status, date, time) '\
                      'VALUES (%s, %s, %s, %s, %s, %s, %s, %s) '
        for i in arr:
            cur.execute(insert_data, i)
        conn.commit()
    except Exception as error:
        print(error)
    finally:
        if cur is not None:
            cur.close()
        if conn is not None:
            conn.close()
```

Figure 5.3.4.2 Pass Data into** the **Linen Log Database Table Function.

5.4 Implementation Issues and Challenges

In Project 1, The first implementation issue is faulty hardware. The RFID reader is not stable with the signal and network, causing the data can't be passed thru the reader to the laptop. The IP address and setup configuration of the RFID reader has gone. However, it is solved by the lab assistant, Dr Goh, and the MDT Innovation Sdn Bhd's IT support during week 10. The second implementation issue is my faulty laptop's ethernet port, which I discovered in week 11. My ethernet port went wrong since it was not used for a long time. I have decided to buy a USB-ethernet converter to replace the laptop's ethernet port. To avoid delaying progress, I have used assumption for the information of RFID Tag by random generator python function and stored it into the array.

In Project 2, I did not have any technical hardware issues, but I faced some challenges while implementing the code or program. Due to the assumption case in my project 1, I was required to modify my Python program to make all the functions connected and work practically. I used up a whole month to understand all the algorithms between the scanner, coding, database, and dashboard. Furthermore, primary key violation when there is absolutely no data in the database table is also one of the most challenging issues I faced. I tried creating a table without a primary key, but it resulted in duplicated data. I have googled and found someone is having the same issue as me. It certainly appears that the data files may be somehow being cached by the PostgreSQL database. To solve this issue, I deleted the database table, which violated the primary key and tried to extract data from another database table that functions well.

Chapter 6

System Evaluation And Discussion

In this chapter, I will briefly discuss my system evaluation. In order to determine its effectiveness, efficiency and overall quality of the system, I have analyzed and assessed the performance of my system. System evaluation is critical for ensuring the RFID linen tracking system functions effectively and efficiently and can be used in UTAR hospital.

6.1 System Testing and Performance Metrics

I have conducted a few system performance tests to measure the performance of the RFID Linen Tracking system. I have utilized the performance metrics, including response time, latency, and data accuracy, to perform multiple performance testing on the RFID linen tracking system. This system performance testing will ensure that the RFID linen tracking system meets the required performance standards set by the UTAR hospital and can handle the expected workload.

6.1.1 Linen Data Accuracy (Accuracy Testing):

I have performed the linen data accuracy testing by comparing the number of tags in real life, program, and database to ensure that the system can accurately identify and track linen items and that the data captured by the system is correct and reliable. For instance, I compare the number of tags placed near the antenna, the number of tags scanned, and the number of linen logs recorded in the database. I put an exact amount of RFID tags on the antenna, then ran the program to determine how many RFID tags were being scanned and how many linen logs were recorded in the database. To ensure the system functions well, I tested 20 times for different data and recorded the output of the system performance.

No.	Number of tags placed near the antenna	Number of scanned tags	Number of logs recorded in database (tracktable)
1	21	23	23
2	40	40	40
3	13	15	15
4	29	29	29
5	31	32	32
6	27	27	27
7	28	28	28
8	17	17	17
9	26	26	26
10	18	18	18
11	21	24	24
12	19	19	19
13	41	41	41
14	34	34	34
15	10	13	13
16	25	25	25
17	5	7	7
18	28	28	28
19	19	19	19
20	15	15	15

Table 6.1.1.1 Linen Data Accuracy Table

6.1.2 Duration required by database & dashboard to fetch linens data (Response Time Testing)

I have performed response time testing on the database and dashboard by measuring the count of force refresh needed to fetch the linen data, store them in the database and upload them into the dashboard. For example, I will let the program run to scan RFID tags for 5 seconds. Once the threshold is reached, I will click on the force refresh button in the database and dashboard until the data is uploaded. To ensure the system functions well, I tested different data and recorded the outcome of the system performance over 20 times.

