LAB INVENTORY MONITORING SYSTEM USING LOW CODE PROGRAMMING

BY

ONG TING XI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ts Dr Goh Hock Guan, for providing me the opportunity to engage in the inventory monitoring system development project. It is my first step to build a career in system development field, and I am truly grateful for this invaluable experience. Thank you for your guidance and support.

ABSTRACT

The area of study for this project is web application development. This project aims to develop an inventory monitoring web portal for solving the problems lacking on existing inventory monitoring systems.

Nowadays, the existing systems still have complex interfaces and navigation structures, which causes time consuming and high cost in adapting this kind of system. Moreover, there are many business enterprises still rely on paper works and excel tables in monitoring inventory. This causes a lot of external error occurring during inventory monitoring and affects the accuracy of monitoring and reporting which resulting below standard level. Additionally, the existing systems don't provide further general instruction for addressing abnormal items. This will cause the unnecessary workflows to be repeated persistently and reduce the efficiency of operation.

Furthermore, the proposed inventory monitoring web portal is developed by using low-code programming, which is Node-RED. Then, this web portal is integrated with database and various APIs which consists of Telegram chatbot API, MySQL database, and so on. Besides that, the process of development of inventory monitoring web portal follows the standards of Agile model.

Then, this project has also conducted literature review on 4 existing inventory monitoring systems for finding out their limitations. The proposed system allows improve the accuracy and efficiency in monitoring inventories.

Consequently, this project develops an inventory monitoring web portal by utilizing the low-code development tool to address the key challenges faced by the existing systems.

Area of Study (Minimum 1 and Maximum 2): Web Application Development

Keywords (Minimum 5 and Maximum 10): Inventory Monitoring, Low-code Programming, Web Application, Node-RED, Alert System

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SQL	Structured query language
MVCC	Multi-Version Concurrency Control
PL/pgSQL	Procedural Language/PostgreStructured query language
PL	Procedural Language
DDL	Data Definition Language
API	Application Programming Interface
RSA	Rivest–Shamir–Adleman
SSH	Secure Shell
SSL	Secure Sockets Layer
TLS	Transport Layer Security
AES-256-CBC	Advanced Encryption Standard - 256-bit - Cipher Block Chaining
AES	Advanced Encryption Standard
CBD	Cipher Block Chaining
MD5	Message-Digest Algorithm 5
SHA-3	Secure Hash Algorithm 3
SHA-2	Secure Hash Algorithm 2
SHA-256	Secure Hash Algorithm 256-bit
FPGA	Field-Programmable Gate Array
ASIC	Application-Specific Integrated Circuit
MAC	Media Access Control
SDLC	Software Development Life Cycle
SN/PN	Serial Number or Part Number
ID	Identification
FYP1	Final Year Project 1
FYP2	Final Year Project 2
GUI	Graphical User Interface

Chapter 1 - Introduction

This chapter describes the project background, motivation of this project, the problem statement, and contributions to the inventory monitoring web portal.

With the advancement of technology, inventory monitoring systems have become essential and important tools for most business activities. It can handle the tasks which were completed manually previously. For instance, stock tracking, balance sheet creation, and so on. The business owners can utilize it to reduce efforts, time, and cost in monitoring and managing the inventory [1]. In other words, the purpose of inventory monitoring system is improving efficiency, accuracy, and cost-effectiveness. Then, this kind of the system is suitable for any business whether it is big or small such as retails, warehouses, business that is managing IT assets, and others [2], [3], [4].

Moreover, the poor performance inventory monitoring system should be avoided as it may cause stockouts, overstocking, misplacement of items, inaccurate records, and so on. These inefficiencies will increase the operational costs such as extra storage fees, wasted labor hours, and so on.

Then, for developing this kind of the system application, the developers usually choose to utilize PHP, C#, jQuery, MySQL, and so on [1], [5]. These development tools are high-code programming, which provides flexibility in the system application and database developments.

However, there's a trend nowadays, the developers will switch to low-code programming for development progress. Low-code programming allows the system application development progresses to become more simplified, which can speed up the progress of application development. Then, it also allows the developers who without programming background to be included in the development lifecycle, and still able to maintain productivity in developing the new system applications [6]. Moreover, the new application developed which utilizes low-code programming is expected to occupy 70% of new system applications developed by businesses in 2025 [7].

1.1 Motivation and Problem Statement

In recent years, there have been many inventory monitoring system applications on the market. However, most of them contain complex system application structures. Then, usually they need sufficient user training sessions since they are difficult to understand [8]. In other words, they need to spend a lot of time implementing and adapt inventory monitoring system applications. As a result, these applications are causing time consuming, high cost, and even low accuracy in operating the businesses.

Furthermore, the existing inventory monitoring system applications can detect and alert the abnormalities on the inventory which may include the damaged inventory [9]. However, these applications don't provide further and general instructions in processing progress for addressing abnormal items, which may waste time on handling and tracking the same problem on the items. If the staff doesn't have experience in solving the common issues on the inventory, they will consume unnecessary time on finding other departments or colleagues for the solutions. As a result, the waste of time may cause low efficiency in managing inventory.

Moreover, another critical problem with monitoring and managing inventory is low accuracy which is due to manual methods. Human error is a common mistake which exists in current inventory monitoring systems [10]. Manual methods such as entering items' information into the Excel tables or relying on paper forms require the staff to pay high attention in managing a huge number of items. However, it is impossible to maintain good attention during work so there is a high rate of mistakes.

This project is motivated by the desire to provide the inventory monitoring web portal, which is efficient, ease-to-use with rich features, and high accuracy for solving the common issues in general inventory monitoring system. Besides that, this project is also motivated by exploring the utilization of low-code programming (Node-Red) for web development which is a renovation method to develop inventory monitoring web portal in more simple way compared to traditional or high-level programming languages.

1.2 Project Objectives

The aim of the project is to propose a user-friendly inventory monitoring web portal that able to solve the issues lacking existing inventory monitoring systems. The target users of this system are the students, supervisors, staff, and head of department of university.

1.2.1 To provide clear and user-friendly user interfaces and navigations in monitoring inventories

The duration of learning and adapting to a new system is critical to the operation of business. The proposed system is aimed to provide a user-friendly environment for monitoring the inventory effectively. For example, the position of menu, the options' sequence within the menu, the design of lists, and so on.

1.2.2 To minimize the external errors in managing inventory records

By providing the restrictions, rules, and automation on inventory monitoring web portal, it can reduce the external errors such as human errors, missing paper forms, typo error on excel files and so on.

1.2.3 To improve efficiency in handling inventories

General instructions for addressing issues on inventory is a key action, it can save time and streamline the process of handling issues. Moreover, the tracking of progress for handling each item is also critical to the operations as it can improve efficiency in monitoring the status of inventories.

1.3 Project Scope

The main scope of the project is providing inventory monitoring web portal that is effective, high accuracy, cost effective, and ease for implementation and maintenance. Besides that, the project also has the following scopes:

1.3.1 User accounts management

The developed system allows administrators and lab staff to register accounts, assign roles with different levels of access control. For instance, only lab staff and admin can manage the item details while the others can't perform any action on modifying the item details. Then, the students are also allowed to register accounts, but the registration of accounts needs to be

approved by lab staff. Moreover, the history of editing also able to show the name of the staff who created and last modified the data. As a result, this system provides data integrity within the inventory stock.

1.3.2 Inventory data registration and management

This project provides a user-friendly interface and navigation for managing inventory details. The management consists of adding, modifying, and removing. Moreover, the users are also able to manage the information of locations where the items are stored and manage the list of information types. They can change location, category, or item type for a group of items at once by changing the respective name of location, category, or inventory type through the locations management module, categories management module, and inventory types management module of this application.

1.3.3 Application for loan items

The system also provides a list of various types of items and forms for students and supervisors to apply for loan the selected items such as monitors, projectors, and so on. For the item list, they can gain the detailed information of each item through the list, which includes item name, status, images of respective items, and so on. This can help them to select the item for loan correctly and efficiently. Then, the supervisors will evaluate form from students while administrators will evaluate the form from supervisors to decide whether to approve or reject the application. Additionally, the lab staff will perform the final confirmation on the applications. Lastly, the lab staff will update the application's status when the loaned item has been returned. The old returned item records also can be removed.

1.3.4 Tracking of abnormal inventory

The users can provide comments regarding processing progress according to the status of abnormal items. This allows high-level users to perform tracking processing progress conveniently about those abnormal items.

1.3.5 Automation on alert and notification

This system can send out alerts and notification automatically through detecting events like application of loan, approaching deadline, and so on. For example, when a student applies to

an application for loaning an item, the lab staff will receive notification regarding this application through Telegram.

1.4 Main Contributions from the Project

1.4.1 Web-based Accessibility

This project proposes an enhanced inventory monitoring web portal using low-code programming, Node-Red, which provides simple and clear user interface and navigation in managing inventory records and applications. This web portal is responsive for various devices with different screen resolutions which are accessible from any device such as tablet, desktop, and so on.

1.4.2 Effective Abnormality Handling

This proposed system able to report and track the processing progress of abnormal items which allows the users to reduce effort in monitoring this kind of inventory through manually asking for the respective progress.

1.4.3 Enhanced Data Accuracy and Integrity

The proposed system ensures the accuracy and integrity of items' information by providing restrictions, rules, and automation in particular sections when entering or modifying the items' information, applications, and user accounts. For example, the lab staff must enter the tag number of the item during the registration to prevent any important data from missing. Moreover, the system will also detect and avoid the duplicated value for registering items to ensure that each item is unique and can be identified.

1.5 Organisation of the Report

The following chapters present the details of this study. The details of literature reviews on technologies and previous works are presented in Chapter2 which regarding inventory monitoring system. Then, Chapter 3 describes the system methodology of this project. Then, Chapter 4 describes the details about the whole system design of this project. Furthermore, the system implementation is presented in Chapter 5. Besides that, the details of system evaluation and discussion are presented in Chapter 6. Lastly, the last chapter, Chapter 7 includes the conclusion and recommendations about this project.

Chapter 2 - Literature Review

2.1 Review of the Technologies

2.1.1 Database

PostgreSQL

An open-source object-relational database that extends the traditional relational model with object-oriented features. It follows the SQL standard as closely as possible although some way may be implemented with minor variations [11].

One of PostgreSQL's primary advantages is supporting various programming languages like C++ and Java, and offers cross-platform compatibility. Then, it supports complex queries, primary/foreign keys, triggers. Updatable views, and a comprehensive transaction concept [12]. Additionally, it supports Multi-Version Concurrency Control (MVCC) for efficient handling of concurrent database access [12]. Furthermore, it also supports advanced features like custom data types, user-defined functions, and even procedural languages (PL/pgSQL, PL/Python, Java, and others), making it highly extensibility. Its ability to include DDL commands in transactions that simplifies the transactions management in a complex relational application [13].

However, PostgreSQL has some drawbacks. Being a traditional relational database, it requires rigid schema definitions, meaning each table record must conform to a predefined structure. This limits the flexibility compared to NoSQL databases like MongoDB. Furthermore, as an open-source system, PostgreSQL lacks centralized ownership, commercial support, or warranties, which may present risks for enterprise users. This decentralization can also lead to inconsistencies in user experience across different deployments [13].

MongoDB

According [14], MongoDB is a free and open-source NoSQL database system designed as an alternative to conventional relational databases. It is particularly effective for handling large volumes of distributed data. Moreover, it offers support for wide range of programming languages, such as Java, Python, C, C++, C#, and so on.

MongoDB offers several benefits, particularly its schema-less design, which allows for flexible data storage without predefined schemas. This flexibility makes it easier to scale databases compared to the relational databases. As a document-oriented database, MongoDB stores data as documents that map to native data types in various programming languages, reducing the

need for complex joins and improving performance. Additionally, its horizontal scalability through sharding makes it well-suited for big data applications. MongoDB also supports third-party storage engines and provide the pluggable APIs, offering customization options.
However, MongoDB also has some drawbacks. The failover process for continuity can take up to a minute since it relies on a single master node. The write performance is also limited as the volume of information that written to the database is relied on the capacity of its single master node. MongoDB lacks referential integrity, as it does not enforce foreign key constraints, which may impact data consistency.

MySQL

This project has applied MySQL database for managing records about inventories. According [15], the MySQL is a open-source relational database. One of its greatest strengths is the large and active community that surrounds it. Developers can easily access countless resources, tutorials, forums, and documentation, making troubleshooting and optimization.
Moreover, MySQL stands out as one of the fastest-performing databases, offering excellent speed and efficiency. Its support for multi-threading contributes to better optimization and responsiveness. Another key advantage of MySQL is its platform independence. Whether deploying an application on Linux, Windows, UNIX, or other operating systems, MySQL ensures the satisfaction for the needs,
However, its performance may decline under heavy workloads. This makes it not suitable to support for large-scale operations.

2.1.2 Programming language

Power Apps

Microsoft Power Apps is a low-code programming language that offers a rapid development environment in building custom apps based on the needs [16]. It allows developers to develop application through user-friendly method likes drag and drop method and also includes some coding. For coding the elements on the application, it has its own unique programming language which is named Power Fx [17]. This programming language is usually used to handle the events of each page, database connection, form configuration, calculation on specific data, and so on. Moreover, it supports wide range of data source, such as SharePoint lists, SQL server, Microsoft 365, and so on. Additionally, it provides integration with Power BI and

Dynamic 365, which provides a platform for designing the dashboard for monitoring the business operations [16].

According to [18], one of the weaknesses of Power Apps is it only allows one developer to edit the app at a time. This means that each project is only suit to a single developer as it is time consuming for a large project with a team of developers to utilize it in the development of system or application. Moreover, it is also challenging to debug as it will only display the error in the term of notification and the given information sometime is not clear. For example, there are different causes of error, but this platform displays the same information about the error. This will cause developers spend a lot of time for testing and debugging, which reduce the efficiency in development of application.

Gilde

Glide is a low-code platform that enables users to transform spreadsheets into functional mobile and web applications [19]. It allows developers to use pre-built component and templates to develop applications or systems in a simple and easy way, and also developers to tailor the design and functionality to their specific needs [20]. The way to design webpage is through adding the component into the webpage by just clicking the preferred components and adjust the position of components by arranging the list of components. It will automatically set each component in responsive style, which makes developers reduce efforts in producing the responsive web pages [21].

One of the advantages of using Glide is its data integration. It supports smooth integration with SQL, Google Sheets, BigQuery, and so on. Then, it allows developers to develop the full responsive application across different devices, from mobile phones to desktops, ensuring user experiences. Then another advantage is ease-to-use, the developer doesn't require coding language for developing the applications.

However, Gridle has limited customization. Due to apply responsive web page automatically, Gridle does not provide many options to developers for adjusting and designing. Most of the components are only able to design based on the pre-built styles, and usually the number of styles is not more than 4. Additionally, Gridle does not allow exporting source code, reducing future scalability or migration options [20].

Node-RED

Node-RED is the low-code programming language that is utilized in this project. It is an open-source flow-based development tool [22]. It also provides drag-and-drop interface which simplifies development, even for user with minimal coding experience. Moreover, Node-red is highly versatile as it is compatible with on various platforms such as Windows, Linus, and so on. Additionally, it has strong community that shares flows, provide help, and contribute to ongoing improvements. Then, its strong integration capabilities make it easy to connect with various services and protocols, allowing users to build complex and interconnected systems with minimal effort. Furthermore, it allows developers to debug the nodes of specific flow, which provide accurate, clear, and efficient information during testing and debugging processes.

However, as the workflows grow in size and complexity, managing and maintaining them can be more difficult, raising scalability concerns. Furthermore, even though it minimizes the efforts for coding, some features and designs still rely on JavaScript and CSS styles. This may pose a challenge for developers without a technical background [23].

2.1.3 Encryption Algorithm

RSA

Two keys are used by the asymmetric encryption technique RSA to encode and decode data. Public key is used to encrypt the data while the secret key is used to decrypt the data [24]. Then, this provides a high security level for data communication as no need to exchange keys beforehand. The private key is kept confidential, but the public key is freely exchanged. Moreover, this algorithm ensures that the data will remain unchanged during the transition due to the missing private key that used for decryption [25]. In other words, it enhances data integrity in data communication. It is widely adopted on many protocols such as SSH, SSL/TLS, and so on [24].

However, this algorithm requires intensive mathematical operations to generate keys with large size for security purposes. This is making it much slower than the symmetric encryption algorithm. As a result, this algorithm is not suitable for processing the large data set and heavy resource consuming [25].

Blowfish

Blowfish is symmetric encryption algorithm which utilizes the same key for data encryption and decryption [26]. The key used by Blowfish can range in length from 32 to 448 bits. This algorithm involves subkey generation for speeding up the encryption process, 16 rounds of complex substitution and permutation that ensures the strong encryption, separation of data into two 32-bits halves for processing, and creation of 64-bit ciphertext blocks through the recombination of data. The decryption of data is a reverse process of encryption [27].

As this algorithm involves fewer operations, the generation of ciphertext block is fast, effective, and safe in general [27]. Then, it also supports wide range of programming languages such as C#, Python, PHP, Java, Kotlin, and others [28]. However, it can be cracked easily through Birthday Attacks due to its small block size, 64-bit. This causes this algorithm to have low security level in ensuring the data integrity [26].

AES-256-CBC

This algorithm is also a symmetric encryption algorithm, which applies to the same key with fixed key length, 256-bit for data encryption and decryption. AES is incorporated into both software and hardware to guard against eavesdropping and assaults on digital data, audio, video, and others [29]. For the encryption process, it also involves expanding secret key into multiple round keys and 14 rounds of transformation with the final round omitting mix columns. The complexity of each transformation round involves round key addition, row shifting, column mixture, and non-linear byte replacement. Then, the final output is the ciphertext blocks and a random generated initialization vector. On the other hand, it requires the same key and respective unique random initialization vector for the decryption process [30,31]. This makes the hacking process more insane. Moreover, the produced 128-bits encrypted blocks needs about 2^{128} attempts to be cracked [29]. This algorithm secures data integrity in a very save way. Additionally, it also supports a wide range of programming languages such as NodeJS, PHP, Python, Ruby, and so on [28].

Although it provides strong protection on data, it still has a weakness which is the consistent key for encryption and decryption. Since every data block is encrypted in the same way, the key must be stored in a proper way to prevent the broken encryption [29].

2.1.4 Hashing Algorithm

MD5

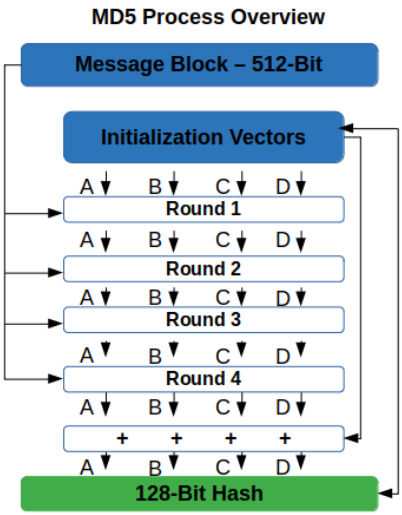


Figure 2.1.1: MD5 Algorithm

A hash algorithm which was published in 1992, one of the earliest and popular algorithms [32]. MD5 can create a fixed 32 hexadecimal digits length or 128 bits hash values from any input, such as file or string [32,33]. It simplifies the verification by comparing the hash values of input and the hash values of target file or other string, but this only suitable for small hash values comparison [33]. Therefore, it has wide adoption in various use cases such as the encryption of the passwords. Besides that, MD5 is lightweight and efficient, using minimal processing power and memory, which makes it ideal for resource-limited devices and quick integrity checks [33,34].

However, since MD5 has been available for about 3 decades, it can be broken through various attacks. The common way to crack this algorithm is by using the collision attack, which uses the different input that can generate the same hash values [35]. As a result, this hashing algorithm has a low security level in ensuring the data integrity.

SHA-3

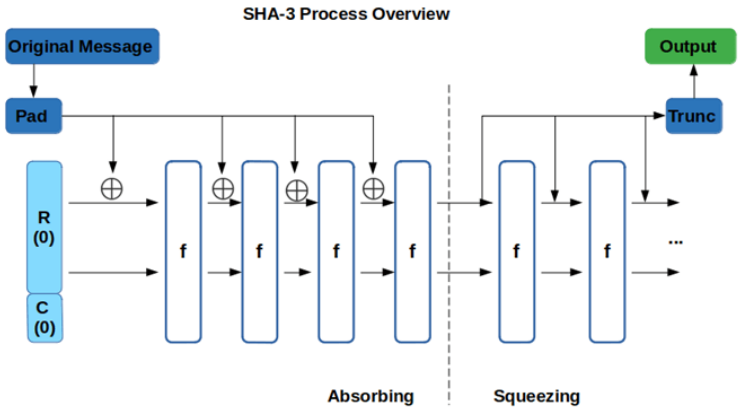


Figure 2.1.2: SHA-3 Algorithm

An algorithm that uses different approach for hashing data compared to SHA-2 algorithm family, which known as sponge construction. It's a means to increase the whole hash algorithm toolkit's resilience, and not the full replacement on SHA-2 [32]. It also has high security level in protecting the credentials data. Moreover, it is different with SHA-256, the developers can use it to generate different hash value length such as 224 bits, 384 bits, and others. In security aspect, it is more security than SHA-2 as it is the version that use different approach that also has fixed the weaknesses of SHA-2 [36].

However, it has limited adoption that may lead to problem in general computing environments. This is because it mainly delivers superior performance when implemented on hardware specifically optimized for its design, such as FPGA or ASIC. Then, this also means that it has lower performance in general environment compared to SHA-2 [37]. Additionally, it can support applications such as domain hashing, MAC address generation, stream-based encryption, and randomized hashing [32].

SHA-256

A member of the SHA-2 algorithm family which is available since 2002 [32]. It is used to convert a string or a set of numbers into a fixed 256-bit hexadecimal string regardless the length of original data. There are several intricate bitwise operations and transformations involved in this hashing process. As a result, this algorithm provides high security level and is also ensuring the data integrity of important stored data. It supports a wide range of use cases, which are including blockchain systems, password protection, secure communications, digital certificates, and so on [38]. Moreover, according to [37], SHA-256 has high speed performance on general

CPUs, the time taken for hashing the data is generally shorter than SHA-3. Therefore, this hash algorithm is applied in the purposed system. However, relative to more modern algorithms, SHA-256 may demand higher computational resources, which becomes more evident in large-scale deployments [36].

2.1.5 Summary of Technologies Review

Table 2.1.1: Summary of Technologies Review

Technologies	Characteristics	Advantage	Disadvantage
PostgreSQL	• Open-source • Object-relational database • Apply SQL standards	• Cross-platform & language support • Rich SQL & ACID features • Extensible (custom types, PL/pgSQL, etc.) • Efficient concurrent database access	• Rigid schema definition • No single commercial owner/support
MongoDB	• Open-source NoSQL	• Schema-less design • Document-oriented • Horizontal Scalability • Support third-party storage engines	• Inefficient failover process • Low writing performance • Lacks referential integrity
MySQL	• Open-source • Relational database	• Large & active community • Multi-threading • Platform independence	• Low performance under heavy workloads
Power Apps	• Low-code • Drag & drop method • Using Power Fx	• Rapid development environment • Ease-to-use • Supports wide range of data source • Ease to set dashboard	• Single-developer editing bottleneck • Inefficient debugging
Glide	• Low-code	• Wide data integration	• Limited Customization

	• Turns spreadsheets/SQL into web & mobile apps • Pre-built components	• Instant responsiveness • Ease-to-use	• Can't export source code
Node-Red	• Open-source flow-based • Drag-and-drop method	• High versatile • Strong community • Strong integration capabilities • Good node-level debugging	• Difficult to manage large and complex flows • Complex features still need JavaScript and CSS
RSA	• Asymmetric encryption approach, public key for encryption, private key for decryption • No need to exchange secret keys beforehand	• High security level • Ensures confidentiality and data integrity • Public key can be freely shared • Ensure data integrity • Widely used in protocols like SSH, SSL/TLS.	• Requires intensive mathematical operations • Slower than symmetric encryption • Not suitable for large datasets
Blowfish	• Symmetric Encryption algorithm • 32 to 448 bits key length • Generate 64-bit ciphertext block	• Fast and efficient • Wide programming language support	• Vulnerable to Birthday Attacks due to small block size (64-bit)
AES-256-CBC	• Symmetric Encryption algorithm • 256-bit key length • Complex operations • Generate and utilize initialization vector and ciphertext block	• Strong encryption • Compatible with wide programming language • Ensure strong data integrity	• Consistent key for encryption/decryption
MD5	• One of the earliest hash algorithms.	• Lightweight and efficient (low resource usage)	• Low security level • Vulnerable to collision attacks

	• Produces 128-bit hash values (32 hexadecimal digits). • Commonly used for verifying file/string integrity and password encryption.	• Ideal for resource-limited devices • Support quick integrity checks	
SHA-3	• Sponge construction • Supports multiple output lengths (224, 256, 384, 512 bits) • Designed to fix weaknesses of SHA-2 and provide extra resilience	• Flexible output lengths. • Higher security compared to SHA-2 • Optimized mainly for specialized hardware	• Limited adoption • Lower performance than SHA-2 on general-purpose CPUs
SHA-256	• Member of SHA-2 family • Produces a fixed 256-bit hash value. • Involves complex bitwise operations and transformations • Widely used in blockchain, password hashing, secure communication, and digital certificates	• High security level • Broad adoption across industries and systems • High speed performance on general CPUs	• Requires relatively high computational resources

2.2 Review of the Existing Systems

2.2.1 Web-Based Inventory Monitoring System (YII Framework)

The previous study [39] proposed a web-based inventory monitoring system for an Indonesia manufacturing company that developed utilizing Model View Controller technique with the YII Framework. This system included 3 key modules which were management of inventory items, calculation of net stock, and inventory stock reporting.

The management of inventory items module provides additional feature which is not only recording the existing inventory items but also including the ingoing and outgoing inventory items. This could enhance the accuracy of management of inventory item by eliminating the confusion in recording stock items. For instance, some user will register the incoming items as existing items, which causing confusion in generating reports.

Furthermore, this study utilized an algorithm in calculation of net stock module which support the decision making. This algorithm is used for determining the quantity of net stock following the warehouse demand and supply. It used the quantity of initial stock, incoming stock, and outgoing stock to get the net stock in the warehouse. For instance, the formular is like below:

Net stock = Initial stock – Incoming stock + outgoing stock

Lastly, this system provided inventory stock reporting which the reporting date can be chosen according to the user's desire. This module can provide user or owner a quick review on the existing inventory stock and make the decision effectively such as whether need to demand or supply more stock items.

2.2.2 Web-based Inventory Management System

There's another study [40] proposed a web-based inventory management system that includes 3 key features, which are management of inventory, management of purchase requests, and generation of reports.

However, this study mainly focused on management of purchase requests and production of reports. Firstly, for management of purchase requests, it included view, add, edit, delete, approve, and reject purchase requests. It provided a complete feature in managing the purchase requests, which is important for controlling the amount of stock items.

Moreover, this proposed system also provides reports based on the chosen date. The system can produce reports for different categories regarding inventory such as orders, sales, and products. This allows the users or owners to perform decision making through analysis different categories of the business.

2.2.3 Review on AssetTiger [41]

AssetTiger is an online inventory management system which is opened to various kinds of business enterprises for managing their assets or stock item. Then, AssetTiger provides one-stop service in setup the inventory management system, which is convenient for business owners.

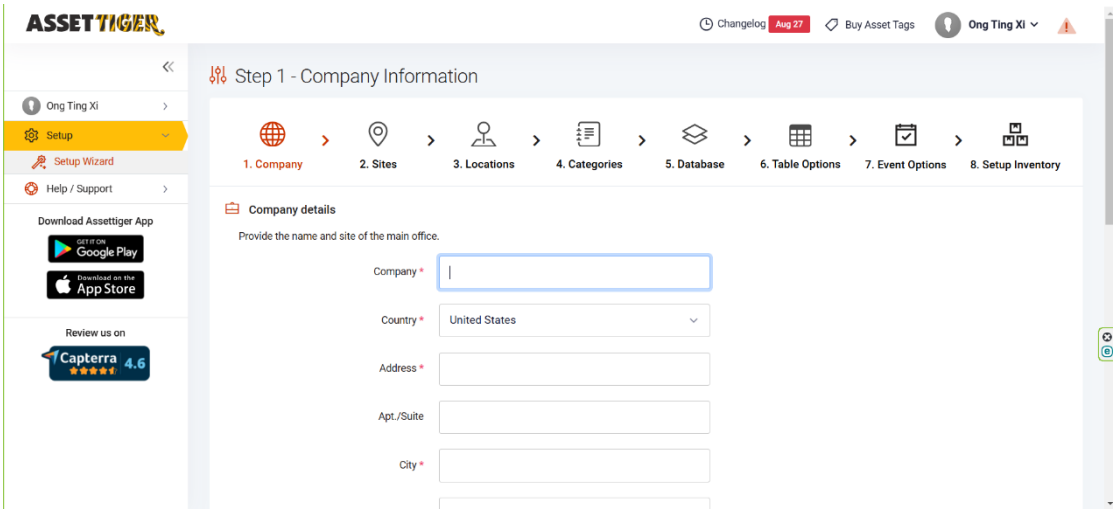


Figure 2.2.1: Interface for setup inventory management site

The one-stop service includes:

1. Register Company details, which including the company name, address, time zone, and so on.
2. Registers sites of company, this is useful for the company with multiple branches in managing the stock items.
3. Setup categories of assets, which can simplify the management of inventory.

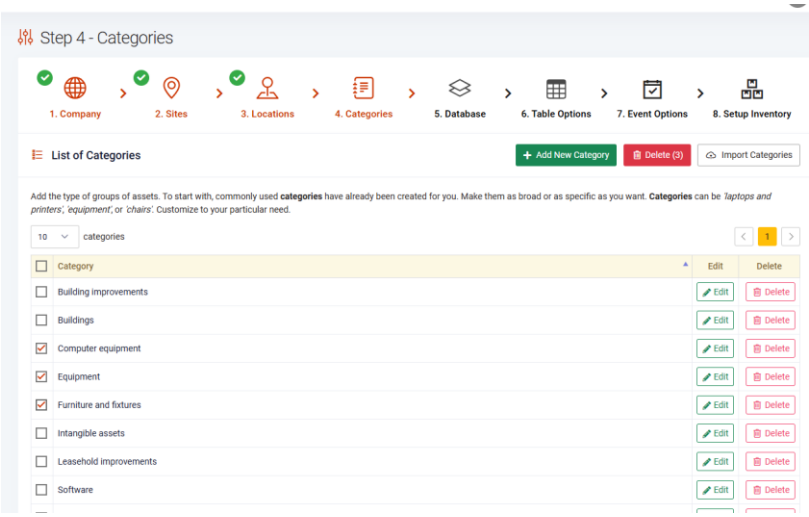


Figure 2.2.2: Setup categories of assets

4. Setup database, it provides simple setup of database which just needs to register the fields that contain necessary information of that asset.

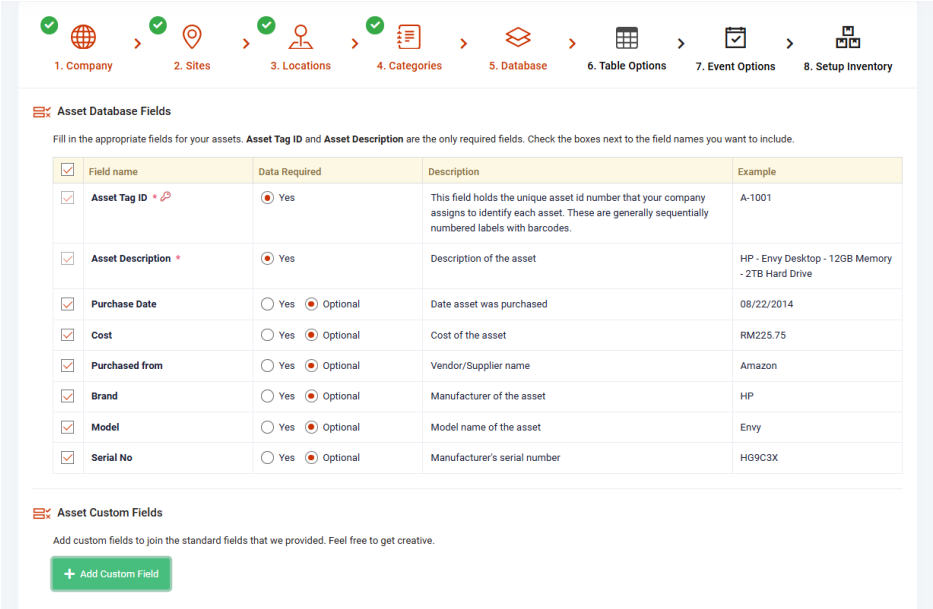


Figure 2.2.3: Setup database

5. Setup properties of inventories. User can label assets which are having warranties, need to be maintenance in specified period, need to perform audit, and so on.
6. Register event for inventory. User can setup what information needs to be entered in the form when take part in specify events. For instance, when check-out the asset, the user needs to fill the details for that asset which includes the owner of the asset, email address, department, and so on.

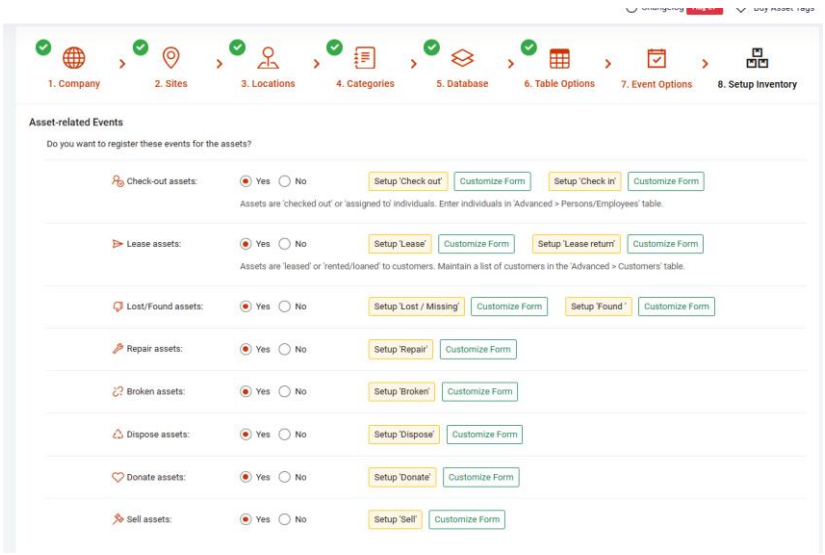


Figure 2.2.4: Register event for inventory

7. Setup inventory fields, which include the format of asset ID, unit, and so on.

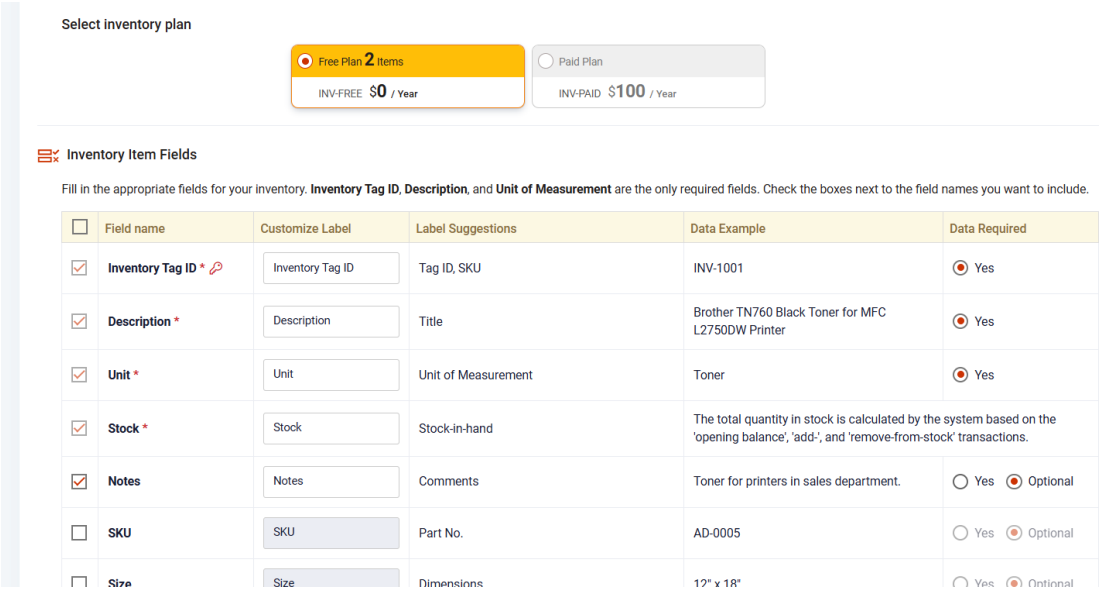


Figure 2.2.5: Setup inventory fields

Then, the interface for entering the inventory information according to the previous setup is shown as below:

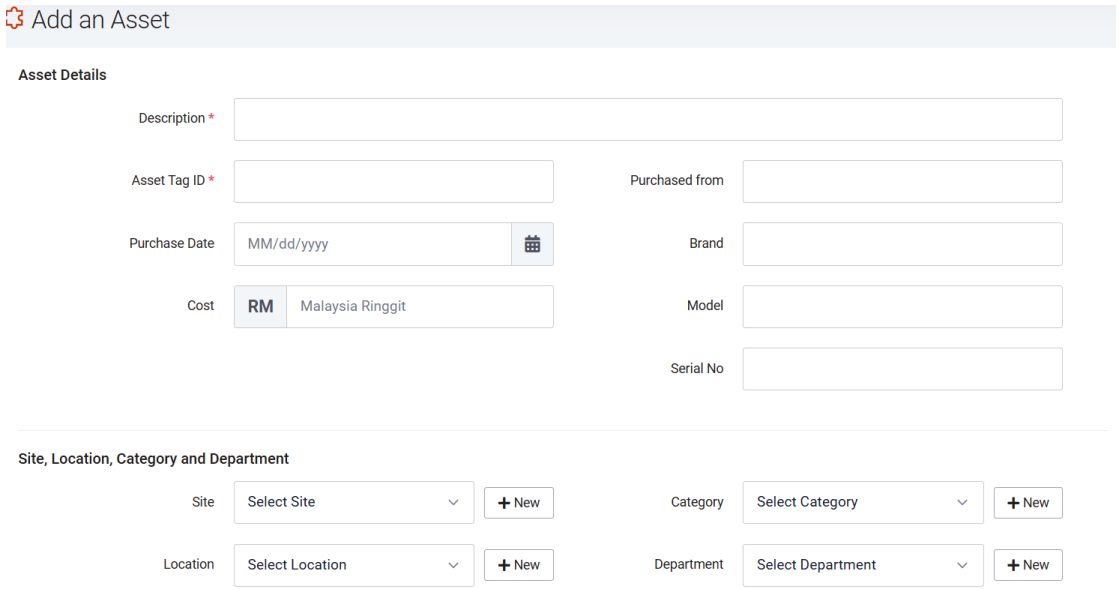


Figure 2.2.6: Record inventory

After entering and saving the inventory information, we can do more action for this data like editing, delete, and so on. It provides rich features in managing inventory details.

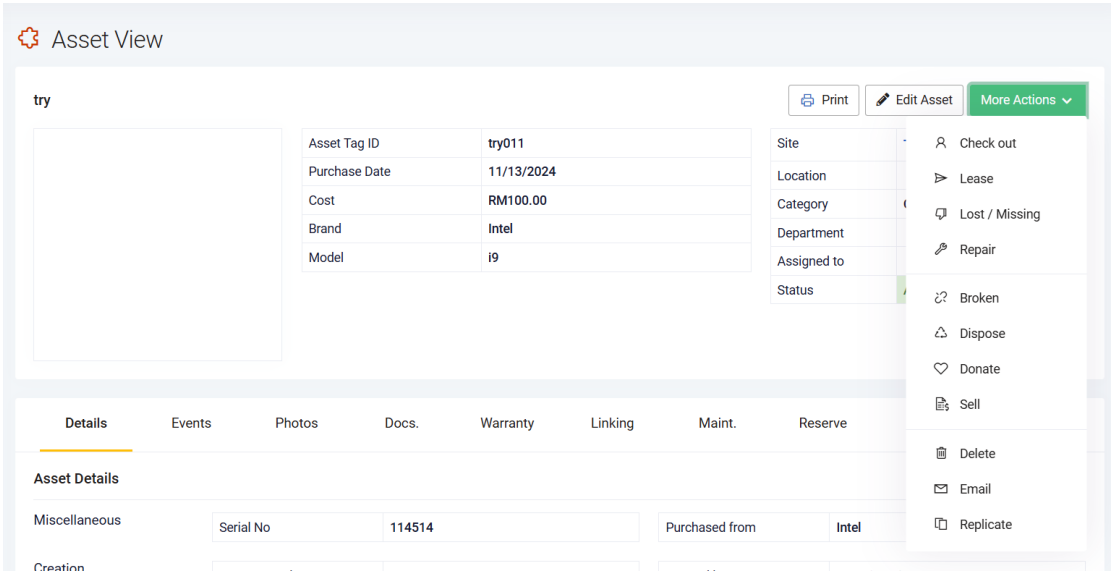


Figure 2.2.7: Management of inventory details

Moreover, AssetTiger also provides editable dashboard for the users. User can adjust the position of the elements of dashboard. Then, it allows users to have the dashboard that they think is effective.

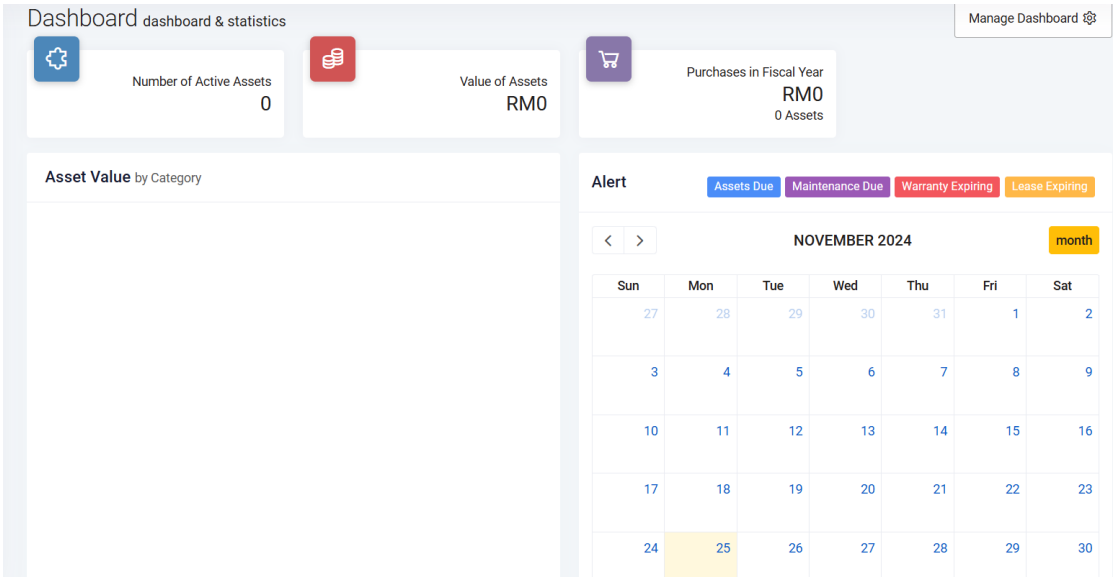


Figure 2.2.8: Dashboard in AssetTiger

2.2.4 Review on Odoo [42]

An open-source inventory management system called Odoo is also making appearance on websites. It is offering functionality for managing the stocks such as register stocks' information, stock tracking, setup reordering rules, barcode scanning, and so on.

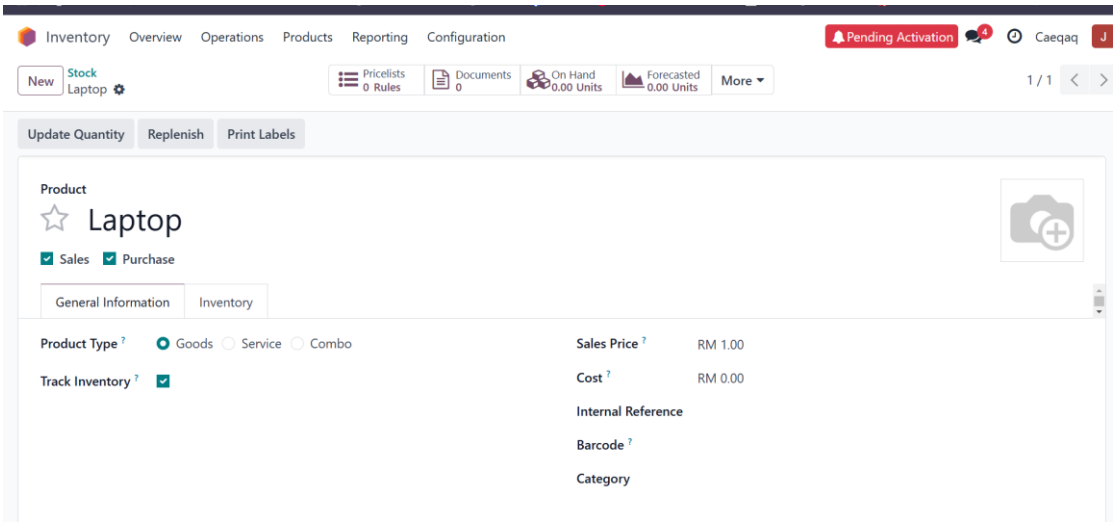


Figure 2.2.9: Registration of stocks

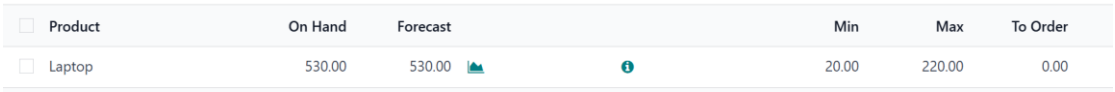


Figure 2.2.10: Setup reordering rules when reaching specific minimum quantities

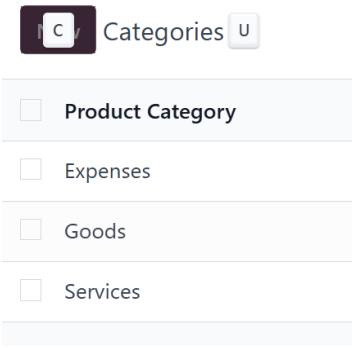


Figure 2.2.11: Configuration of the category of products

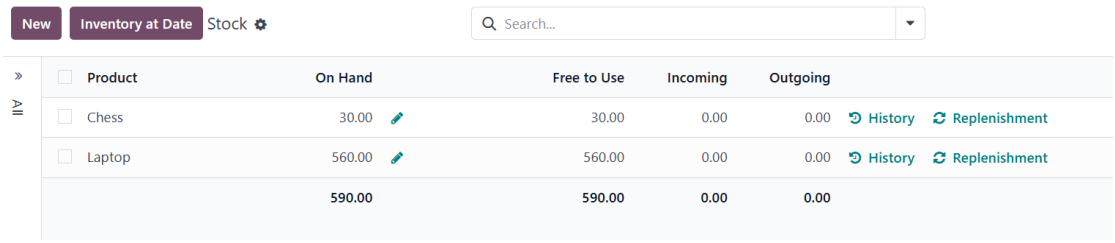


Figure 2.2.12: List of products for tracking

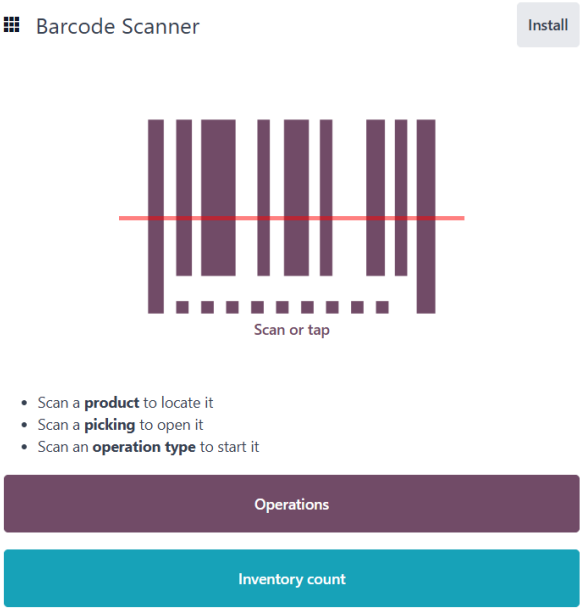


Figure 2.2.13: Barcode scanning

Moreover, it is also offering rich integrations with various core modules such as sales, services, accounting, and so on. These modules allow users to have the desired additional functionalities on the system.

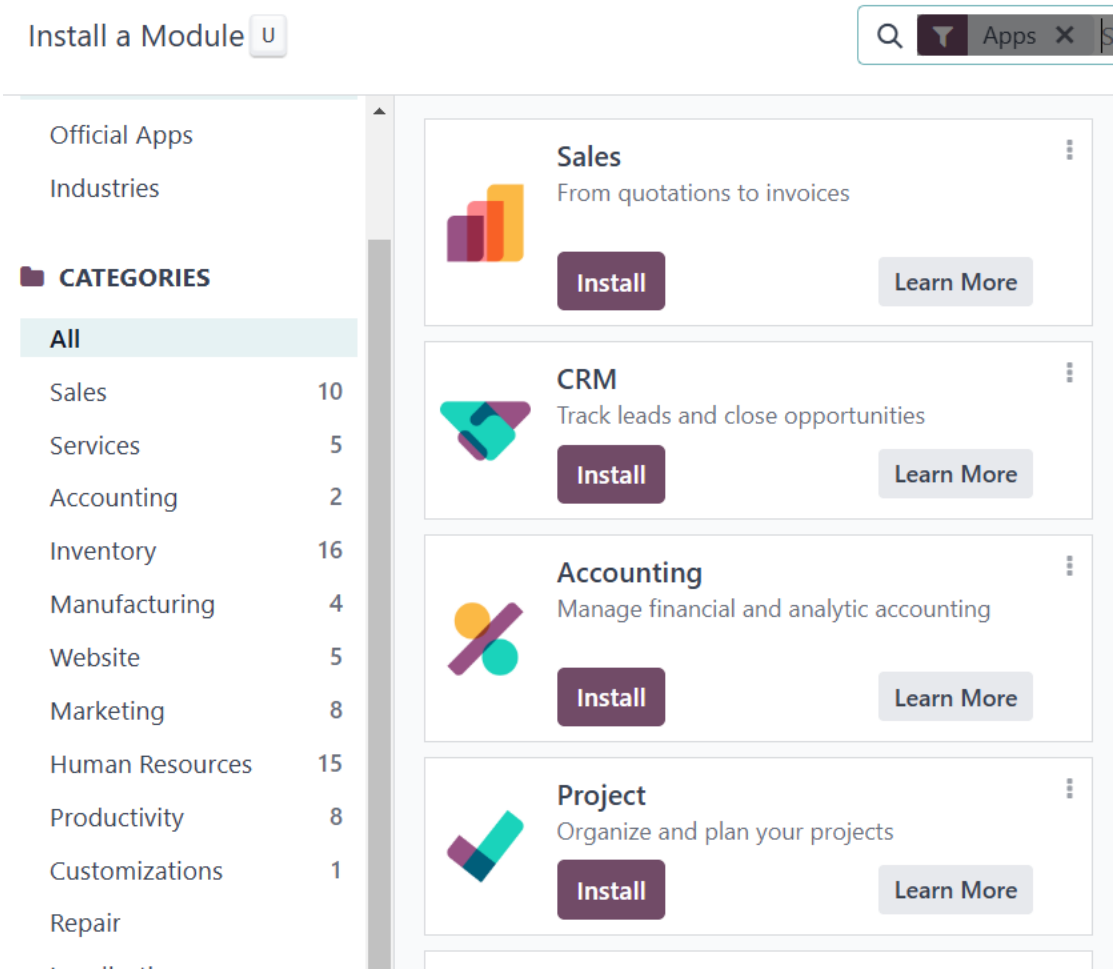


Figure 2.2.14: List of modules

2.3 Weaknesses of Previous Studies

2.3.1 Web-based inventory monitoring and management systems

The weaknesses of previous studies' systems were unable to simplify the progresses of development and implementation of inventory monitoring systems due to the utilizing of traditional development tools such as C++, JavaScript, PHP, HTML, MySQL, and so on. These development methods are required professional knowledges and complex development life cycles. For instance, the integration of these platforms is insane for the complex system requirements and structure. Moreover, the maintenance of the systems would be difficult due to the same reason. Then, the proposed system might not be extendable anymore, and the IT professionals also hard to understand the coding created by previous developer. As a result, this causes the high cost and time consuming in developing, implementing and maintaining the systems.

Moreover, for the system developed by previous study [39], it didn't provide automated alert, and the user roles management. This is causing this system has low efficiency in handling key issues because the users may fail to perform action immediately, and the user is unable to check the history of the modification or registration for identifying the causes of issues.

Lastly, the previous study [40] developed the similar system that is weak in basic inventory management such as add, edit, and remove the inventory data. This is because it too focused on the management of purchase requests which caused other system scopes to be reduced.

2.3.2 AssetTiger

This inventory management system web portal is unable to simplify the navigation design due to its rich features. Sometimes, too many detailed features are not a good practice. This may cause users have a deep learning curve in using those systems. As a result, this causes users have low effectiveness in adapting this inventory management system.

Additionally, this inventory management system can't display interface with completely responsive on mobile devices as it is restricted by its features. Most of the features are only suitable for desktop resolution.

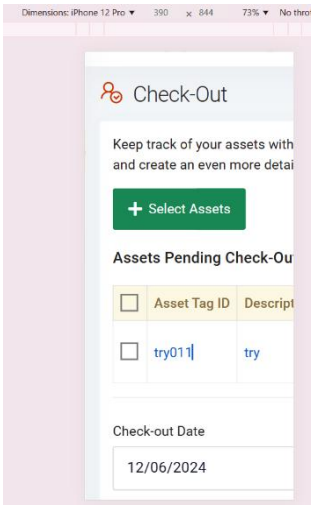


Figure 2.3.1: Interface with not completely responsive

Lastly, this system may complicate the simple managing and monitoring processes. This is because it provides rare and detailed features which is only suitable for big business enterprises such as the recording of depreciation on assets. Then, these kinds of features are unnecessary for small and medium businesses.

2.3.3 Odoo

This inventory management system also has weak navigation for user to setup the automated messaging for specific events such as the reminder of low level of stock. For example, the user needs to create the inventory rules for specific product to get the automated messages about the reordering products.

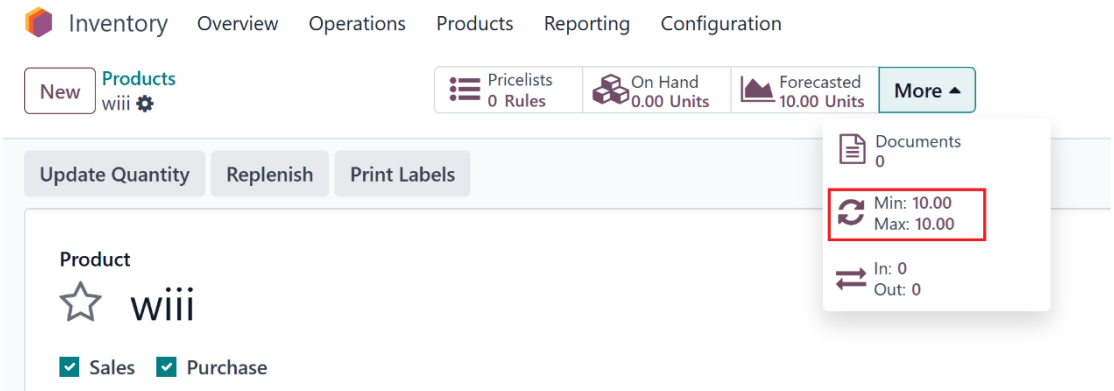


Figure 2.3.2: Way to set the automated messages for low-level stocks

Moreover, there's not a single page to manage all the automatic messages. The user needs to go to specific category for setup the automatic messages. This causes users to pay a lot of effort in setup the automatic messages for necessary events.

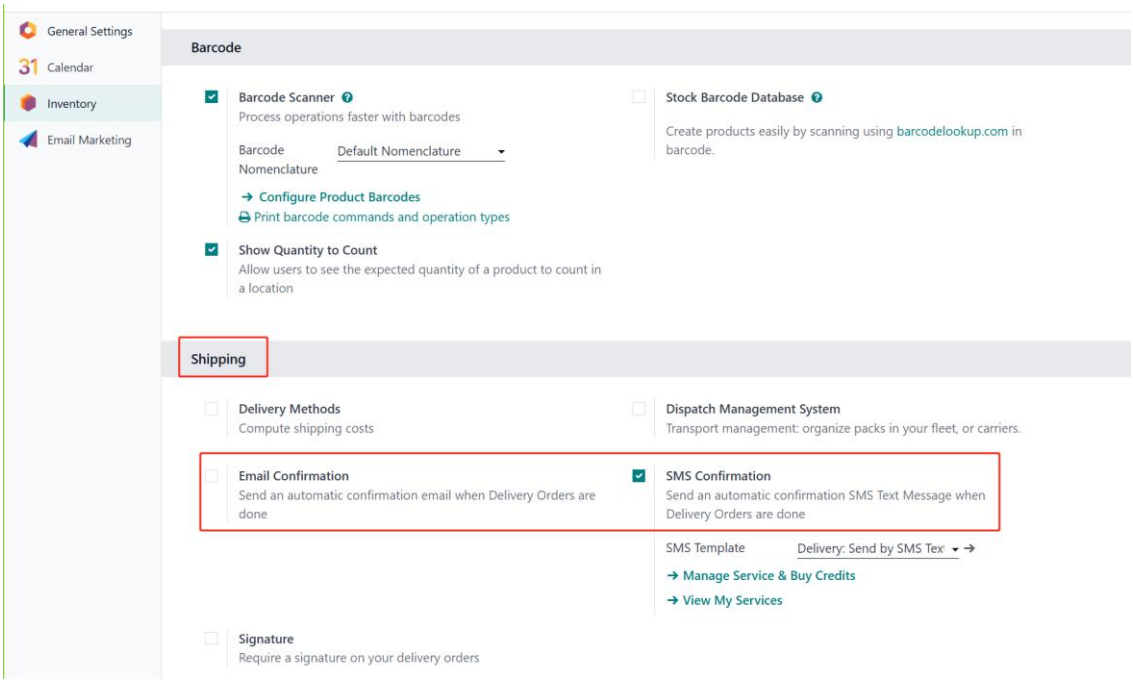


Figure 2.3.3: Setting of automatic messages regarding shipping

2.3.4 Summary of the existing systems

Table 2.3.1: Summary of the existing systems

Existing systems	Advantages	Disadvantages	Critical Comments
Web-Based Inventory Monitoring System (YII Framework)	• Net stock calculations • Track incoming and outgoing stock • Dashboard for review purpose	• Lack of automated alert feature • Lack of user roles control	• Low efficiency in handling key issues
Web-based Inventory Management System	• Detailed report generation • Flexible in managing purchase requests	• Weak in basic inventory management	• Too focus on managing purchase requests
Review on AssetTiger	• Convenient for setup inventory monitoring system • Customizable positions of dashboards • Rich features	• Complex user interface and navigation • Provide rare features which may unnecessary • Not responsive	• Not suitable for small and medium business enterprises • May complicating simple managing and monitoring process
Review on Odoo	• Rich features on inventory management • Rich integrations with other applications	• Weak navigation • Inefficient automated messages management	• Difficult to setup automated messages

2.4 Concluding Remark

In this chapter, there are 3 database platforms, 3 low-code programming language, 3 encryption algorithm, 3 hashing algorithm, and 4 previous works on inventory monitoring system be reviewed. Each of these reviews are mainly focused on characteristics, advantages, and disadvantages to find out the critical aspects that would be focused within this project.

Chapter 3 - System Methodology

3.1 System Development Models

3.1.1 Waterfall model

According to [43], one frequent name for the Waterfall Model is a linear-sequential life-cycle model. It is straightforward and easy to grasp. Within this model, each phase must be fully finished before the subsequent one starts, and the phases never overlap.

There are 6 phases within this model:

Phases 1, Requirement Gathering and Analysis

During this opening stage, every potential need for the proposed system is identified and recorded in a requirements-specification document.

Phase 2, System Design

In this phase, the requirement specifications from the previous phase are examined and the system design are used to identify the overall system architecture.

Phase 3, Implementation

Build system first in units, which are tiny programs that are integrated into the subsequent phase, using inputs from the system design. Each unit is coded and verified for correct operation through Unit Testing.

Phase 4, Integration and Testing

Integrate all the validated units into a system after each unit is tested. After integration, the whole system undergoes comprehensive testing to uncover defects or failures.

Phase 5, Deployment of System

Once all functional and non-functional tests pass, the finished product is installed in the client's environment or released to the market.

Phase 6, Maintenance

In the client environment, some problems might arise. Patches are published to address these problems. Additionally, improved versions are offered to enhance the product. Ongoing maintenance ensures these updates reach the customer smoothly.

3.1.2 Agile Model

Agile model is a type of incremental model. The product is created in quick, successive iterations. Each short cycle produces a small release that extends the features delivered in earlier ones, and every release undergoes rigorous testing to keep quality high [44].

The phases of this model also has 6 similar phases. However, Agile Model does not need the comprehensive planning to start a project, unlike Waterfall Model. Moreover, due to the frequency of new increment, the implementation of new changes can be low cost. The developers are revisiting only a few hours or, at most, a couple of days of development work for introducing a new feature [45].

3.1.3 Rapid Application Development Model (RAD Model) [46]

Rapid Application Development Model is also a type of incremental model with concise development cycle. It suits project whose requirements are already clear and that can be built with a component-based strategy. The main feature of this model is it divides a project into small modules which each module can be distributed independently to separate teams. Once completed, the modules are integrated into the overall system. Within every module, work still follows the familiar Waterfall steps such as analysis, design, coding, and testing. Then, RAD includes 4 basic stages: Requirements Gathering, User Description, Construction, and Cutover.

Stage 1, Requirements Gathering

Begin by eliciting needs through methods such as brainstorming, task and form analysis, user scenarios, or FAST workshops. Develop a structured outline that identifies the key data to capture, explains how it will be gathered, and maps out how it will be processed into an initial model.

Stage 2, User Description

This phase involves collecting feedback from the end-users, re-validates and fine-tunes the information from previous phase, and establish the prototype. In this phase, the data attributes are also clarified and documented.

Stage 3, Construction

Refine the prototype into a functioning product. Using the high-productivity automated tools, convert the processes and data models into final product and applying any required changes and enhancements along the way.

Stage 4, Cutover

Conduct thoroughly test the interaction among the separate modules produced by different teams. Once integration tests pass, conduct user acceptance testing and prepare the system for release.

3.1.4 Spiral Model [47]

Spiral Model is an SDLC framework that guides development through a series of repetitive cycles, visualized as the turns of a spiral. Each loop of the spiral represents one phase of the project and there is not a fixed total number of loops. Uniquely, this model is a risk-driven model which emphasizes controlling and mitigating risks through successive iterations of the development cycle. Then, each phase in the Spiral Model is broken down into separate quadrants include defining the aim, analysing and addressing risks, developing the next product version, and reviewing and planning for the next phase.

Quadrant 1, Object defined

Gather user requirements, set the target of project which includes functional and non-functional requirements.

Quadrant 2, Risk analysis and resolving

Identify and evaluate the risks regarding the project. Then, compare the alternatives, choose the best option, and built the prototype for solution that is possible the best.

Quadrant 3, Develop next version of product

Develop, test, and verify the chosen features for delivering the next version of the software.

Quadrant 4, Review and plan for next phase

The developed version of the software is evaluated by the customer. Then, the team capture the feedback and start planning for the next phase at the end of this phase.

3.1.5 Selected Model

Consequently, the final decision on the model of this project is Agile Model. One of the reasons is the planning to start this project is very limited. The developed inventory monitoring web portal requires frequent of user requirements and feedback to provide the updates frequently. Moreover, the quality of each updated or added feature is required to be ensured in high quality. Furthermore, this model also allows this project to redesign or remove the features based on the discussions with the users in each development cycle.

Then, for following this model, this project consists of six phases. The first phase is collecting the requirements, which understand the features and workflows that are needed to be included in the system through discussion with target users. Secondly, the next phase is designing the system. The overall architecture of the system, wireframes of user interfaces, and Unified Modeling Language diagrams are designed within this phase to describe how those requirements will be applied in the system.

Moreover, the construction or iteration of system is in the third phase. The features of the systems will start to be developed according to the identified system designs. However, not all the features will be developed at once before entering the next phase. Only a few features will be developed for each development life cycle.

Furthermore, the next phase is testing the developed features from previous phase. This testing is conducted internally and not externally. The end users of this system will not take part in this phase. Each developed feature is tested to ensure that it can work well in the system before deploying to the end users' environment.

Then, the next phase is deploying, which includes integrating the features as the latest version of the system and deploying it into the end users' environment. Then, the end users can start to use the latest version of the system and verify that this system archives their expected performance levels.

Lastly, the final phase is feedback collection. End users would provide feedbacks regarding the problems or suggestions of the newest system version. Then, these feedbacks will be collected, analyzed, and aids in conducting the next development life cycle for improving or upgrading the system.

3.2 System Requirement

3.2.1 Hardware

The hardware involved in this project are laptop, lab computer, and mobile phone. Laptop provides the environment for the development of inventory monitoring web portal. It is issued for running low-code platform and MySQL Workbench, implementing web portal, and performing self-testing for web portal. Additionally, it is also used for recording the necessary notes, remarks during the development process.

Besides that, lab computer is utilized for user testing purposes and deployment of the lab inventory monitoring system's final version. Lastly, the mobile phone is used for testing and deploying the automated notification and Telegram chat ID generation features.

Table 3.2.1: Specifications of laptop

Description	**Specifications**
Model	Gigabyte G5 AX200NGW
Processor	Intel Core i5-10500H
Operating System	Windows 11
Graphic	NVIDIA GeForce RTX 3060 6GB
Memory	32GB DDR4 RAM
Storage	4TB SSD

Table 3.2.2: Specifications of lab computer

Description	**Specifications**
Model	Acer Veriton M2640G
Processor	Intel Core i5-6400 CPU @ 2.70GHz
Operating System	Windows 11 Pro
Graphic	AMD radeon R5-310 2GB DDR3
Memory	8 GB RAM
Storage	480GB SSD

Table 3.2.3: Specifications of mobile phone

Description	**Specifications**
Model	POCO F6 Pro
Processor	Qualcomm Snapdragon 8 Gen 2
Operating System	Android 14
RAM	12 GB
Battery Capacity	5000 mAh
Storage	256 GB

3.2.2 Software

This project includes Node-RED, Telegram, and MySQL for developing the front-end elements and backend logic of inventory monitoring web portal.

Firstly, Node-RED is a low-code programming development tool which is capable for design backend logic with different workflows to handle inventory data and integration with various service platforms, APIs, and database. Moreover, it is also capable of designing and implementing responsive web pages through utilizing the built-in components such as button, form, table, template, and so on. The template allows user to put HTML and JavaScript coding into webpage. Then, the whole development process can be completed through utilizing the palettes provided from various sources. For example, this tool can download various palettes easily, such as node-red-node-mysql which provide a simple way to operate with mySQL database.

Figure 3.2.1: node-red-node-mysql palette

Secondly, Telegram provides chatbot which is integrated with inventory monitoring web portal for notification and alert purpose. Lastly, MySQL is a database that is integrated with web portal from node-red, and it is capable of creating necessary tables for storing inventory data and managing the accesses of inventory data.

3.3 Functional Requirements

This section outlines the functional requirements of the lab inventory monitoring system. These requirements define the core features and operations that the system must support for ensuring

the efficient inventory management, tracking, and loan application management. Each requirement is designed to meet the users' needs such as students, supervisors, lab staff, and administrators.

Table 3.3.1: Functional requirements of Lab Inventory Monitoring System

No	Functional Requirements
1	Student able to register a new account
2	Student able to view the list of available items
3	Student able to apply for loaning items through this system
4	Student able to apply for renew the period of loan item
5	Student able to check the status of applications
6	Supervisor can decide whether to approve student's loan application
7	Supervisor can decide whether to approve student's renew loan application
8	Supervisor can view the list of available items
9	Supervisor can apply for loaning items
10	Supervisor can apply for renewing period of loan item
11	Supervisor can check the status of applications
12	Lab staff can decide whether approve or reject the request of student account's registration
13	Lab staff can register new item on the system
14	Lab staff can modify the existing items in the system
15	Lab staff can report/remove the item from the system
16	Lab staff can view the item records on the system
17	Lab staff can leave comment regarding item's status
18	Lab staff can track the status of the item
19	Lab staff can manage the list of item categories
20	Lab staff can manage the list of inventory types
21	Lab staff can perform final confirmation on applications of loan from students and supervisors
22	Lab staff can update status of applications as returned
23	Lab staff can view the history of all applications
24	Lab staff can remove records from applications history list

25	Lab staff can register account with any role except administrators and students.
26	Lab staff can deactivate account with student role only
27	Lab staff and Administrator can view users list.
28	Administrator can decide whether to approve supervisor's loan application
29	Administrator can view the history of all applications
30	Administrator can manage items
31	Administrator can register account with any role except students
32	Administrator can deactivate account with any role except students
33	Administrator can manage item venues information
34	All users able to log into the system
35	All users able to log out the system
36	System able to encrypt and store password and Telegram chat ID
37	System able to retrieve and decrypt password and Telegram chat ID
38	System able to perform Telegram chat ID distribution and verification.
39	System able to send automated notification to users based on specify activities. (New application of loan, loan period is about to end, result of application, reported item, and so on)
40	System able to automatically cancel any pending applications when the corresponding item has been reported
41	System able to validate the requested loan period
42	System able to validate whether a specify application can be renewed according to the new loan period and chances remaining for renewing.
43	System able to validate the deactivation of account whether the user still have any processing case
44	System able to detect and avoid any invalid data input
45	System able to clear specific user session after 2 minutes when webpage is closed

3.4 Project Milestone

Figure 3.6.1: Gantt Chart for FYP I

The first two weeks were mainly focused on refining the report, which included chapter 1 (project objectives and scope) and chapter 2 (literature review). Then for the following two weeks, the development of registration items feature and modification item feature were conducted. Furthermore, the improvement of both these features was carried out in the next 3 weeks. Then, the following task was simple, which was development of location management module, this process just took around 5 days. Then the self-testing on the system was carried out about 1 week. After done testing, the report writing was conducted around 4 weeks. Lastly, the system demonstration was conducted on the last two week, which within week 13 and 14.

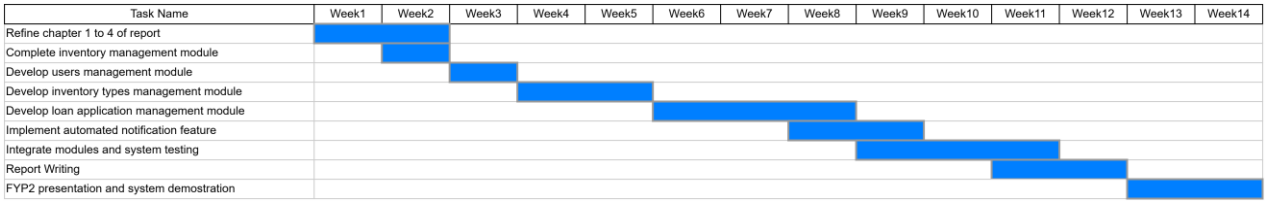


Figure 3.6.2: Gantt Chart for FYP II

According to the Gantt chart above, this project is required to be 100% completed within FYP2. In the new first two weeks, the refinement of report's chapter 1 to 4 and the development of inventory management module should be completed. Then, the following six weeks should complete the development of remaining modules. After developing all modules at week 8, the implement of automated notification feature is also started within the same week, and it takes around two weeks to complete. Then, the integration of modules and system testing is conducted by around week 9 to week 11. Then the report writing is conducted from week 11 to week 12. Lastly, the presentation and system demonstration will be carried out within week 13 and week 14.

3.5 Concluding Remark

In this chapter, the selected system development model, Agile Model has been stated with reasons. Moreover, this chapter has also brief about the hardware and software that are utilized in this project. Then, the expected system testing and performance for each existing feature have been explained in the tables. The expected challenges are also stated in this chapter. Lastly, the project milestone has been shown in a Gantt charts within this chapter, which includes refine report, develop modules, improve modules, testing, report writing, and system demonstration.

Chapter 4 – System Design

4.1 System Architecture

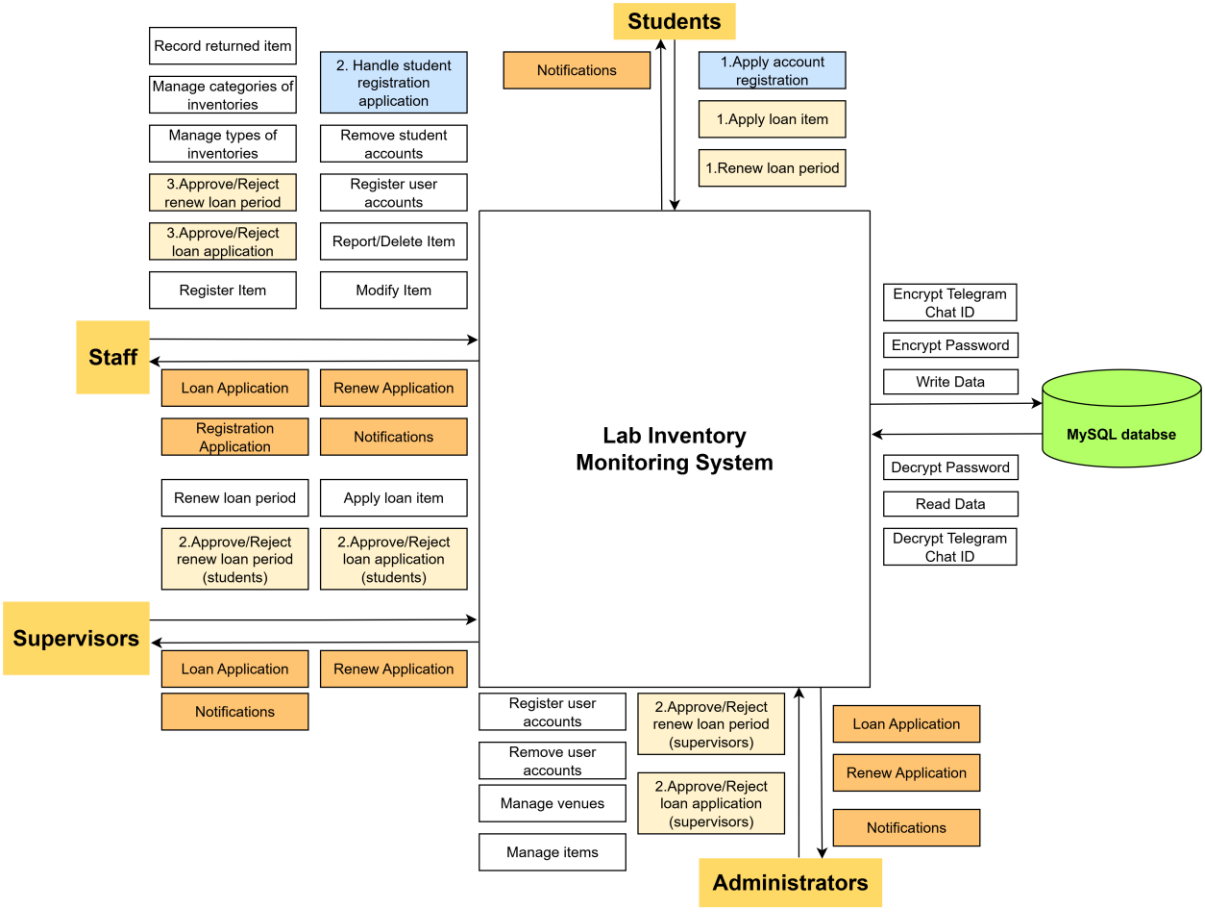


Figure 4.1.1: Block diagram of Lab Inventory Monitoring System

The block diagram shown above has display the actions that can be taken by each user role. It shows the flow of how the applications of new student account, loan item, and renew loan period are generated and handled. Then, each user role will receive notification from the system according to the specific event occurs. Moreover, it states the flow about how the data is input and write into the database. It also shows that the flow about how the data is retrieved from database and output to users.

4.2 Functional Modules in the System

The modules that are included in system:

- User management module
- Inventory management module
- Location management module
- Item types management module
- Item categories management module
- Applications management module

Then, the use case diagram for these modules is shown as below:

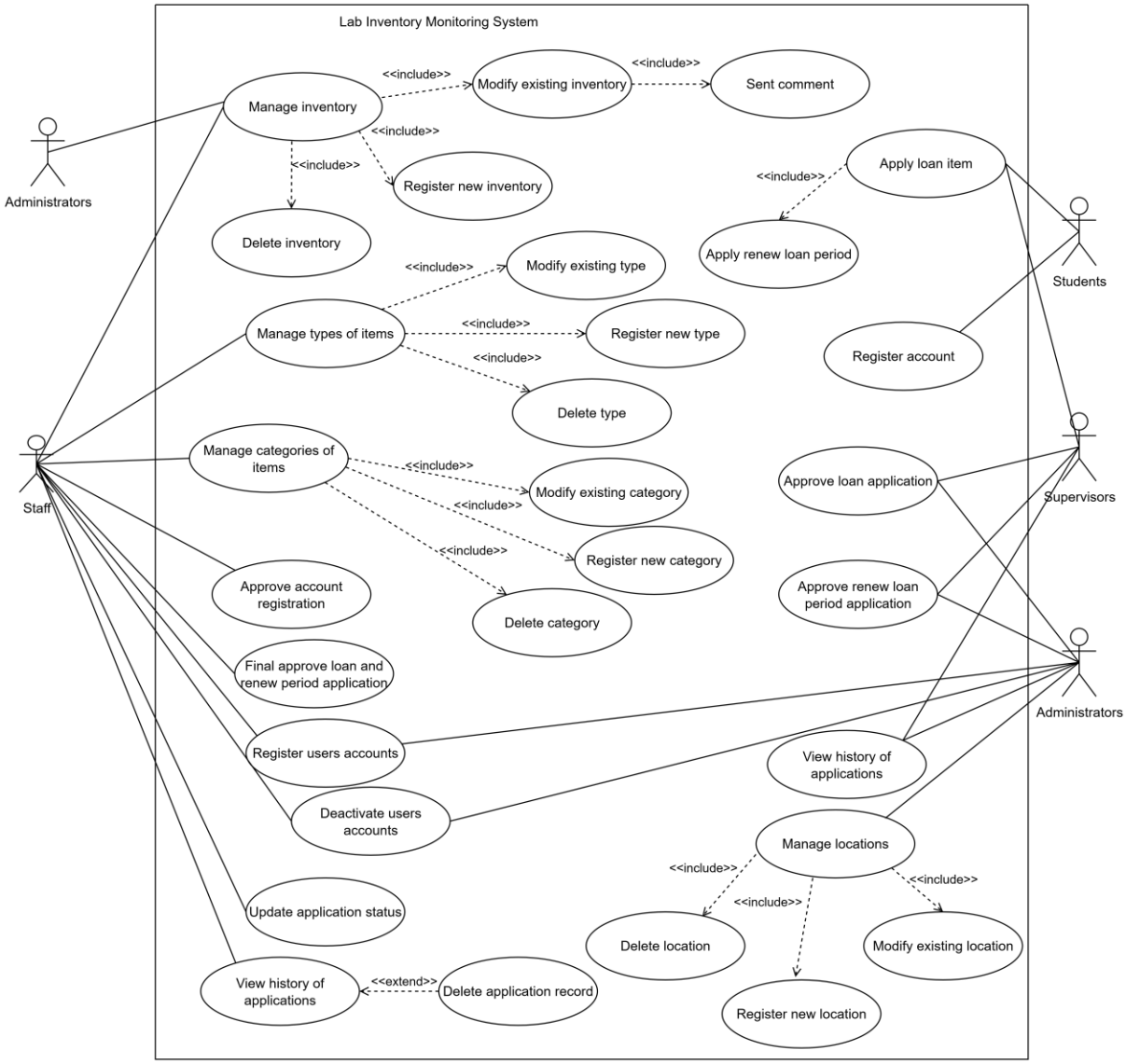


Figure 4.2.1: Use Case Diagram of Lab Inventory Monitoring System

According to the use case diagram above, staff utilizes all modules within the system except locations management module. In user management module, staff can approve registration of student's account and register account for other users except for administrators and students. Furthermore, they can deactivate user accounts with student role only. Then, they can handle the applications of loan item or renew loan period by performing the final confirmation. Furthermore, they are able to update application status when the loaned item is returned and check the history of applications. The old history records can be removed by staff. Additionally, they also can manage inventories, item categories, and item types.

Furthermore, for students, they can register account for themselves but needs the approvement from lab staff. They also can request to loan item and renew the respective loan period.

Besides that, supervisors are responsible for handling the applications from students. Then, they are also allowed to loan item and renew the respective loan period. Moreover, the administrators are responsible for handling the applications from supervisors. They are also allowed to register and deactivate accounts for other users with any role except students. Moreover, supervisors and administrators are also allowed to view the history of applications. However, supervisors only able to view the history of applications of their students while administrators can view all applications records. Lastly, administrators can manage the locations of items through locations management modules.

4.3 System Flow

4.3.1 Flow Chart

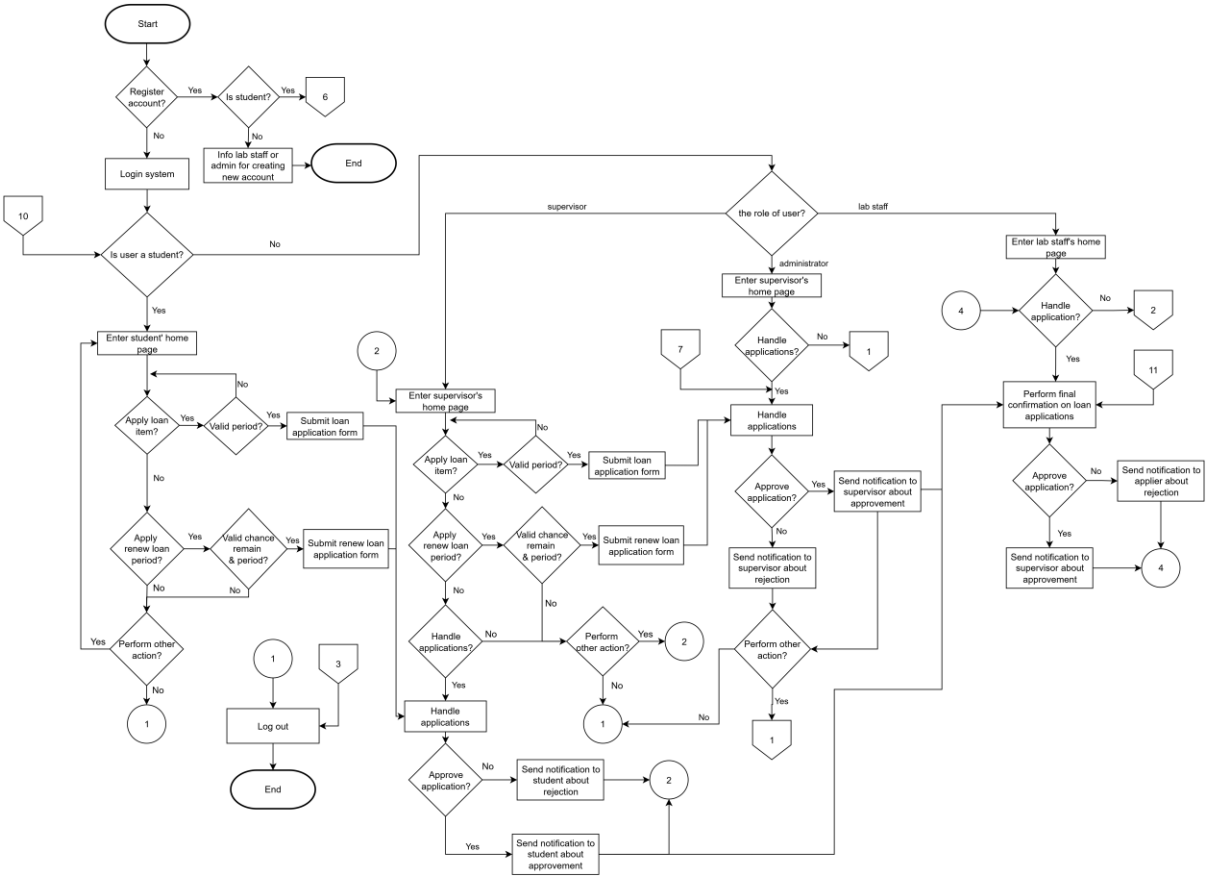


Figure 4.3.1: Flow Chart of Lab Inventory System (Part 1)

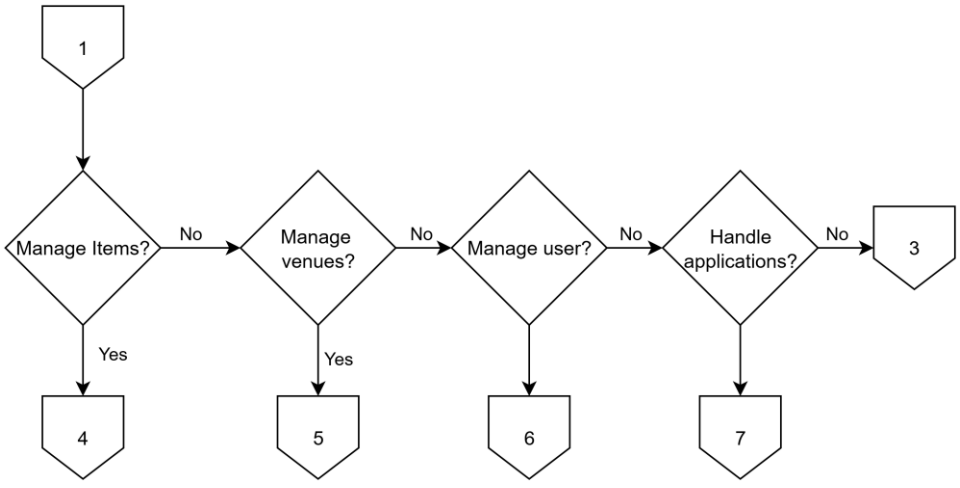


Figure 4.3.2: Flow Chart of Lab Inventory System (Part 2)

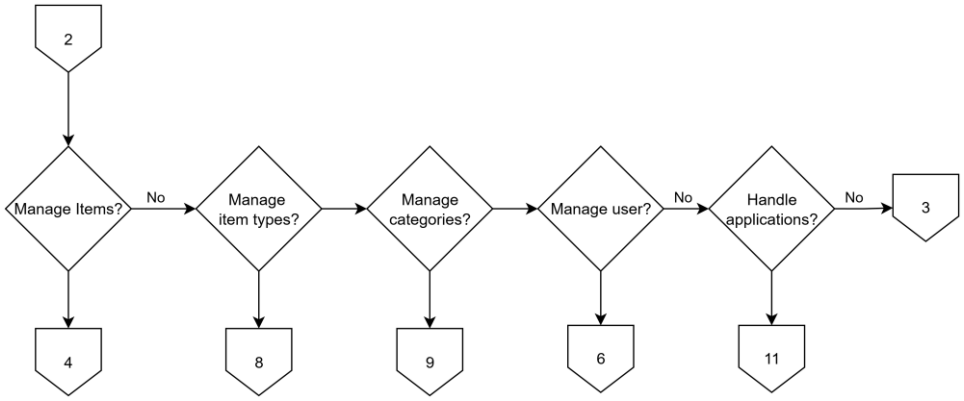


Figure 4.3.3: Flow Chart of Lab Inventory System (Part 3)

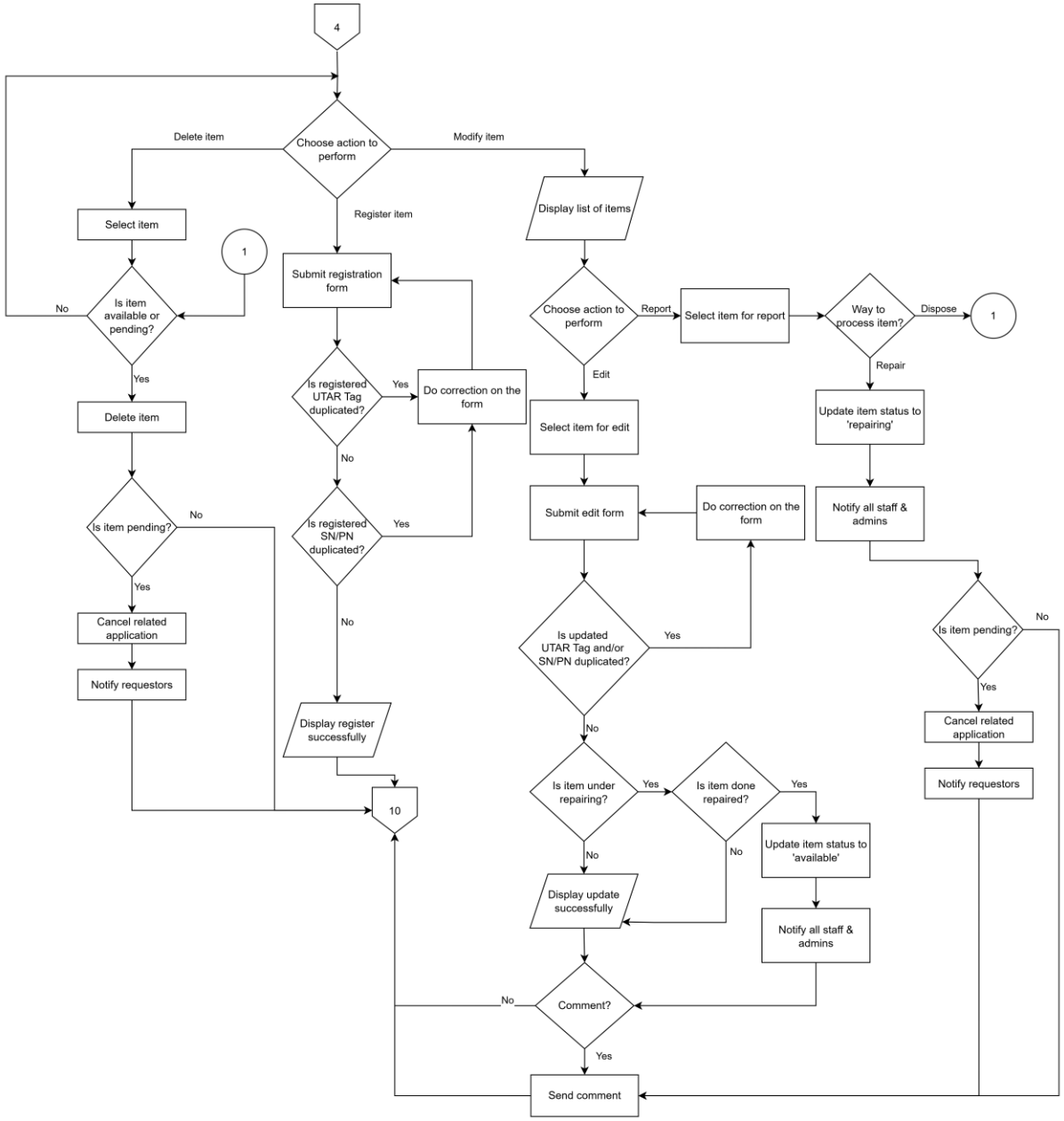


Figure 4.3.4: Flow Chart of Lab Inventory System (Part 4)

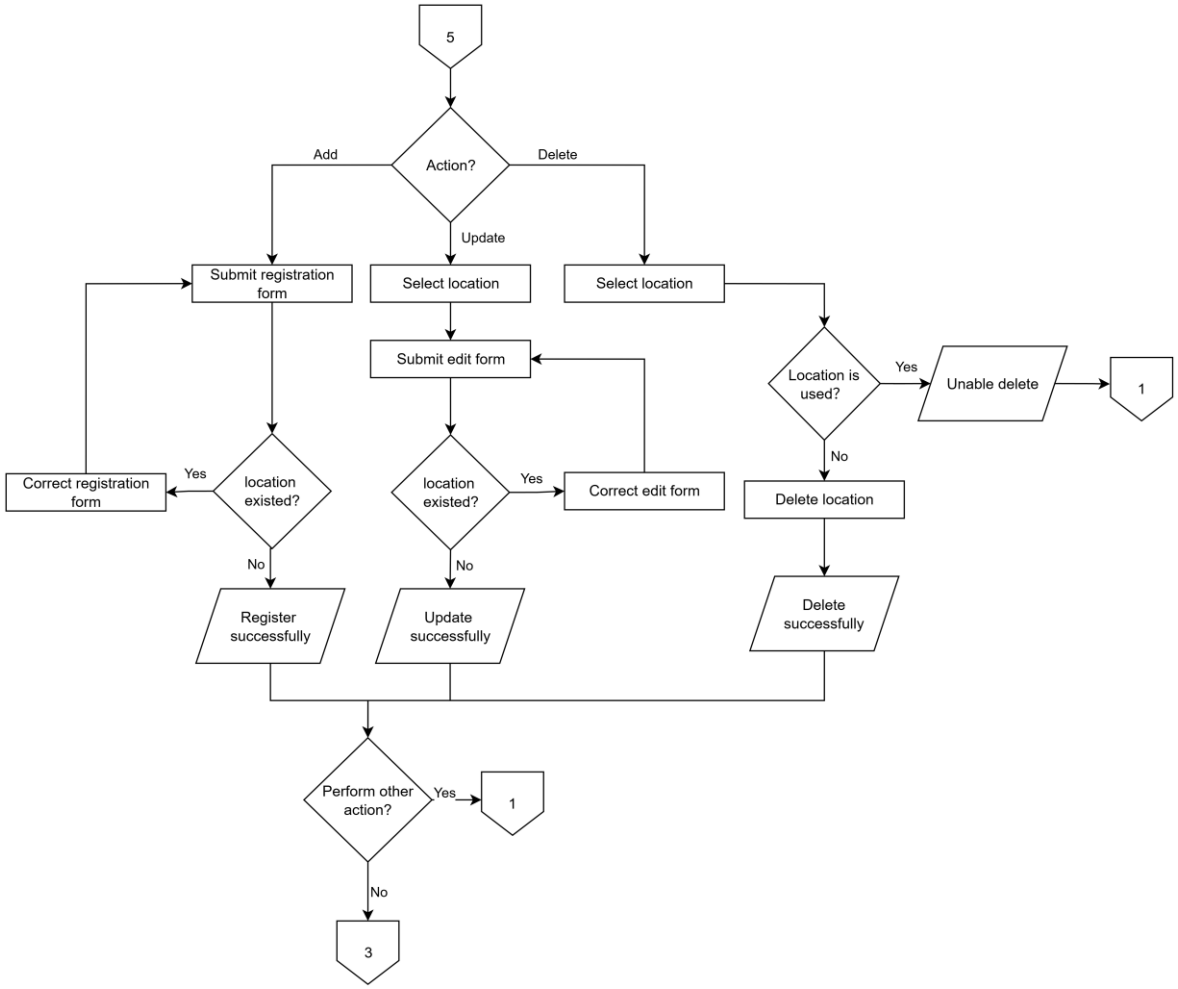


Figure 4.3.5: Flow Chart of Lab Inventory System (Part 5)

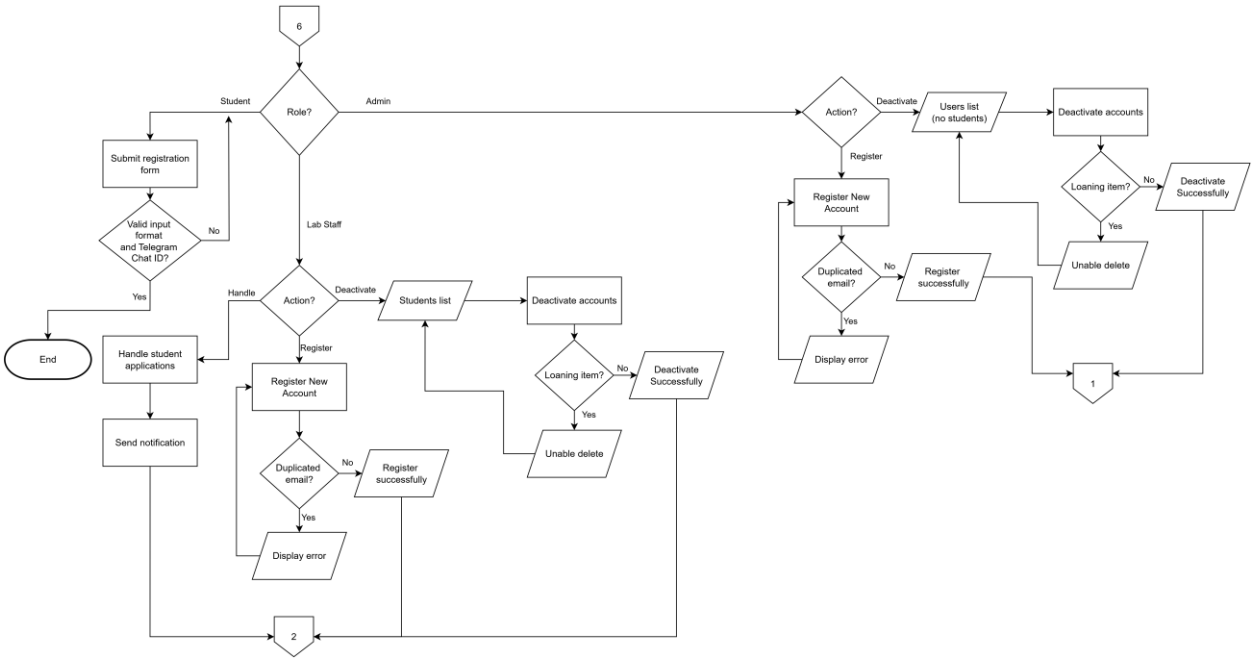


Figure 4.3.6: Flow Chart of Lab Inventory System (Part 6)

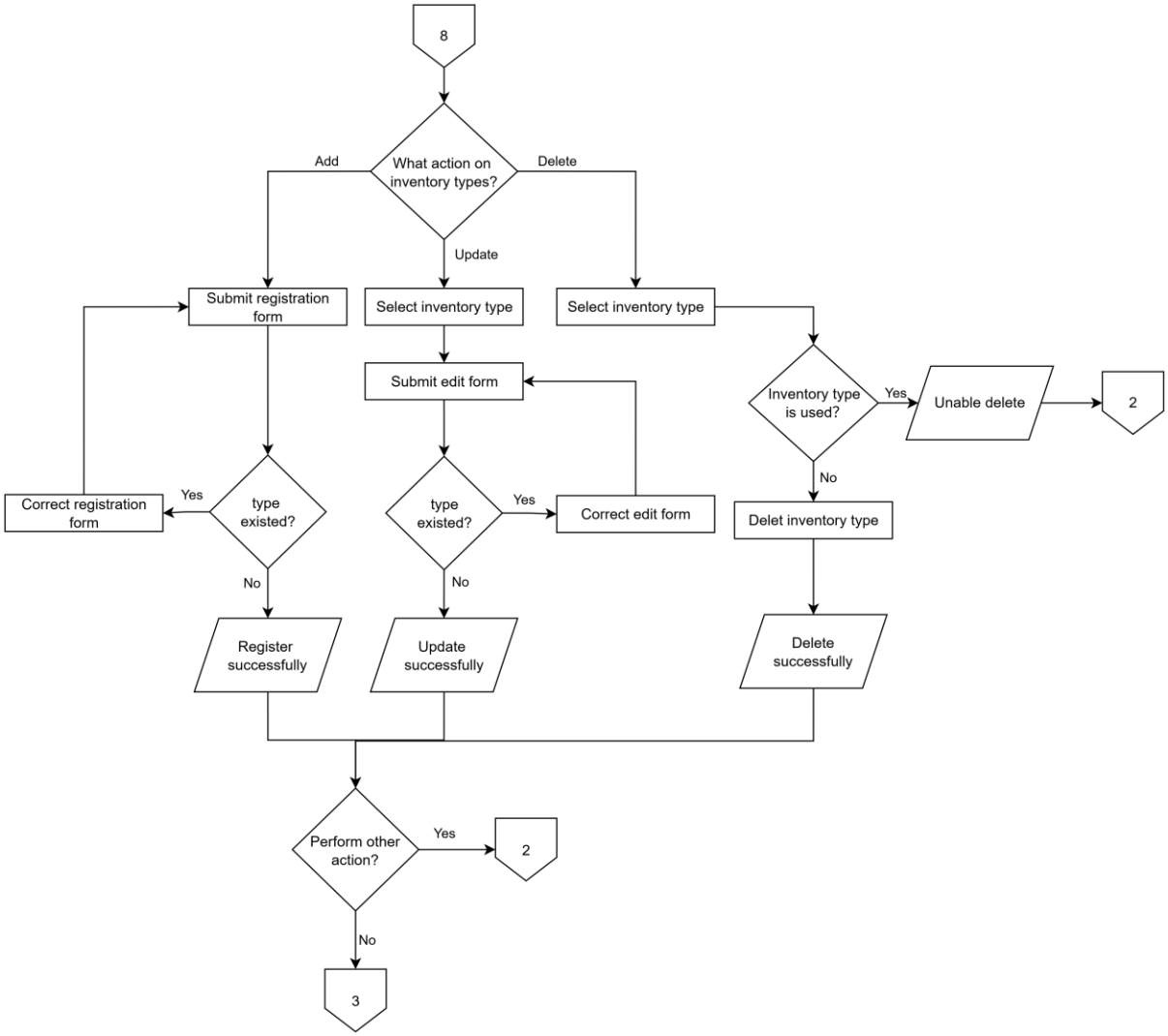


Figure 4.3.7: Flow Chart of Lab Inventory System (Part 7)

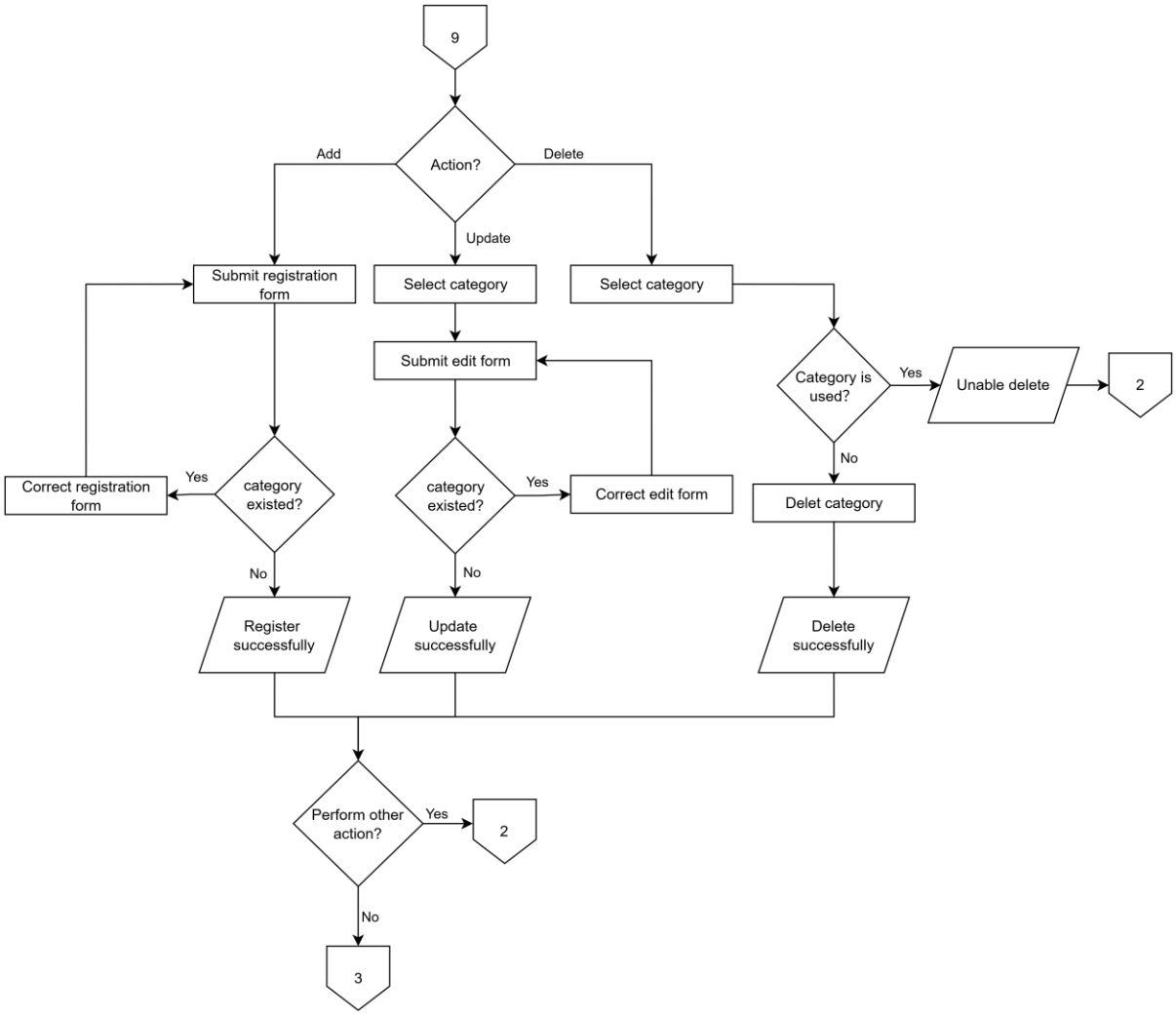


Figure 4.3.8: Flow Chart of Lab Inventory System (Part 8)

According to flow chart (part 1) above, it shows the workflows about how the loan application is applied, renewed, and handled among student, supervisor, admin, and lab staff. Then, the part 2 states the actions that can be perform by administrator while the part 3 show the actions that can be perform by lab staff. Then, the remaining flow charts present the workflows about other modules which are explained in the table below:

Table 4.3.1: Explanations of flow charts for parts 4, 5, 6, 7, and 8

Parts	**Modules**	**Description**
4	Inventory Management Module	Flows about how item is registered, edited, and removed by following the rules
5	Location management module	Flows about how location is registered, edited, and removed by following the rules

6	User management module	Flows about how user account is registered, edited, and removed by following the rules
7	Item types management module	Flows about how item type is registered, edited, and removed by following the rules
8	Item categories management module	Flows about how item category is registered, edited, and removed by following the rules

4.4 Database Design

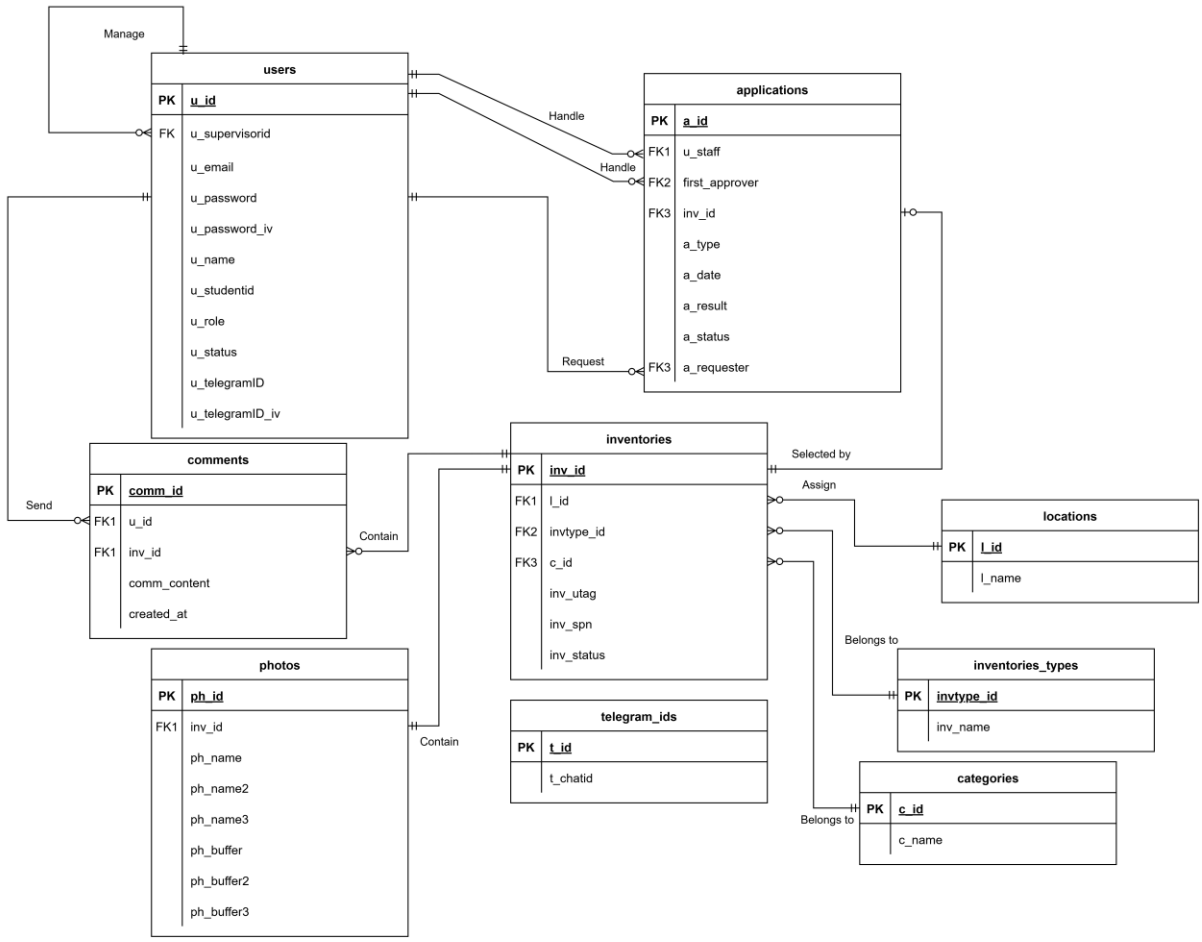


Figure 4.4.1: Entity Relationship Diagram of Lab Inventory Monitoring System

According to the ERD diagram above, the inventories entity is used to store information about the items, which includes UTAR tag, SN/PN, status, and so on. Then, the locations entity, inventories_types entity, and categories entity are used to make the inventory management

more efficient as they can help users to modify the location, category or type of group of items at once, no need edit them row by row in inventory management.