No.	Number of tags	Force refresh count on each platform	Database Data	Dashboard data
1	21	1	Updated	Updated
2	40	2	Updated	Updated
3	13	1	Updated	Updated
4	29	1	Updated	Updated
5	31	2	Updated	Updated
6	27	1	Updated	Updated
7	28	1	Updated	Updated
8	17	3	Updated	Updated
9	26	1	Updated	Updated
10	18	1	Updated	Updated
11	21	1	Updated	Updated
12	19	1	Updated	Updated
13	41	2	Updated	Updated
14	34	1	Updated	Updated
15	10	1	Updated	Updated
16	25	1	Updated	Updated
17	5	1	Updated	Updated
18	28	2	Updated	Updated
19	19	1	Updated	Updated
20	15	1	Updated	Updated

Table 6.1.2.1 Refresh Count by Database and Dashboard

6.2 Performance Testing Result

The performance testing objective was to determine the system's effectiveness, efficiency and overall quality. The testing was conducted using my laptop with an Intel Core i7, 8GB RAM, a PostgreSQL database and a Microsoft Power Bi Dashboard. There are 55 RFID tags ready to be scanned by the antenna, read the RFID reader and display the linen data on my laptop. I have conducted two performance tests: linen data accuracy testing and response time testing with the duration required by the database and dashboard to fetch linens data. Each of the tests will be conducted 20 times.

Throughout the multiple system performance tests for the linen data accuracy, I could state that the data linen accuracy is quite high. Although there are some differences in data for the number of tags placed near the antenna and the number of scanned tags in a few tests, I think it shouldn't be a problem for the system. This is because I was unable to predict or measure the exact range of the antenna that can detect RFID tags. So, the antenna may detect some of the RFID tags placed further away from it.

While for the response time testing on the database and dashboard, the system provided quick and efficient data for inventory tracking management, with an average one force refresh attempt for data retrieval of the database and data updates in the dashboard. Due to the poor network bandwidth, the system may need more than 1 force refresh to retrieve the linen data into the database and dashboard. But it was not a big issue that would affect the performance of the system.

All in all, no significant performance issues or bottlenecks were identified during the performance testing. In general, the results of the performance tests unquestionably showed that the system has met the expected levels of services, as well as delivering the hospital employees a positive user experience.

6.3 Project Challenges

RFID Linen Tracking System is a technology-based and user-friendly system. It is mainly functioned to monitor and track linen inventory movement in UTAR hospital. Nevertheless, there are an array of difficulties and obstacles that will be faced throughout the route of completion and implementation of the linen tracking system. It is important to foresee potential project issues that would face along and learn to deal with them when they arise to build resilience capacity. In this section, I will describe the project challenges and, most importantly, the solutions for conquering them.

The first struggle is high initial investment cost. Implementing an RFID linen tracking system can be expensive because it necessitates the installation of hardware like as RFID tags, readers, and related software for a large hospital. Worst still, a higher cost is required to cover as, in this project, there are two hospitals which are TCM and WM. Hence, it is important to perform a cost-benefit analysis to estimate and enumerate the expected return on investment (ROI) during the installation of the RFID linen tracking system. The analysis should take into account of the potential benefits such as greater productivity, efficient inventory counts, lower losses and reducing yearly linen purchase and labor saving.

The second bump throughout the route is privacy and transparency concerns. RFID technology utilizes radio frequency for communication is at risk of interception by unauthorized parties. This can pose questions about the privacy and security of the transmitted data. Thus, an adequate security and standard security measure should be practiced avoiding unauthorized access and safeguard the data.

The third problem that will be encountered is staff training. A pre-training is required to implement of an RFID linen tracking system. Employees should be trained and equipped with the essential skills for the utilization of the hardware components, interpretation of data and rectification of common problems that may encounter in future.

In short, implementing an RFID linen tracking system and digitalizing linen management in hospitality could be challenging; however, on the other side of the coin, this system yields significant benefits, including reduction of costs, optimization of inventory flow and quick identification of variances. To overcome these obstacles and ensure the effective use of the system, it is crucial to meticulously plan the implementation process through organizing pre-training for hospital staffs and ensuring that the system is equipped with measures should be equipped with proven and standard security measures.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

In a nutshell, this report was about the final year project that developed a gantry-based UTAR Hospital Linen Tracking through RFID using Python that enhanced the efficiency of the existing linen tracking system. RFID solutions offer a more enhanced version of inventory management and accounting procedures that make the transition of linen smoother and more efficient by replacing manual input or tracking each valued asset.

In addition, a gantry-based linen tracking system using RFID technology tends to boost the linen tracking system's transparency to eliminate linens' losses, reduce restocking costs, and compensate for the losses. RFID solutions enable linen tracking systems to trace all the linen, dramatically reducing linen loss. Besides that, the gantry-based UTAR Hospital linen tracking system through RFID using Python also aims to minimize the workload of the hospital staff. RFID solutions allow hospital staff to relieve the burden of manually recording and uploading the linen data. The laundry plant suppliers can check the daily outsourcing linen amount from the hospital via the user-friendly dashboard directly connected to the database.