Furthermore, the photos entity is used for storing the set of photos that belongs to each inventory. Besides that, comments entity stores the comment contents regarding the items' status. Then, the most complex entity within this system is users entity. For ensuring the flexibility and efficiency of user management, the information of students, supervisors, lab staff, and administrators will be stored in this entity. One of the reasons for applying this structure is when checking the duplicated email address or telegram chat ID, this process will be simple and easy. The attributes u_password_iv and u_telegramID_iv store the unique initialization vectors which are used for decryption of password and Telegram chat ID. Additionally, telegram_ids contains Telegram chat ID that has been hashed which used for verification of Telegram chat ID during the account registration. Then, the table below explains the relationship among users, inventories, comments, and applications entities.

Table 4.4.1: Explanations of relationships among users, inventories, comments, and applications entities

Relationships	**Explanations**
users and users (manage)	• Each supervisor manages 1 or more students • Each student is managed by 1 and only 1 supervisors • Lab staff and administrators don't need the supervisors
users and applications (request)	• Each student or supervisor requests 0 or more loan applications/ renew loan period applications • Each application is requested by 1 and only 1 student or supervisor
users and applications (handle)	• Each lab staff (u_staff) handles the final confirmation of 0 or more applications • Each supervisor (first_approver) handles 0 or more applications

	• Each administrator (first_approver) handles 0 or more applications
applications and inventories (selected by)	• Each application selects 1 and only 1 inventory • Each inventory is selected by 0 or 1 application
users and comments (send)	• Each user can send 0 or more comments • Each comment is sent by 1 and only 1 user
comments and inventories (contain)	• Each inventory contains 0 or more comments • Each comment is contained by 1 and only 1 inventory

4.5 GUI Design

This section shows the wireframes for sign in page, sign up page, home page and each module.

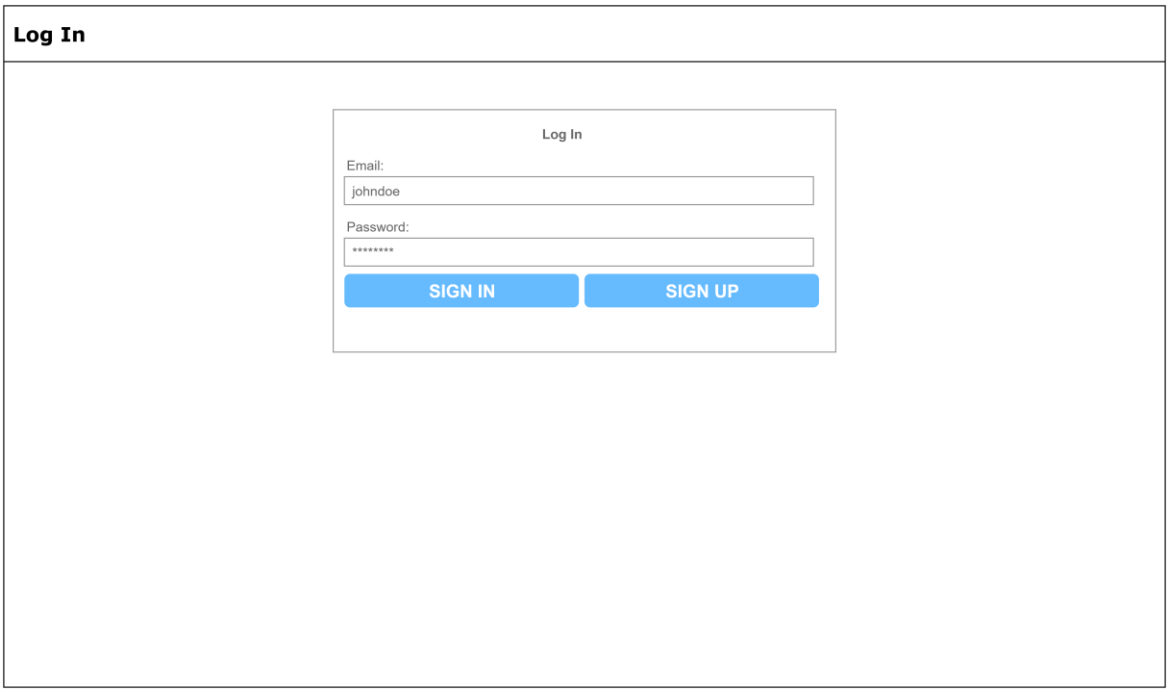


Figure 4.5.1: Interface for sign in and sign up

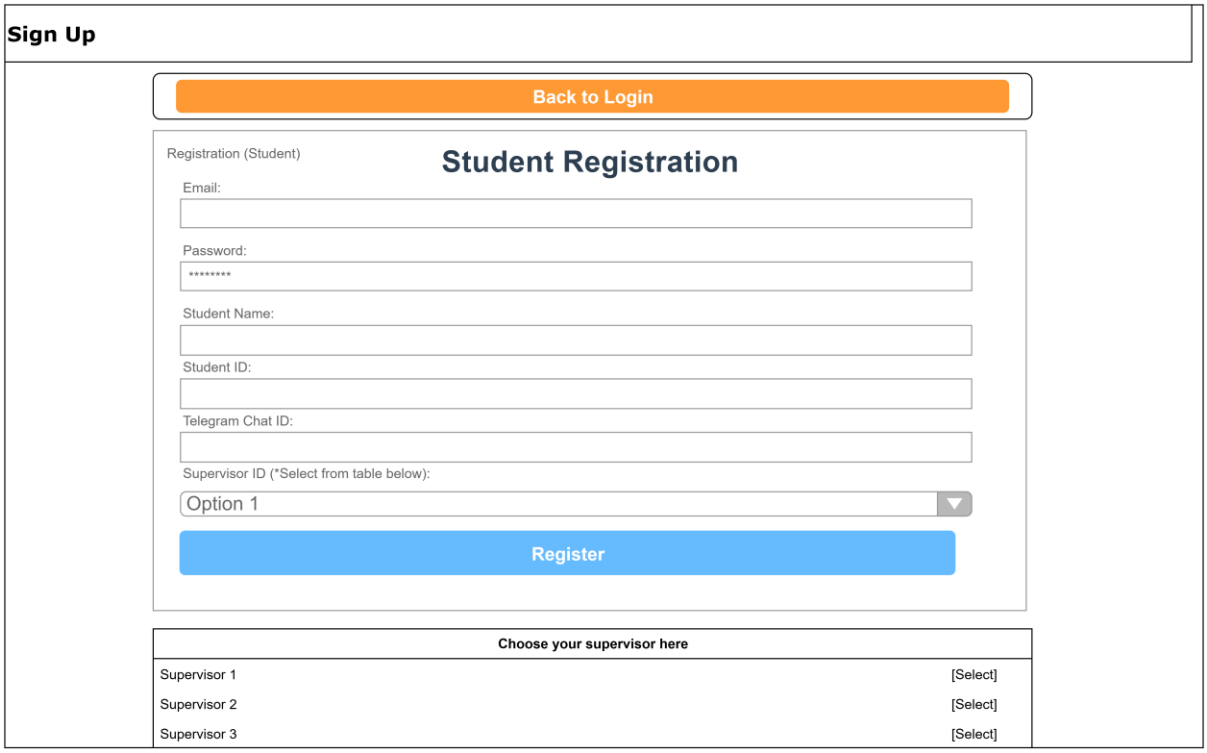


Figure 4.5.2: Interface for sign up form

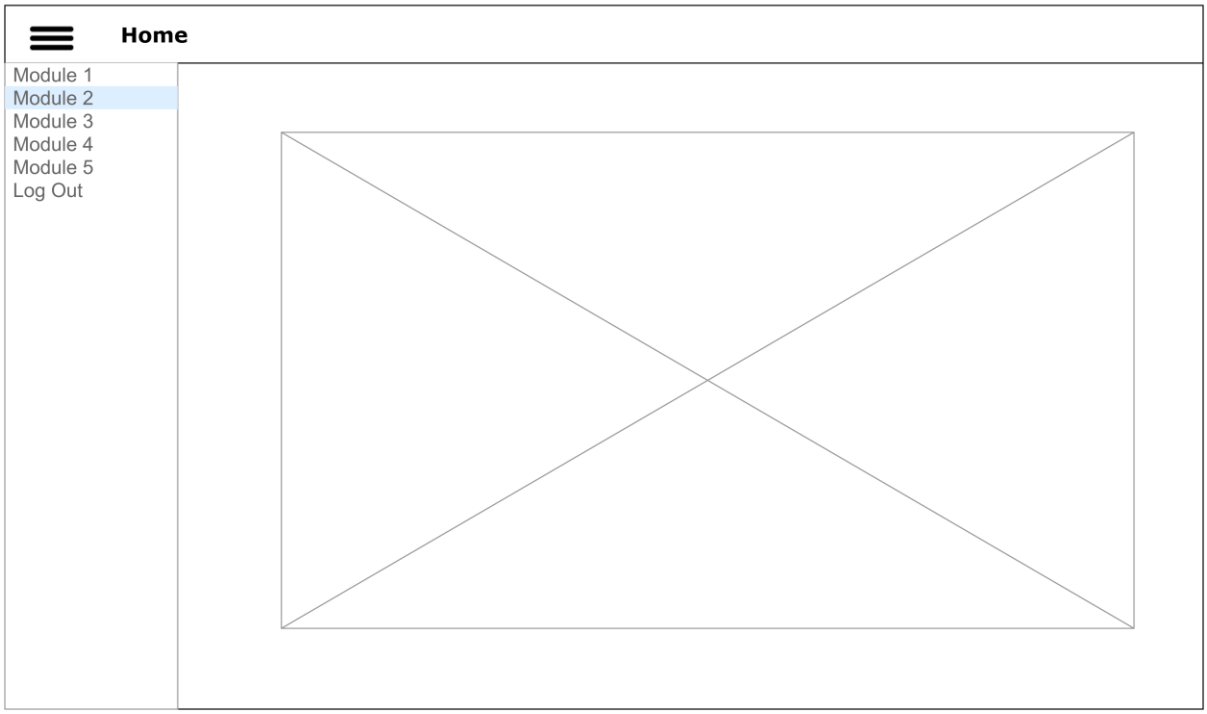


Figure 4.5.3: Home page for all user roles

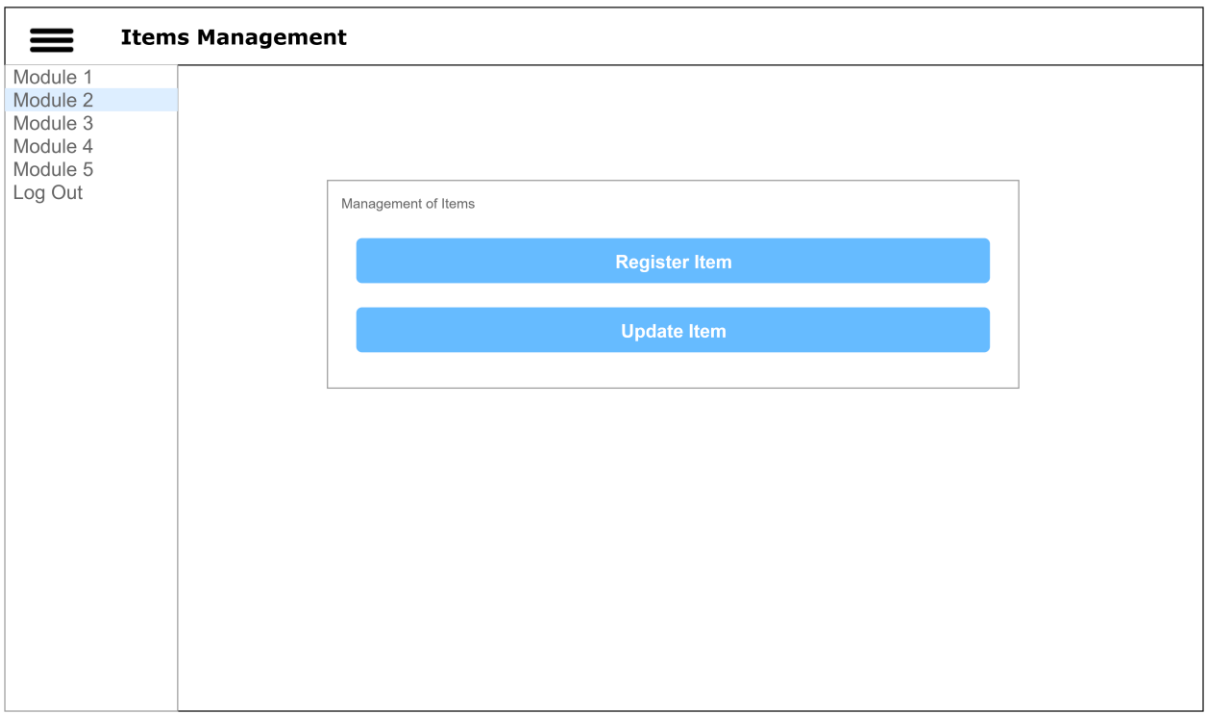


Figure 4.5.4: Main page of inventory management module

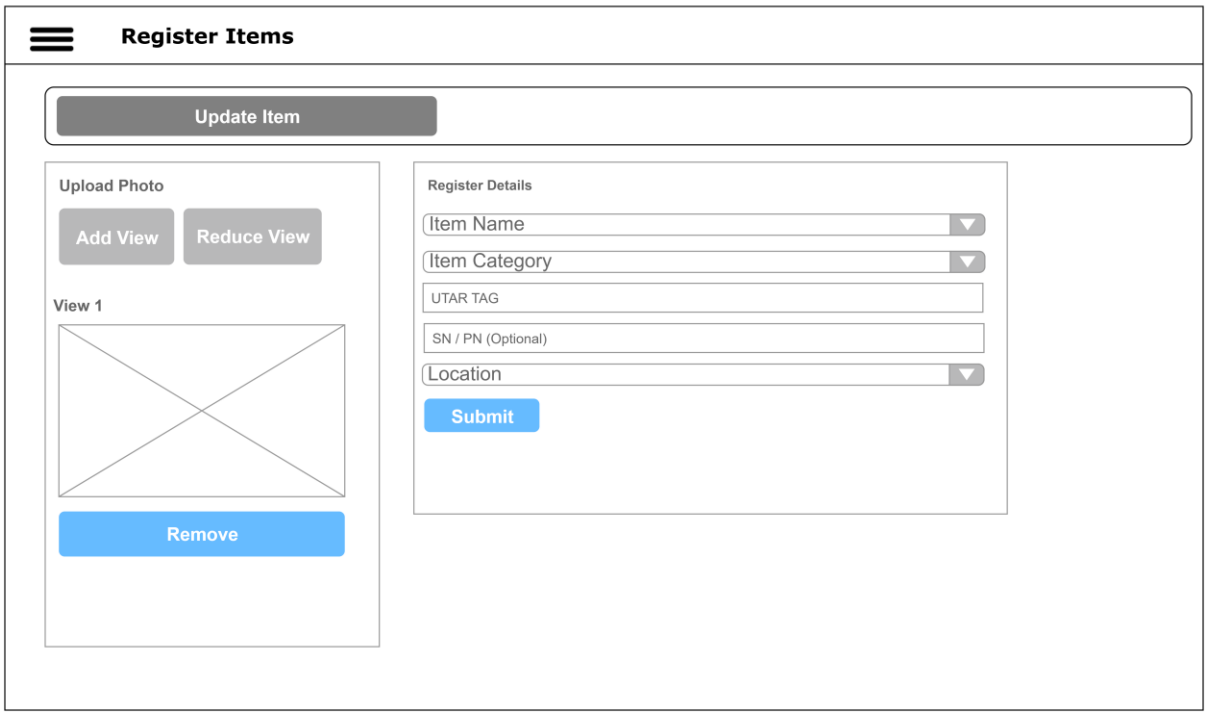


Figure 4.5.5: Interface for registering new inventory

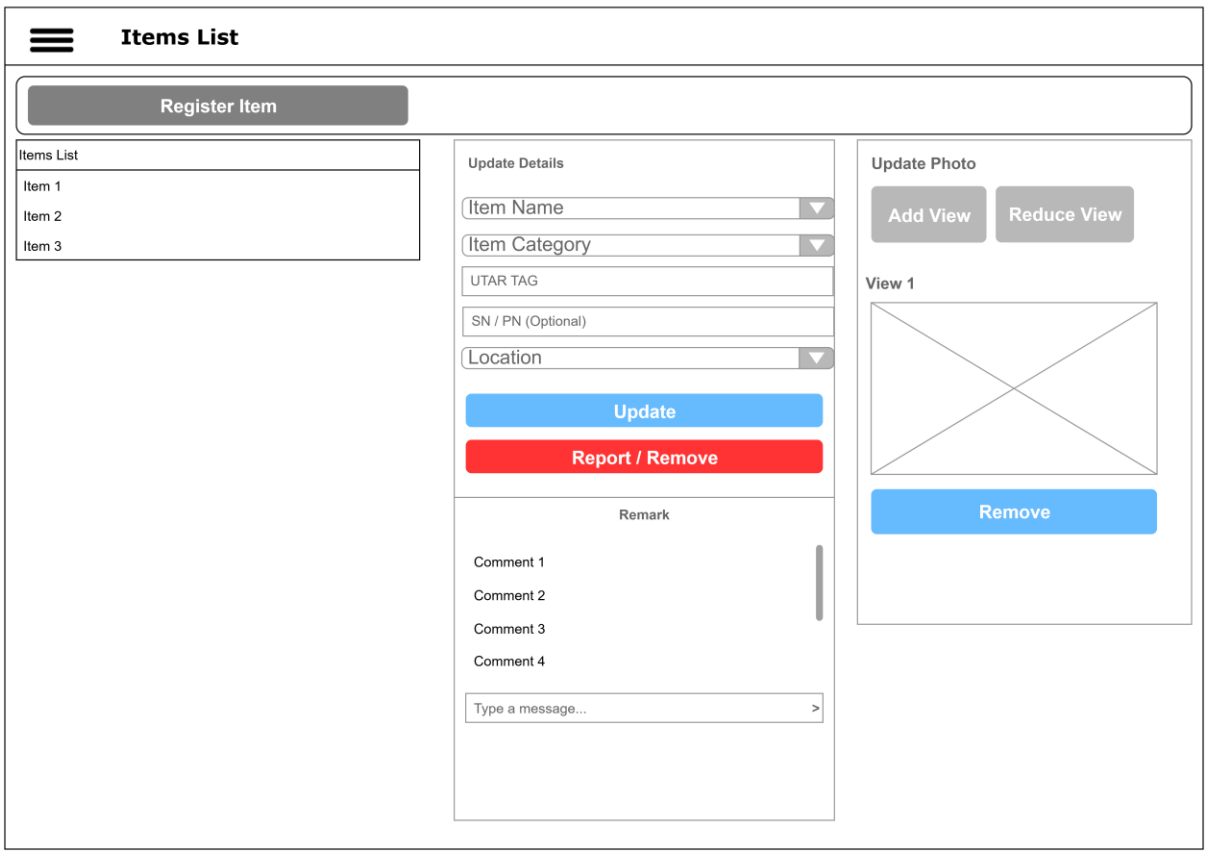


Figure 4.5.6: Interface for modifying and commenting on existing inventory

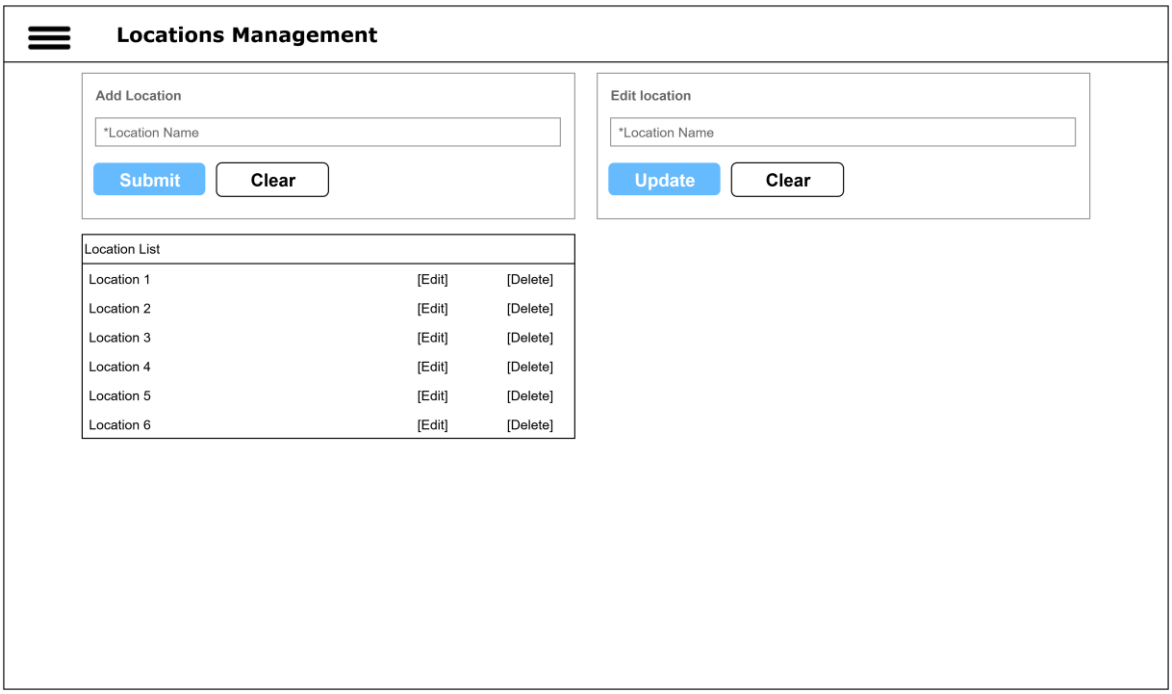


Figure 4.5.7: Interface of location management module

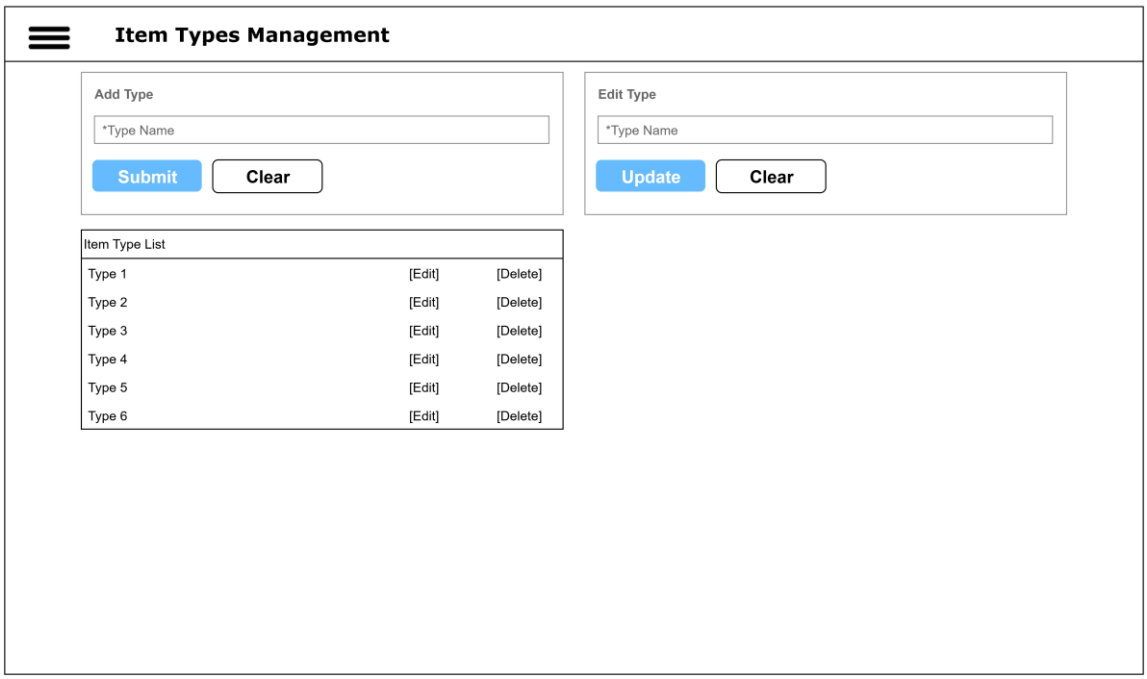


Figure 4.5.8: Interface of item types management module

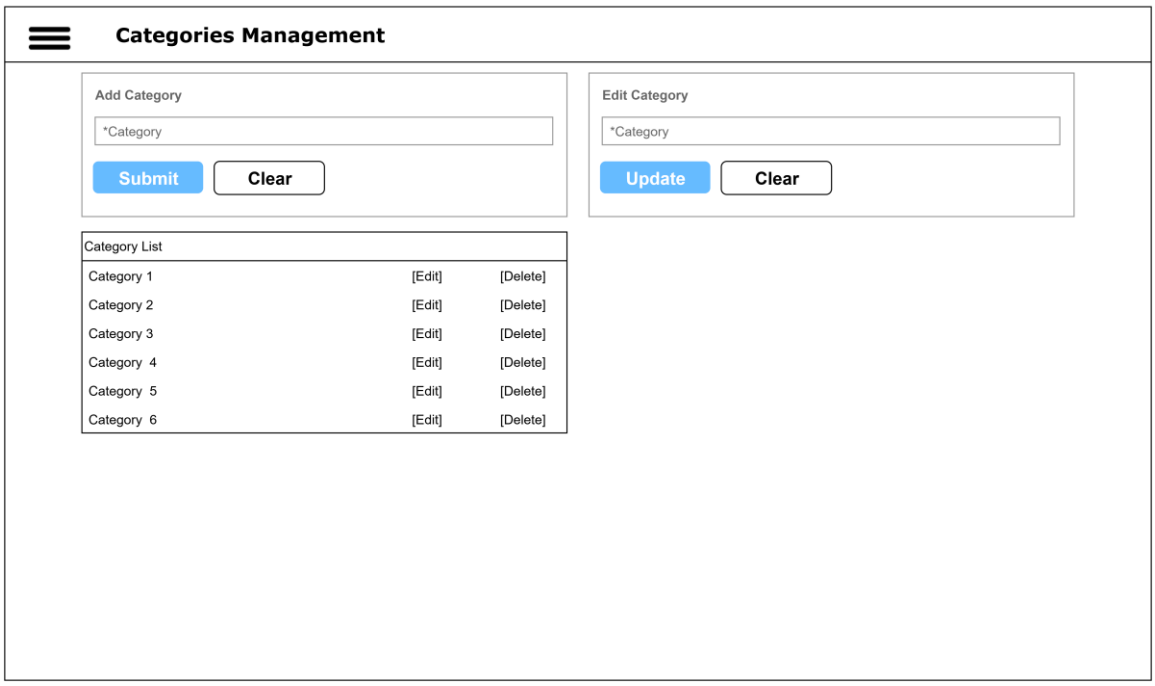


Figure 4.5.9: Interface of item categories management module

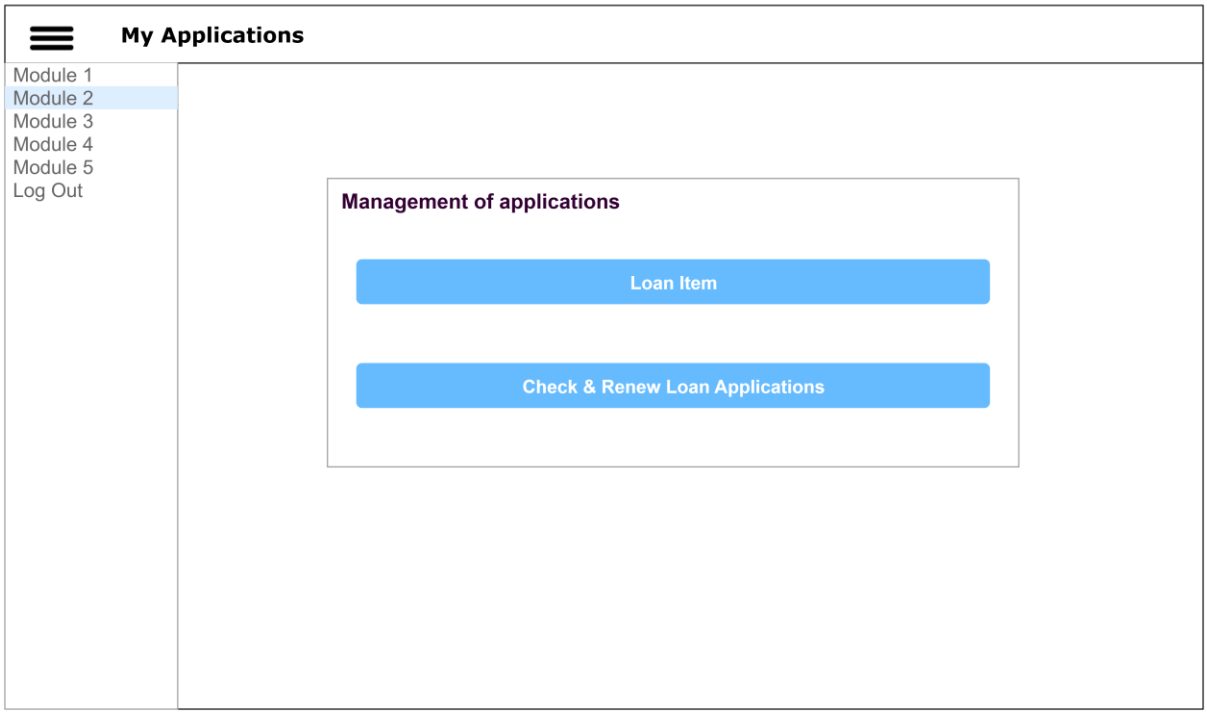


Figure 4.5.10: Interface of "My Application" (For Students & Supervisors)

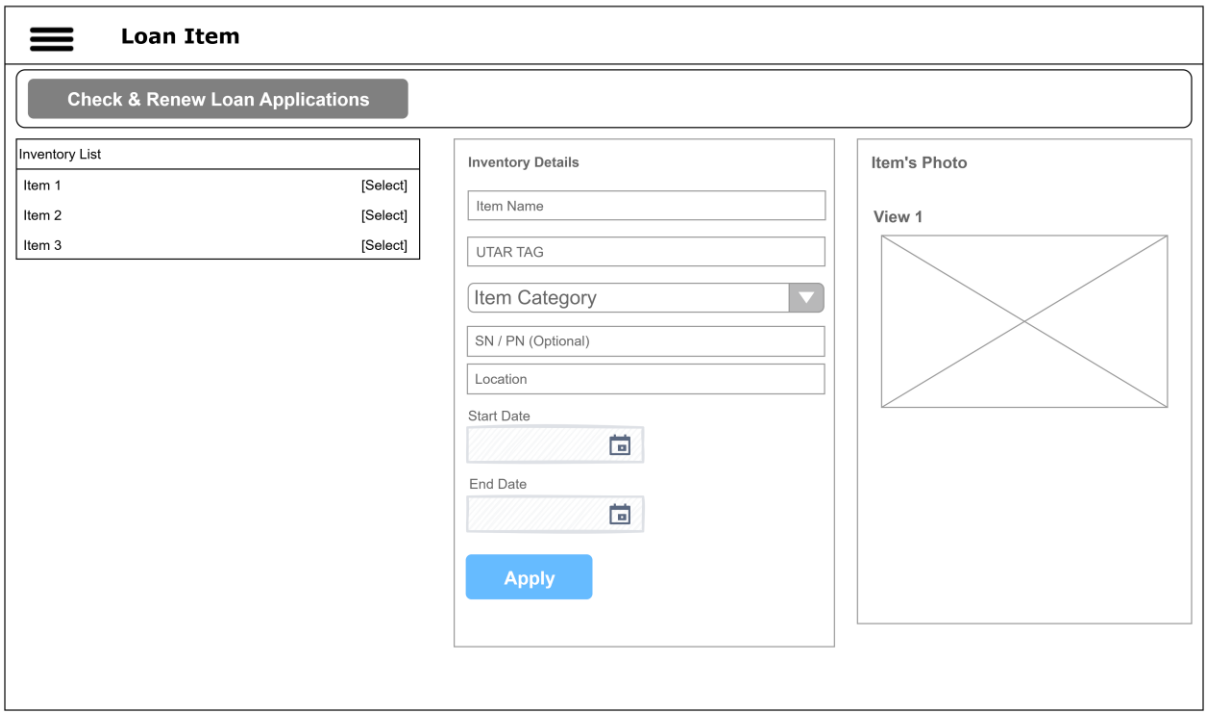


Figure 4.5.11: Interface of applying loaning item

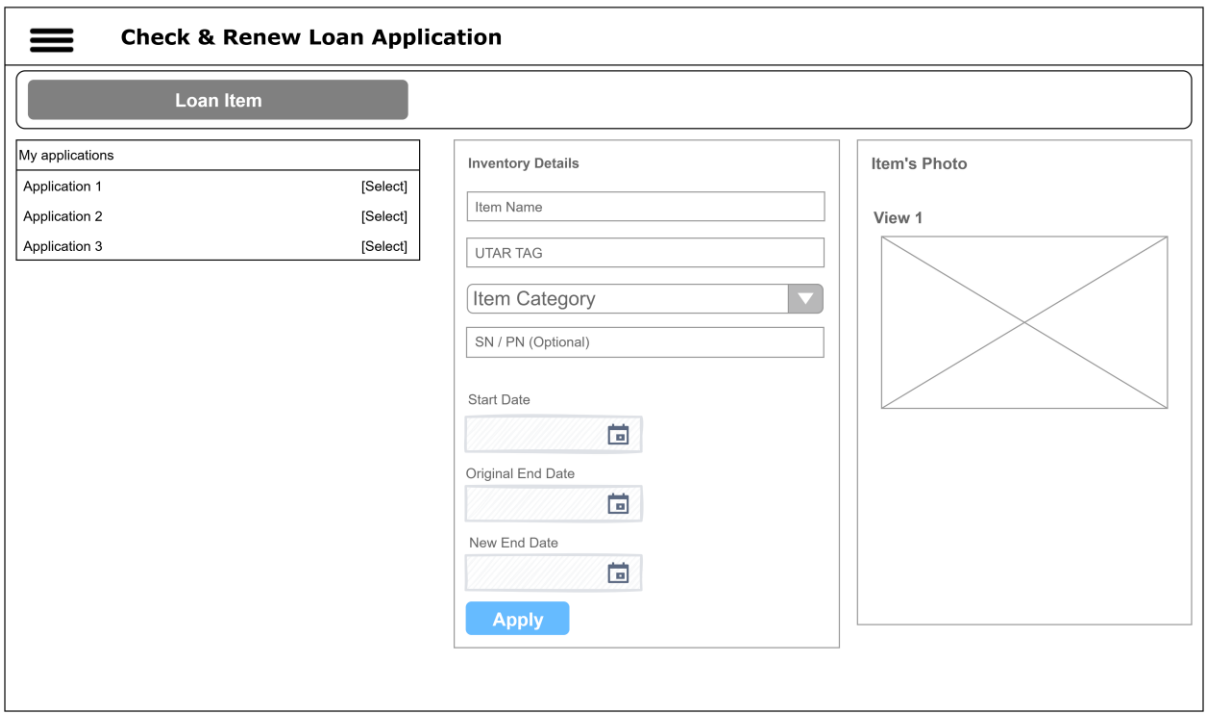


Figure 4.5.12: Interface for checking applications' status and renewing application

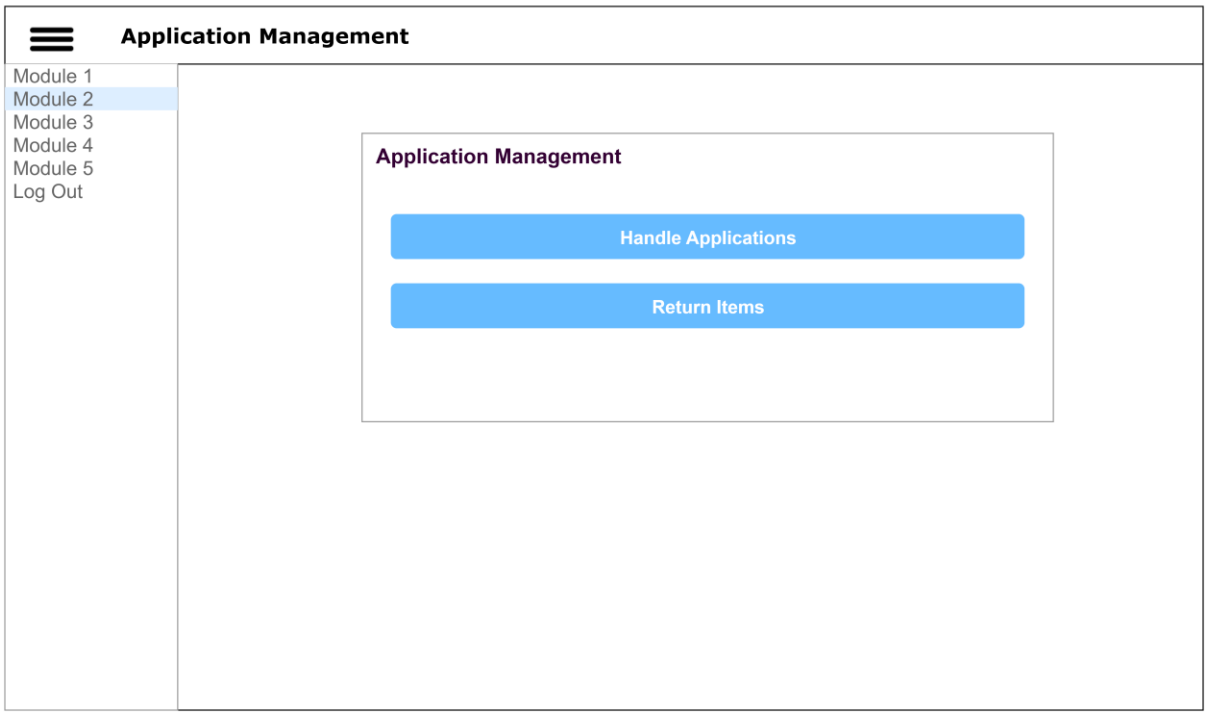


Figure 4.5.13: Main page of applications management module (For Lab Staff)

Supervisors can access both main parts (My Applications and Handle Applications) of the loan application management module. This is because they can apply loaning items, renew applications, and handle applications from the students.

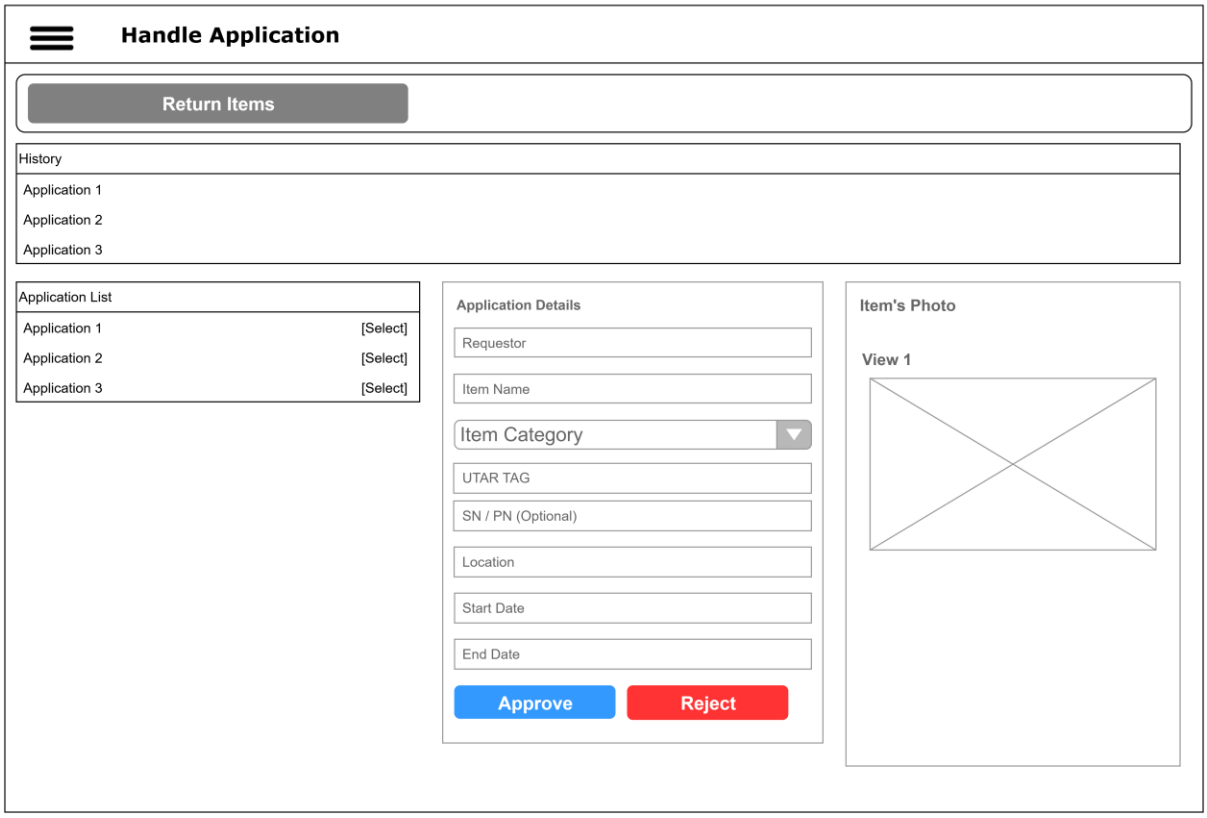


Figure 4.5.14: Interface of approve/reject applications (For Lab Staff)

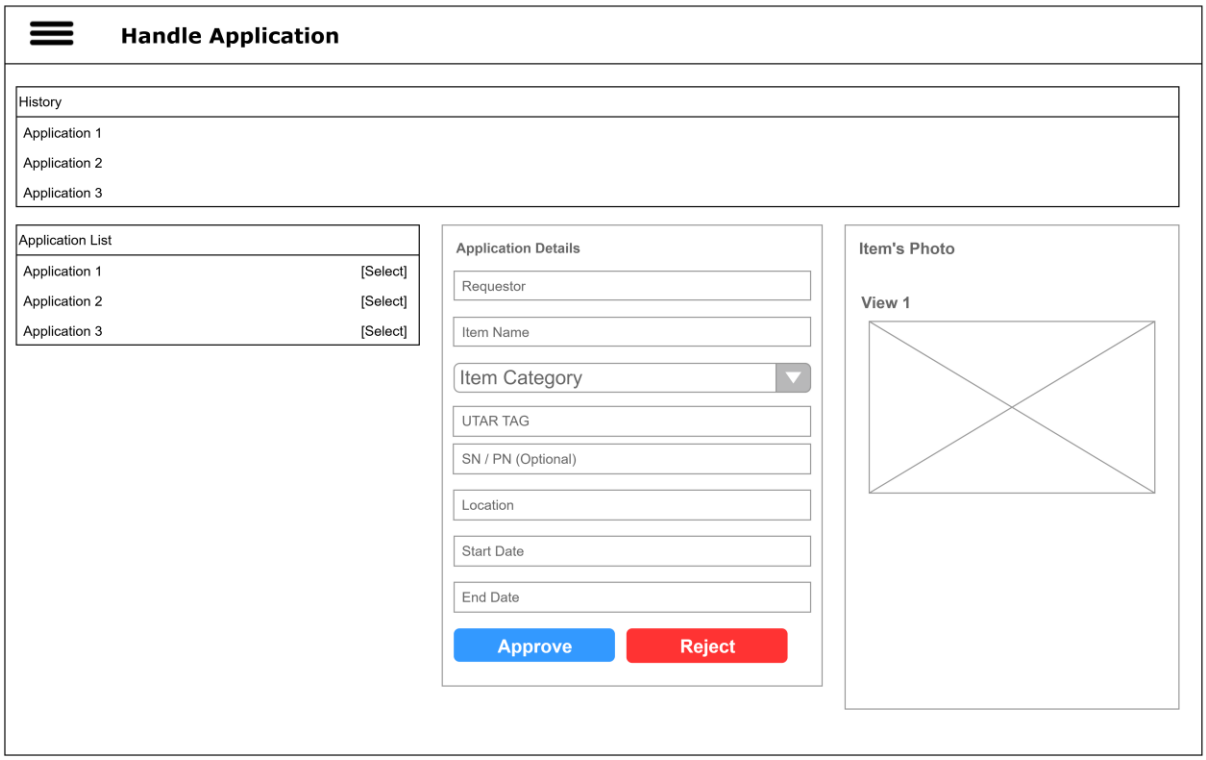


Figure 4.5.15: Interface of approve/reject applications (For Supervisor and Adminstrator)

The form in “Handle Application” section will be changed dynamically according to the type of selected application. Moreover, the application list of this section will also be changed dynamically according to the current user role. The application list will show applications of supervisors for admin while the applications of students will be shown to corresponding supervisors.

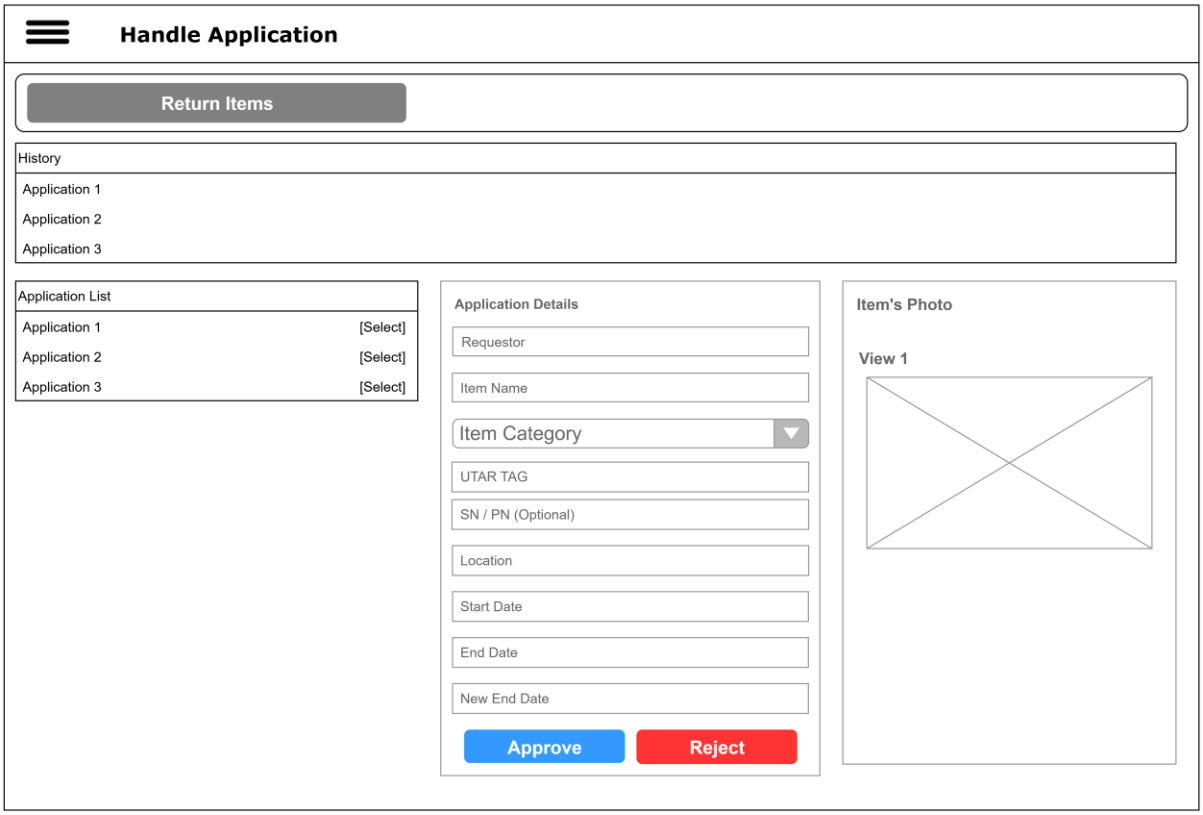


Figure 4.5.16: Interface of approve/reject the renew period applications (For Lab Staff)

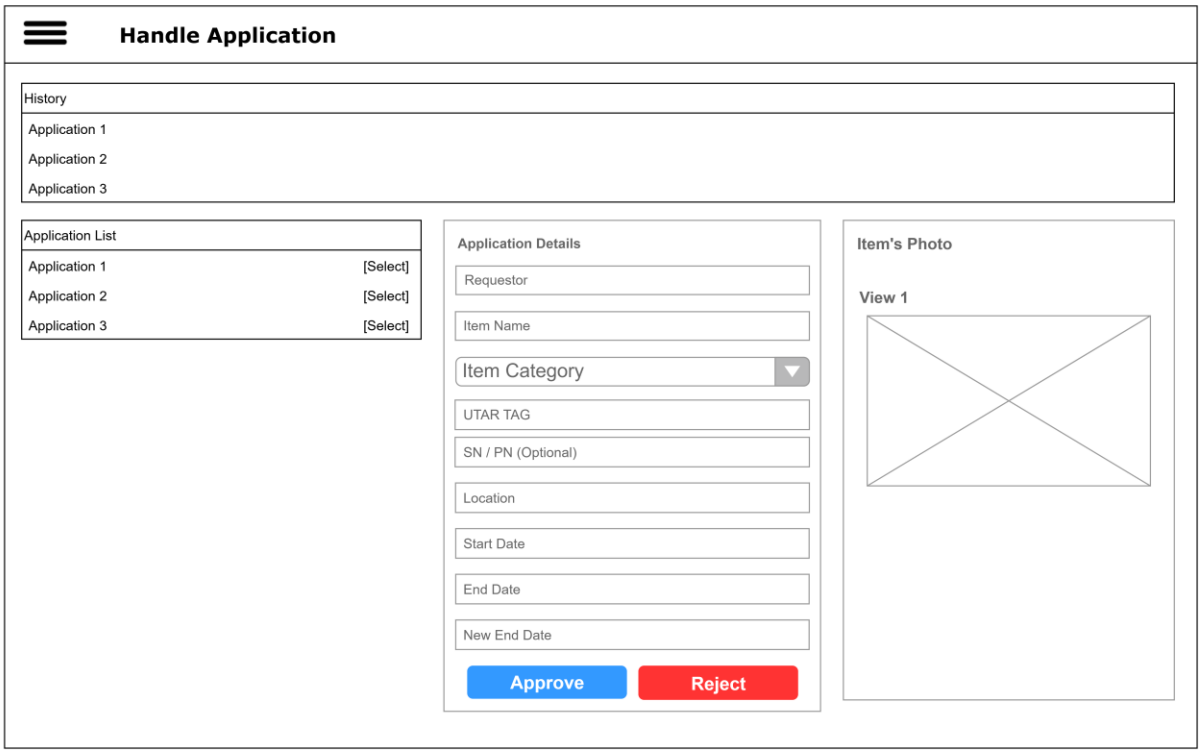


Figure 4.5.17: Interface of approve/reject the renew period applications (For Supervisor and Adminstrator)

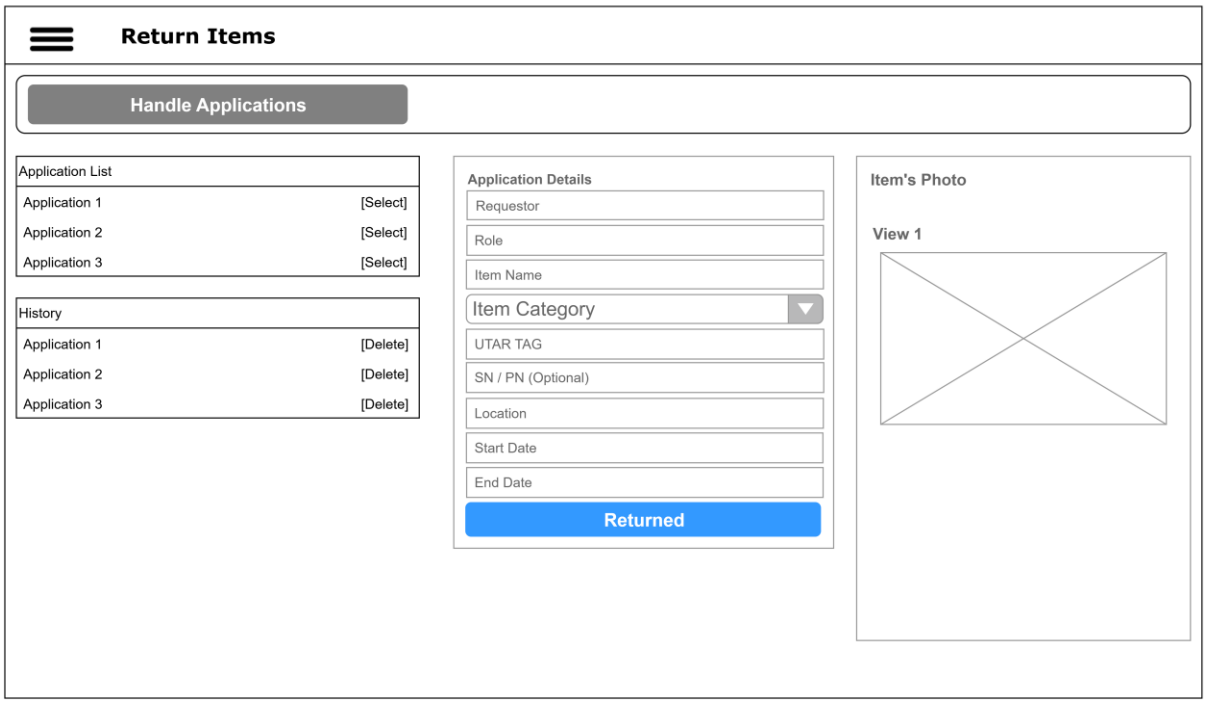


Figure 4.5.18: Interface of Return Items (For Lab Staff)

The main page of user management module will be changed according to the role of the users.

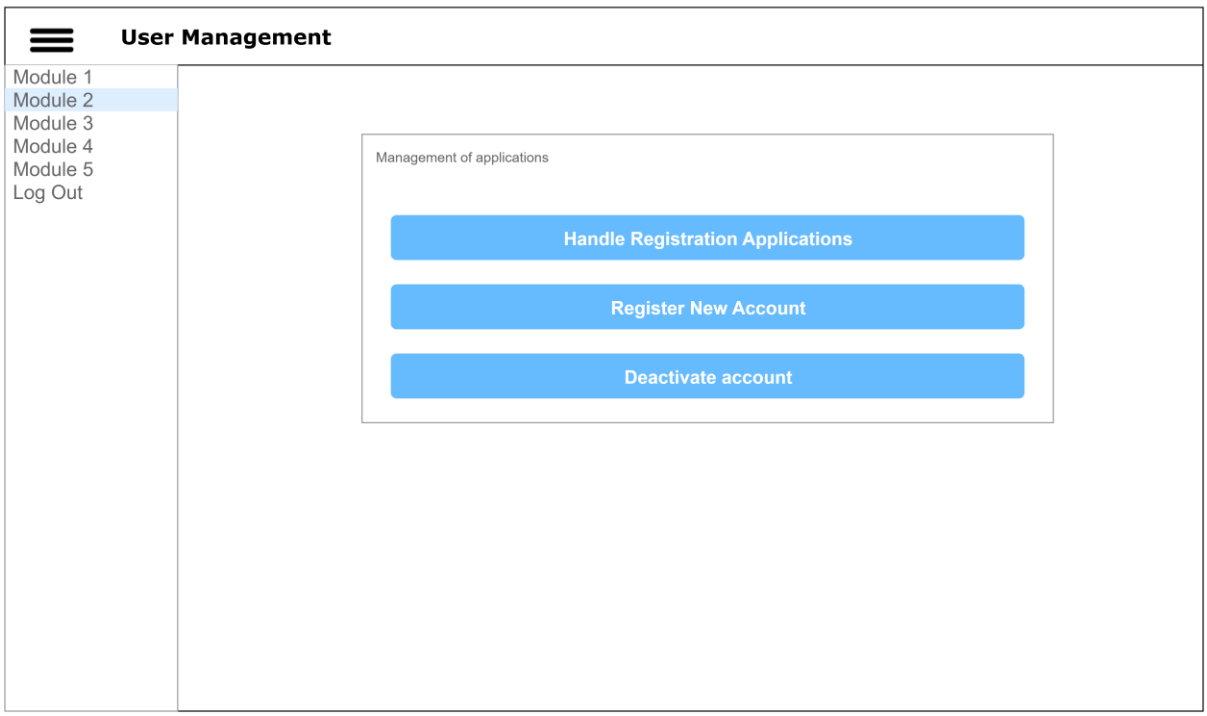


Figure 4.5.19: Main page of user management module (For Lab Staff)

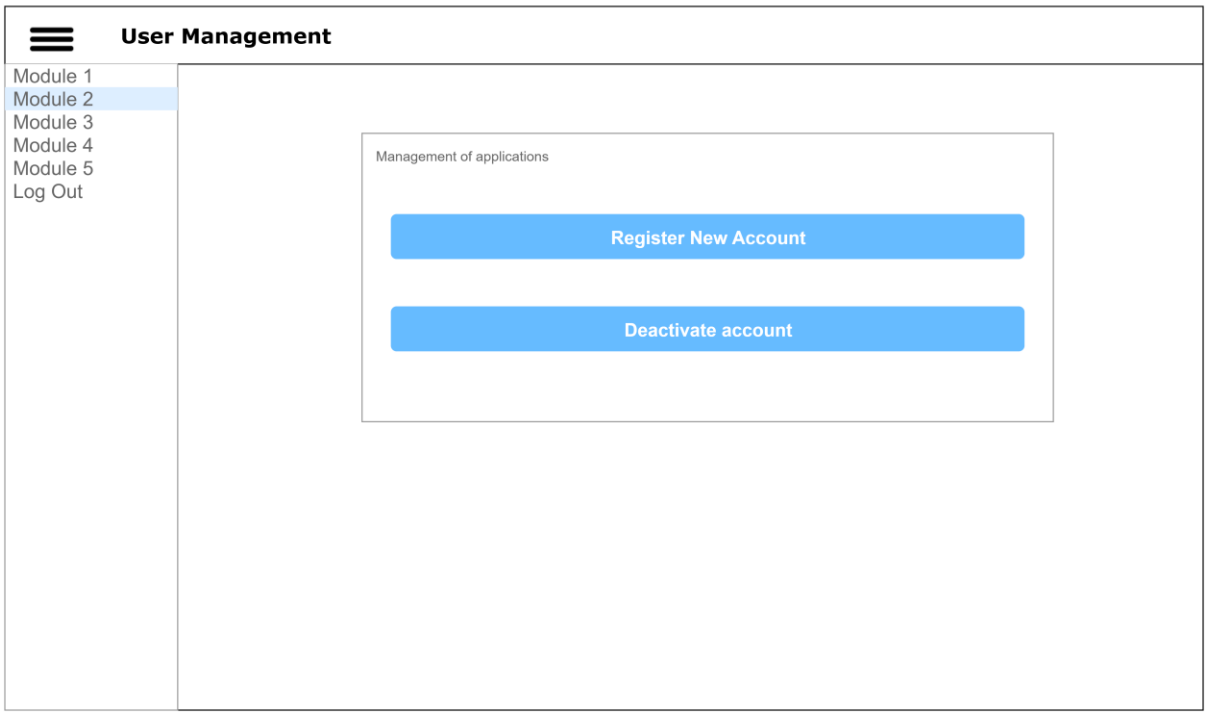


Figure 4.5.20: Main page of user management module (For Administrator)

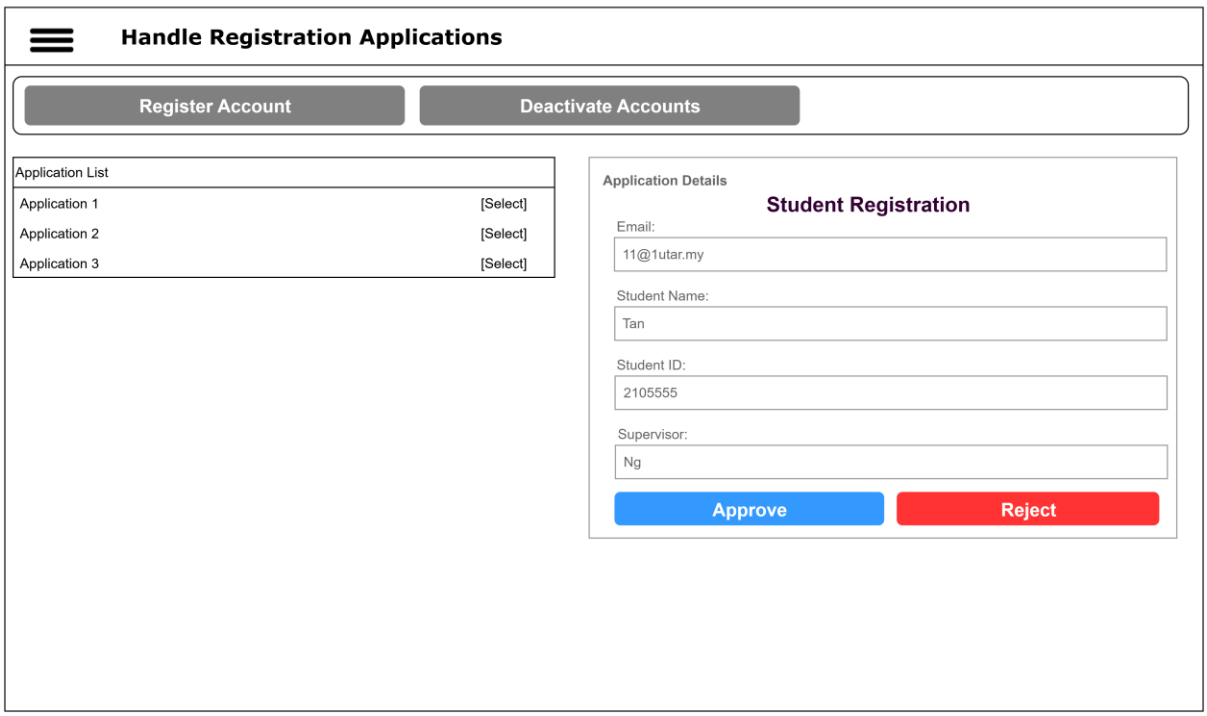


Figure 4.5.21: Interface of handling registration applications for student's account

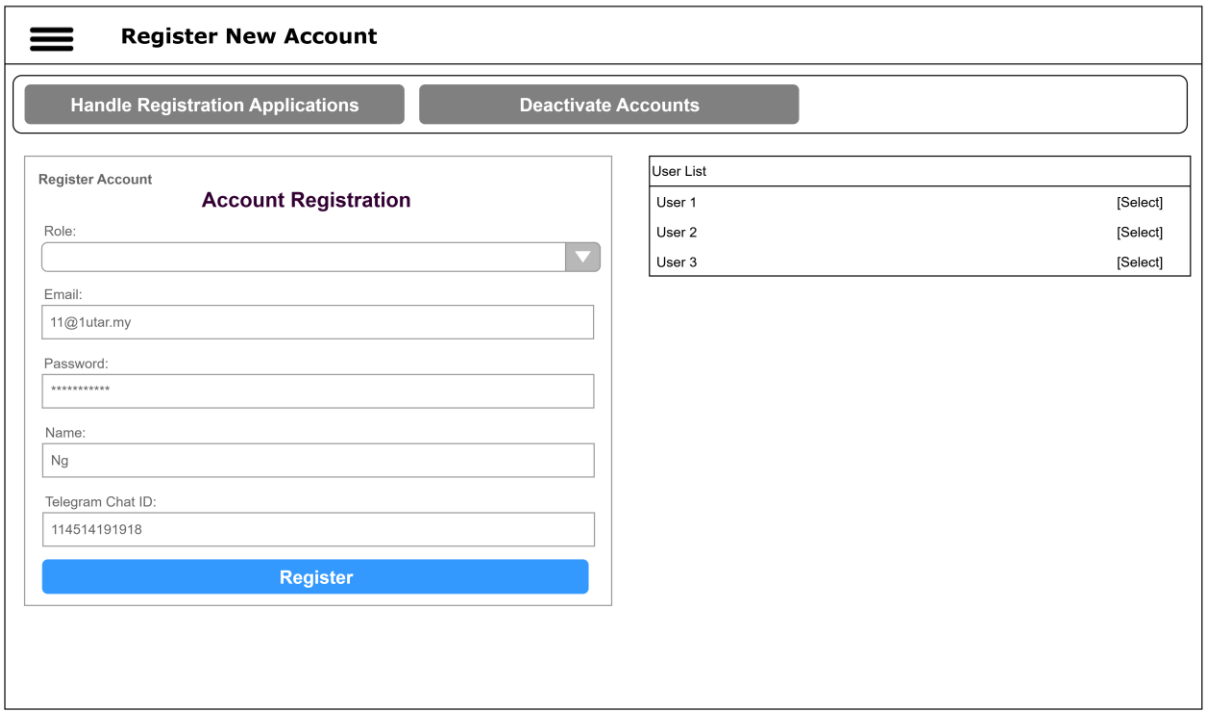


Figure 4.5.22: Interface for registering new account (Lab Staff)

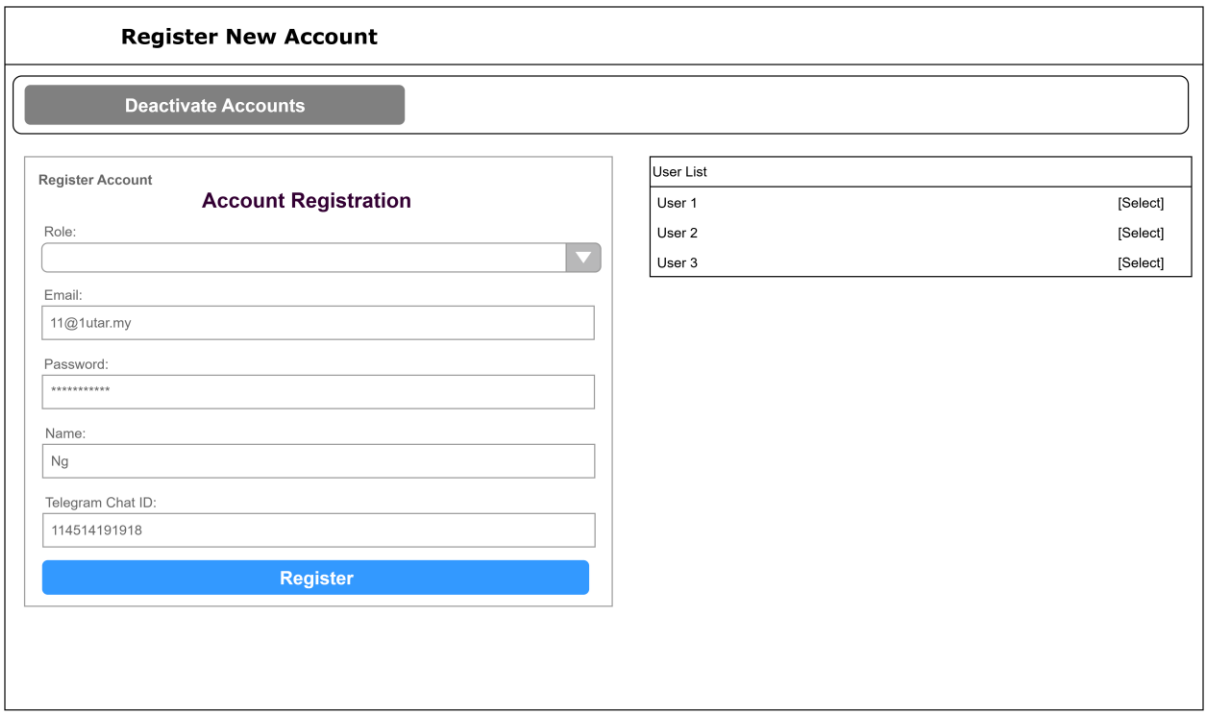


Figure 4.5.23: Interface for registering new account (Administrator)

Lab staff are only allowed to deactivate student accounts while administrator can deactivate any accounts with any role other than student.

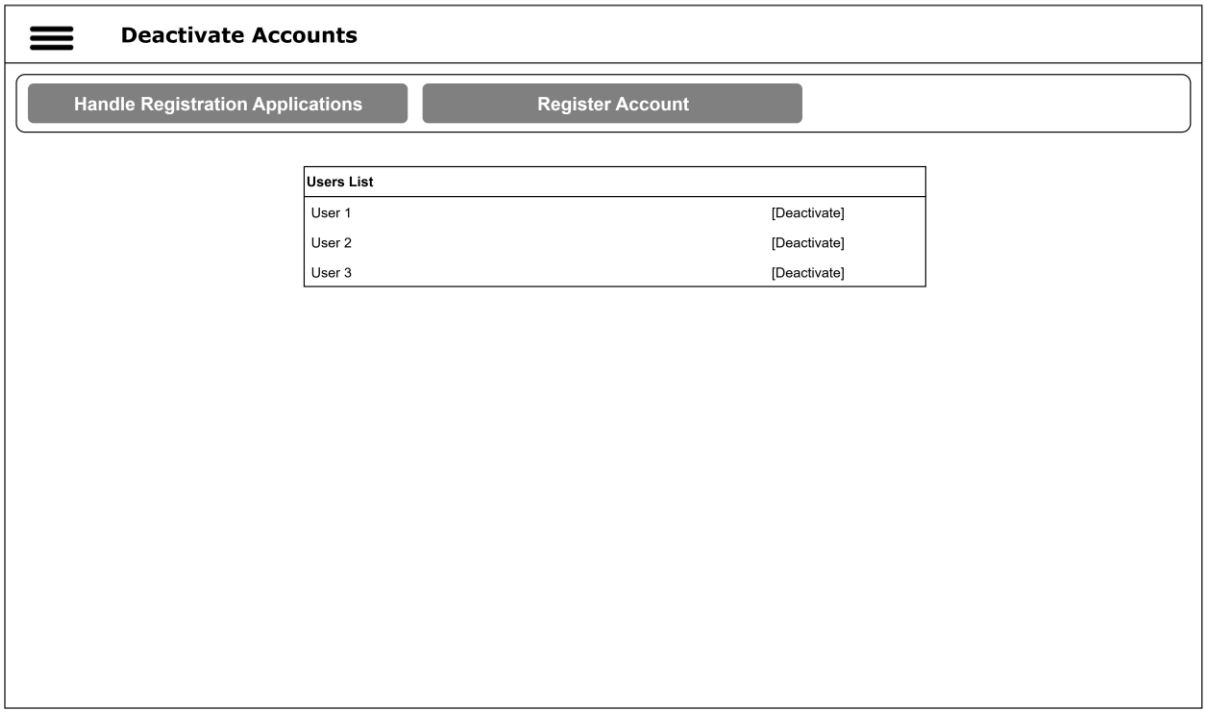


Figure 4.5.24: Interface of deactivation of accounts (Lab Staff)

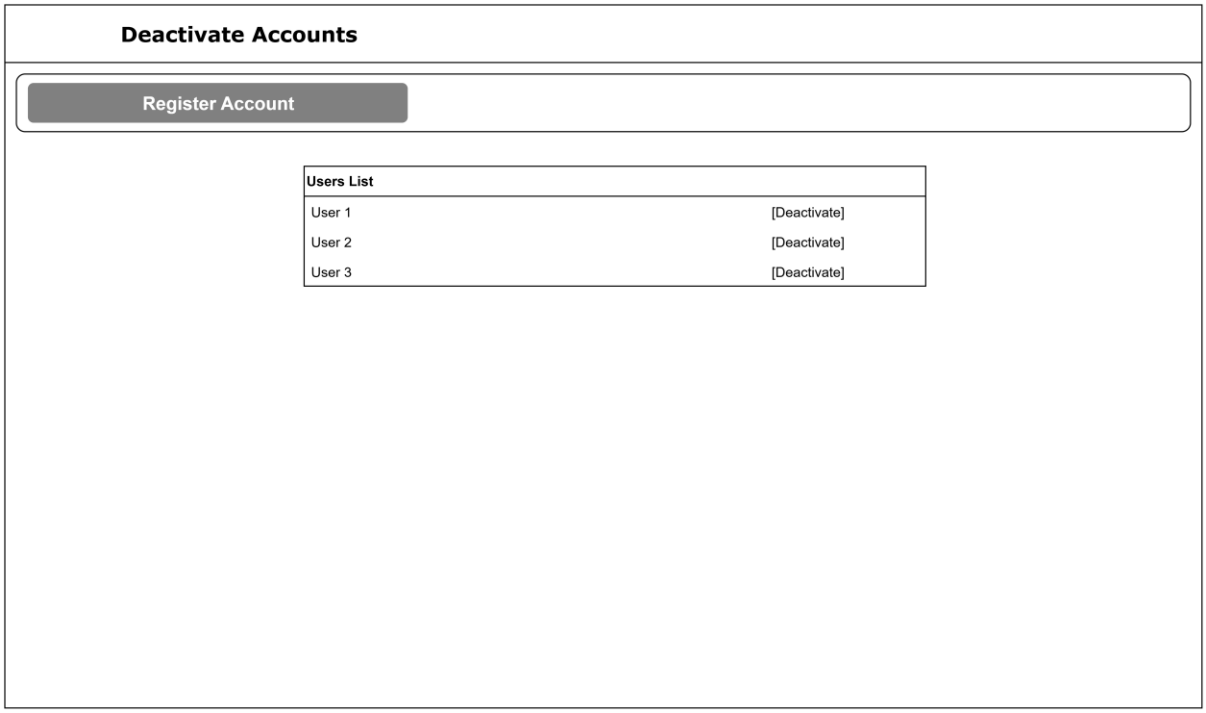


Figure 4.5.25: Interface of deactivation of accounts (Administrator)

4.6 Concluding Remark

This chapter mainly describes the system design of lab inventory monitoring system which including the interactions of different user roles with each module within the system, the structure of database, and the overall graphical user interface of this system.

Chapter 5 - System Implementation

5.1 Hardware Setup

This project involves three hardware which are laptop, lab computer, and mobile phone. For the laptop, the Node-RED, MySQL workbench, and Telegram are installed for developing the system and enhancing the system after testing. For the desktop, it is also installed Node-Red, MySQL workbench, and Telegram but it is for testing and deployment purposes. It just needs to import the developed JSON file into Node-RED and import the SQL file which contains structured tables and data into the database to make the system could be function. Moreover, it is connected to a specific local area network, which enables the users within this network to access and utilize the lab inventory monitoring system. On the other hand, the mobile phone connects to internet to ensure that it can receive the notifications at anywhere from the Telegram bot.

Table 5.1.1: Specifications of laptop

Description	**Specifications**
Model	Gigabyte G5 AX200NGW
Processor	Intel Core i5-10500H
Operating System	Windows 11
Graphic	NVIDIA GeForce RTX 3060 6GB
Memory	32GB DDR4 RAM
Storage	4TB SSD

Table 5.1.2: Specifications of desktop

Description	**Specifications**
Model	Acer Veriton M2640G
Processor	Intel Core i5-6400 CPU @ 2.70GHz
Operating System	Windows 11 Pro
Graphic	AMD radeon R5-310 2GB DDR3
Memory	8 GB RAM
Storage	480GB SSD

Table 5.1.3: Specifications of mobile phone

Description	**Specifications**
Model	POCO F6 Pro
Processor	Qualcomm Snapdragon 8 Gen 2
Operating System	Android 14
RAM	12 GB
Battery Capacity	5000 mAh
Storage	256 GB

5.2 Software Setup

This project utilized 3 software, including Node-Red, MySQL Workbench, and Telegram.

Node-Red

The way to setup Node-Red is simple, just execute the given installation command from official website in the command prompt. After the installation is completed, execute command “node-red” on command prompt also to launch it.

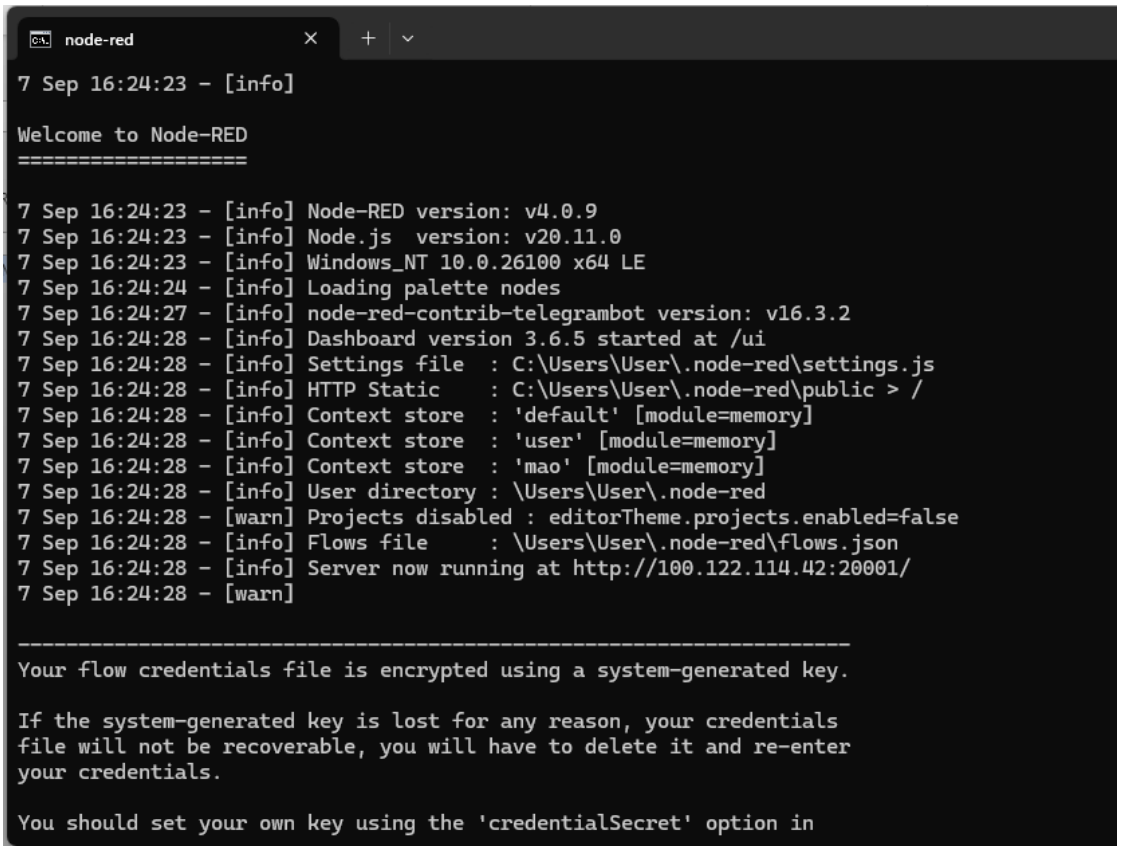


Figure 5.2.1: Prompt for launching Node-RED

MySQL Workbench

Download installer from official website and launch it. Then, select the destination folder for installation.

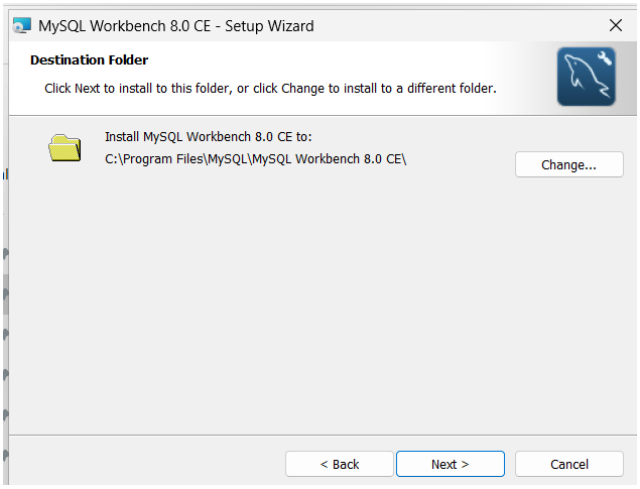


Figure 5.2.2: Select folder for installation

After selecting destination folder, choose complete installation and make the double confirmation on the installation. Then, wait for the installation of MySQL Workbench to be completed.

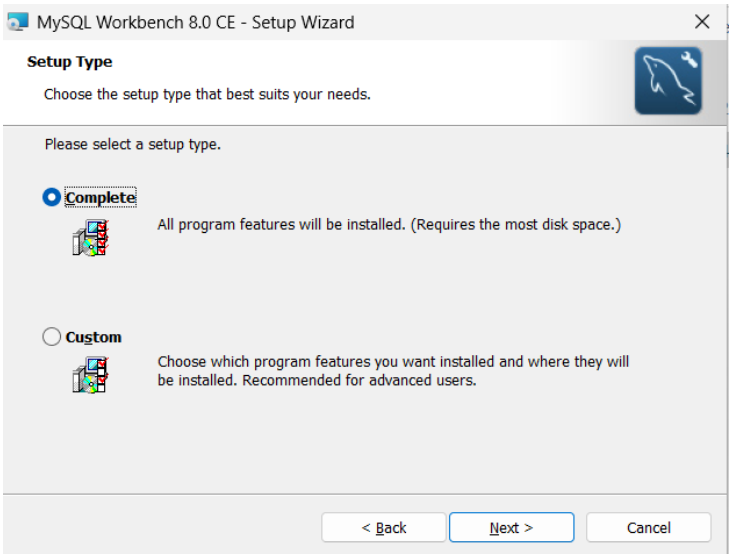


Figure 5.2.3: Choose complete installation

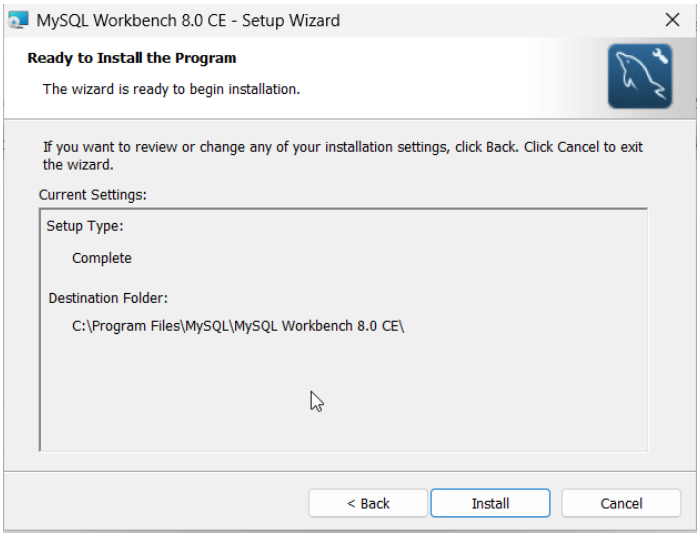


Figure 5.2.4: Review settings before installation

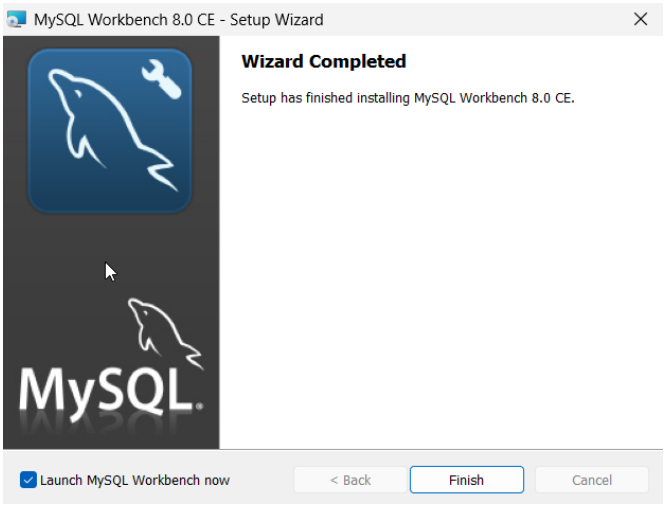


Figure 5.2.5: Completed installation

Telegram

Download and launch Telegram from Google Play. Then, log in Telegram through telephone number. Then, search and enter “BotFather”. In the chat room, press “Start”

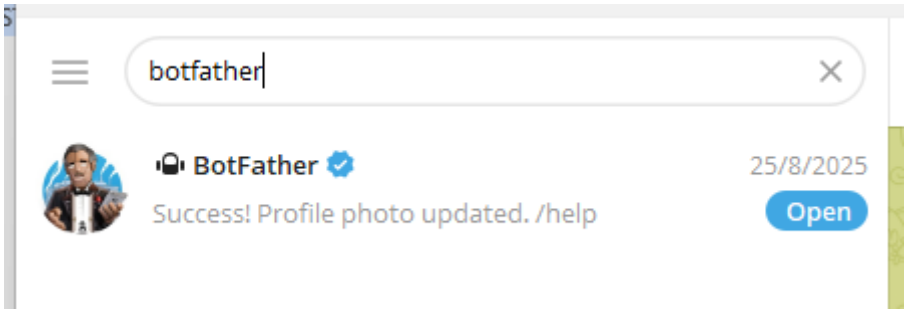


Figure 5.2.6: Search BotFather

After that, create a new bot through sending "/newbot" and intended name to this channel. If the bot is created successfully, the token of the new bot will be given. The token will be used to integrate Telegram bot with Node-RED.

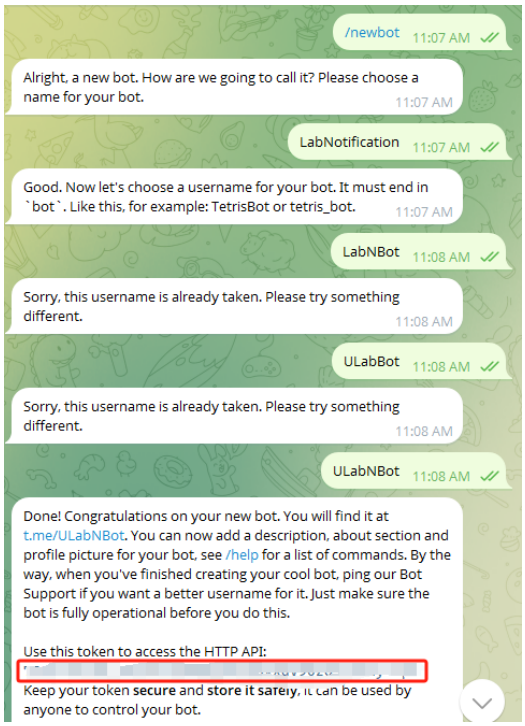


Figure 5.2.7: Create new Telegram bot

5.3 Setting and Configuration

Node-Red

Firstly, create a new file named 'public' in '..\.node-red\' directory and put a picture into this file which will be used for background of home page.

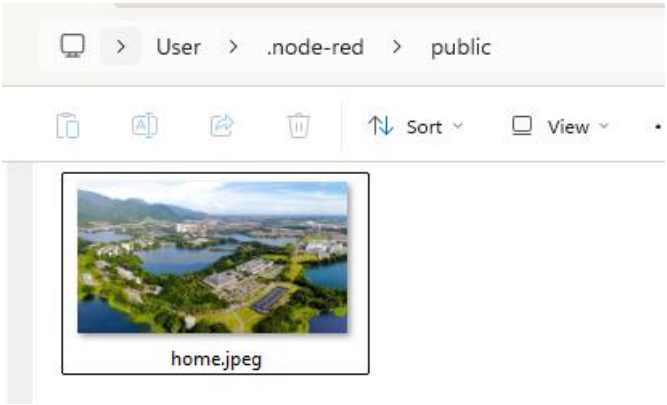


Figure 5.3.1: Path for home page background image

Then, go to the settings file (settings.js) of Node-Red whose the path location is shown at the command prompt when launching Node-Red. This file can set the server's IP address, port number, dashboard authentication, global context, and so on.

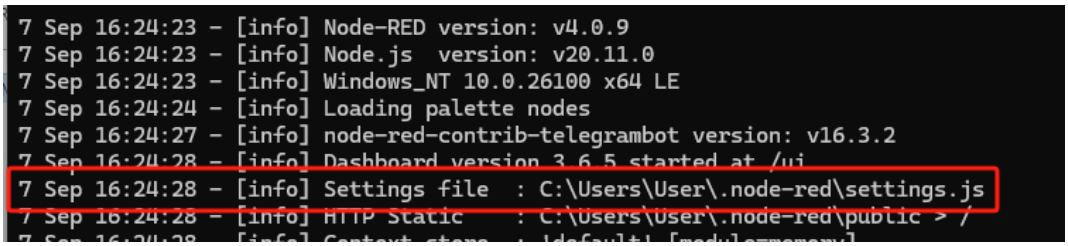


Figure 5.3.2: Path for setting.js

In setting.js, there are 5 elements needs to be configured:

httpStatic:

Enter the path where contains the home's background picture. This configuration allows Node-red to retrieve local image easily within CSS script.

```
httpStatic: 'C:\\Users\\User\\.node-red\\public\\',
```

Figure 5.3.3: Configuration on httpStatic

```
.v-main {
    background-image: url('/home.jpeg') !important;
```

Figure 5.3.4: CSS script for getting image for home page background

uiHost:

Set IP address (192.168.1.4) of the system server.

```
 */
//uiHost: "127.0.0.1",
//uiHost: "172.16.5.181",
uiHost: "192.168.1.4",
//uiHost: "100.122.114.42",
```

Figure 5.3.5: Set server IP address

uiPort:

Configure the port number to 20001 to prevent any collision with other servers.

```
/** the tcp port that the Node-RED web server is listening on */
uiPort: process.env.PORT || 20001,
```

Figure 5.3.6: Set server port number

functionGlobalContext:

Set require("crypto") into "myhash" global context to allow Node-Red to utilize the build-in Node.js module, cryptographic functionality for hashing, encrypting, and decrypting the data. On the other hand, set the environment variable, process.env.AES_KEY into "AES_KEY" global context which is used as a key for encryption and decryption of AES.

```
 */
functionGlobalContext: {
    myhash:require("crypto"),
    AES_KEY:process.env.AES_KEY,
},
```

Figure 5.3.7: Configuration of global contexts

adminAuth:

Before configuring this section in setting.js, the hashed password must be generated first through executing "node-red admin hash-pw" and enter intended password in command prompt.

```
C:\Windows\System32>node-red admin hash-pw
Password: $2y$08$kcIS8UiMYBjT9Fnq4z8RoO64FZUot1tB2PtpUqrkrbTvQ.NBmV1W2

C:\Windows\System32>
```

Figure 5.3.8: Generate hashed password

Then, paste this generated bycrypt hash into this section as password. At the same time, configure username and permissions. Assign all permissions to the user by "*".

```
adminAuth: {
type: "credentials",
users: [{
    username: "admin",
    password: "$2y$08$kcIS8UiMYBjT9Fnq4z8RoO64FZUot1tB2PtpUqrkrbTvQ.NBmV1W2",
    permissions: "*"
    }]
},
```

Figure 5.3.9: Configuration of account in adminAuth

After configuring all necessary settings, save the setting.js and restart Node-RED to take effect of the configurations. Then, install necessary palettes by using Node-RED editor.

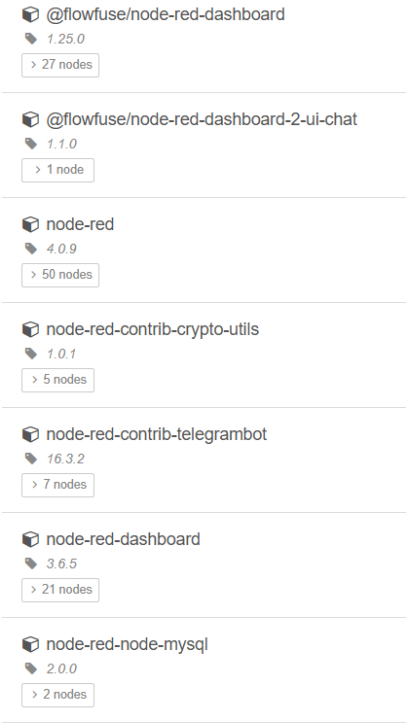


Figure 5.3.10: List of installed palettes in Node-RED

Table 5.3.1: Functions of installed palettes

Palettes	**Functions**
@flowfuse/node-red-dashboard	The main palette used to build user interfaces and handling backend codes
@flowfuse/node-red-dashboard-2-ui-chat	Provide interface to allow users to send comments on a specific item
node-red	Provide built-in nodes to support various development processes such as debugging, trigger button for automated tasks, and so on.
node-red-contrib-crypto-utils	Provide AES, RSA, and sha256 nodes to simplify the encryption and decryption processes
node-red-contrib-telegrambot	Integrate Telegram bot with Node-Red through using the token of specified bot. Support the automated notification messages to the users.

node-red-dashboard	Unremovable default palette which is installed together with the installation of Node-red. Don't have any utilization in this project.
node-red-node-mysql	**Integrate MySQL server into Node-Red through** host IP address, port number, username, password, and database name.

MySQL Workbench

Click the add button in main page to add and configure the connection name, connection method, host IP address, port number, username, and password. After configuring the connection, test the connection to ensure that the configuration is valid.

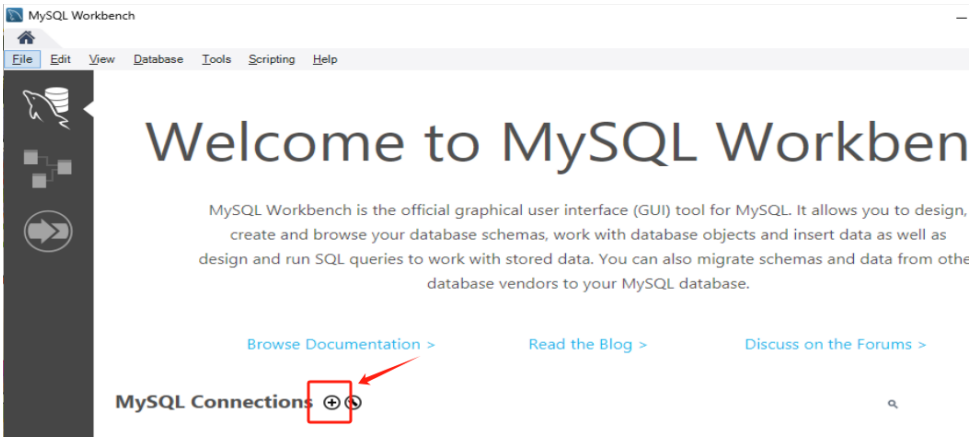


Figure 5.3.11: Main page of MySQL Workbench

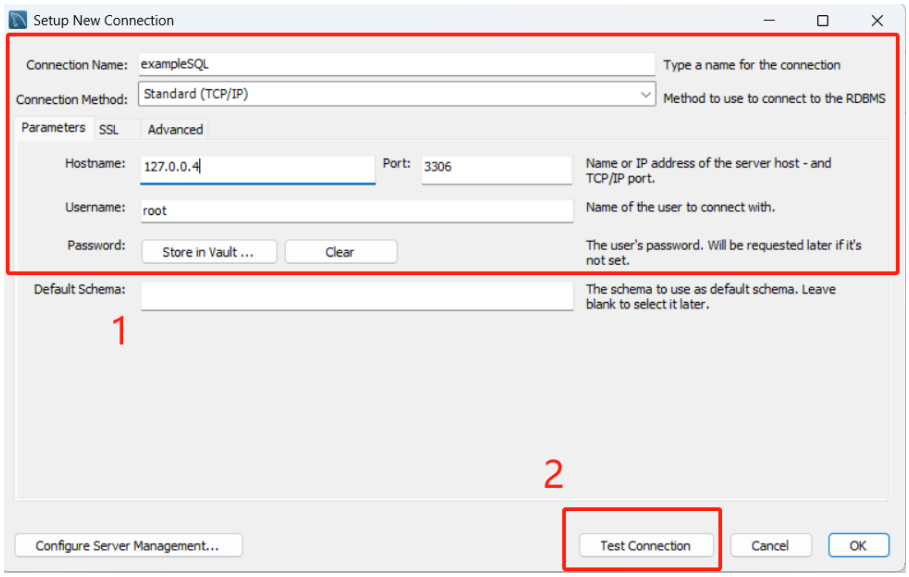


Figure 5.3.12: Configuration of connection

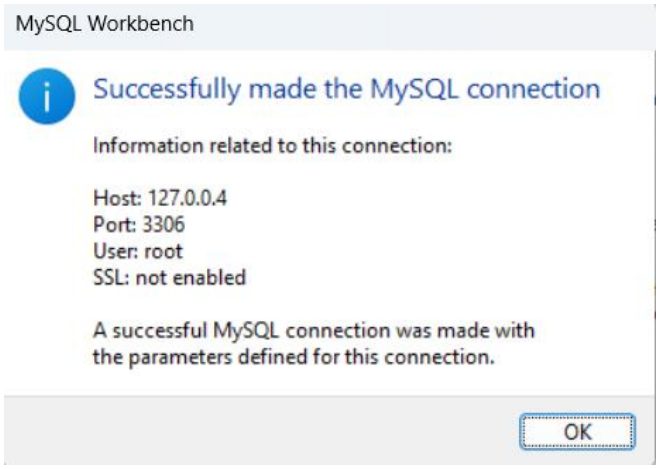


Figure 5.3.13: Test connection successfully

After ensuring the connection is valid, enter the connection using the registered username and password. Choose the 'Server' from the tool bar and select data import option.

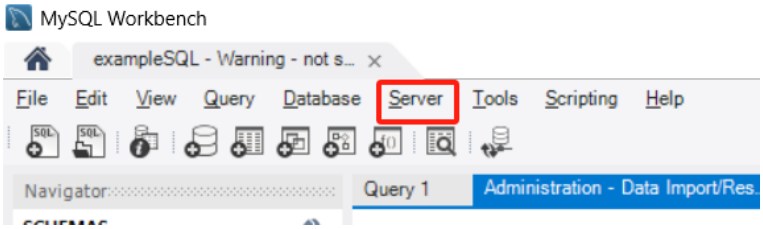


Figure 5.3.14: Tool bar

Import the data using the prepared SQL file that contains structured tables and data into the new target schema (database). Before starting data import, set the data import method as 'Dump Structure and Data' to ensure that all the tables, constraints, and data would be imported to the selected database at once.

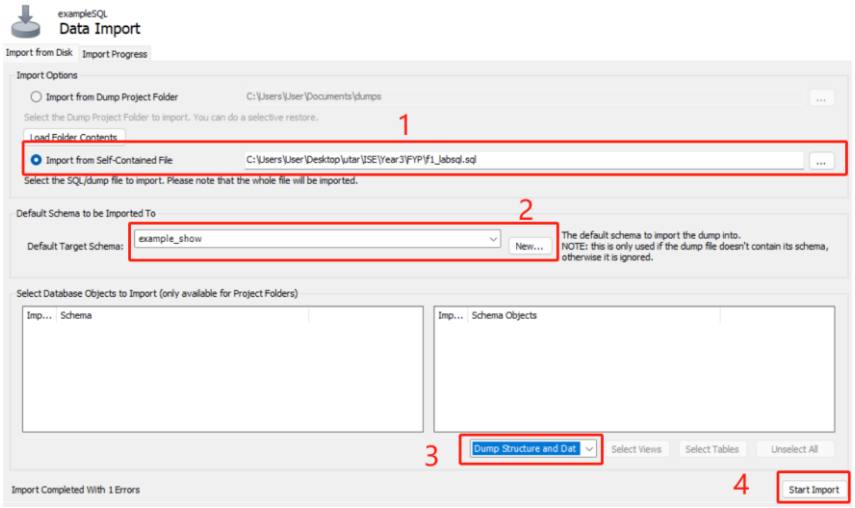


Figure 5.3.15: Configuration of data import

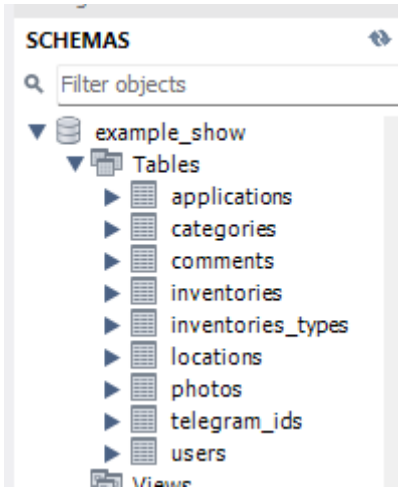


Figure 5.3.16: Imported tables and data

Lastly, configure the database connection in MySQL node by using the specifications that has been configured in MySQL Workbench.

Figure 5.3.17: mysql node in Node-RED

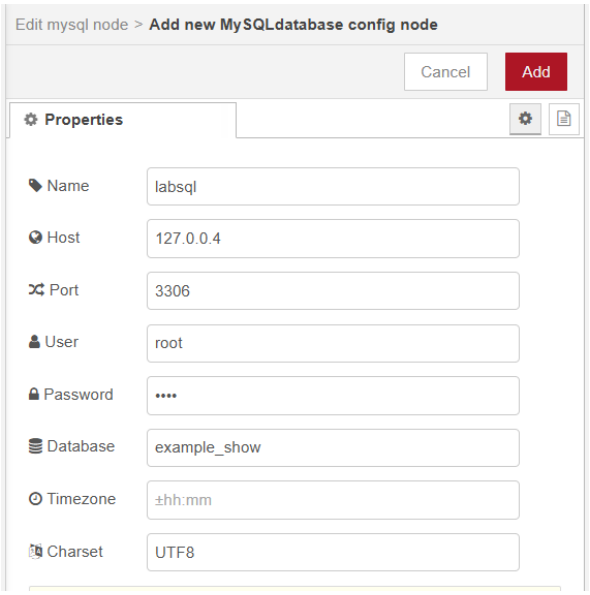


Figure 5.3.18: Configuration on mysql node

Telegram Bot

Configure the bot name and profile picture through commands '/setname' and '/setuserpic'.