Throughout this final year project, I gained interesting deep knowledge about RFID technology and embedded skills. I have integrated my embedded expertise and research on RFID technology to develop this final-year project. I was rewarded with an opportunity to gain experience in embedded system development related to IoT by setting up the hardware and software.

In order to achieve all the objectives, I will undergo progress checking every week to ensure that I'm on track and that my product is related to the objective. Therefore, I'm able to submit this project's report on time during week 12 and present the product on week 12. I hope my final year project will assist in the development of UTAR Hospital.

7.2 Recommendation

Based on the findings of this entire project, there are a variety of recommendations for future work to enhance the system's efficiency and effectiveness. First of all, the RFID linen tracking system can be integrated with other systems, like laundry management systems and inventory management systems, to provide a more comprehensive tracking and management solution for linens. By integrating with other systems, the RFID system is able to share data and work together to deliver a seamless user experience. For instance, integration with a laundry management system can allow the RFID system to track and monitor the entire lifecycle of linen, from its initial dispatch to the laundry facility to its return to the hospital. This integration may boost the visibility and transparency of the linen management process and facilitate more accurate and efficient inventory management.

In addition, a mobile application can be developed to provide staff with access to the RFID linen tracking system and real-time linen status updates. This mobile application can improve linen management's efficiency, mobility, and accuracy by enabling staff to update linen status from anywhere in the hospital swiftly and effortlessly. The mobile application is capable of providing employees with real-time alerts and notifications when linens need to be replenished, or there are linen availability issues. Besides, a mobile application can be modified to provide distinct views and access levels for various user groups, such as nurses, domestic staff, and laundry staff.

Aside from that, staff can be provided with adequate user training and support to ensure that the RFID linen tracing system is used correctly and effectively. This staff training and support can include onsite training sessions, user guides, and online support materials. Training can involve scanning RFID identifiers, updating linen status, and resolving common problems. Employees can confidently utilise the system and maximise its benefits by providing adequate training and support. Training and support can be provided continuously to ensure employees are aware of system updates and new features.

By implementing these recommendations, the RFID linen tracking system can be enhanced to provide more accurate, efficient, and effective linen management. The integration with other systems can provide greater visibility into the linen management process, the mobile application can improve efficiency, mobility, and accuracy, and user training and support can assure staff confidence in using the system.

REFERENCES

[1] "How is the tracking of hospital equipment beneficial? - asset infinity," 06-Apr-2020. [Online]. Available: https://www.assetinfinity.com/blog/tracking-hospital-equipment-beneficial. [Accessed: 31-Aug-2022].

[2] "RFID vs Barcode comparison advantages disadvantages," 21-Nov-2019. [Online]. Available: https://www.peaktech.com/blog/rfid-vs-barcode-comparison-advantages-disadvantages/#:~:text=Cost%20Comparison,more%20expensive%20than%20barcode%20scanners. [Accessed: 31-Aug-2022].

[3] "The shocking price of RFID tags," 11-Sep-2016. [Online]. Available: https://www.advancedmobilegroup.com/blog/the-true-price-of-rfid-tags#:~:text=Passive%20RFID%20tags%20tend%20to,signal%20to%20the%20RFID%20system. [Accessed: 18-Apr-2022].

[4] "What is a barcode? | technical information of automatic identification," Denso Wave. [Online]. Available: https://www.denso-wave.com/en/adcd/fundamental/barcode/barcode/index.html. [Accessed: 31-Aug-2022].

[5] A. M. Peres, L. A. D. Braccialli, S. M. Pirolo, E. de F. Ribeiro Higa, and M. Mielo, "HOSPITAL LINEN AND HEALTH CARE: PERSPECTIVE OF PROFESSIONALS AND STUDENTS," 28-Nov-2017. [Online]. Available: https://docs.bvsalud.org/biblioref/2018/05/883494/53413-231504-1-pb.pdf. [Accessed: 30-Aug-2022].

[6] A. S. Prakash, M. Solayappan, and U. D. Kumar, "Managing Linen at Apollo Hospitals," May-2018. [Online]. Available: https://pdfcoffee.com/apollo-hospitals-pdf-free.html#Sneha+Aggrawal. [Accessed: 18-Apr-2022].