Figure 5.3.19: Update bot name

Figure 5.3.20: Update bot's profile picture

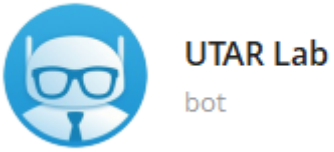


Figure 5.3.21: Updated bot's name and profile picture

5.4 System Operation

5.4.1 Log in and navigations

This section presents the error handling in log in page, the scene after login successfully, and the navigations that is distributed dynamically for different roles.

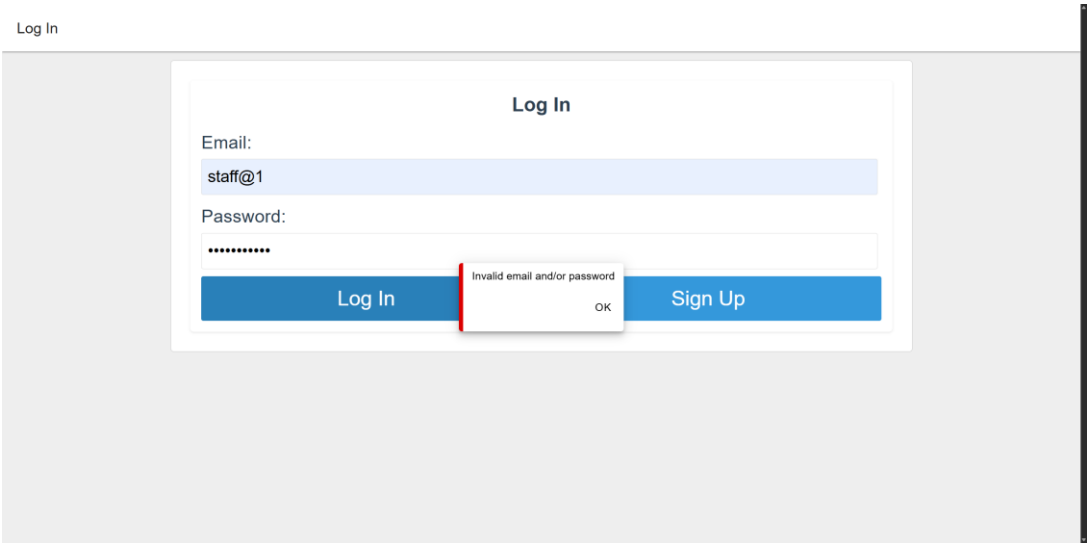


Figure 5.4.1: Log in with wrong password or inactivated account

Figure 5.4.2: Log in successfully will directly come to home page

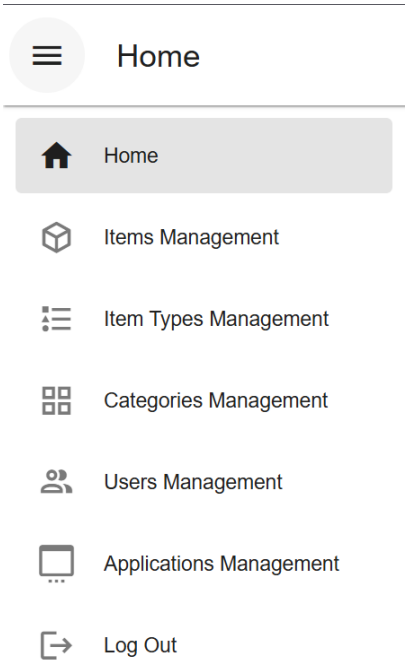


Figure 5.4.3: Menu for staff

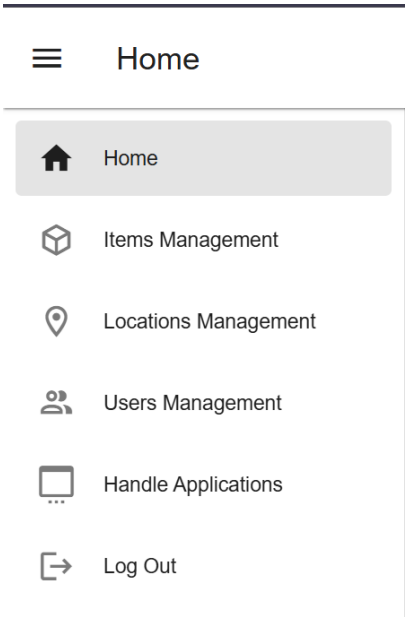


Figure 5.4.4: Menu for admin

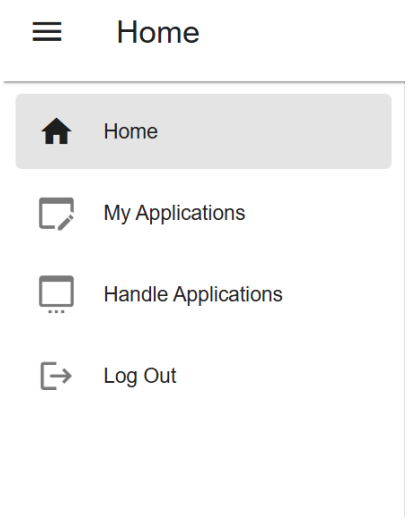


Figure 5.4.5: Menu for supervisor

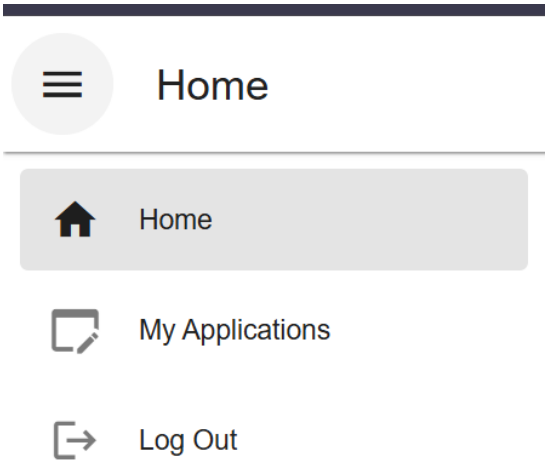


Figure 5.4.6: Menu for student

After 2 minutes the user closes the webpage, the system will clear the user session. Then, the user needs to login again when he or she visit to this system again.

```
14 Sep 17:31:28 - [info] [ui-base:My Dashboard] Disconnected tbqxtmgW5OMtdZFWAAAD due to transport close
14 Sep 17:33:28 - [warn] [function:session] Cleaning flow/global for IP: 10012211442
```

Figure 5.4.7: Clear user session automatically

5.4.2 Inventory Management Module

The system verify the registration, modification, report, and deletion of the items.

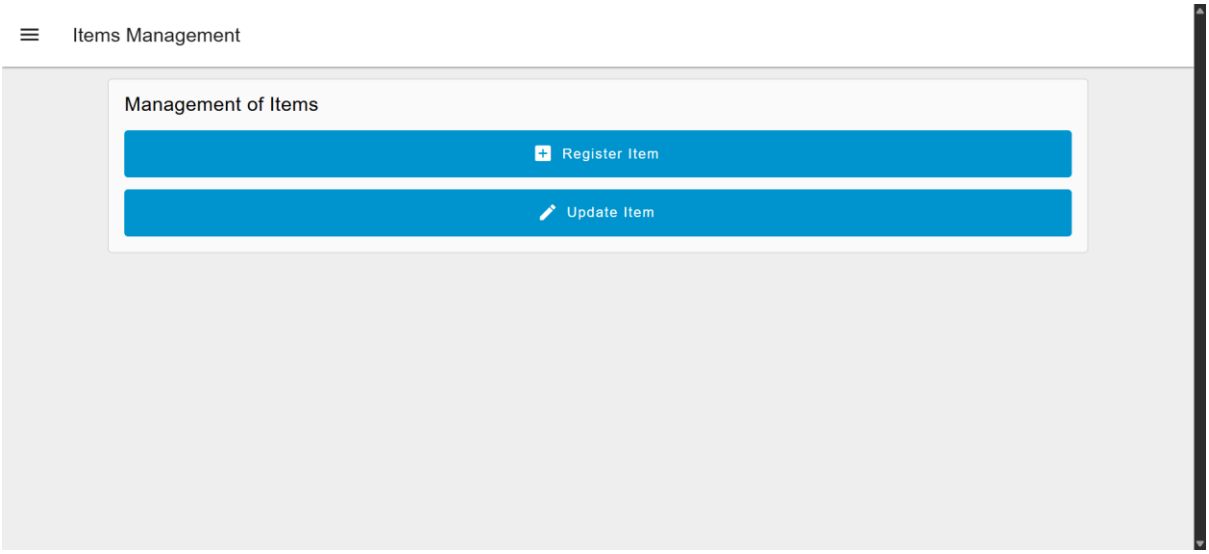


Figure 5.4.8: Main page of Items Management Module

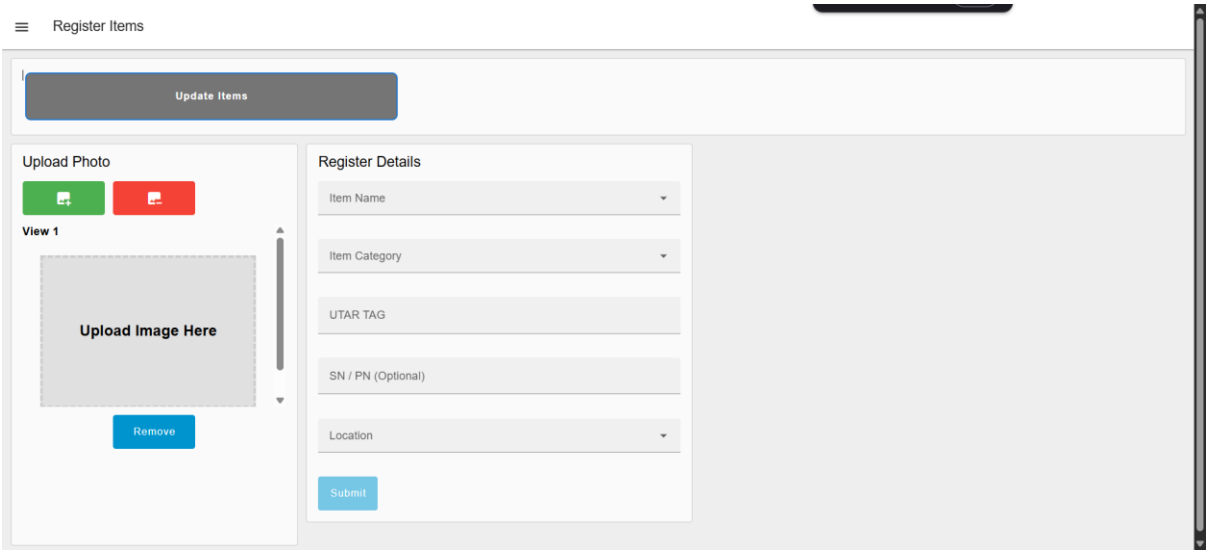


Figure 5.4.9: Interface of Register Items

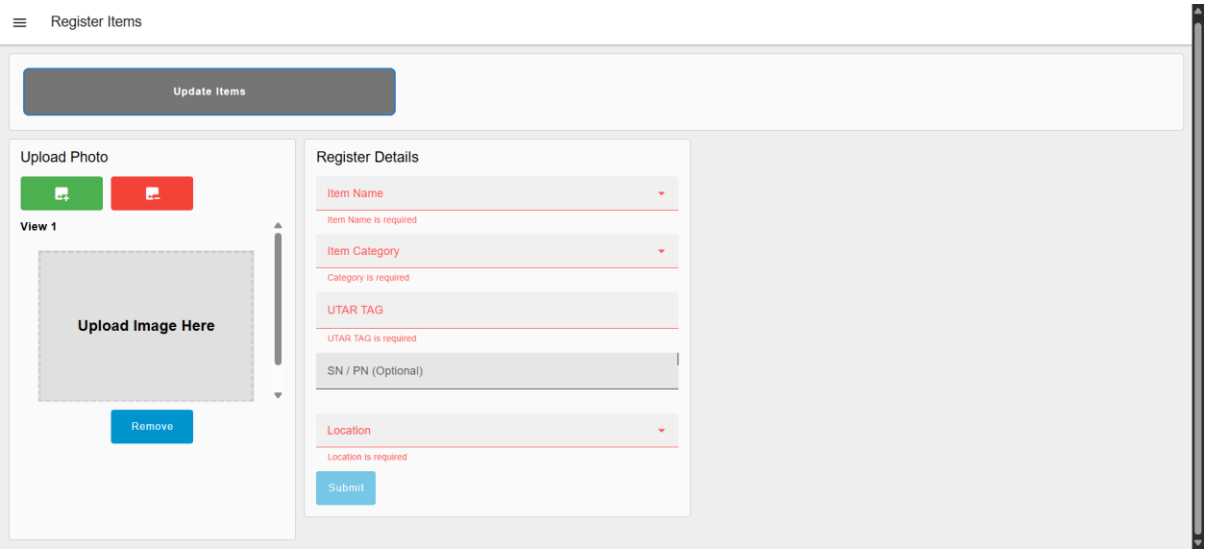


Figure 5.4.10: Necessary fields in form

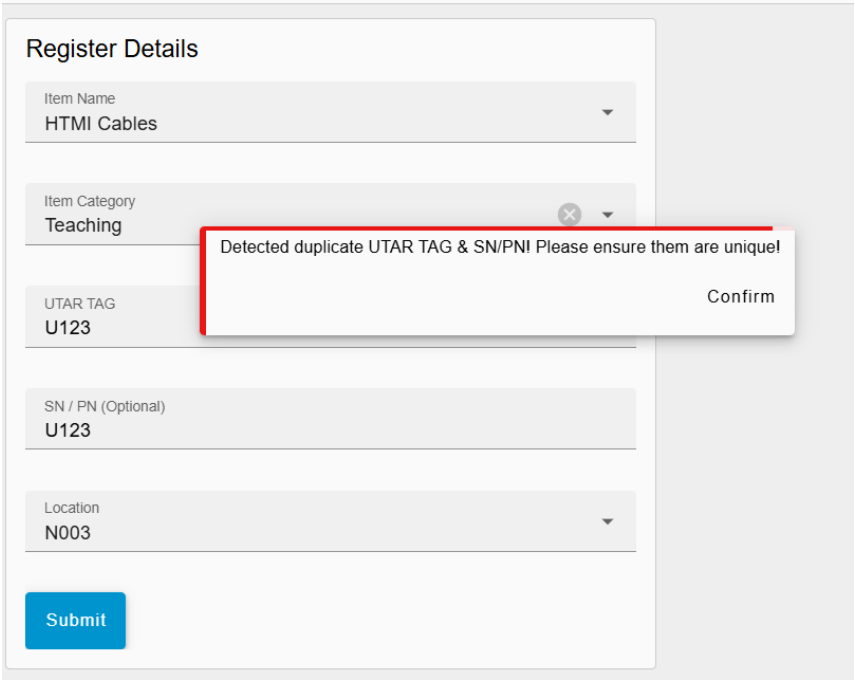


Figure 5.4.11: Register duplicated UTAR Tag & SN/PN

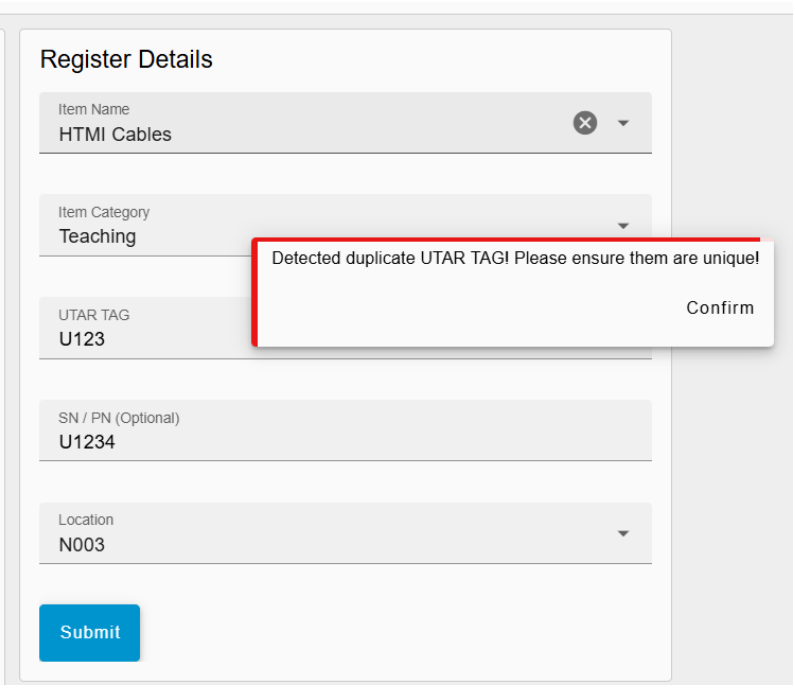


Figure 5.4.12: Register duplicated UTAR tag

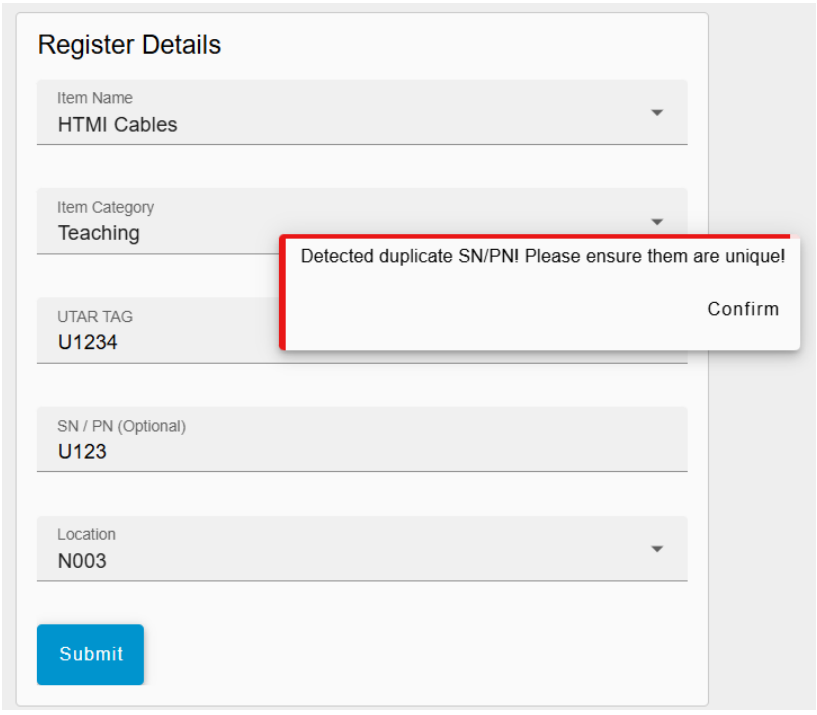


Figure 5.4.13: Register duplicated SN/PN

Users are allowed to register or update items without attach any photo.

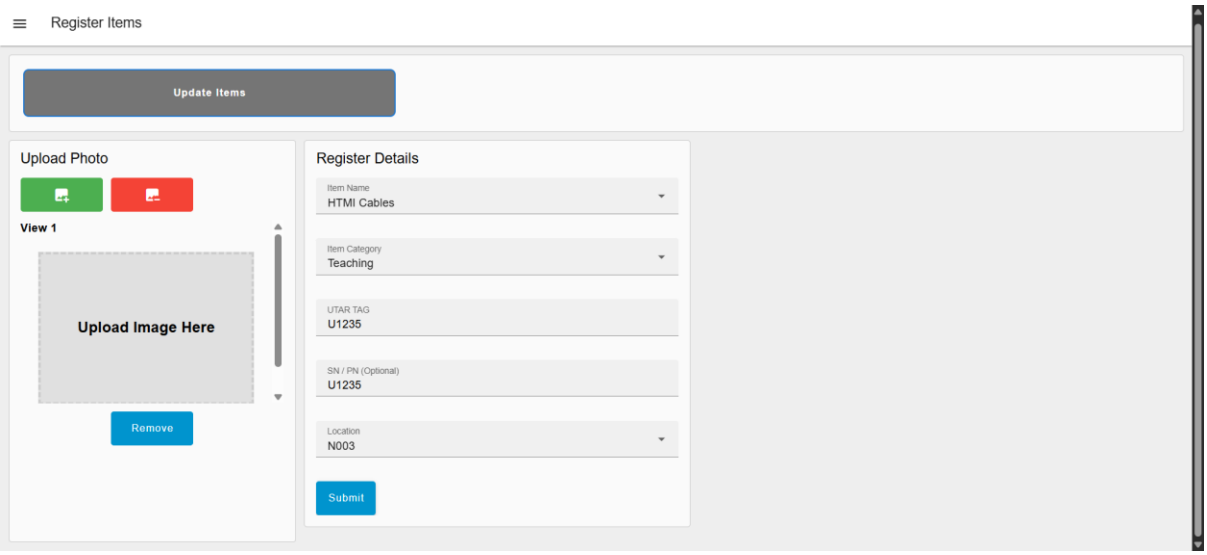


Figure 5.4.14: Register item with unique UTAR Tag, SN/PN, and without photo

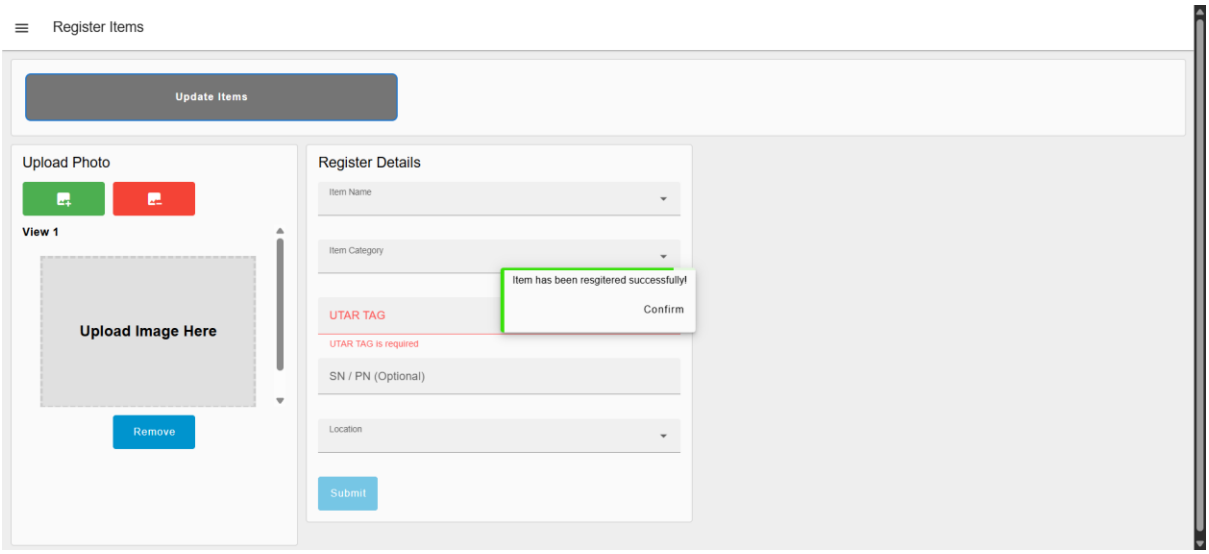


Figure 5.4.15: Result of register item with unique UTAR Tag, SN/PN, and without photo

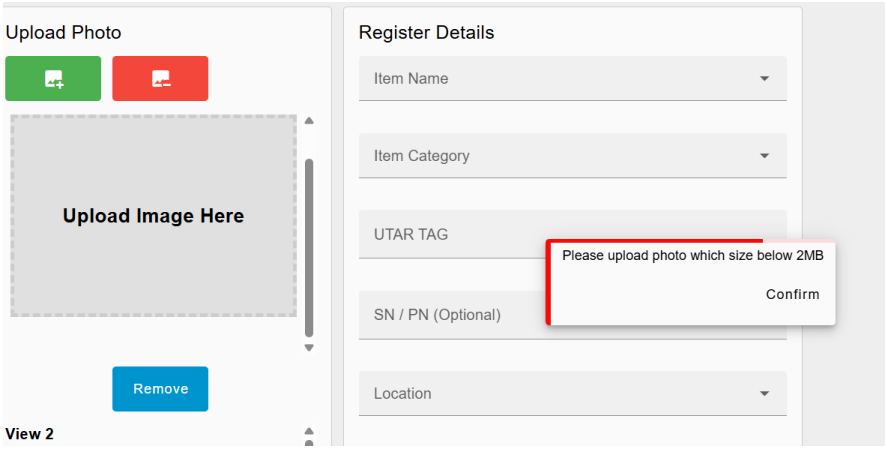


Figure 5.4.16: Upload invalid photo size

Each item can have maximum 3 photos. After add view during registering or updating the item, user is not compulsory to upload photo to the added view as the system accept the null photo. Moreover, the user no need to upload the photo in sequence during item's registration and modification. The system will sort the sequence of photos automatically in ascending order before they are stored in the database. For example, the user leaves the view 1 empty but attach photo on view 2. After the submission, the user will view that the photo will be arranged to view 1. The sample case will be shown in the modification of items.

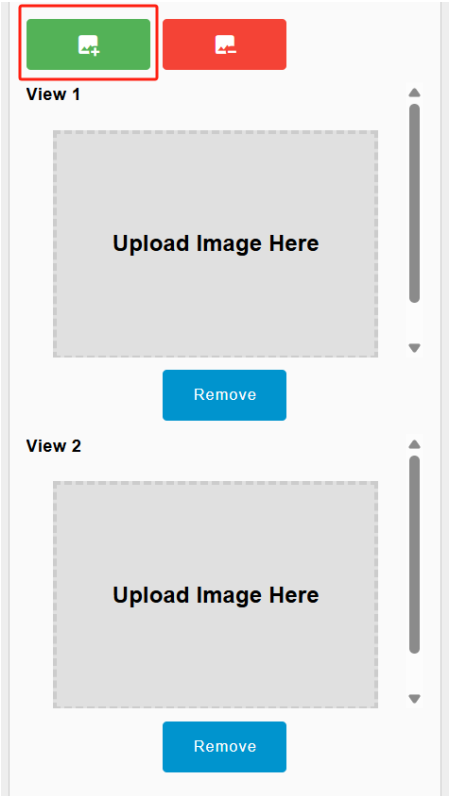


Figure 5.4.17: Press "add view" green button

Figure 5.4.18: Maximum number of views for each item

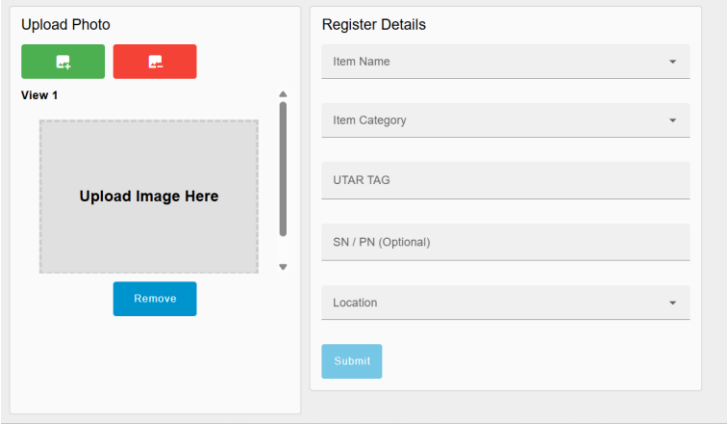


Figure 5.4.19: Press “minus view” red button twice

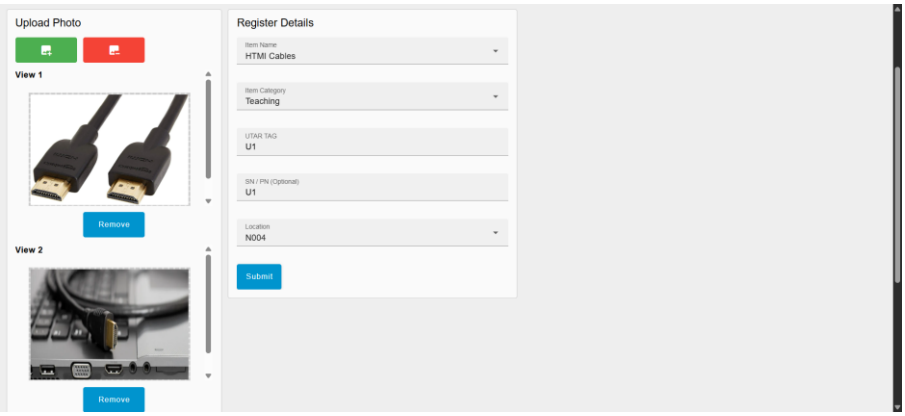


Figure 5.4.20: Register item with photos

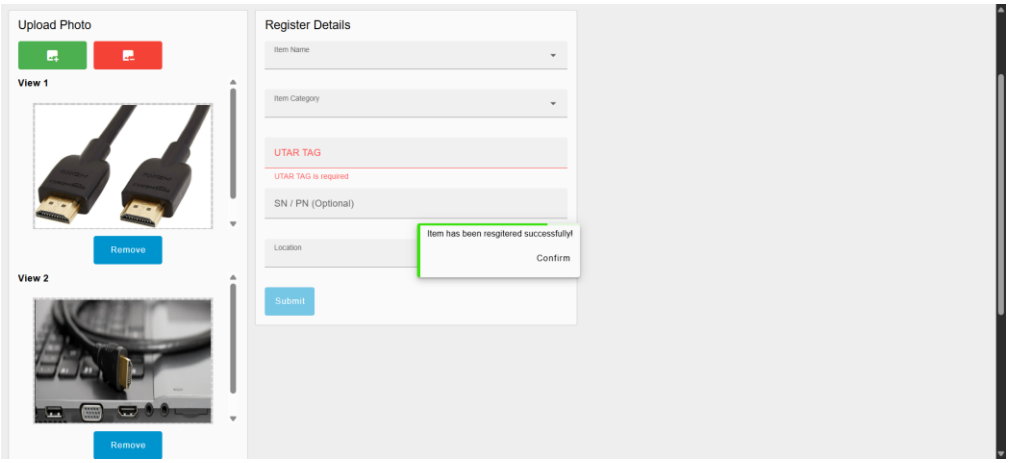


Figure 5.4.21: Result of register item with photos

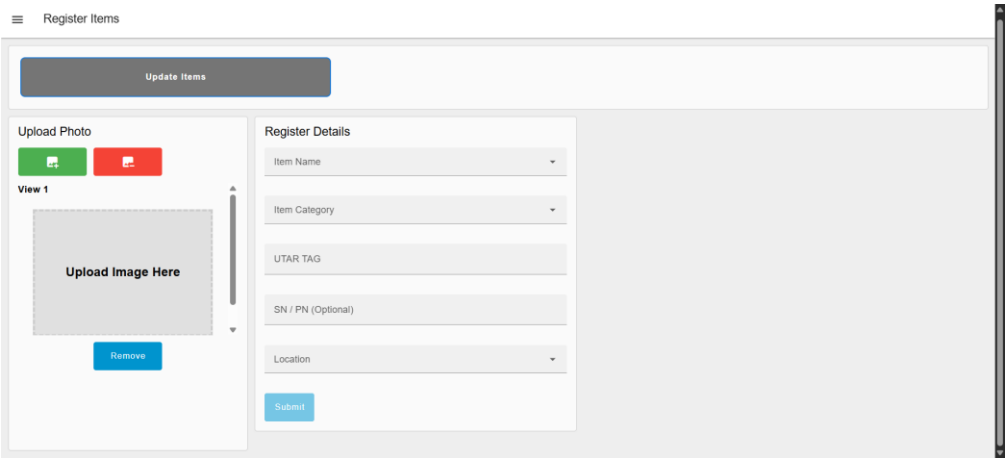


Figure 5.4.22: Refresh page after register item successfully

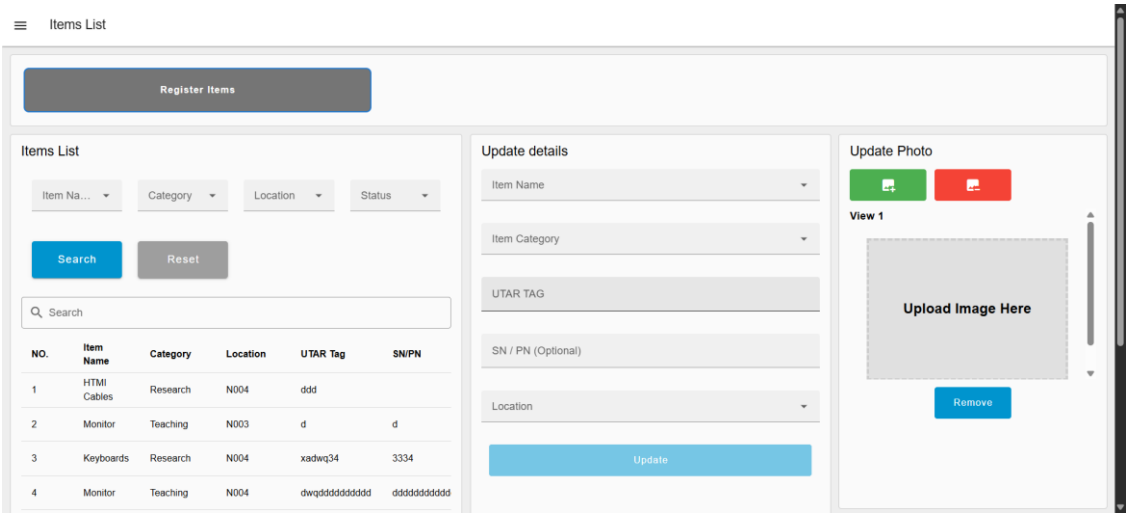


Figure 5.4.23: Interface of Update Items

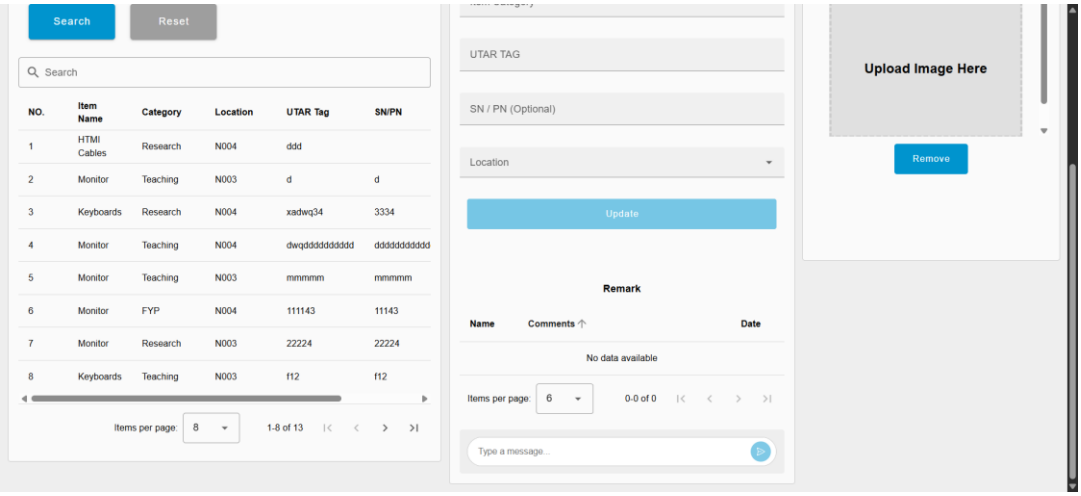


Figure 5.4.24: Remark section of Update Items

Most of the tables within this system provide the filters and searching features.

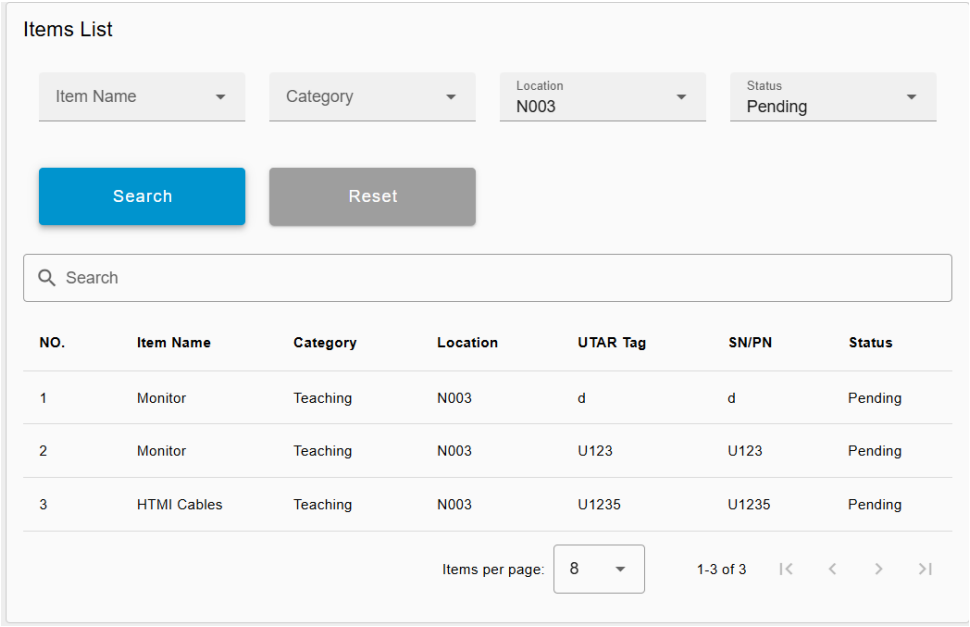


Figure 5.4.25: Filter rows with partial conditions

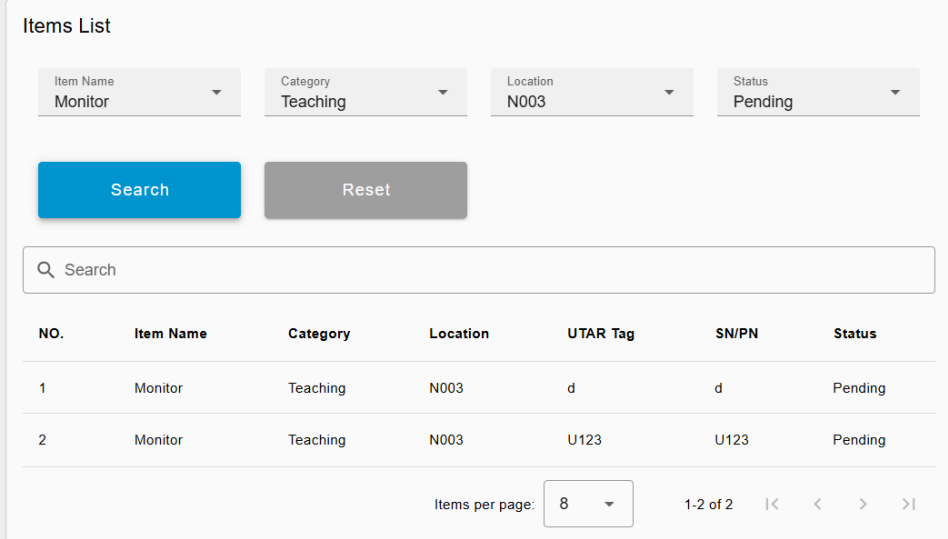


Figure 5.4.26: Filter rows with full conditions

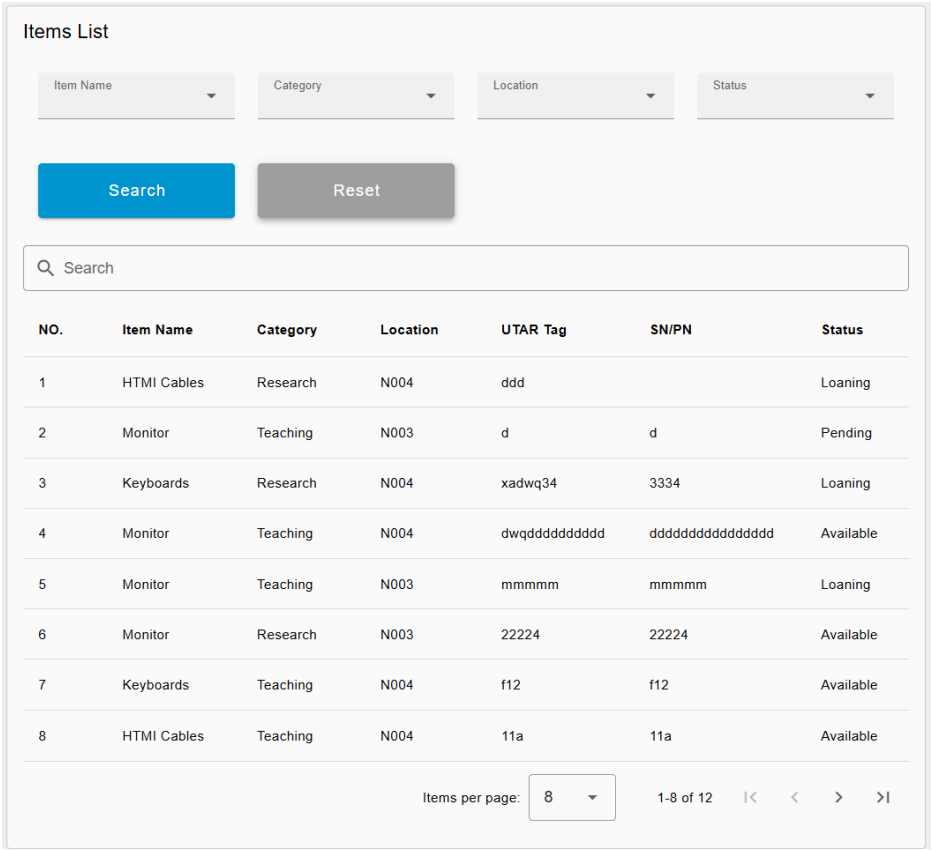


Figure 5.4.27: Press ‘Reset’ button on filter or filter rows with empty conditions

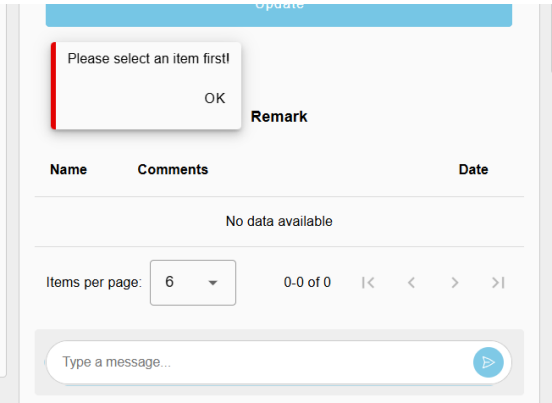


Figure 5.4.28: Send comment without selecting item

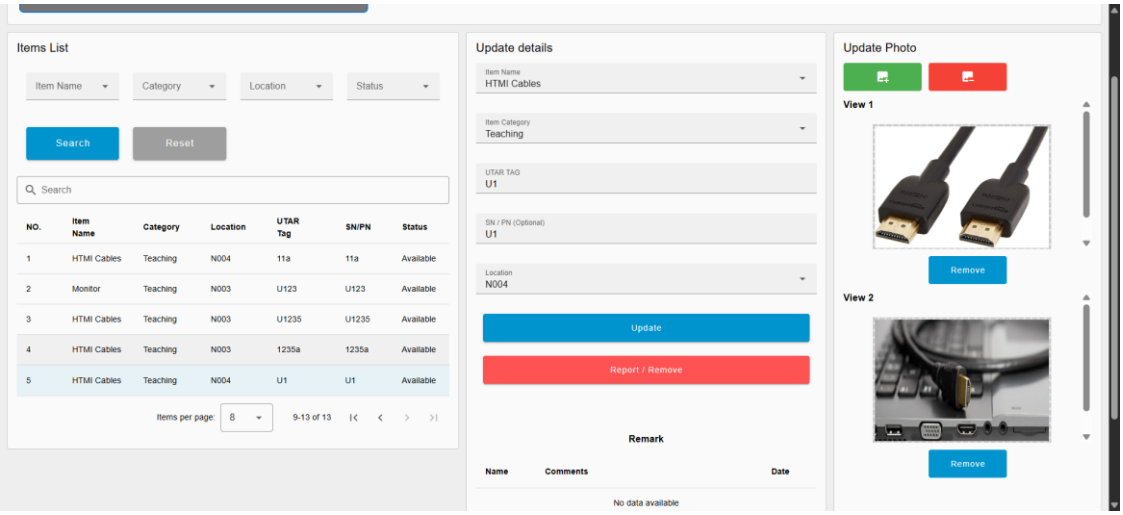


Figure 5.4.29: Select item for editing

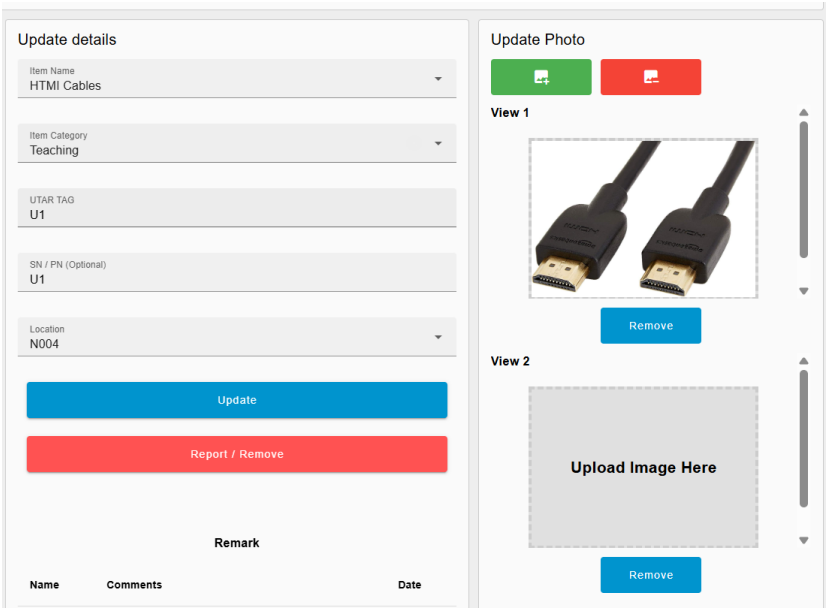


Figure 5.4.30: Remove photo

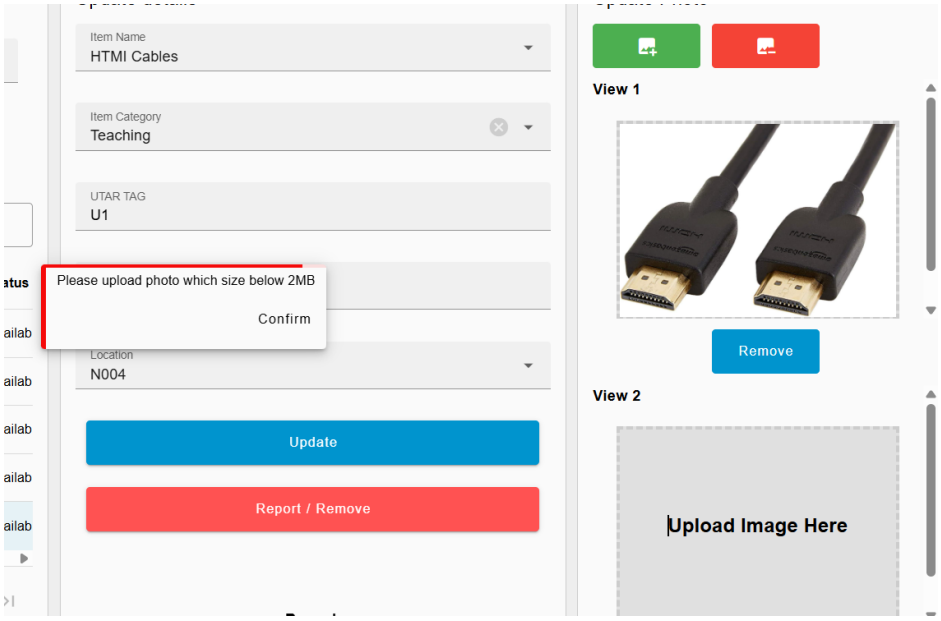


Figure 5.4.31: Update invalid photo size on View 2

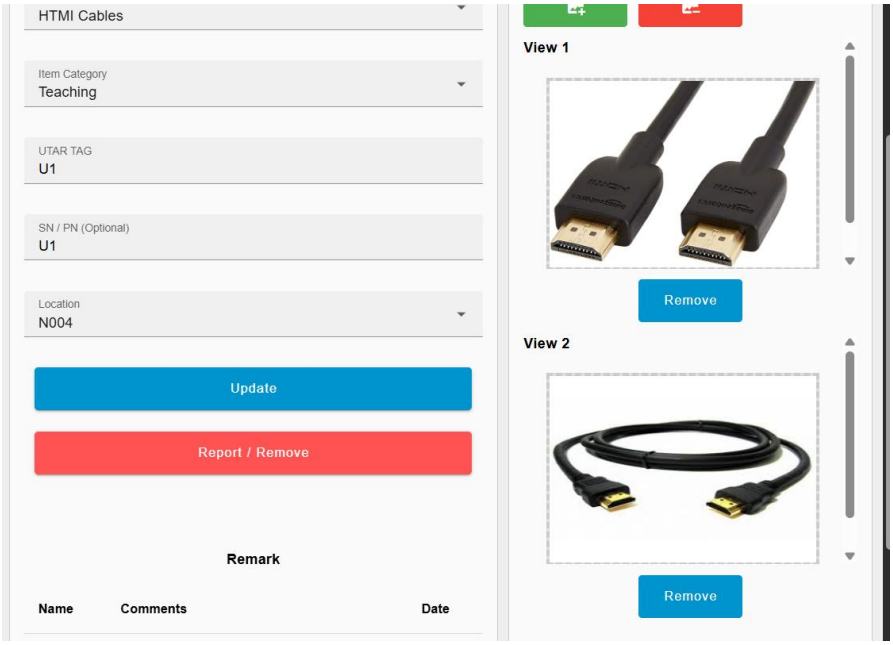


Figure 5.4.32: Update valid photo size on View 2

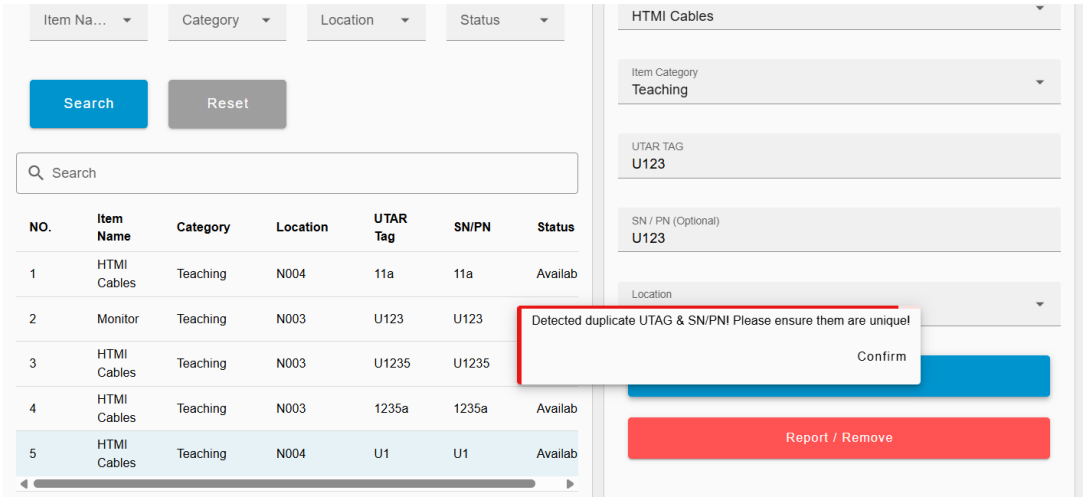


Figure 5.4.33: Update item with duplicated UTAR Tag and SN/PN

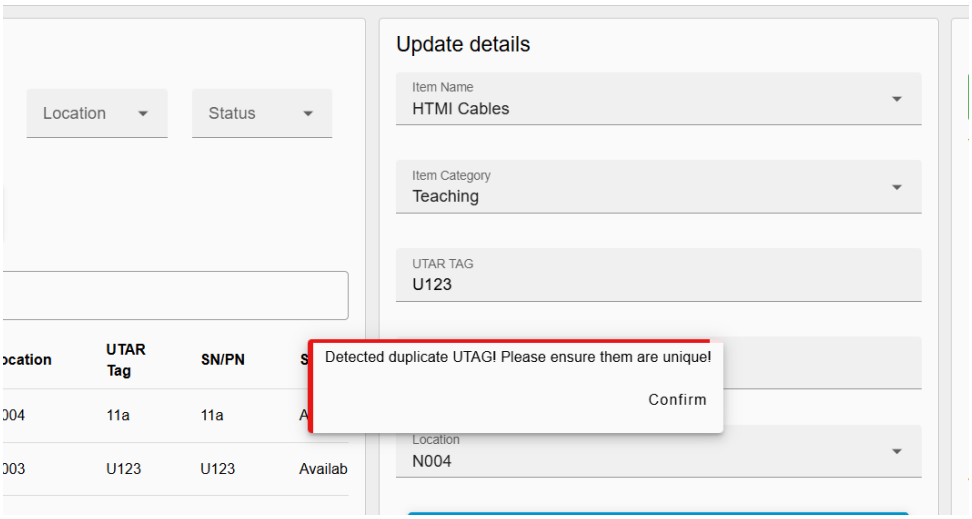


Figure 5.4.34: Update item with duplicated UTAR Tag only

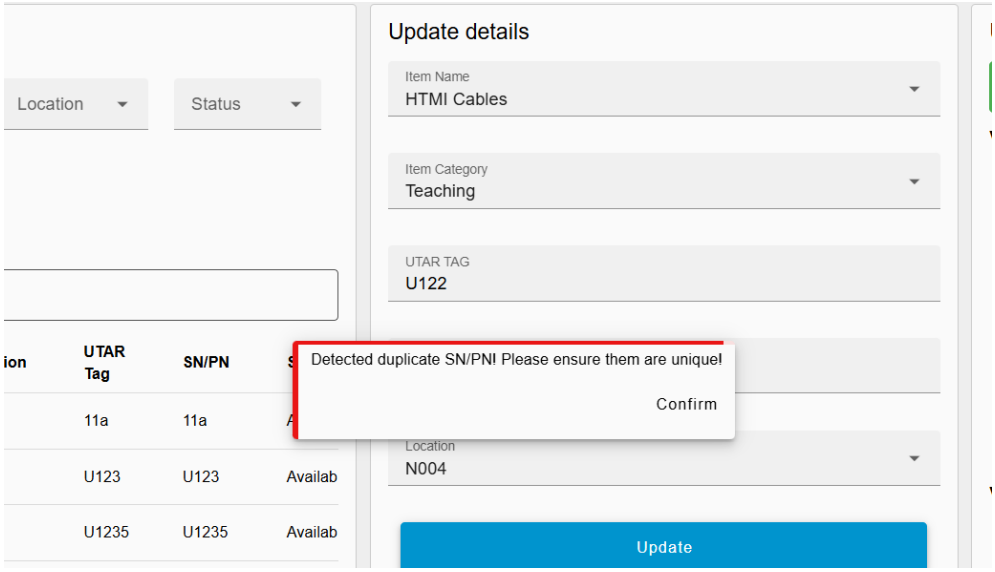


Figure 5.4.35: Update item with duplicated SN/PN only

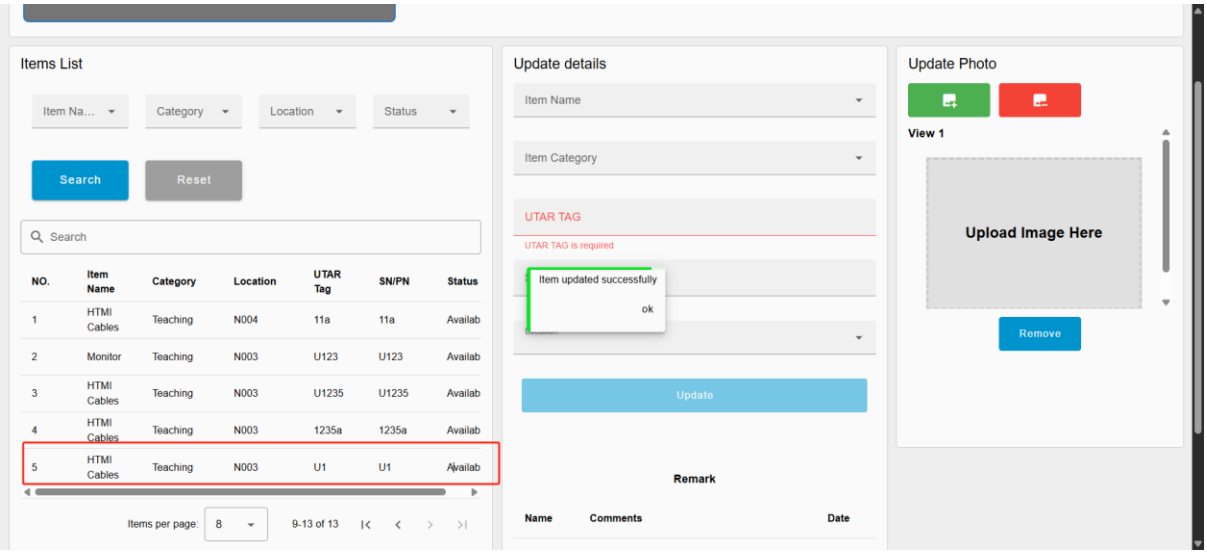


Figure 5.4.36: Update item with original UTAR Tag and SN/PN

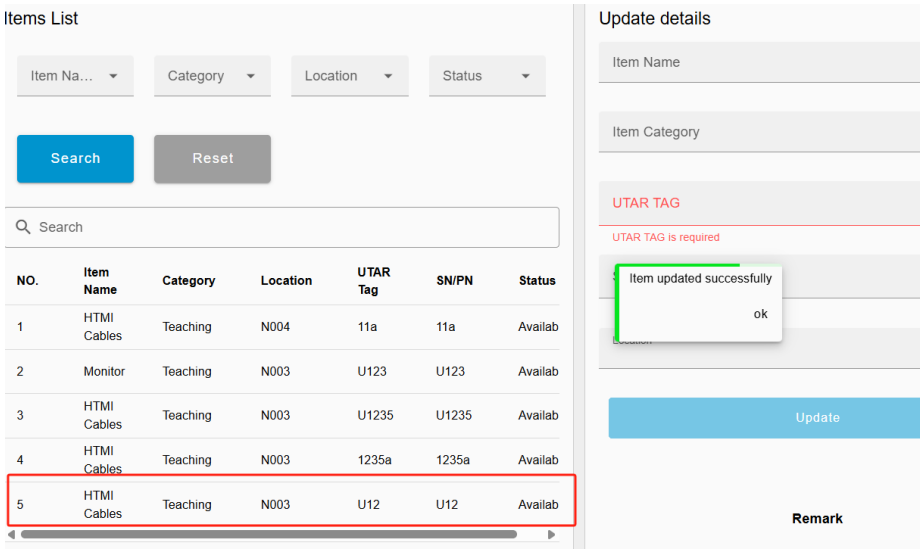


Figure 5.4.37: Update item with unique UTAR Tag and SN/PN

There is not matter about the sequence of uploaded photos.

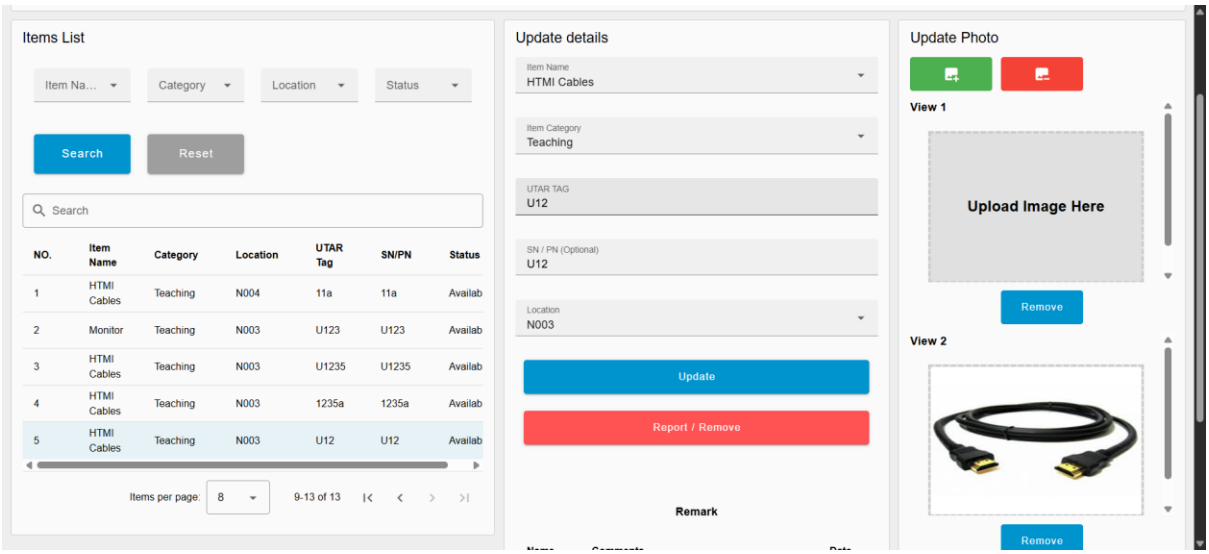


Figure 5.4.38: Update item without View 1 but with View 2

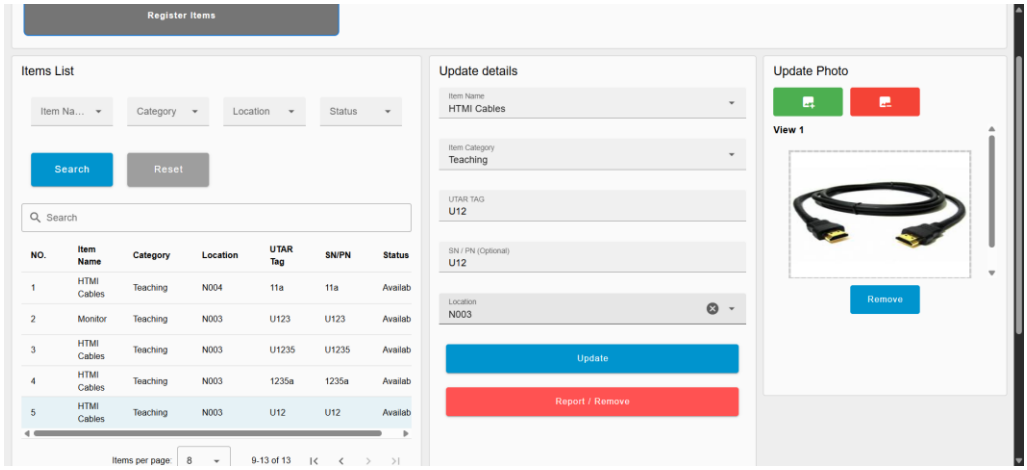


Figure 5.4.39: Result of update item without View 1 but with View 2

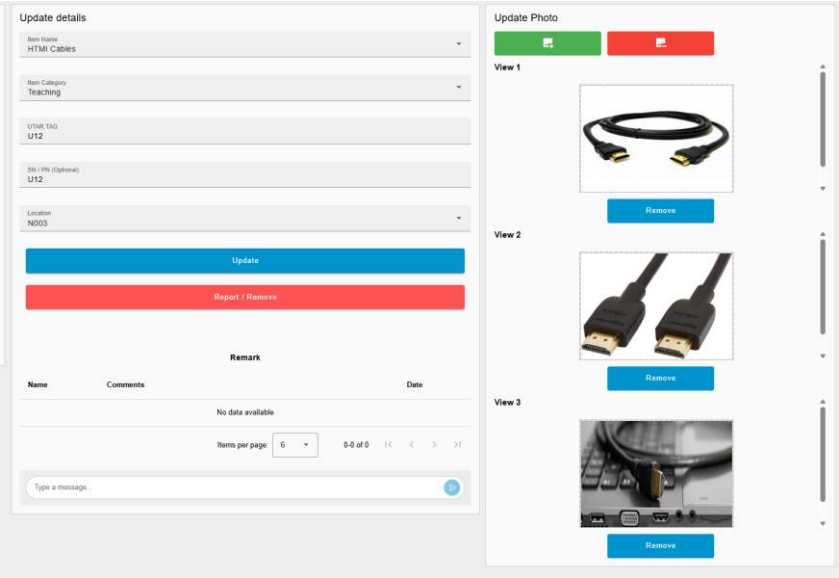


Figure 5.4.40: Result of update item with maximum number of photos

After reporting an item that is to be maintenance, the item's status and remark are updated. At the same time, all lab staff and administrator receive the notification regarding this item.

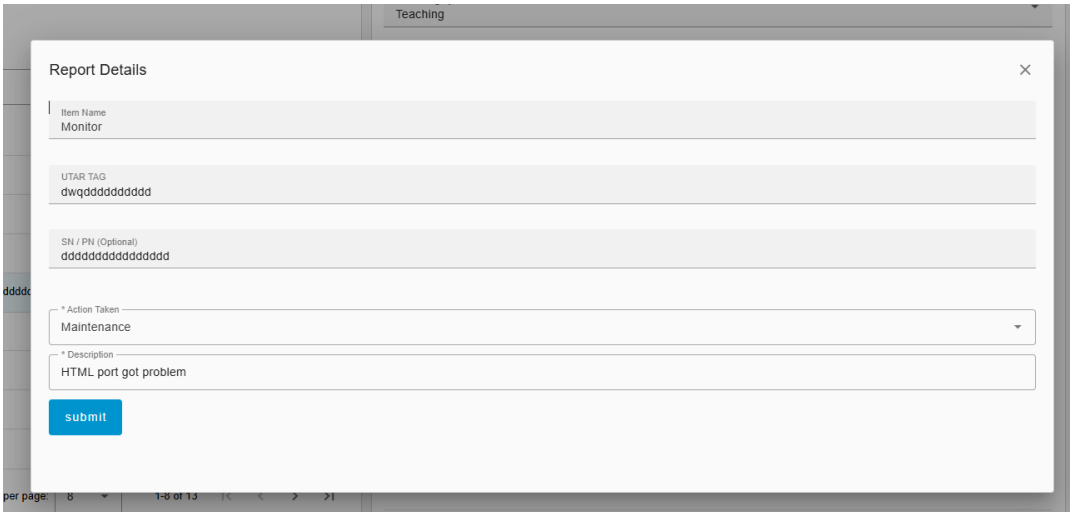


Figure 5.4.41: Select and report item (Maintenance)

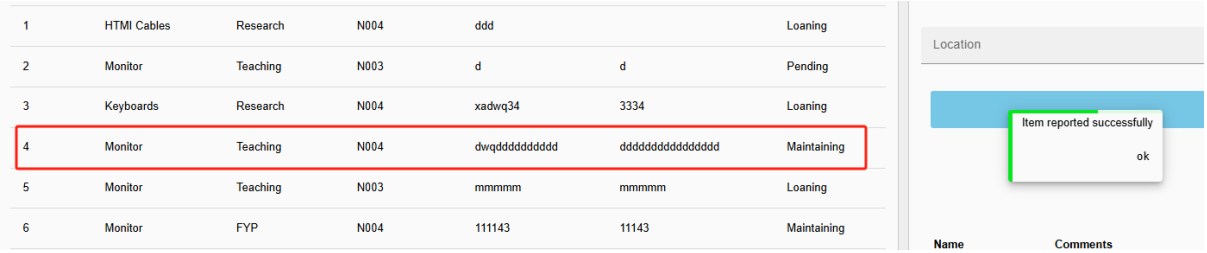


Figure 5.4.42: Submitted report

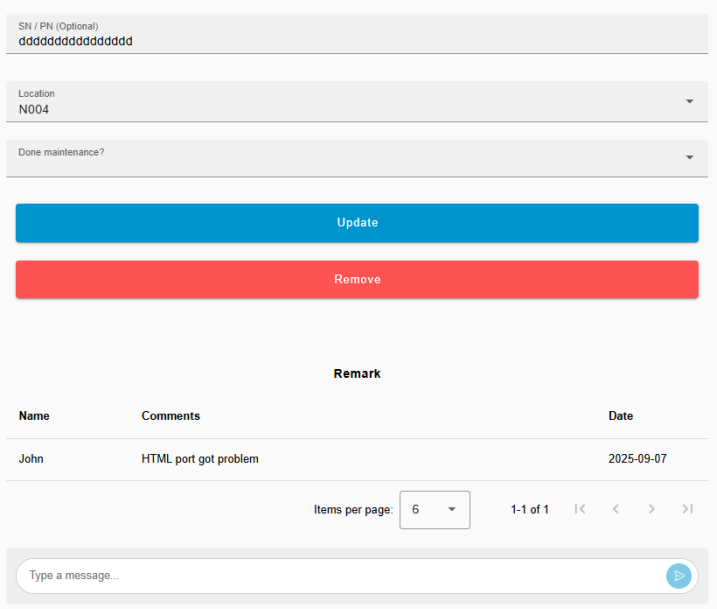


Figure 5.4.43: Updated remark after reporting

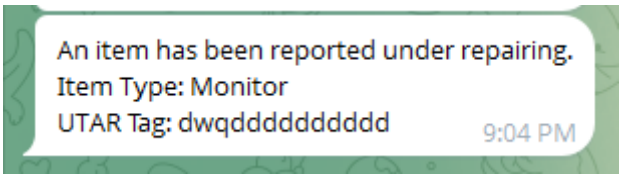


Figure 5.4.44: Notification received by all lab staff and admin (after report)

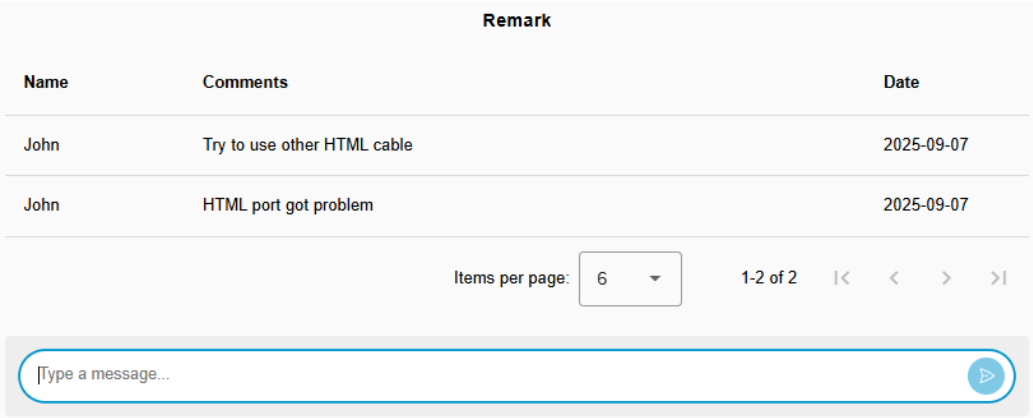


Figure 5.4.45: Send comment

There is no matter to leave the 'Done maintenance' field to be blank. The modification of item would be still successful. This is because the field is also a reminder about this item is under maintenance, and the options still include 'No' due to reduce the probability of selecting 'Yes' accidentally.

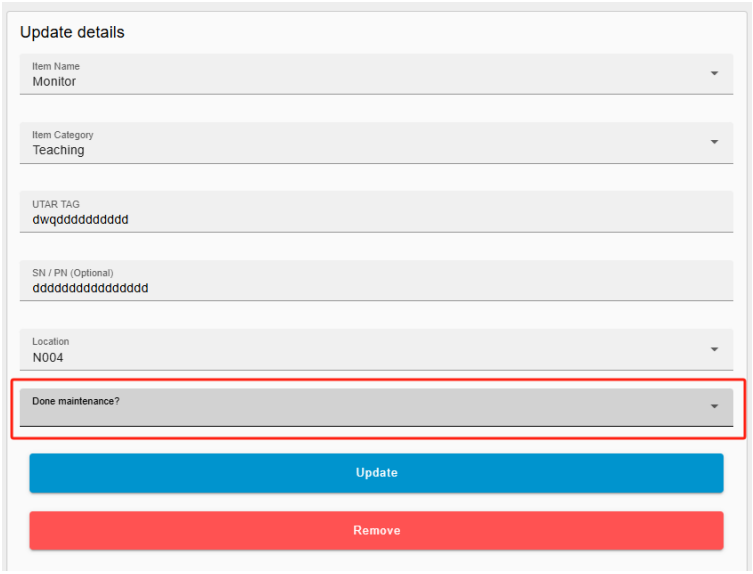


Figure 5.4.46: Select item under maintaining

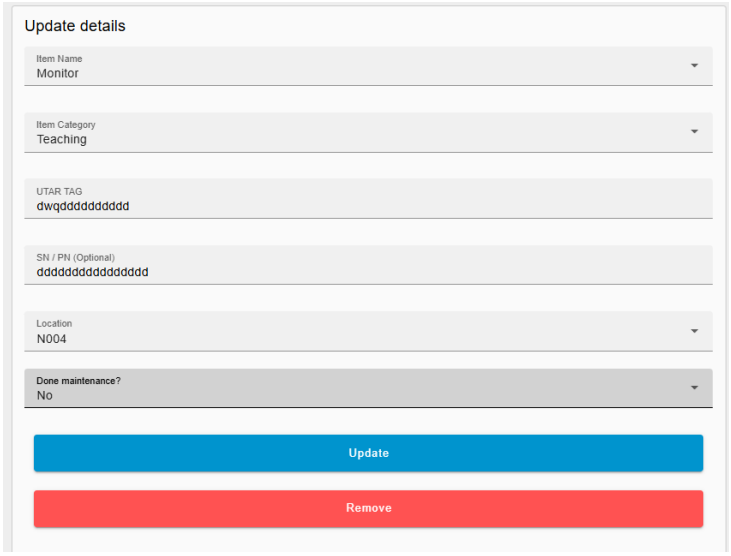


Figure 5.4.47: Update item with incomplete maintenance

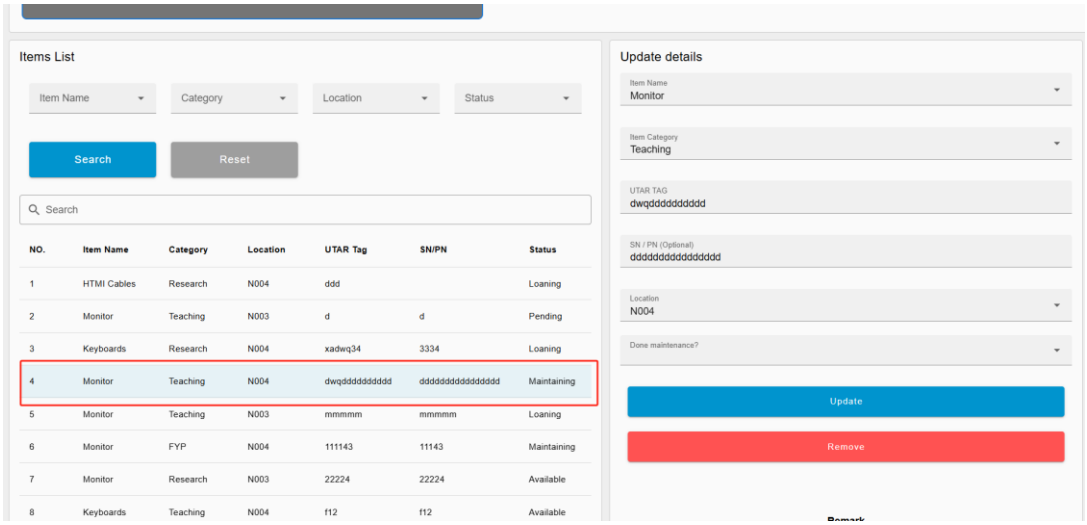


Figure 5.4.48: Result of update item with incomplete maintenance

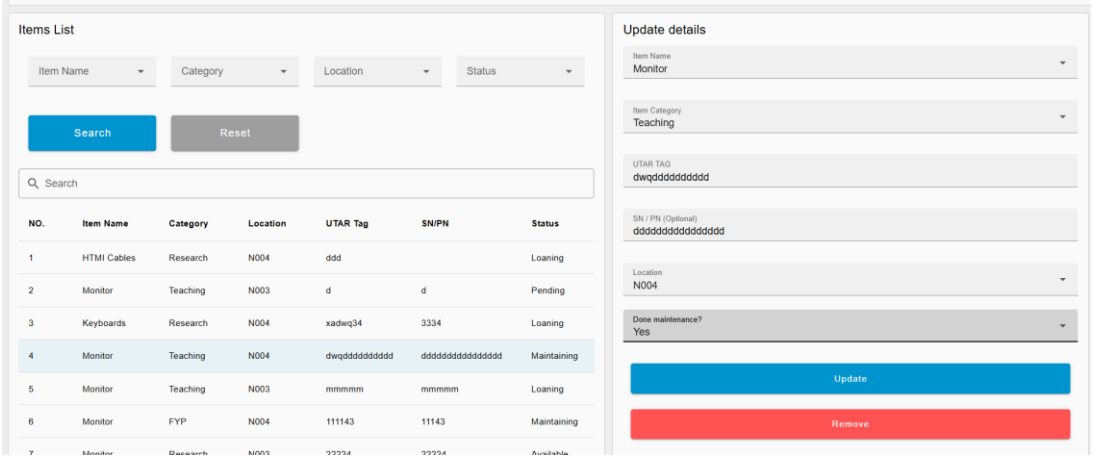


Figure 5.4.49: Update item with completed maintenance

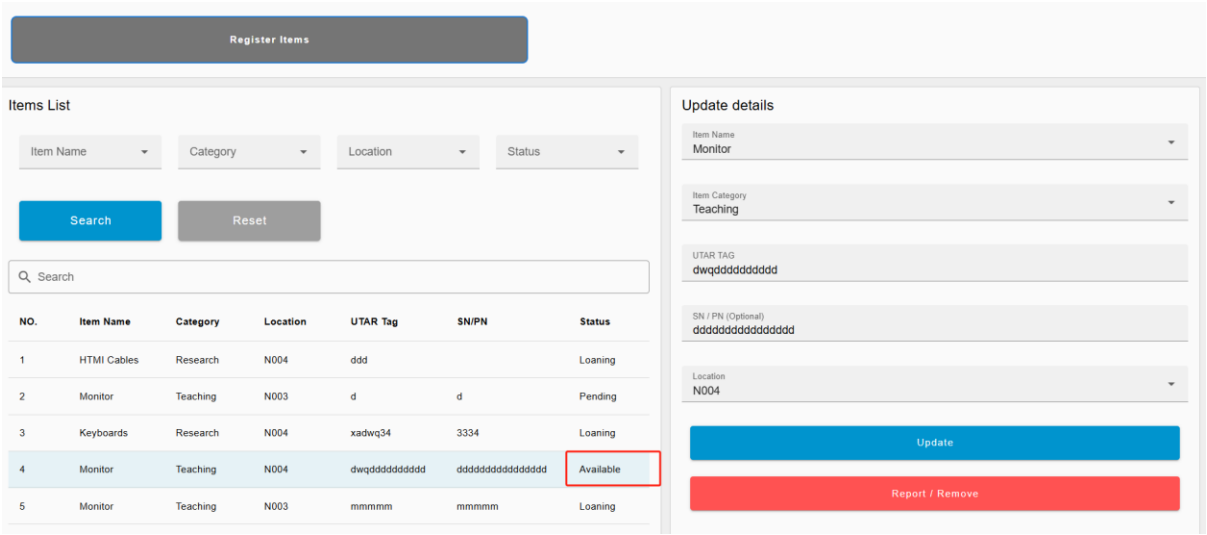


Figure 5.4.50: Result of update item with completed maintenance

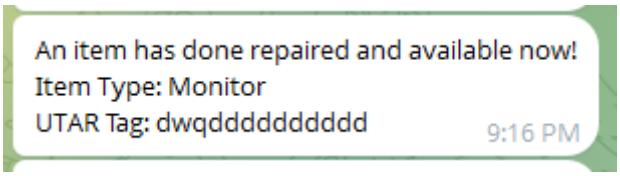


Figure 5.4.51: Notification received by all lab staff and admin (done maintenance)

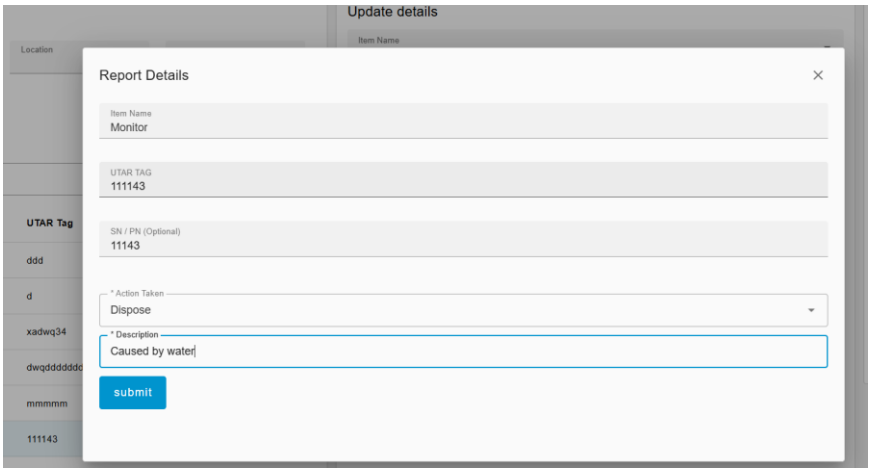


Figure 5.4.52: Report item that can't function anymore

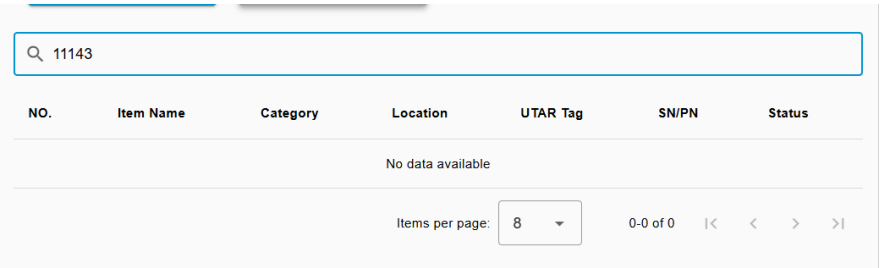


Figure 5.4.53: Result after remove item (can't be searched anymore)

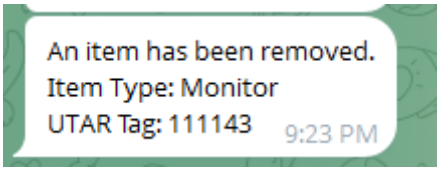


Figure 5.4.54: Notification received by all lab staff and admin (after delete)

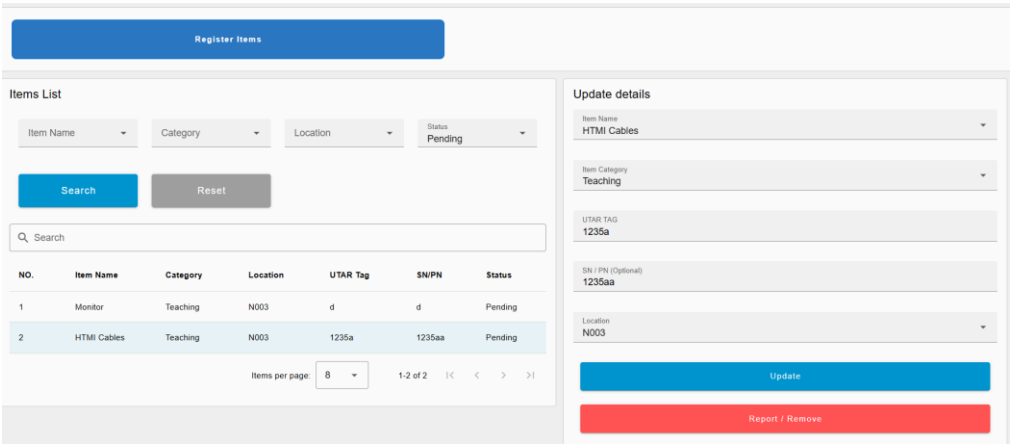


Figure 5.4.55: Report item with pending status

When reporting the pending item, the corresponding applications will be cancelled, and the requestor will receive the relevant notification. Moreover, when the user check the application status, the cancelled application's reject reason has been updated automatically with "Unavailable Item".

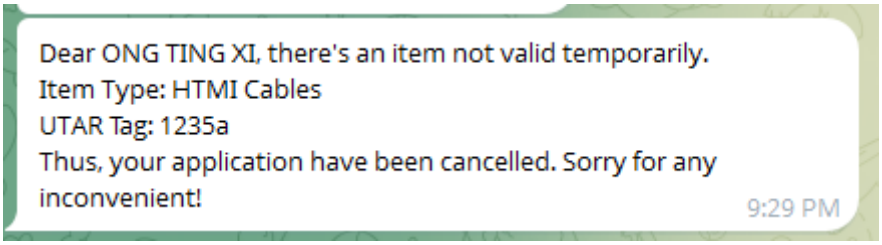


Figure 5.4.56: Notification received by the requestor (item is under maintenance)

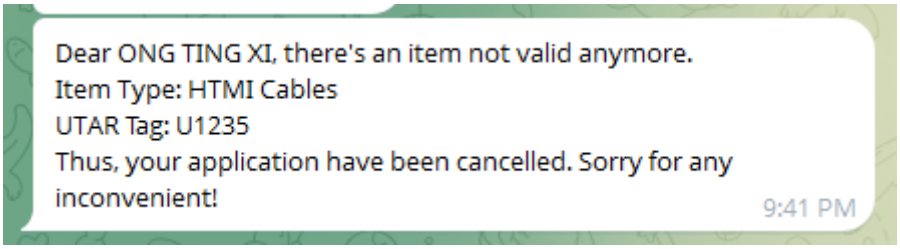


Figure 5.4.57: Notification received by requestor (item is removed)

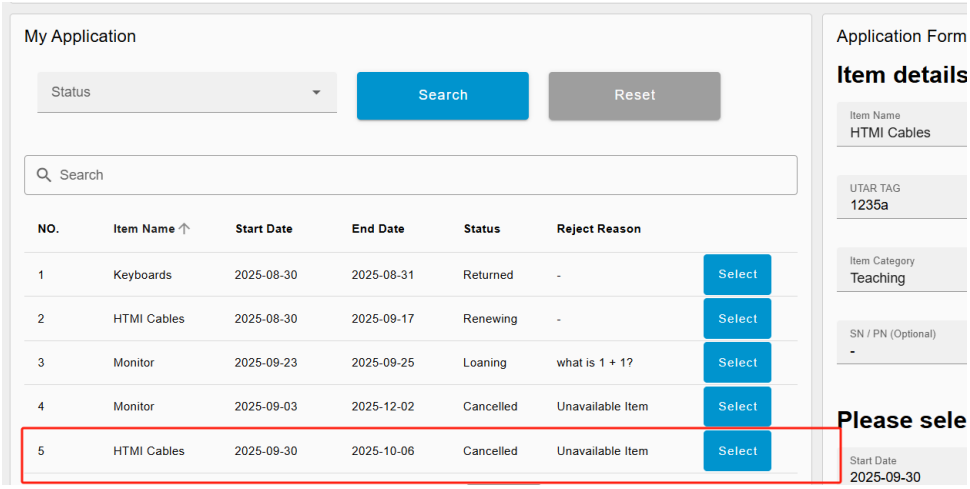


Figure 5.4.58: Requestor's application (Reported)

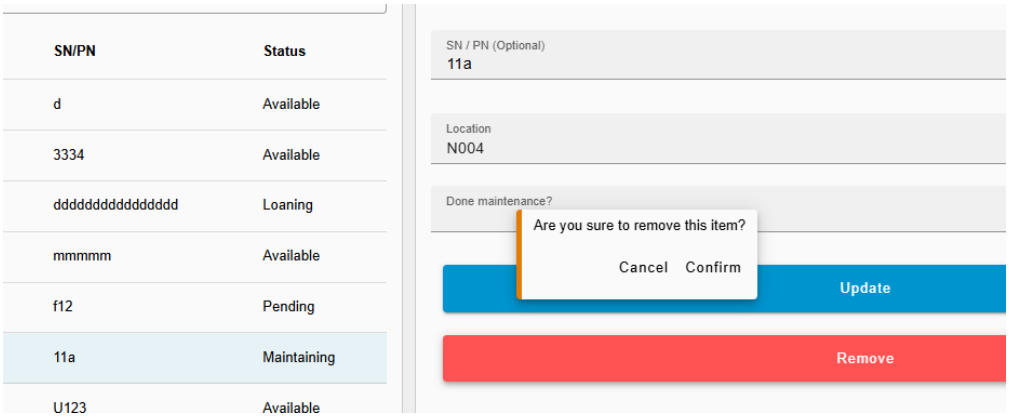


Figure 5.4.59: Remove item which under maintenance

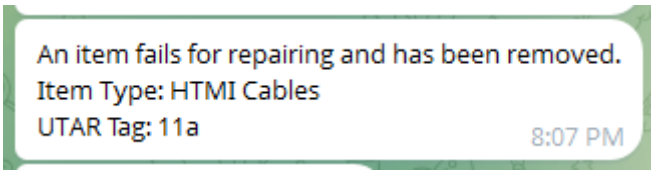


Figure 5.4.60: Notification relevant to failed maintenance and deletion on the item (For Administrator and Lab Staff)

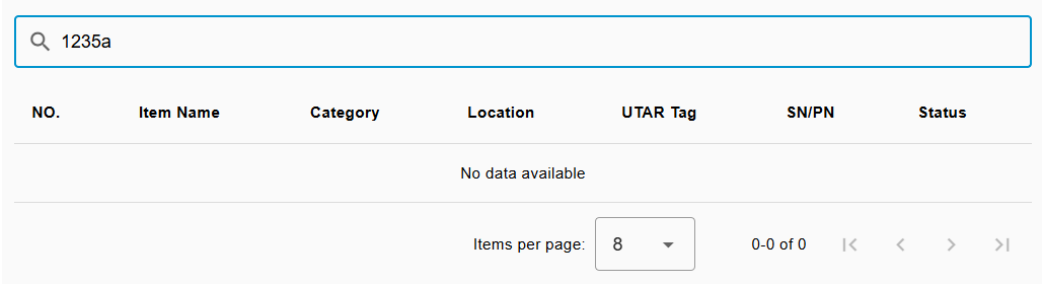


Figure 5.4.61: Result of remove item which under maintenance

5.4.3 Item Types Management Module

The processes of registering, editing, and deleting item types which include error handlings have been presented in this section. The detection of invalid duplicated item types is not case sensitive for preventing any item types that have different formats of the letters but with the same name.

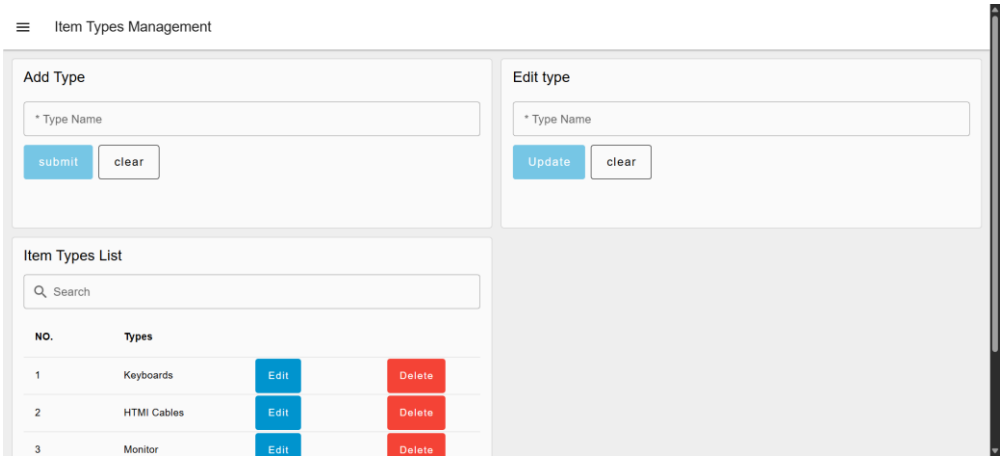


Figure 5.4.62: Interface of Item Types Management Module

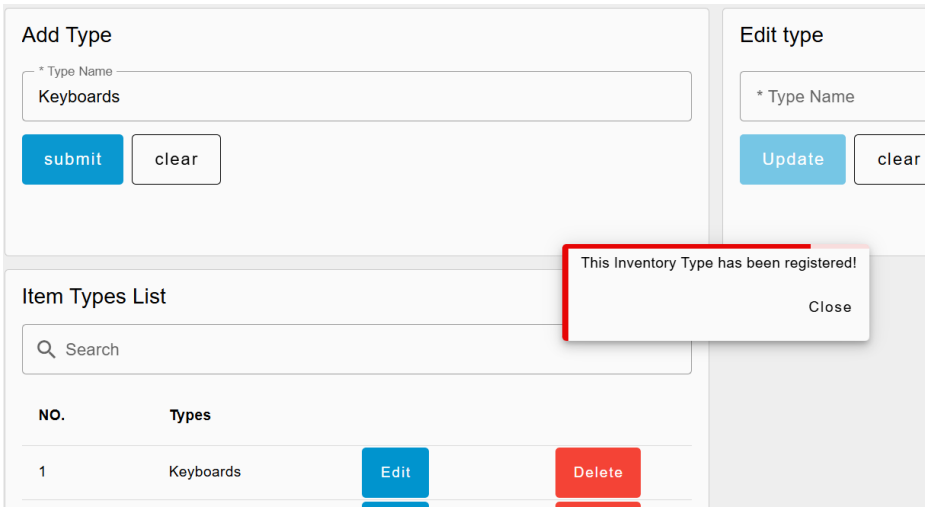


Figure 5.4.62: Add duplicate type

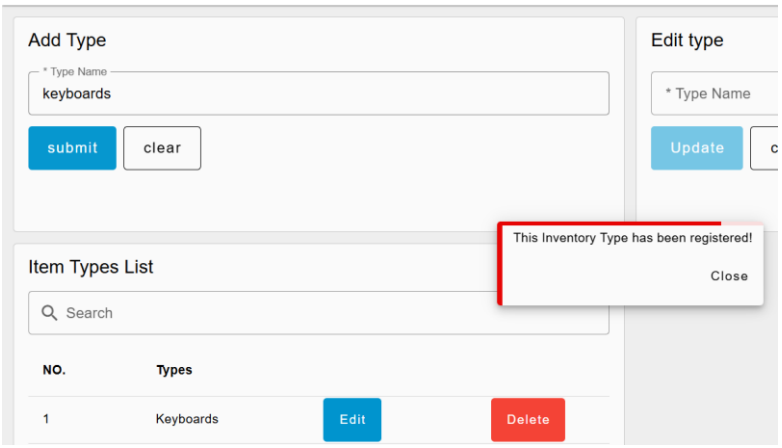


Figure 5.4.63: Add duplicate type with small capital letters

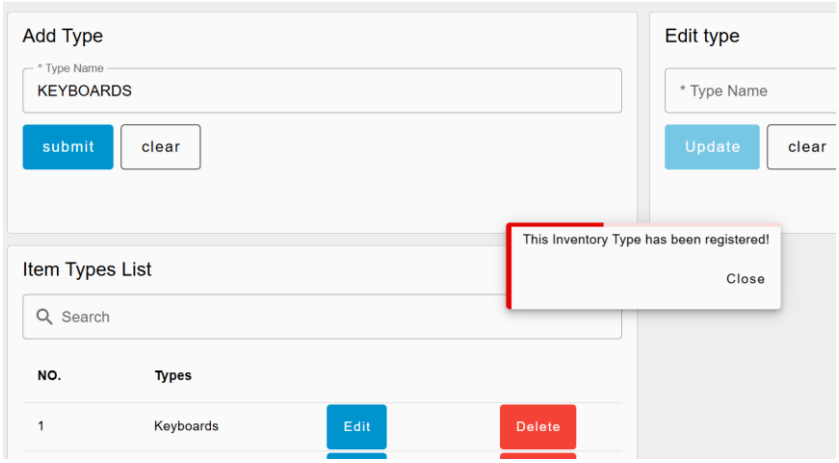


Figure 5.4.64: Add duplicate type with capital letters

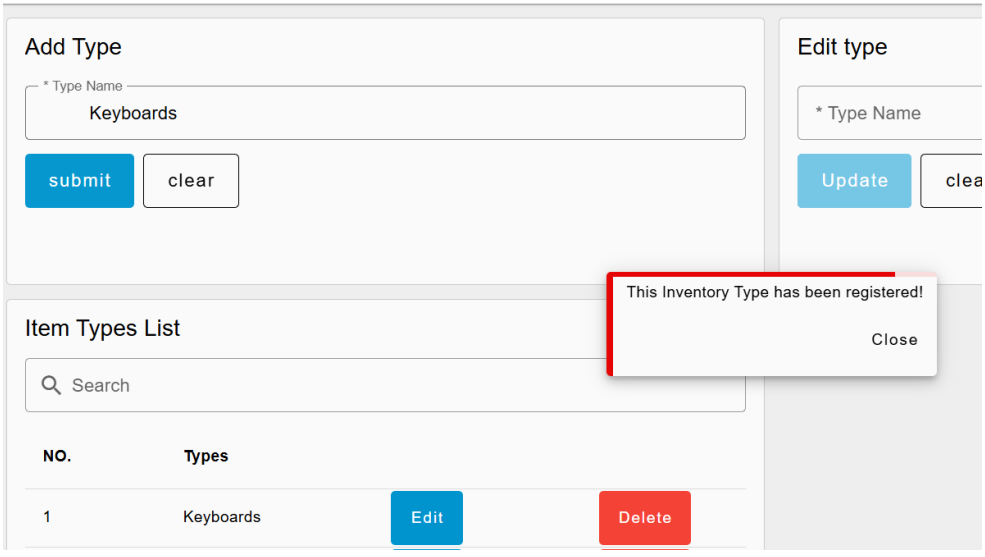


Figure 5.4.65: Add duplicate type with extra spaces

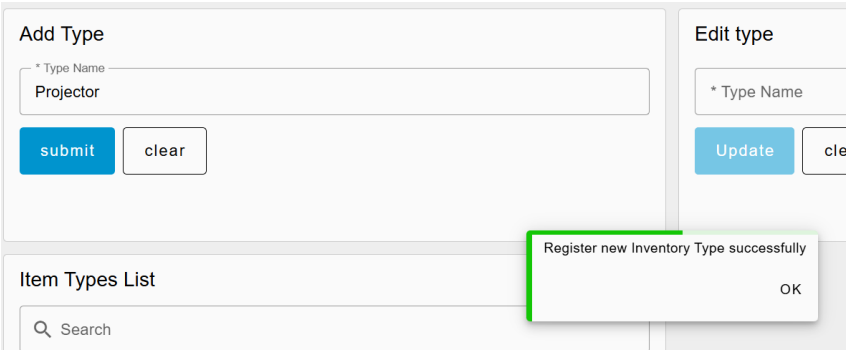


Figure 5.4.66: Add unique type

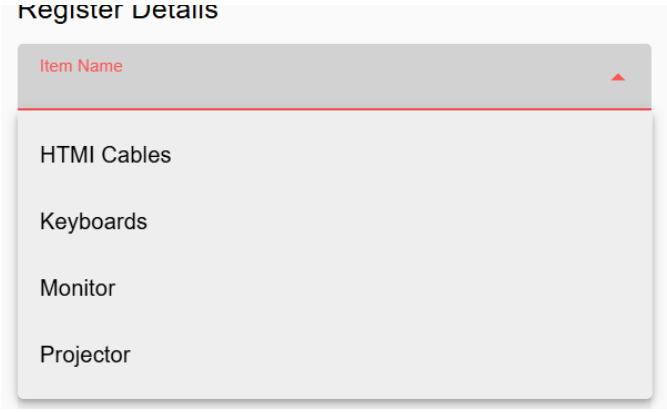


Figure 5.4.67: Updated item name options

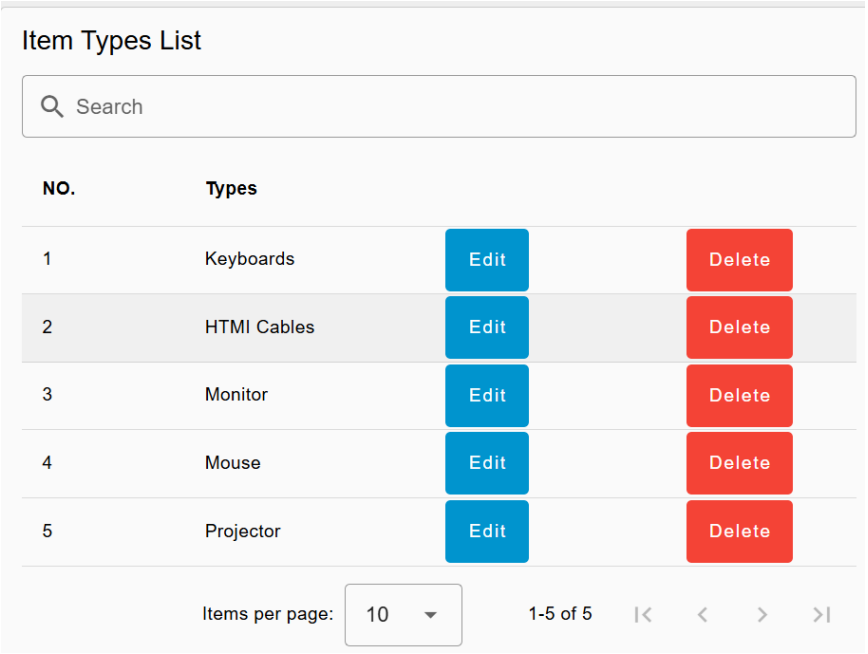


Figure 5.4.68: Updated item types list

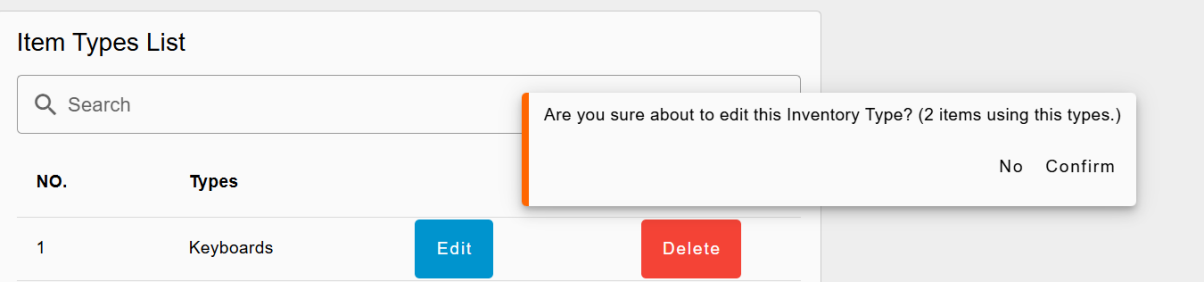


Figure 5.4.69: Edit item which is using

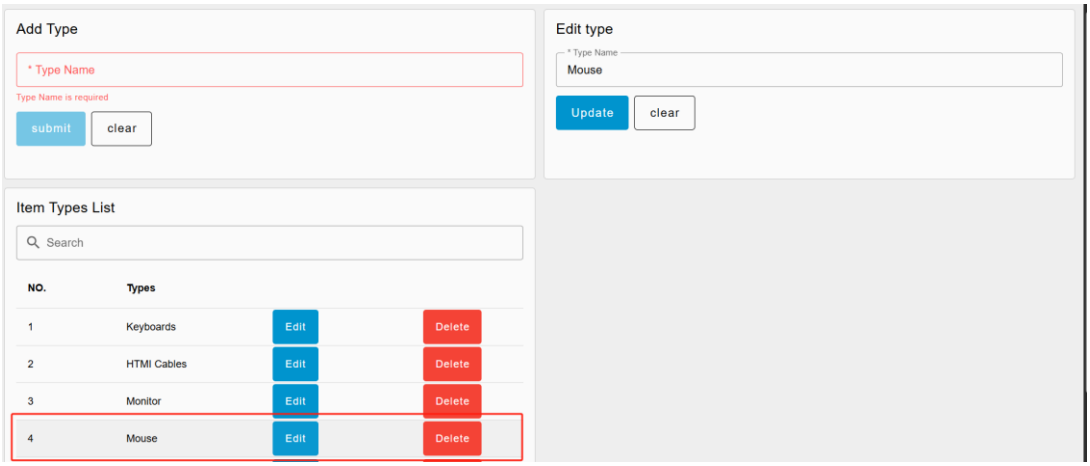


Figure 5.4.70: Edit item which is not used (Directly fill the form)

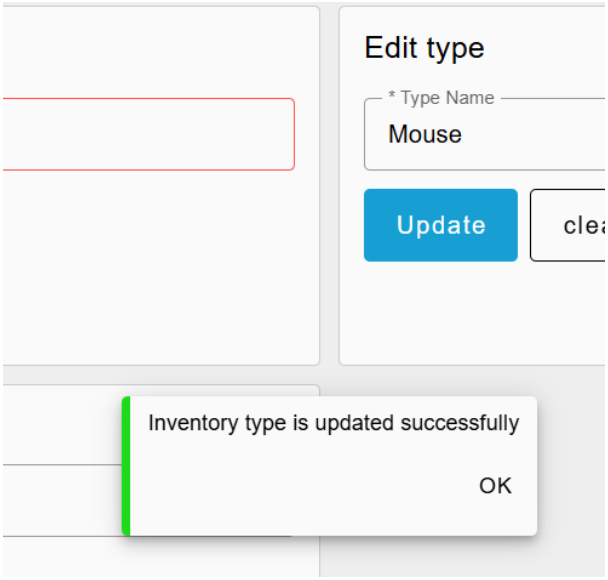


Figure 5.4.71: Update type without any change

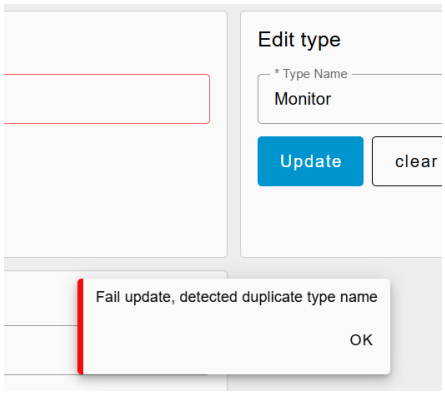


Figure 5.4.72: Update selected type with duplicate type name

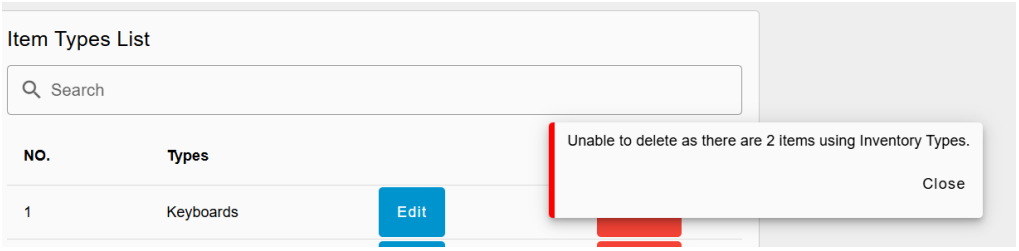


Figure 5.4.73: Unable delete item types that has been used

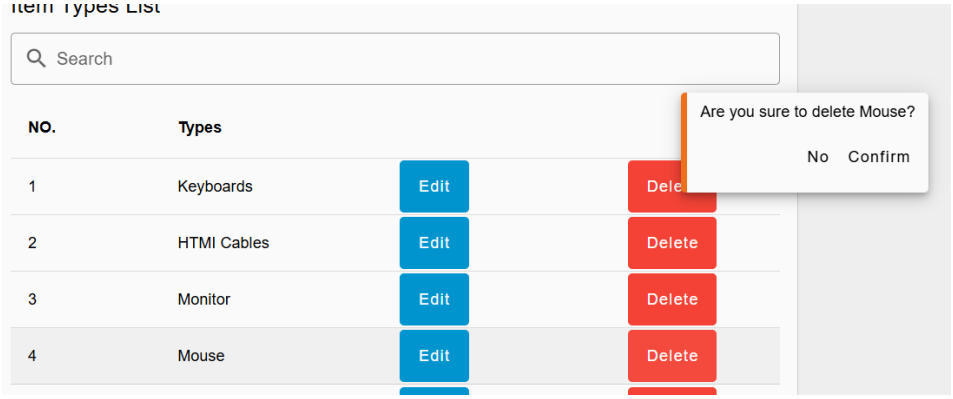


Figure 5.4.74: Delete item types that is not used

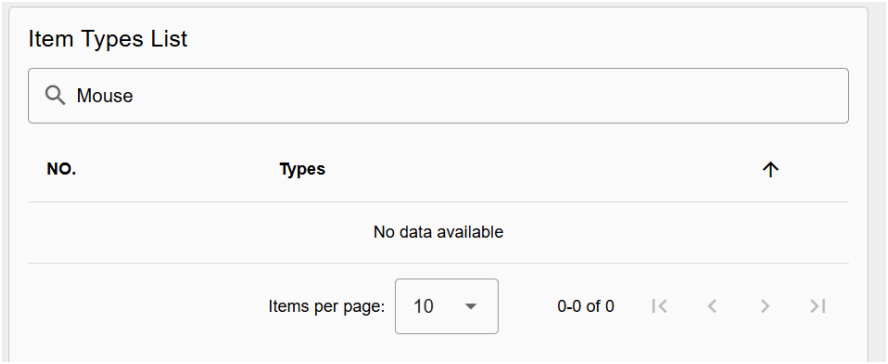


Figure 5.4.75: After delete item

5.4.4 Item Categories Management Module

The processes of registering, editing, and deleting item categories which include error handlings have been presented in this section. The detection of invalid duplicated item categories is not case sensitive for preventing any item categories that have different formats of the letters but with the same name.