[7] C. Finch, "Advantages & disadvantages of RFID," Techwalla. [Online]. Available: https://www.techwalla.com/articles/advantages-disadvantages-of-rfid. [Accessed: 31-Aug-2022]

[8] S. Amsler and S. Shea, "RFID (radio frequency identification)," 31-Mar-2021. [Online]. Available: https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-

identification#:~:text=Every%20RFID%20system%20consists%20of,fixed%20 readers%20and%20mobile%20readers. [Accessed: 30-Aug-2022].

[9] *S. K. Ramasamy and S. Hassan, "Radio Frequency Identification (RFID) adoption in Healthcare Linen Management: A case study at ProLinenCare," 30-Jun-2018. [Online]. Available: https://e-journal.uum.edu.my/index.php/gbmr/article/view/11057. [Accessed: 31-Aug-2022].*

[10] *University of Washington, "RFID fact sheet." [Online]. Available: https://rfid.cs.washington.edu/posters/RFID-fact-sheet.pdf. [Accessed: 30-Aug-2022].*

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	Study week no.: 2
Student Name & ID: Fong Karr Keet 1903625	
Supervisor: Dr. Goh Hock Guan	
Project Title: DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Review the FYP 2 guideline.
- Identify the task.

2. WORK TO BE DONE

- Review back the FYP 1.

3. PROBLEMS ENCOUNTERED

- N/A

4. SELF EVALUATION OF THE PROGRESS

- Spend more time on reading the FYP 2 guideline.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 4**
Student Name & ID: Fong Karr Keet 1903625	
Supervisor: Dr. Goh Hock Guan	
Project Title: DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Review back the Python program in FYP 1.

2. WORK TO BE DONE

- Attempt to change the program from virtual scenario to real-time scenario.

3. PROBLEMS ENCOUNTERED

- N/A.

4. SELF EVALUATION OF THE PROGRESS

- Should start earlier on review the Python program in FYP 1.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 6**
Student Name & ID: Fong Karr Keet 1903625	
Supervisor: Dr. Goh Hock Guan	
Project Title: DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Continue coding part of the project.

2. WORK TO BE DONE

- Attempt to enhance real-time scenario Python program.

3. PROBLEMS ENCOUNTERED

- Stuck in infinite loop for the scanning part.

4. SELF EVALUATION OF THE PROGRESS

- Should start earlier on hands on the hardware.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 8**
Student Name & ID: Fong Karr Keet 1903625	
Supervisor: Dr. Goh Hock Guan	
Project Title: DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Continue coding part of the project.

2. WORK TO BE DONE

- Attempt to solve the infinite loop at the scanning part.

3. PROBLEMS ENCOUNTERED

- N/A.

4. SELF EVALUATION OF THE PROGRESS

- Should solve the infinite loop earlier.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 10**
Student Name & ID: Fong Karr Keet 1903625	
Supervisor: Dr. Goh Hock Guan	
Project Title: DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON	

1. WORK DONE [Please write the details of the work done in the last fortnight.] • Start to write the report for FYP2. • Setup and Connect the Database and Dashboard
2. WORK TO BE DONE • Done the report until Chapter 6.
3. PROBLEMS ENCOUNTERED • Database Primary Key Violation
4. SELF EVALUATION OF THE PROGRESS • Should done all the coding part earlier.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	Study week no.: 12
Student Name & ID: Fong Karr Keet 1903625	
Supervisor: Dr. Goh Hock Guan	
Project Title: DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Finalize Report.

.

2. WORK TO BE DONE

- Compile the code, database, and dashboard.

3. PROBLEMS ENCOUNTERED

- N/A.