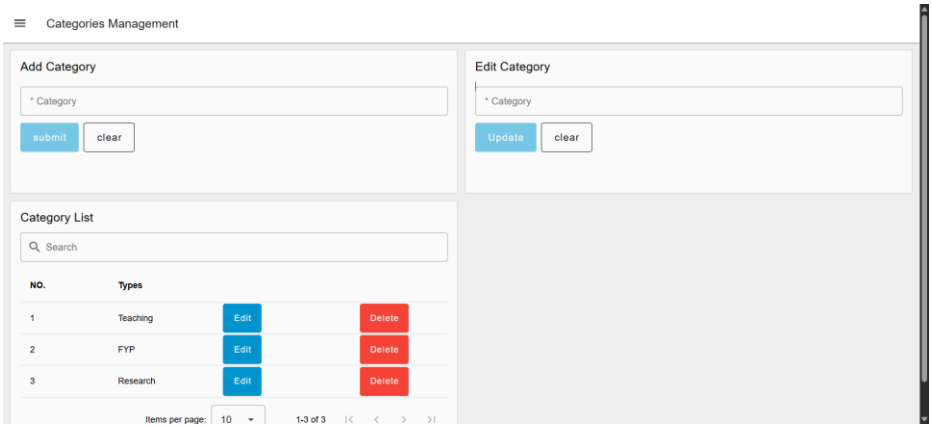


Figure 5.4.76: Interface of Categories Management Module

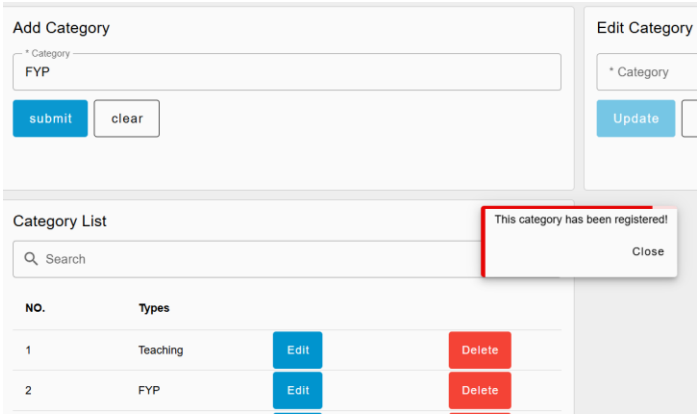


Figure 5.4.77: Add duplicate category

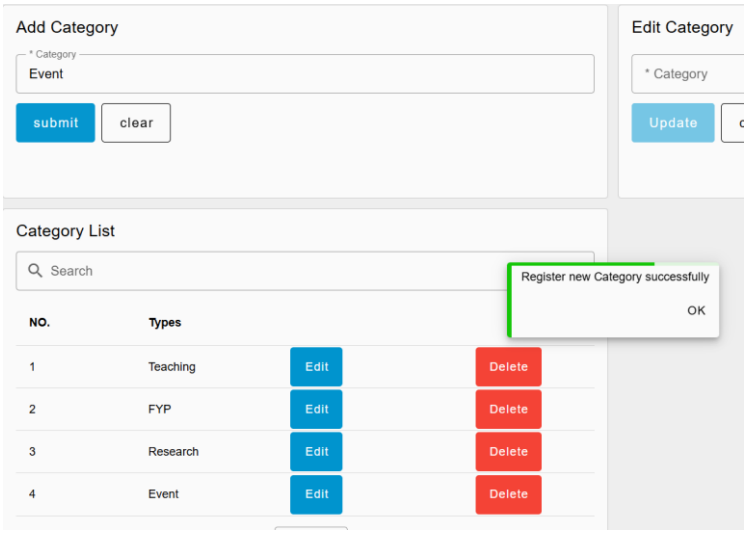


Figure 5.4.78: Add unique category

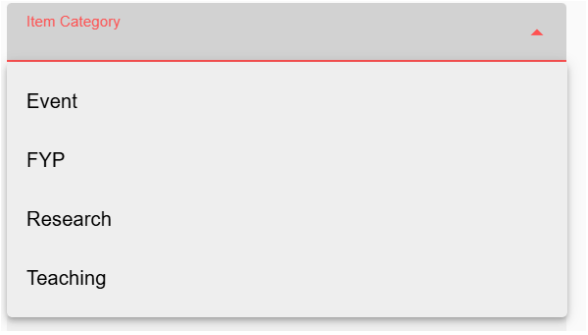


Figure 5.4.79: Updated item category options

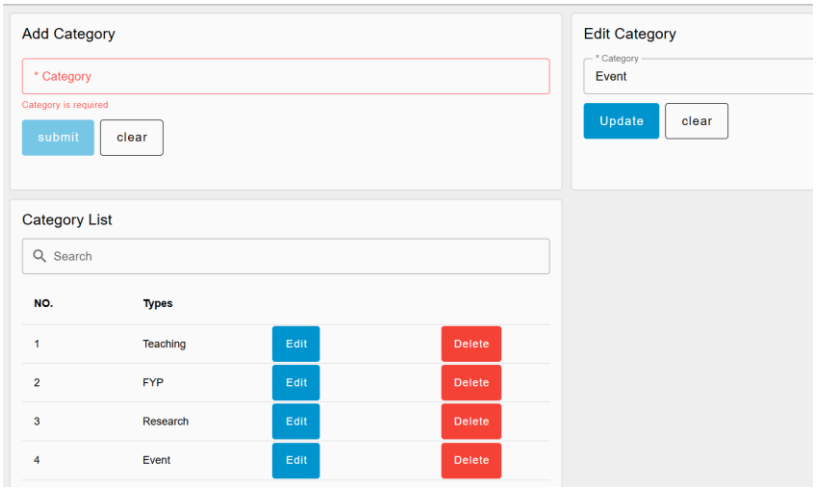


Figure 5.4.80: Select category for edit

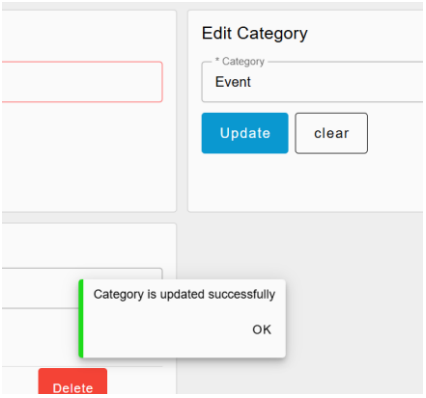


Figure 5.4.81: Update category without any change

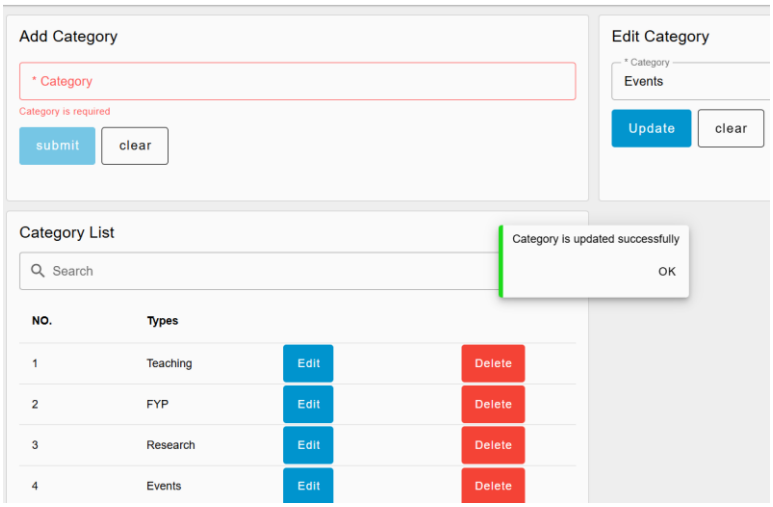


Figure 5.4.82: Update category with unique name

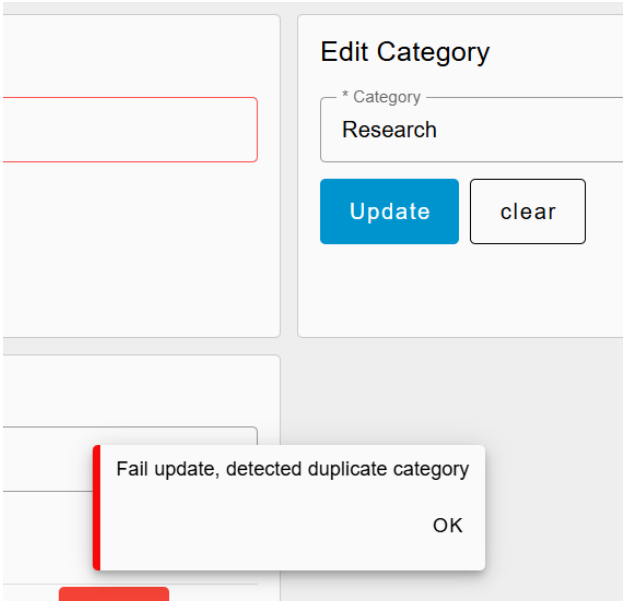


Figure 5.4.83: Update category with duplicate category name

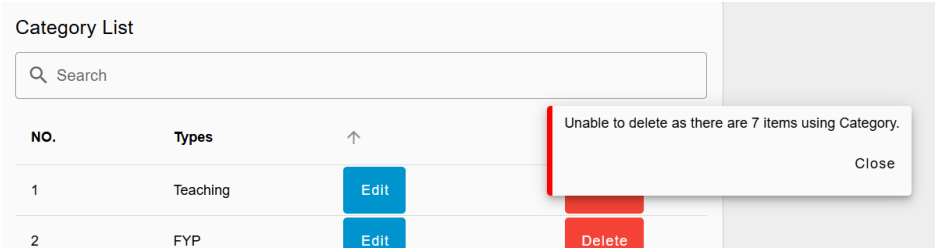


Figure 5.4.84: Unable delete category which is used

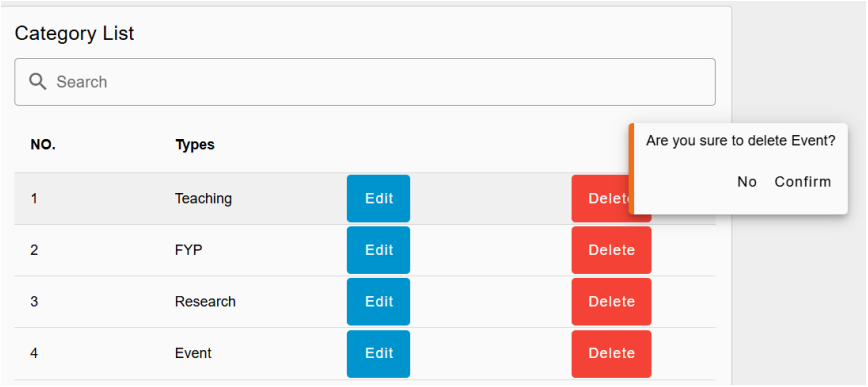


Figure 5.4.85: Delete category which is not used

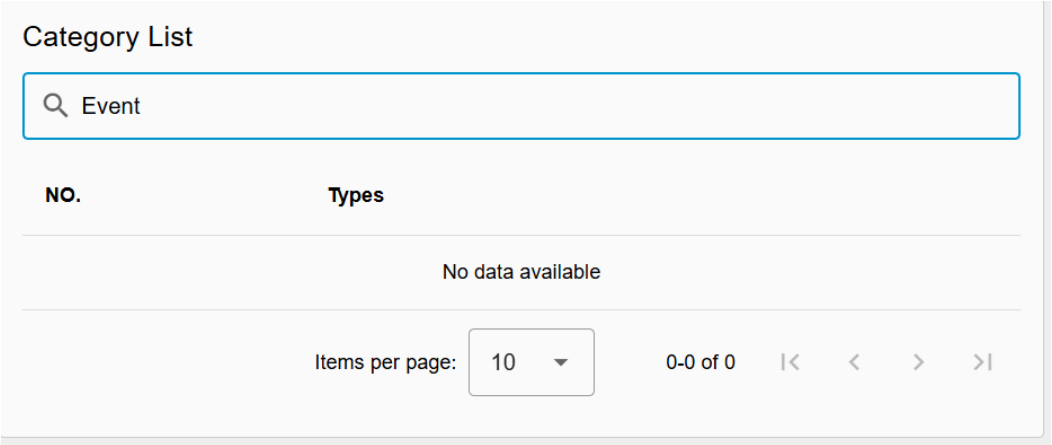


Figure 5.4.86: After delete category

5.4.5 Location Management Module

The processes of registering, editing, and deleting locations which include error handlings have been presented in this section. The detection of invalid duplicated locations is not case sensitive for preventing any locations that have different formats of the letters but with the same name.

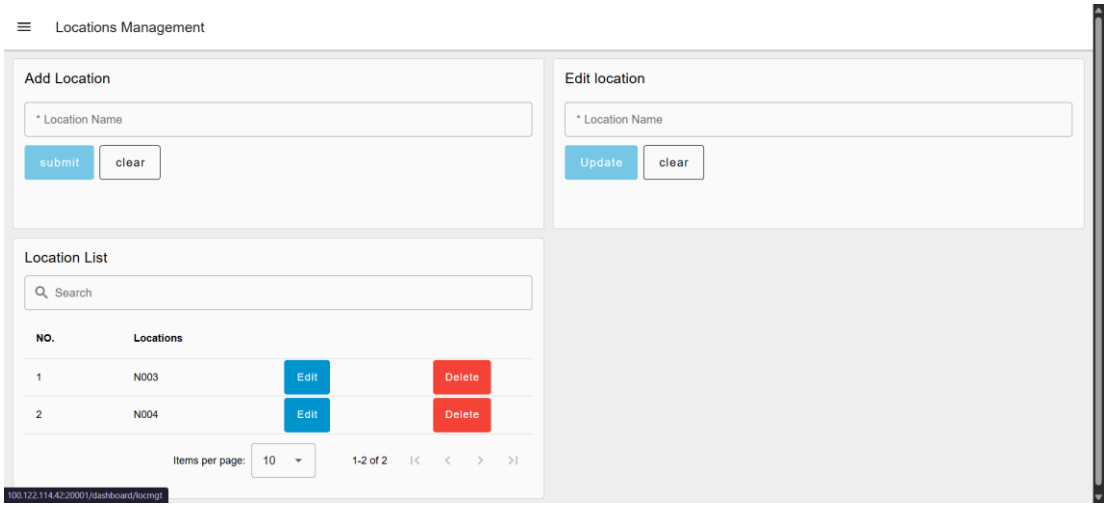


Figure 5.4.87: Interface of Locations Management Module

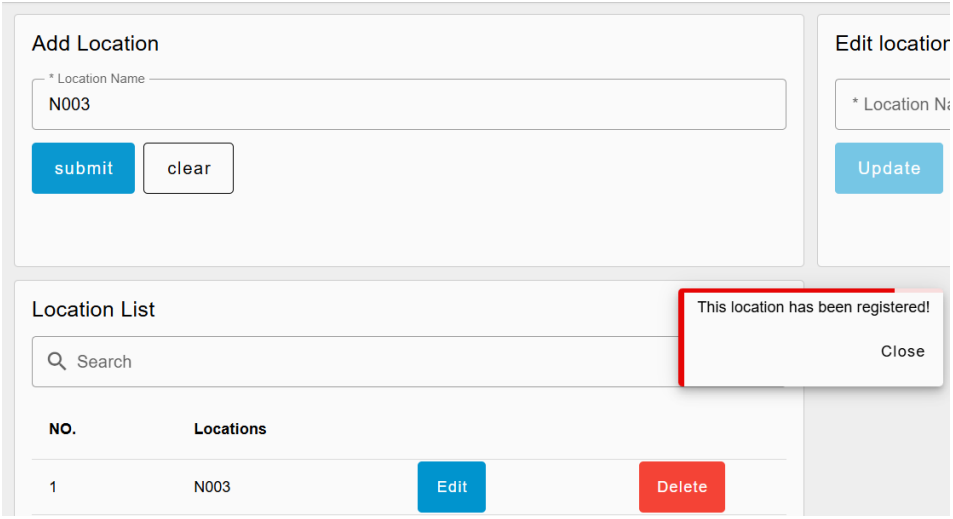


Figure 5.4.88: Add duplicate location

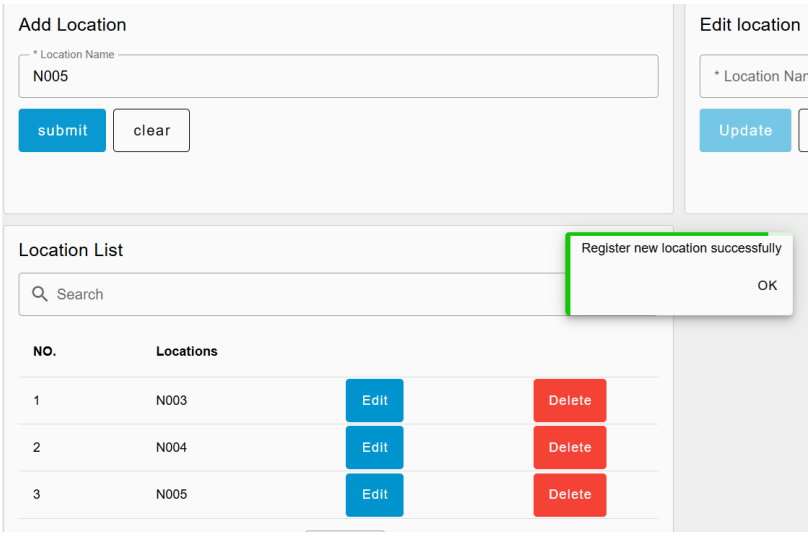


Figure 5.4.89: Add unique location

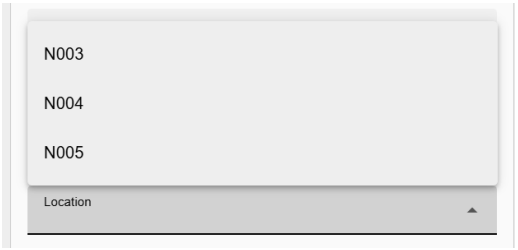


Figure 5.4.90: Updated location options

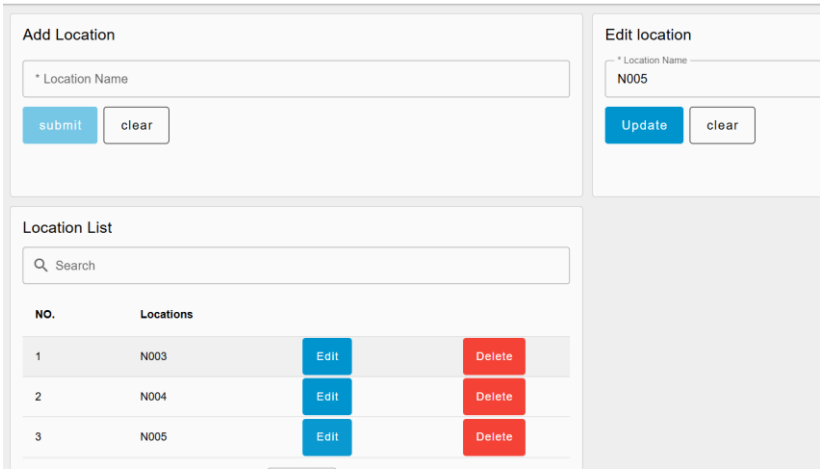


Figure 5.4.91: Select location for edit

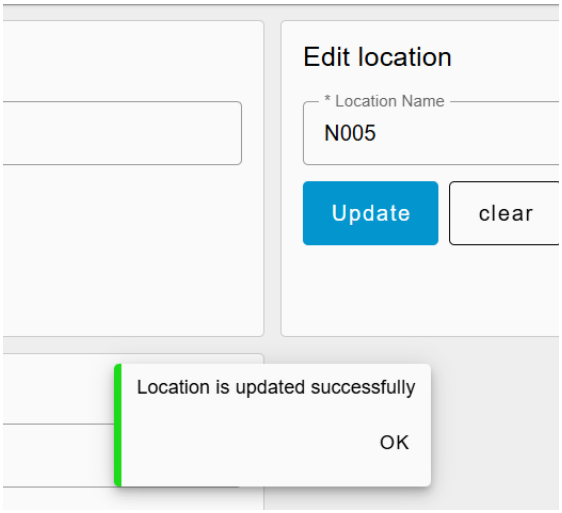


Figure 5.4.92: Update location without any change

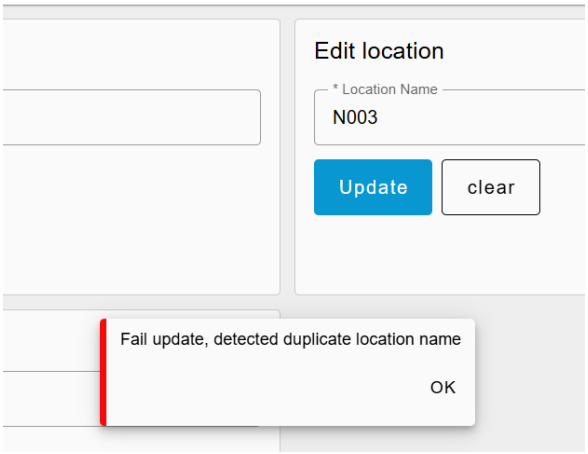


Figure 5.4.93: Update location with duplicate location name

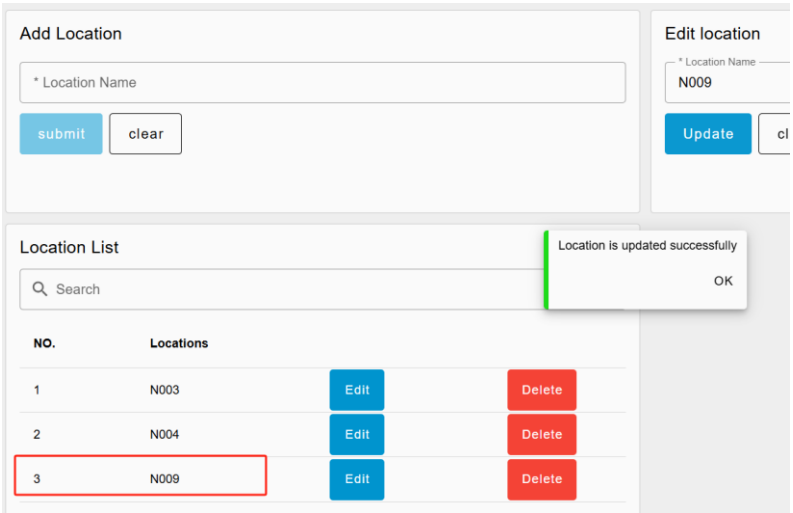


Figure 5.4.94: Update location with unique location name

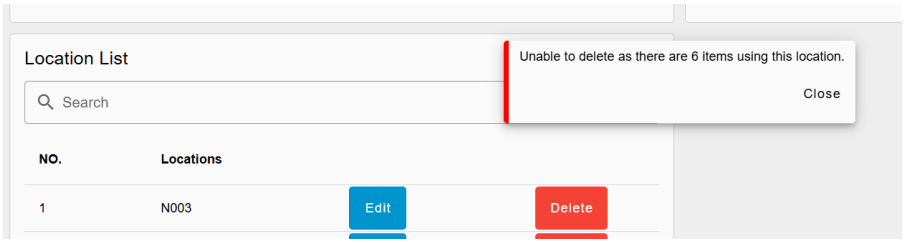


Figure 5.4.95: Delete location which is used

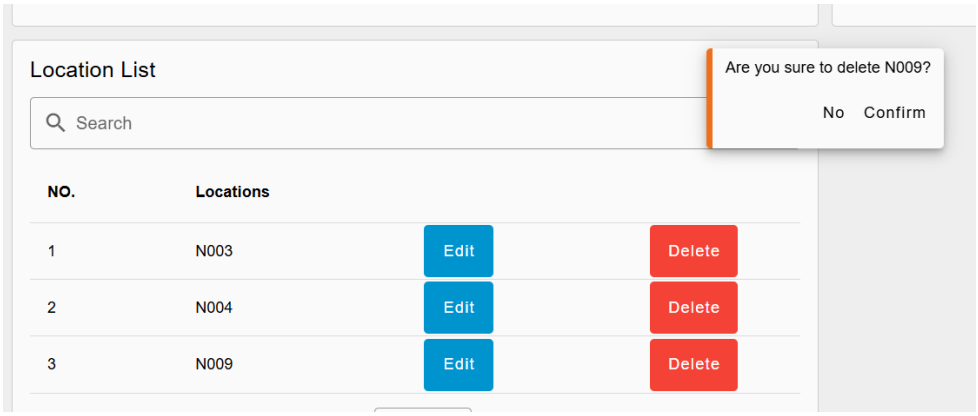


Figure 5.4.96: Delete location which is not used

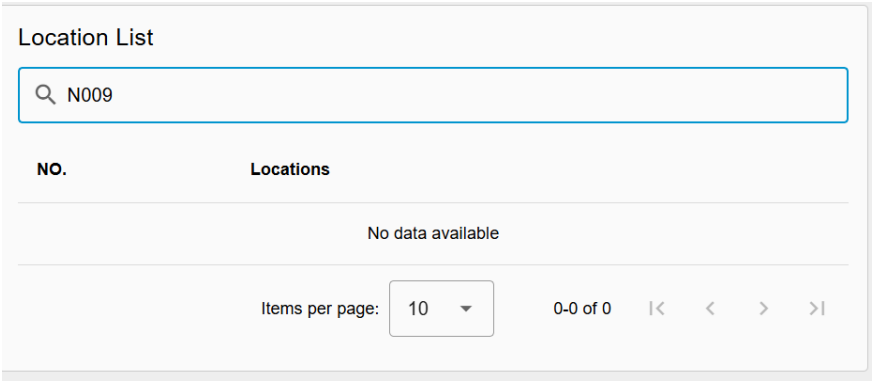


Figure 5.4.97: After delete location

5.4.6 User Management Module

This module provides dynamic interfaces for administrator and lab staff. On the other hand, the student is only allowed to use this module for applying the new account.

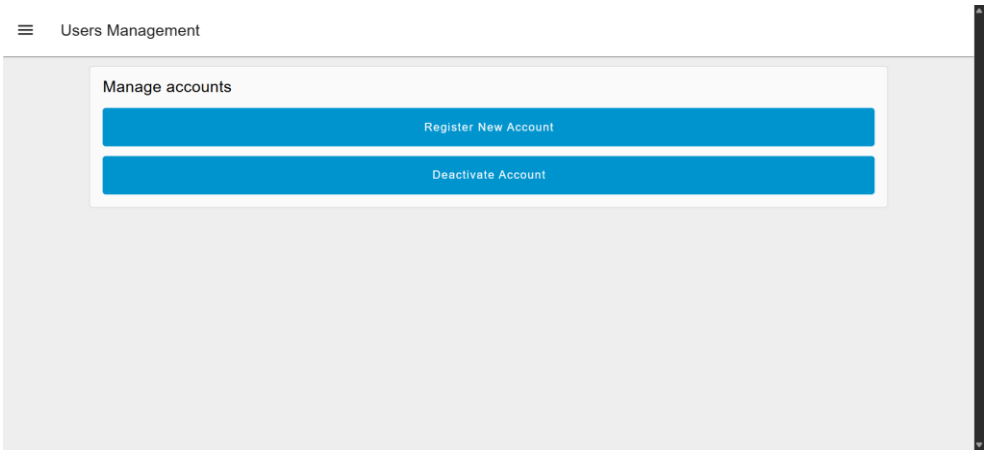


Figure 5.4.98: Main page of Users Management Module (Adminstrator)

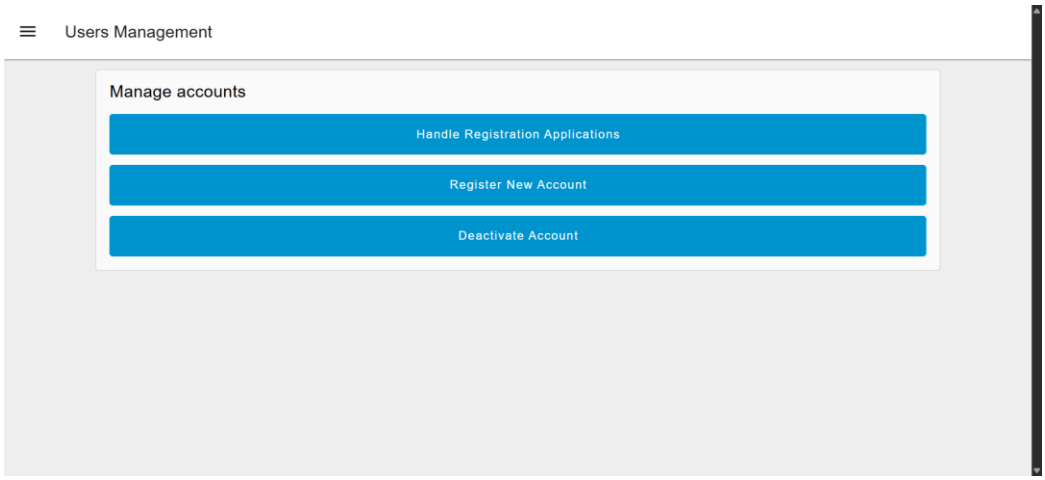


Figure 5.4.99: Main page of Users Management Module (Lab Staff)

During applying and registering the new account, all the fields within the forms must be completed with the required input formats. Then, the registered email and student ID must be unique. Additionally, the reason for not performing checking on duplicated Telegram chat ID is to prevent any Telegram chat ID from being known by others.

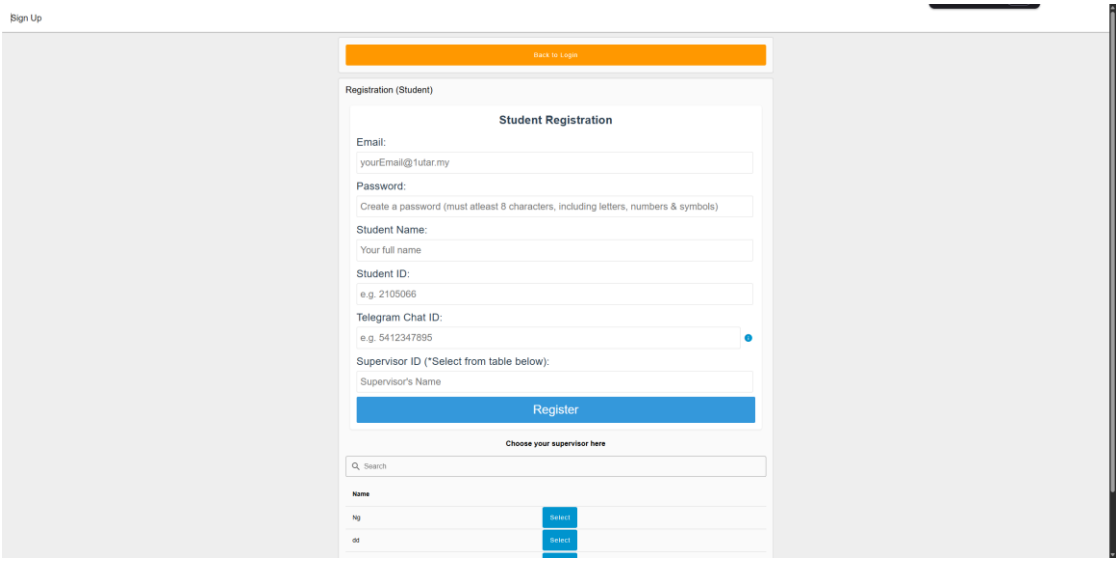


Figure 5.4.100: Interface for student applies new account

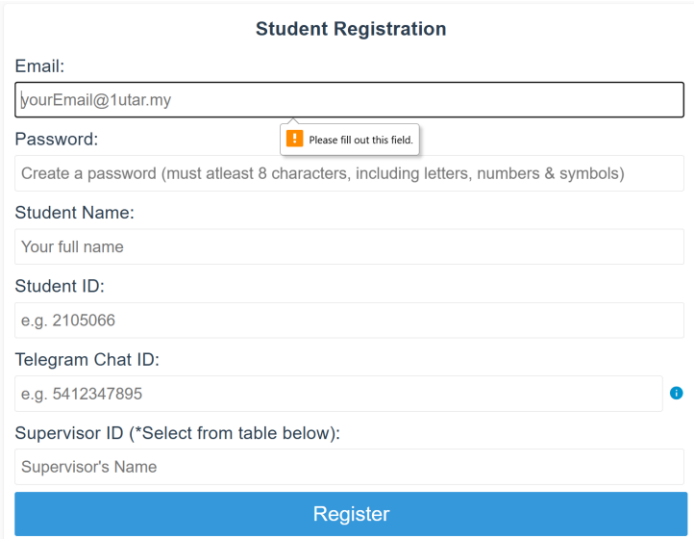


Figure 5.4.101: Detect any empty field

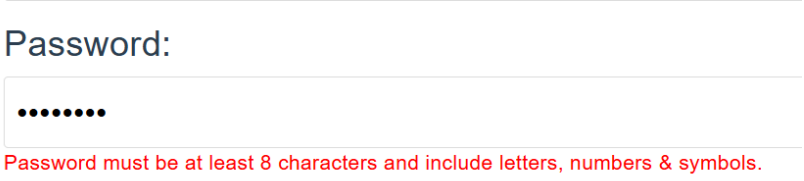


Figure 5.4.102: Password that fail to follow required format

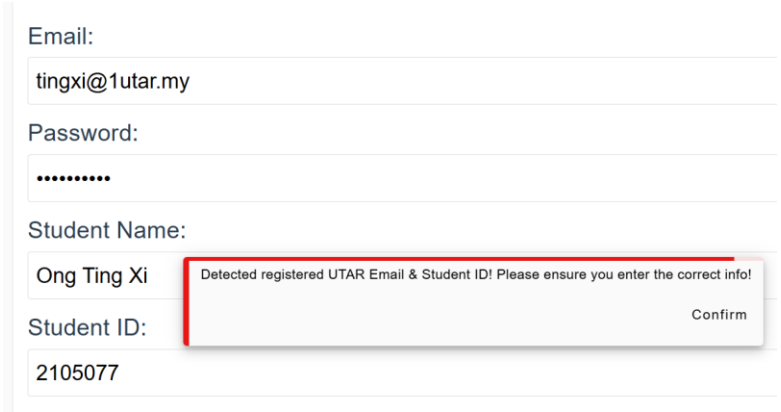


Figure 5.4.103: Register duplicate email and student ID

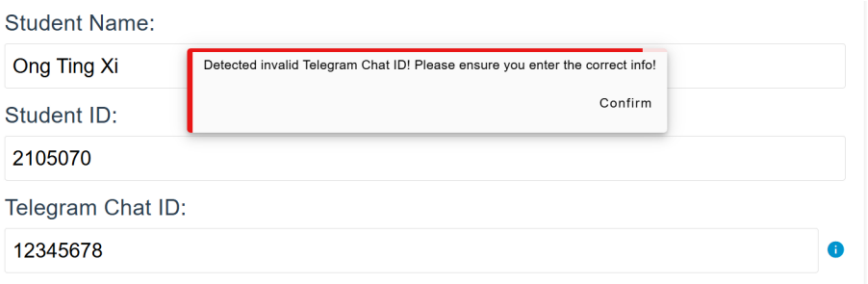


Figure 5.4.104: Detected invalid Telegram Chat ID

Only the student account's registration needs to be approved by lab staff.

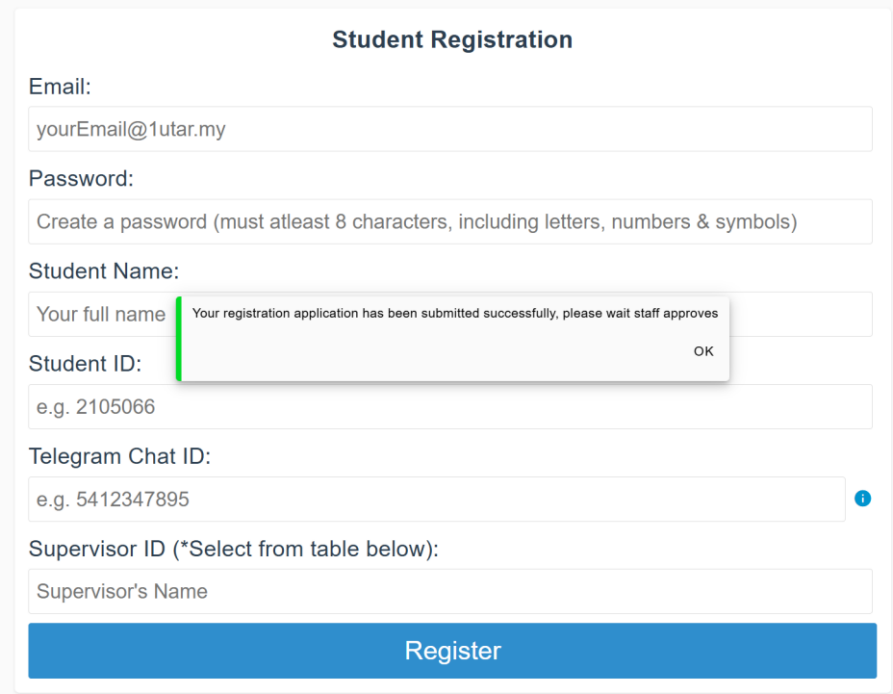


Figure 5.4.105: Submit correct registration form (Student)

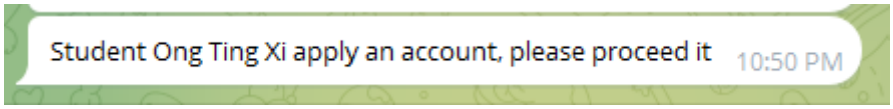


Figure 5.4.106: Notification received by all lab staff

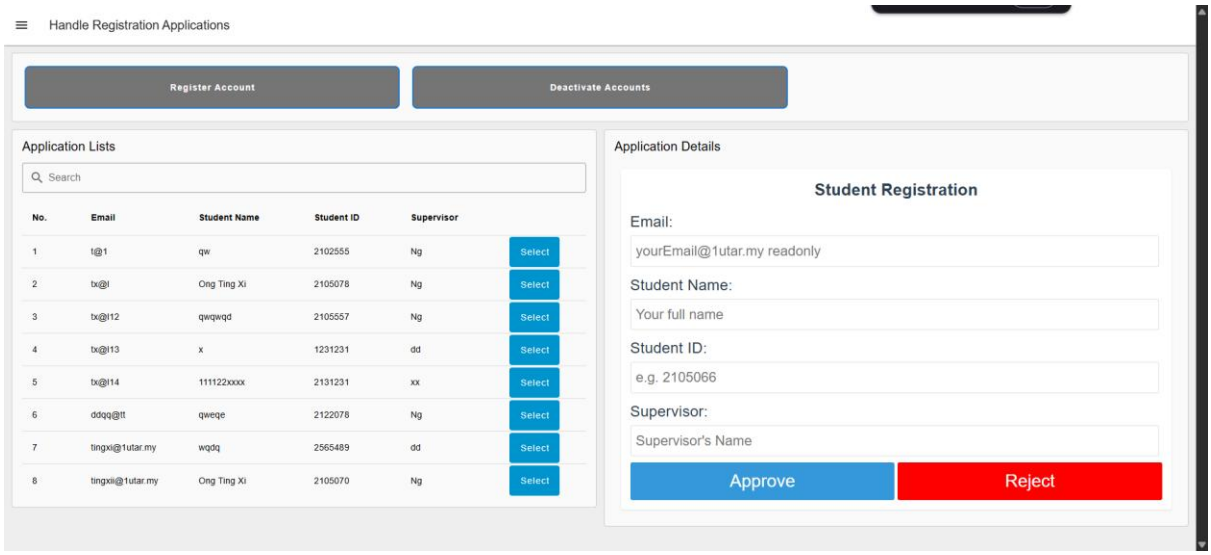


Figure 5.4.107: Interface of handle students' registration applications (Lab Staff only)

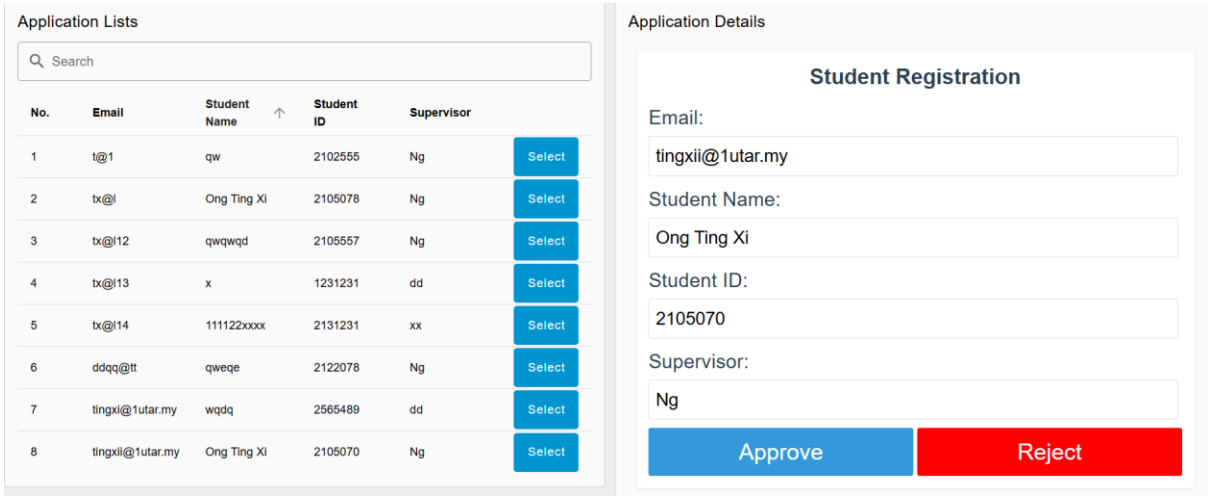


Figure 5.4.108: Select application for handling

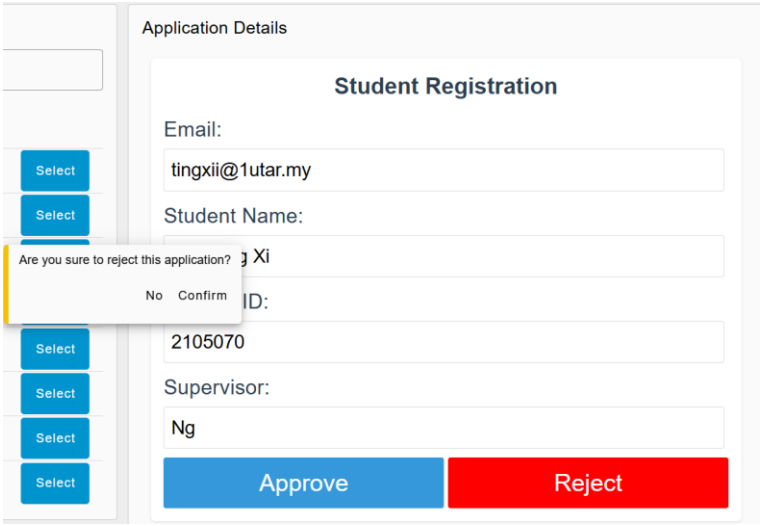


Figure 5.4.109: Double confirm on rejection of application

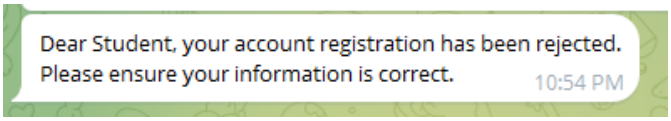


Figure 5.4.110: Notification received by student after rejection

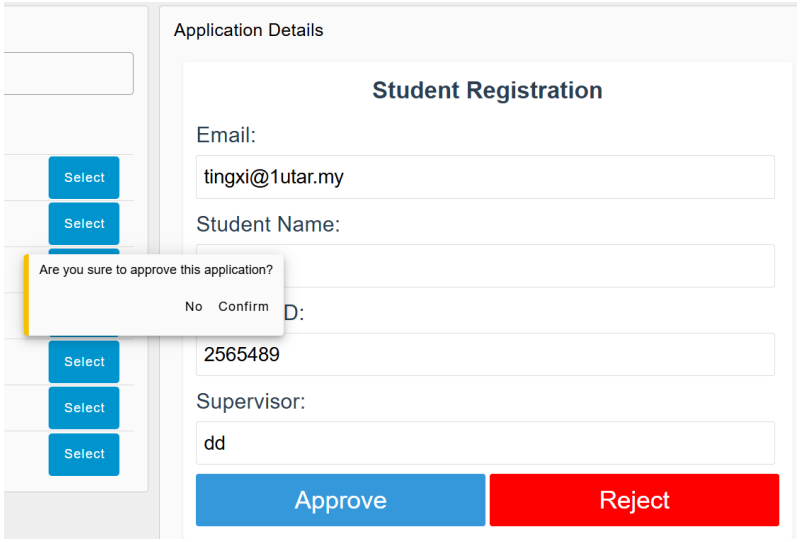


Figure 5.4.111: Double confirm on approvement

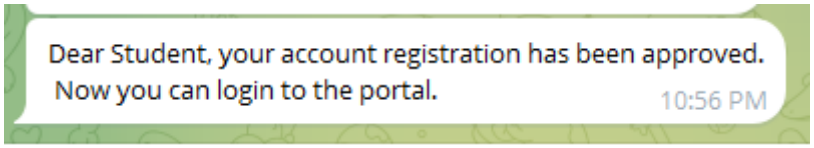


Figure 5.4.112: Notification received by student after approvement

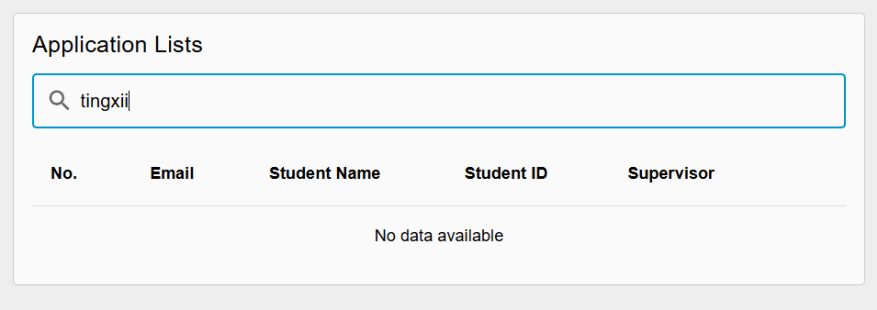


Figure 5.4.113: Application list after rejecting successfully

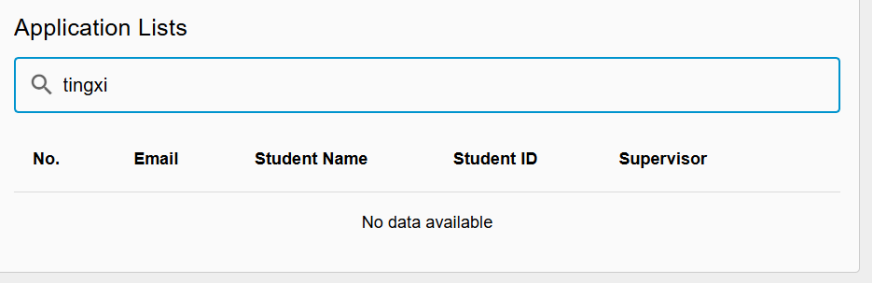


Figure 5.4.114: Application list after approving successfully

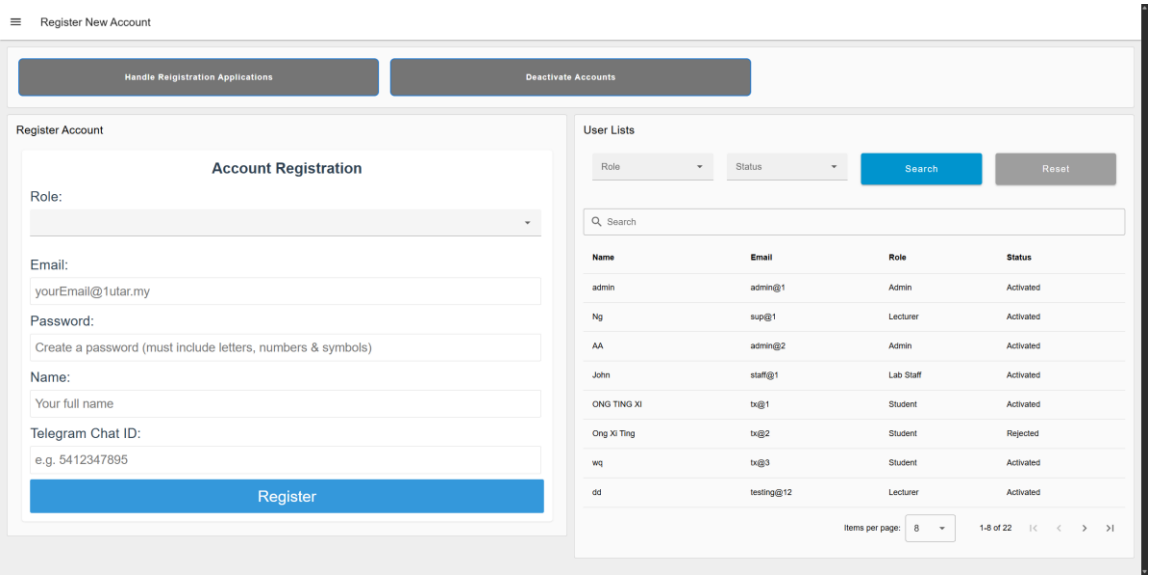


Figure 5.4.115: Interface for account registration (Lab Staff)

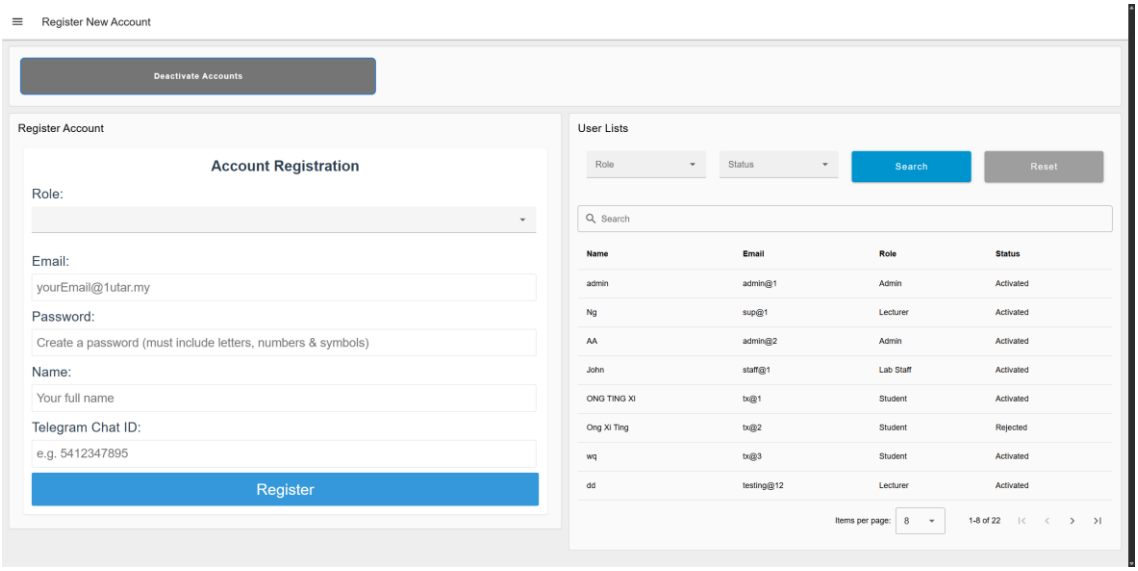


Figure 5.4.116: Interface for account registration (Administrator)

Lab Staff can register new account for other supervisor (lecturer) and lab staff. On the other hand, administrators can register new account for every role other than student.