4. SELF EVALUATION OF THE PROGRESS

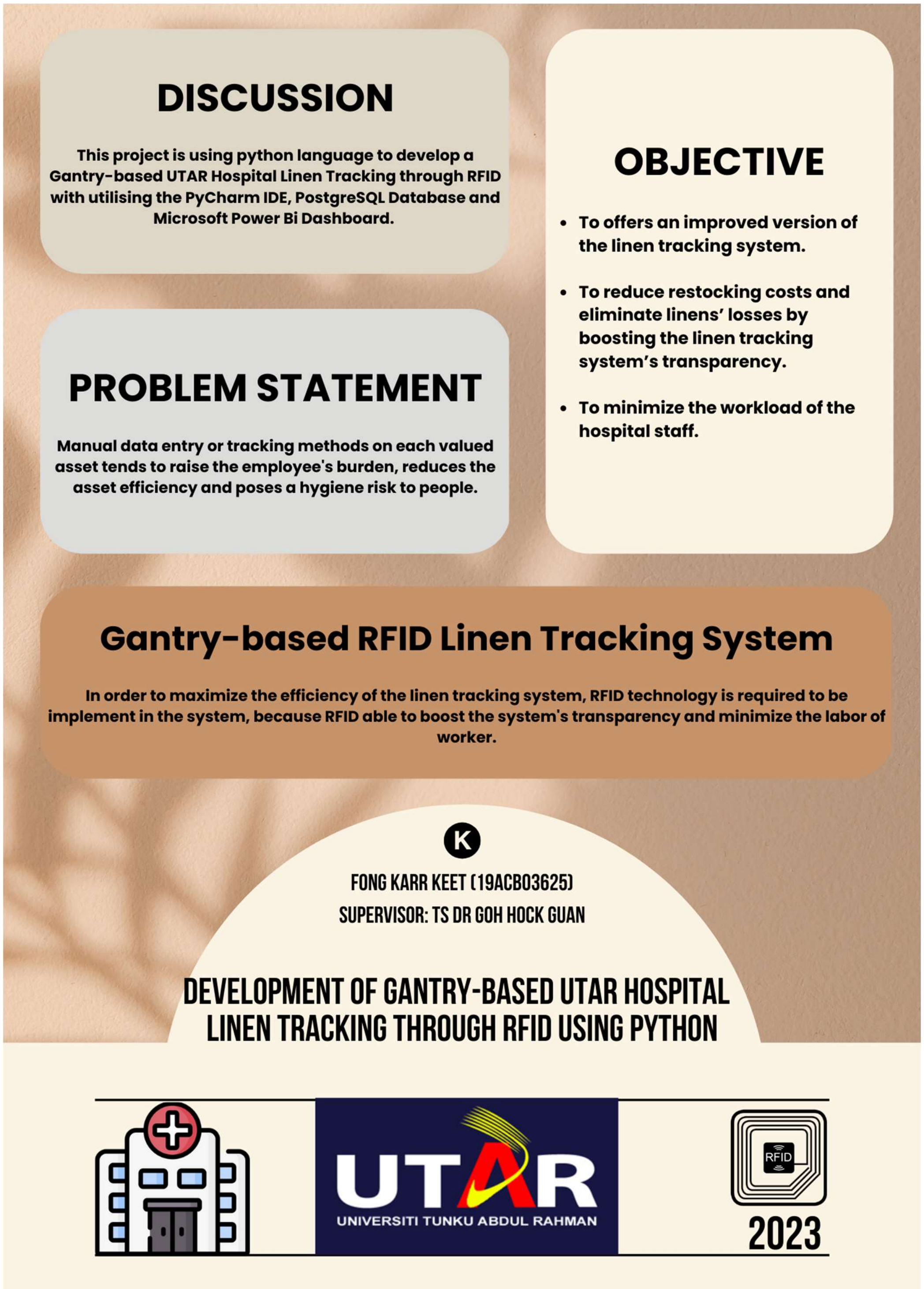
- Should done all the coding part earlier.

Supervisor's signature

Student's signature

APPENDIX

POSTER



PLAGIARISM CHECK RESULT

DEVELOPMENT OF GANTRY-BASED UTAR HOSPITAL LINEN TRACKING THROUGH RFID USING PYTHON

ORIGINALITY REPORT

2%	1%	1%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	e-journal.uum.edu.my Internet Source	1%
2	Submitted to Ngee Ann Polytechnic Student Paper	1%
3	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1%
4	Submitted to Midlands State University Student Paper	<1%
5	Submitted to Dulwich College Singapore Student Paper	<1%
6	Submitted to Griffith College Dublin Student Paper	<1%
7	Submitted to Kaplan College Student Paper	<1%

Exclude quotes On

Exclude matches < 8 words

Universiti Tunku Abdul Rahman			
Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Fong Karr Keet
ID Number(s)	1903625
Programme / Course	Bachelor of Information Technology (Honours) Communications and Networking
Title of Final Year Project	Development of Gantry-based UTAR Hospital Linen Tracking through RFID using Python

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 2 % Similarity by source Internet Sources: 1 % Publications: 1 % Student Papers: 2 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Goh Hock Guan	Name: ____________
Date: 26/4/2023	Date: ____________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	1903625
Student Name	Fong Karr Keet
Supervisor Name	Dr. Goh Hock Guan

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: 25/4/2023