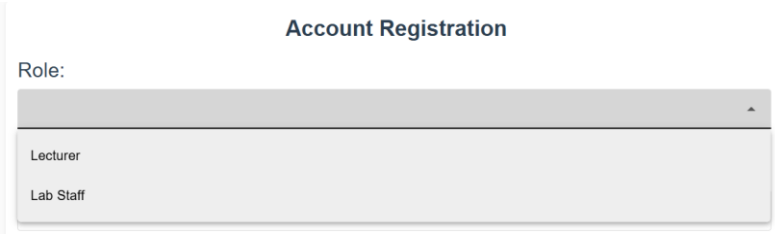


Figure 5.4.117: Role options (Lab Staff)

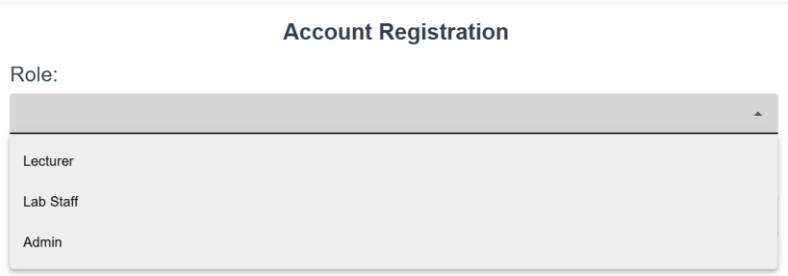


Figure 5.4.118: Role options (Administrator)

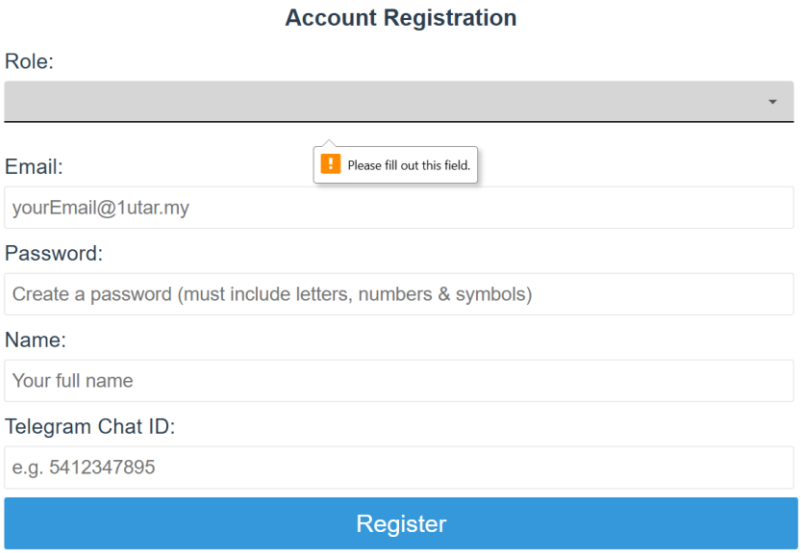


Figure 5.4.119: Detection of any empty field

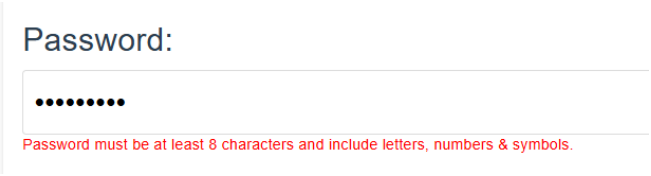


Figure 5.4.120: Password format fails to match requirement

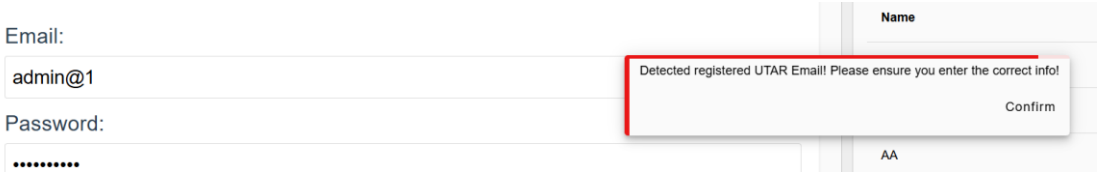


Figure 5.4.121: Register duplicate email

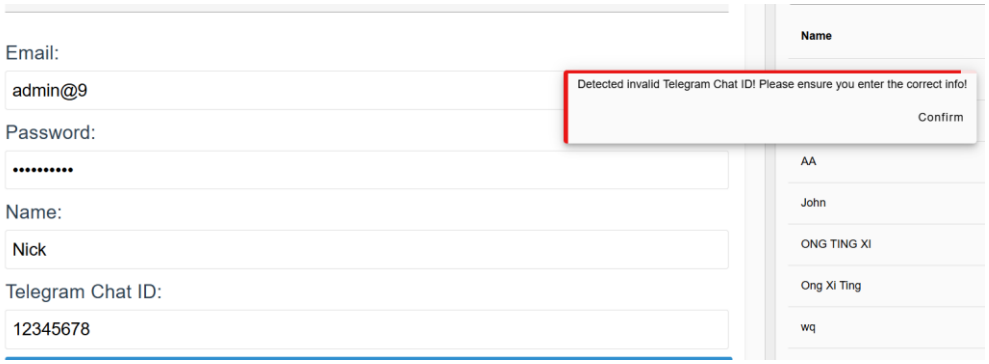


Figure 5.4.122: Register invalid Telegram Chat ID

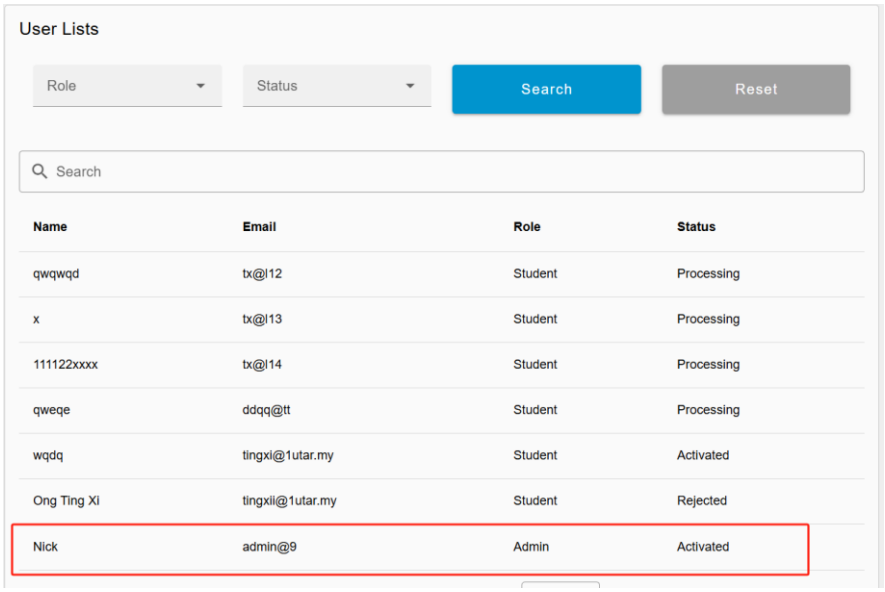


Figure 5.4.123: Updated user list (Register successfully)

Lab staff can deactivate student accounts while administrator can deactivate accounts with any role except student and the first admin account.

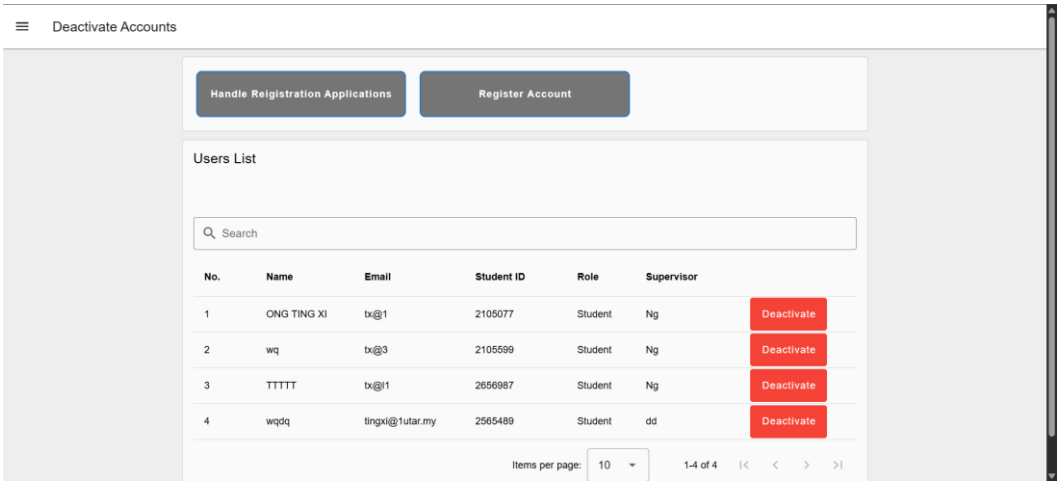


Figure 5.4.124: Interface of deactivate account (Lab Staff)

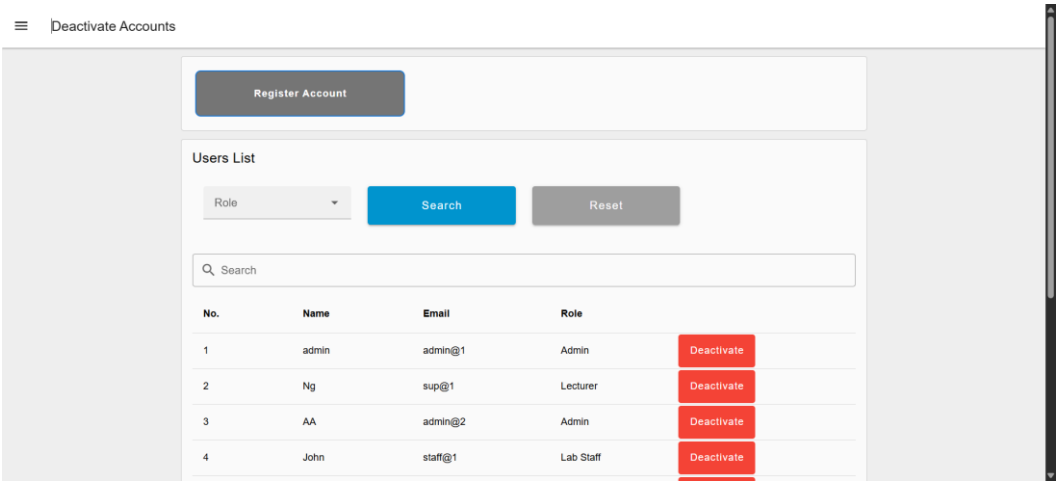


Figure 5.4.125: Interface of deactivate accounts (Admin)

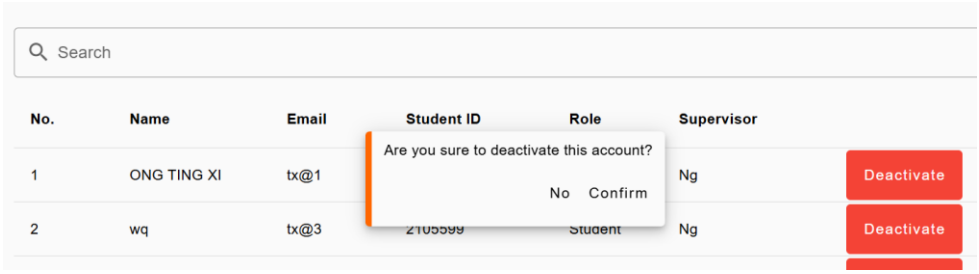


Figure 5.4.126: Double confirm on deactivating account

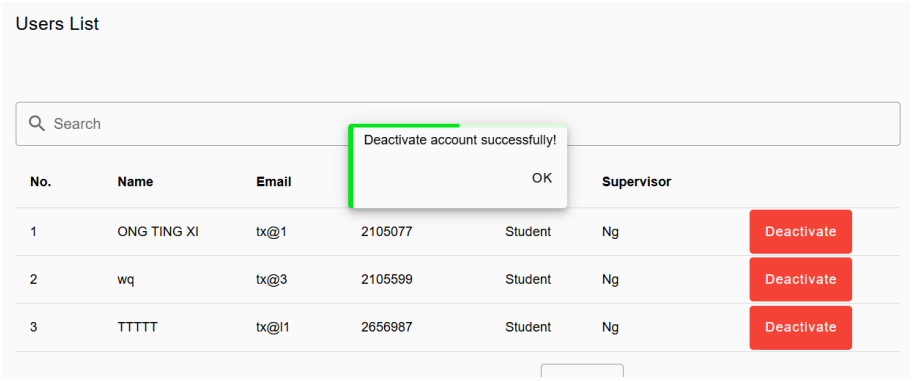


Figure 5.4.127: Deactivate successfully

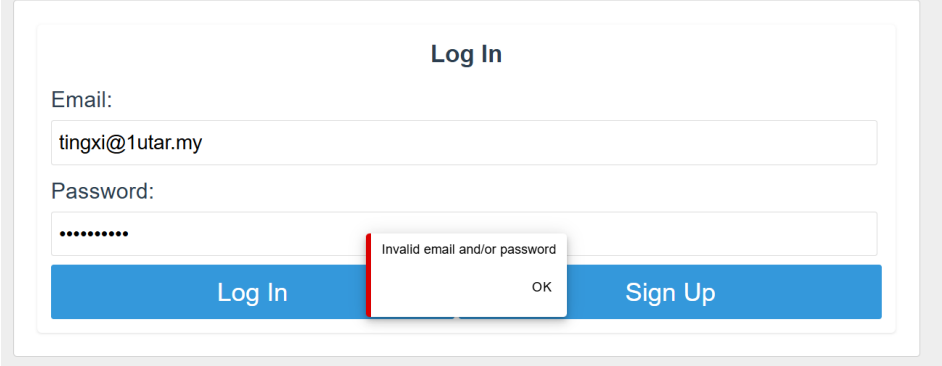


Figure 5.4.128: User unable to login anymore

This system must have at least one administrator account for preventing this system lack of a top-level user.

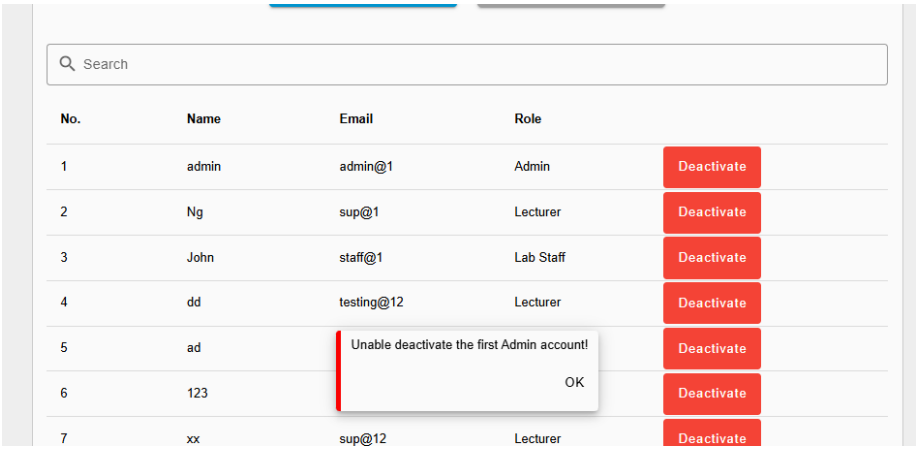


Figure 5.4.129: Unable to delete first administrator account

The users with uncompleted cases are not allowed to be deactivated to ensure the loaning process is following the regulations.

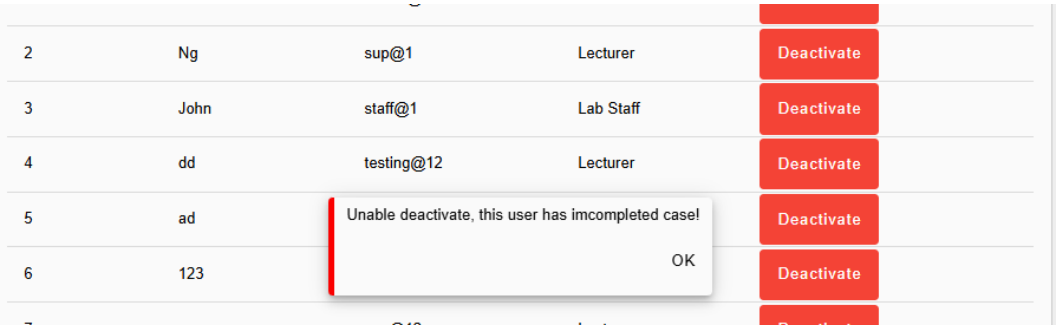


Figure 5.4.130: Unable delete user account that has uncompleted case

5.4.7 Applications Management Module

This module contains two main parts. One part (My Applications section) is for students and supervisors to loan item, while the other part (Applications Management and Handle Applications sections) is for supervisors, administrators, and lab staff to handle the applications.

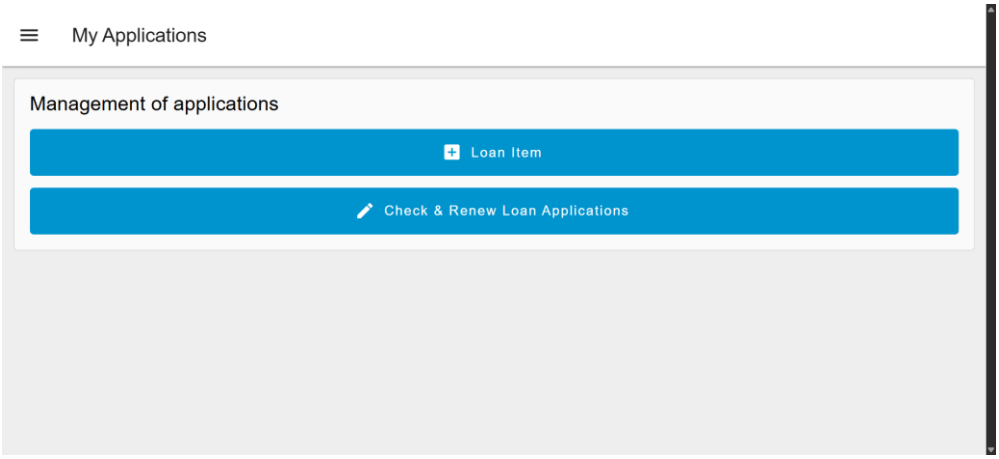


Figure 5.4.131: Main page of "My Applications" (Student & Supervisor)

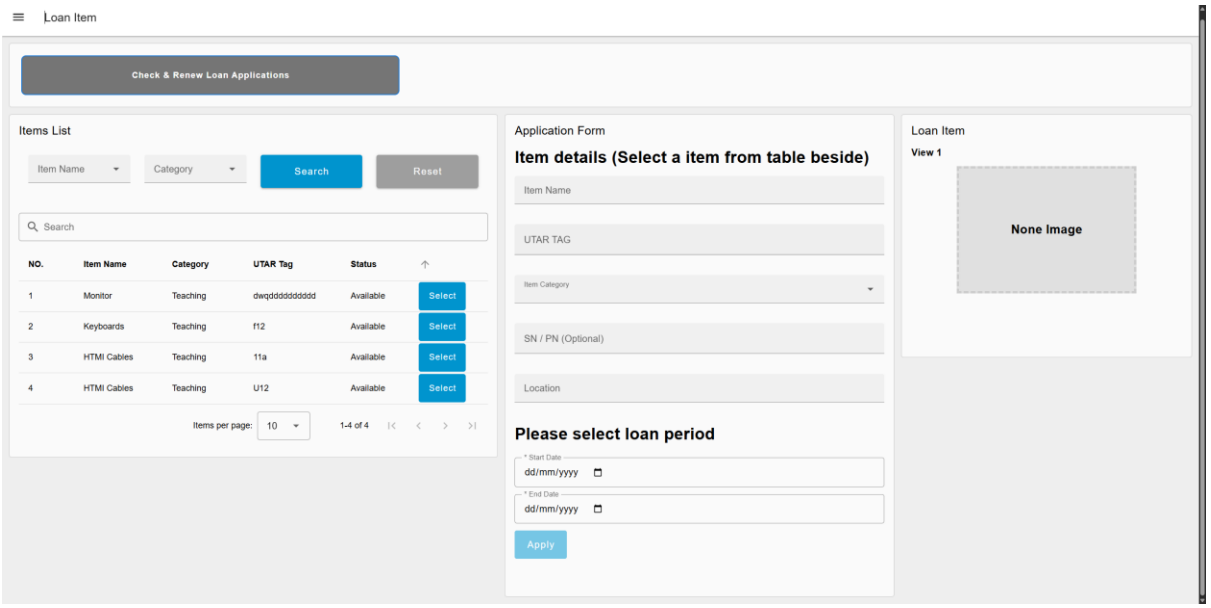


Figure 5.4.132: Interface of loan item

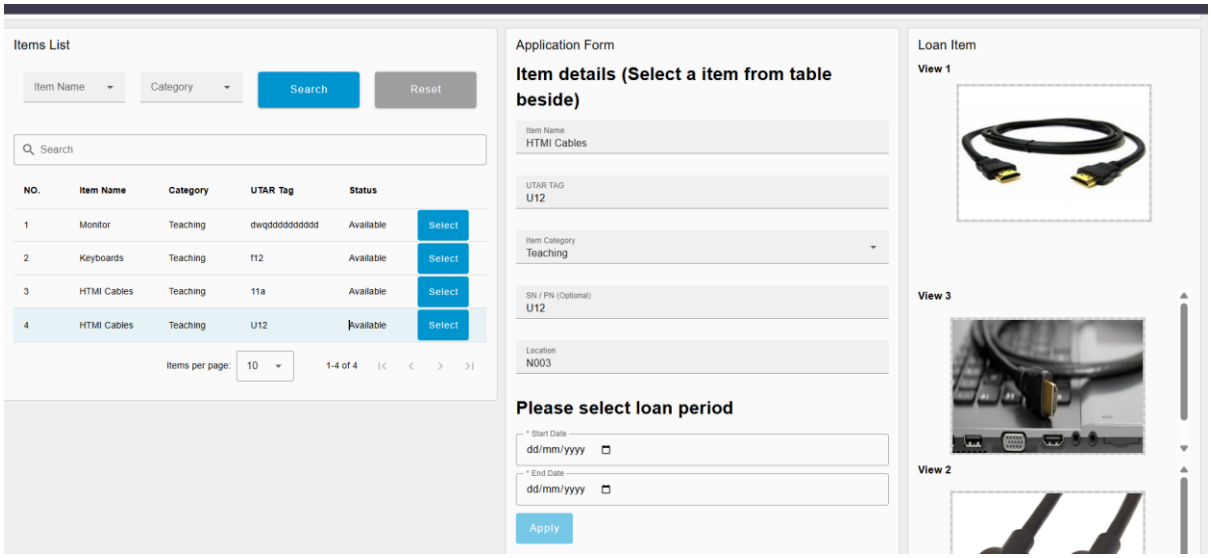


Figure 5.4.133: Select item for loaning

There are several rules on choosing the loan period for preventing any loan application with ambiguous loan period that can cause the system to be confusing in handling the automated checking on the due date of each application.

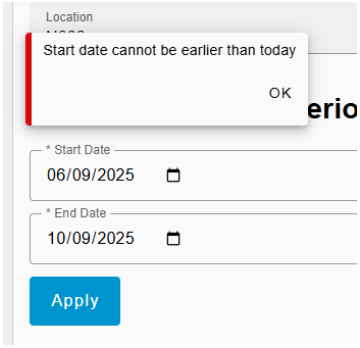


Figure 5.4.134: Select start date that earlier than today

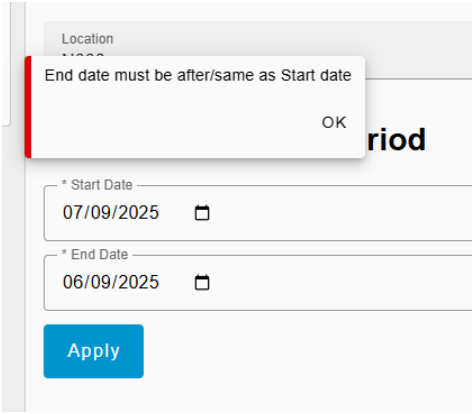


Figure 5.4.135: Select end date that earlier than start date

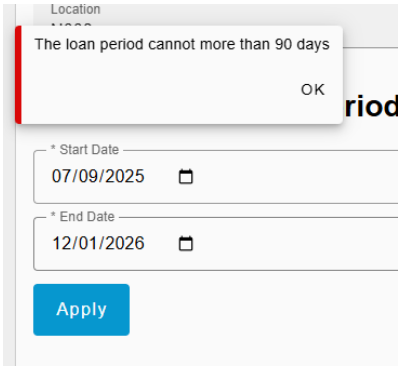


Figure 5.4.136: Select loan period that more than 90 days

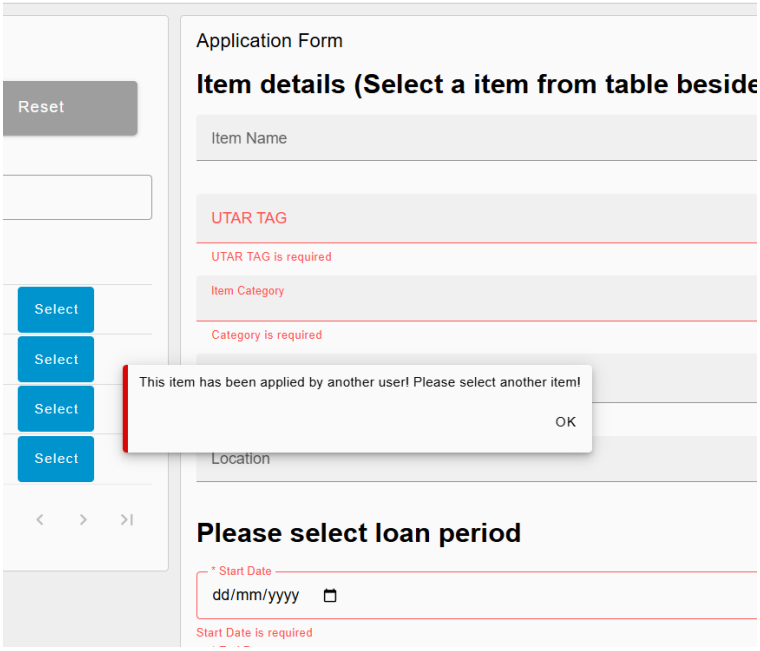


Figure 5.4.137: Situation when the same item is applied by others

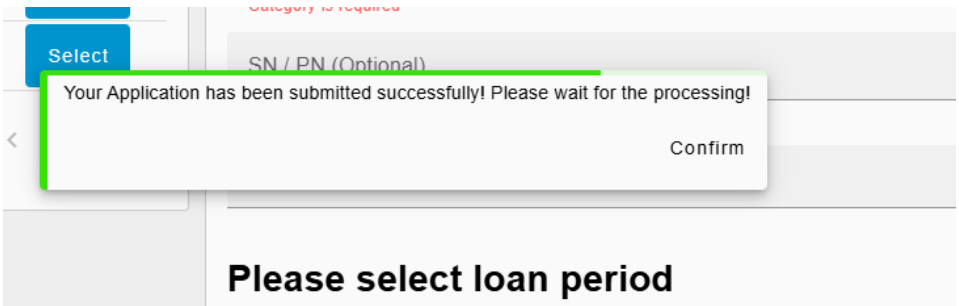


Figure 5.4.138: Submit valid loan period

Supervisors are responsible for handling their students' applications while administrators are responsible for handling the supervisors' applications.

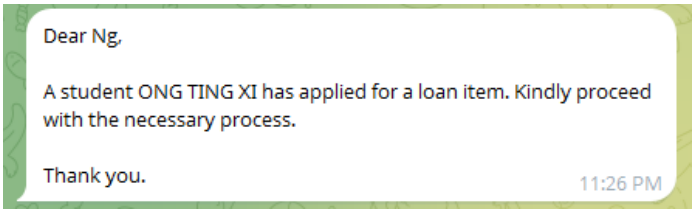


Figure 5.4.139: Notifications received by corresponding supervisor (Student's application)

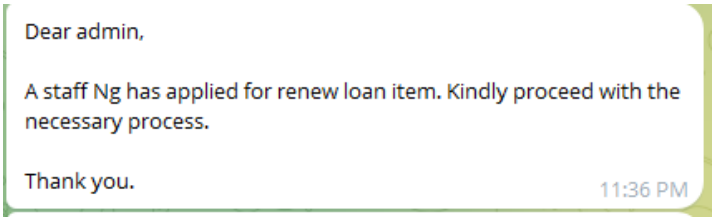


Figure 5.4.140: Notifications received by admin (Supervisor's application)

Supervisor can view the history of his or her students' applications history. On the other hand, administrators can view the applications history that from all supervisor and students. Then, the lab staff performs the final confirmation on these applications.

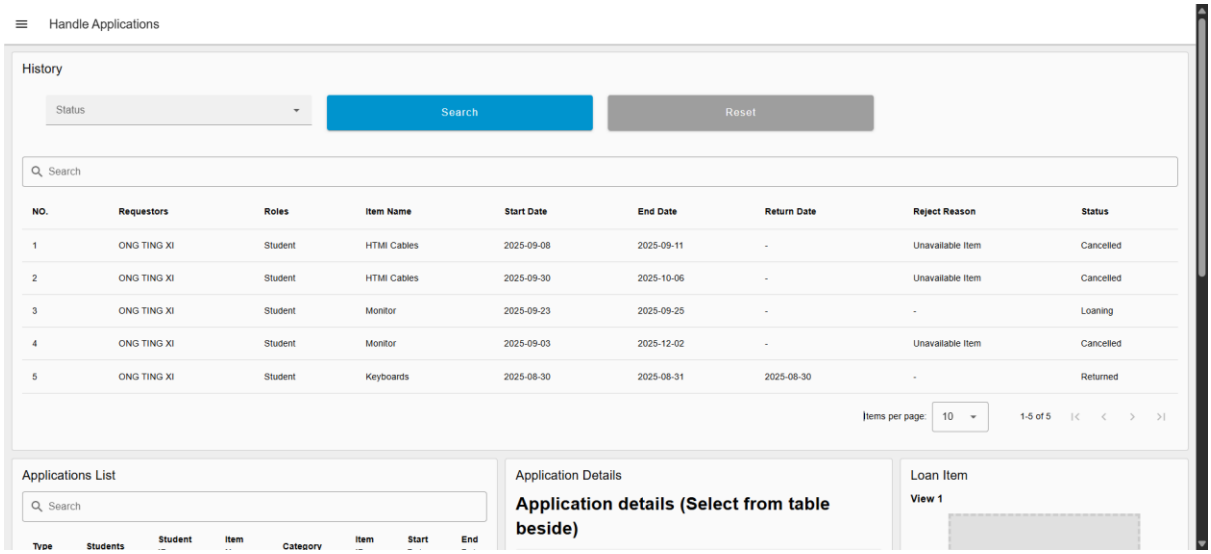


Figure 5.4.141: Interface of handle applications (Supervisor, Part 1)

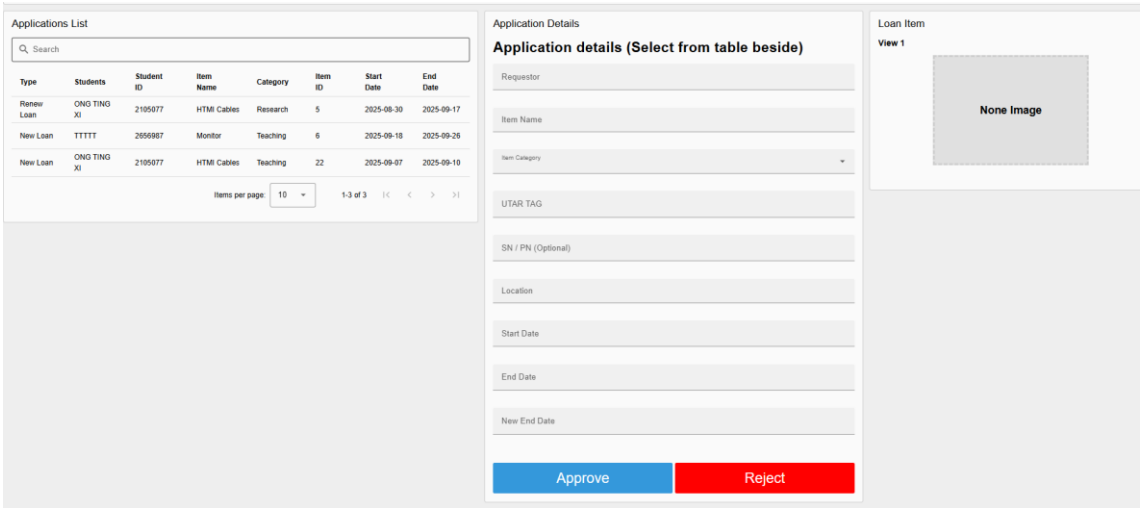


Figure 5.4.142: Interface of handle applications (Supervisor, Part 2)

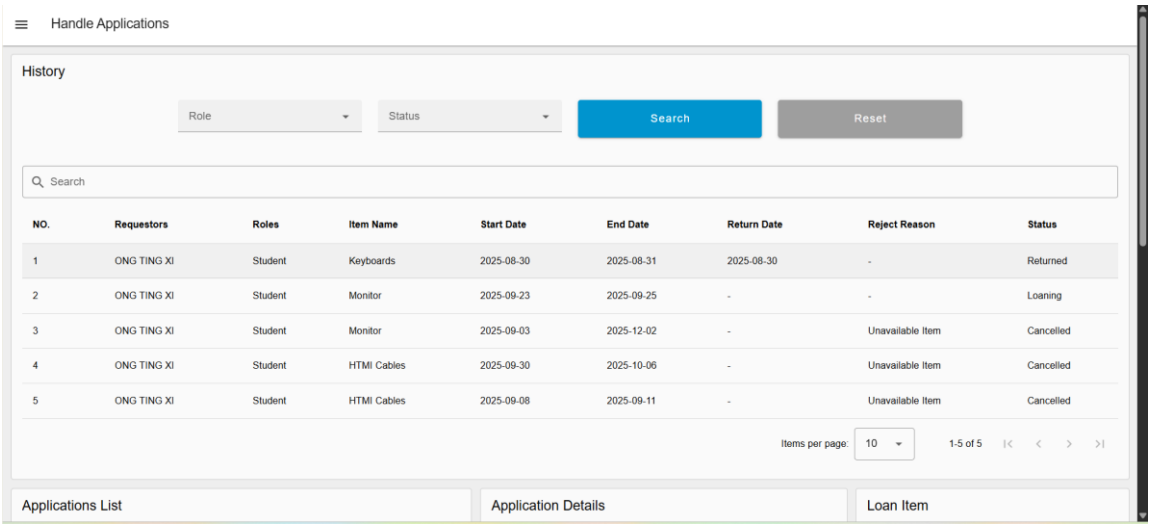


Figure 5.4.143: Interface of handle applications (Administrator, Part 1)

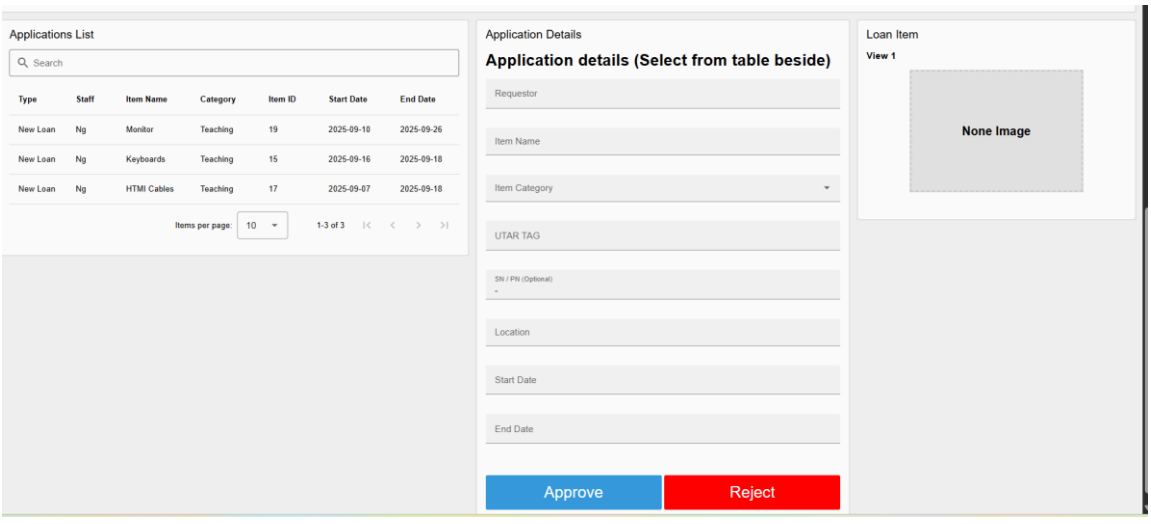


Figure 5.4.144: Interface of handle applications (Administrator, Part 2)

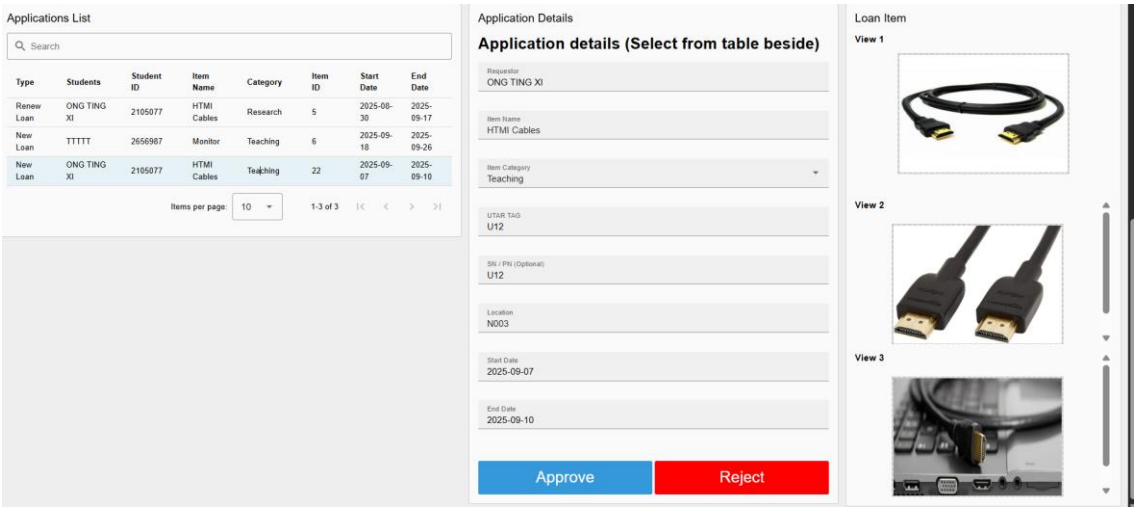


Figure 5.4.145: Select application for handling

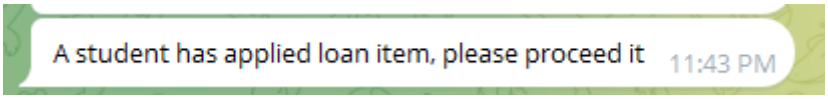


Figure 5.4.146: Notification received by lab staff after approvement from supervisor

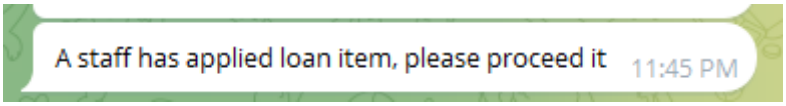


Figure 5.4.147: Notification received by lab staff after approvement from admin

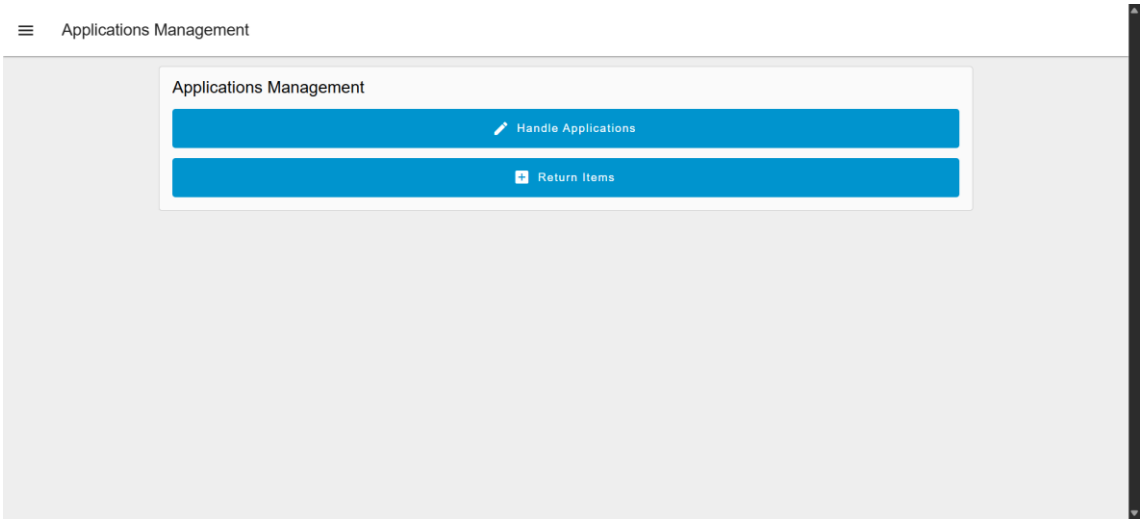


Figure 5.4.148: Main page of Applications Management Modules (Lab Staff only)

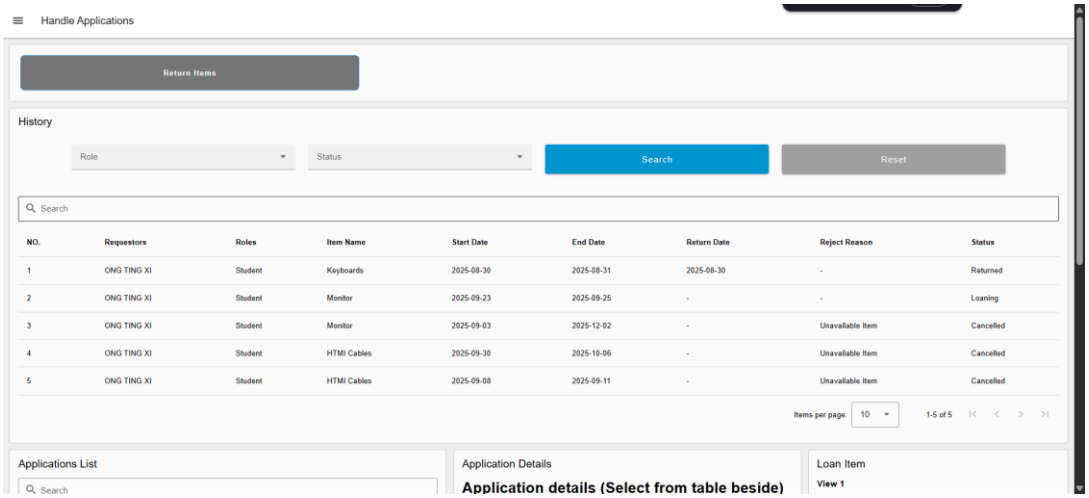


Figure 5.4.149: Interface of handle applications (Lab Staff, Part 1)

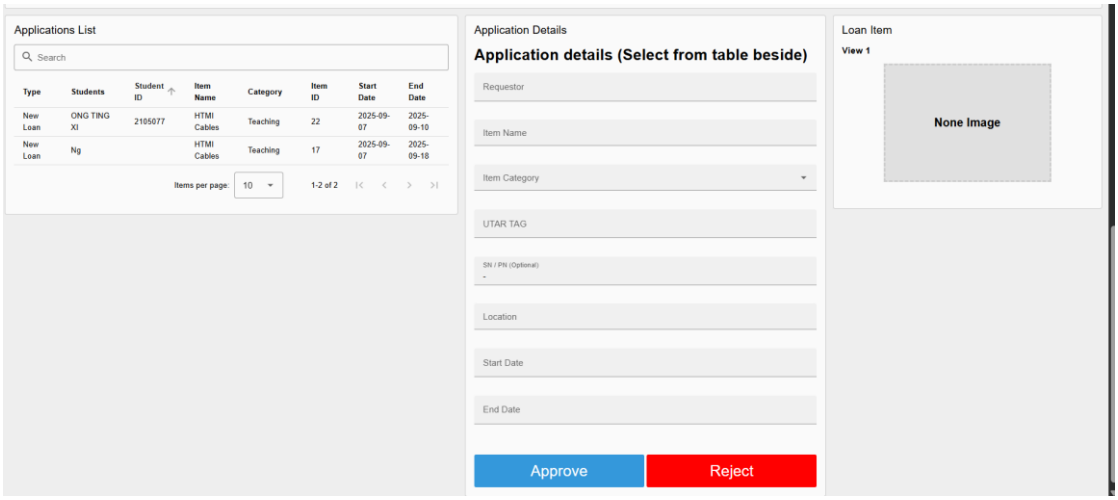


Figure 5.4.150: Interface of handle applications (Lab Staff, Part 2)

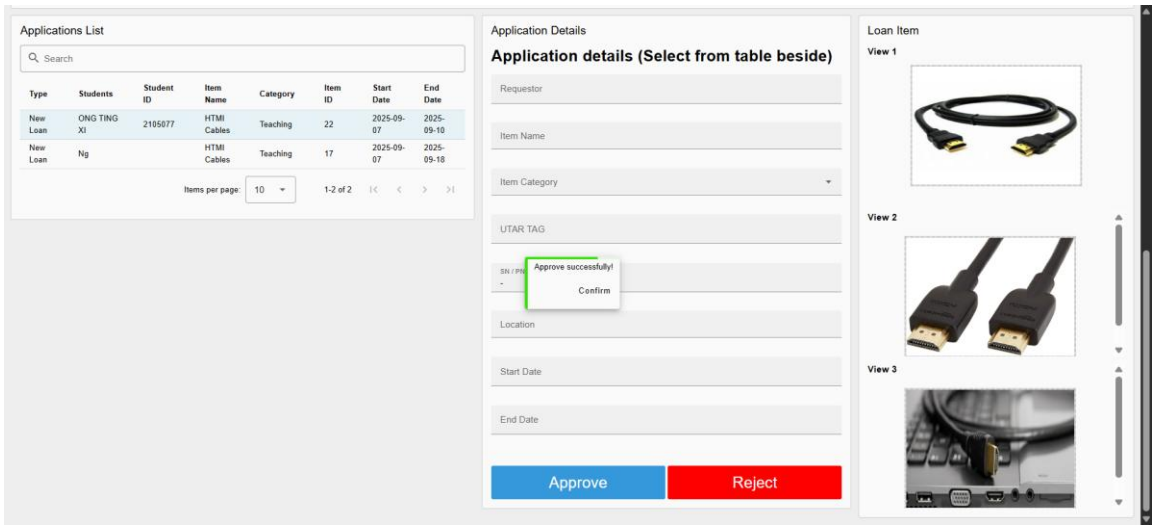


Figure 5.4.151: Approve application

When supervisor, administrator, or lab staff is rejecting the application, they must enter the reason for rejection of that application.

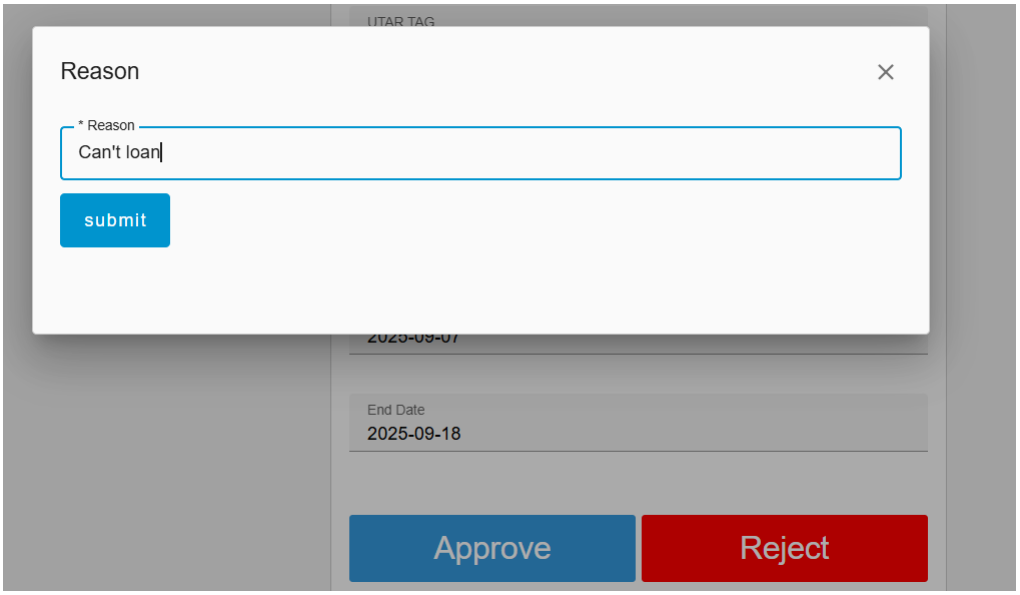


Figure 5.4.152: Rejected application with reason

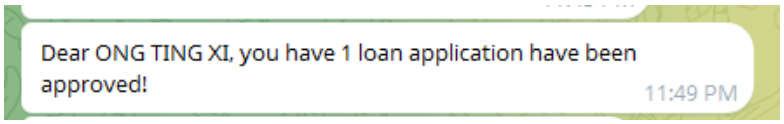


Figure 5.4.153: Notification received by requestor if approved by lab staff

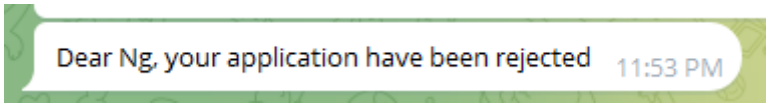


Figure 5.4.154: Notification received by requestor if rejected by lab staff, administrator, or supervisor

The default arrangements of applications history table is based on the user role, and the modified date of the applications in descending order.

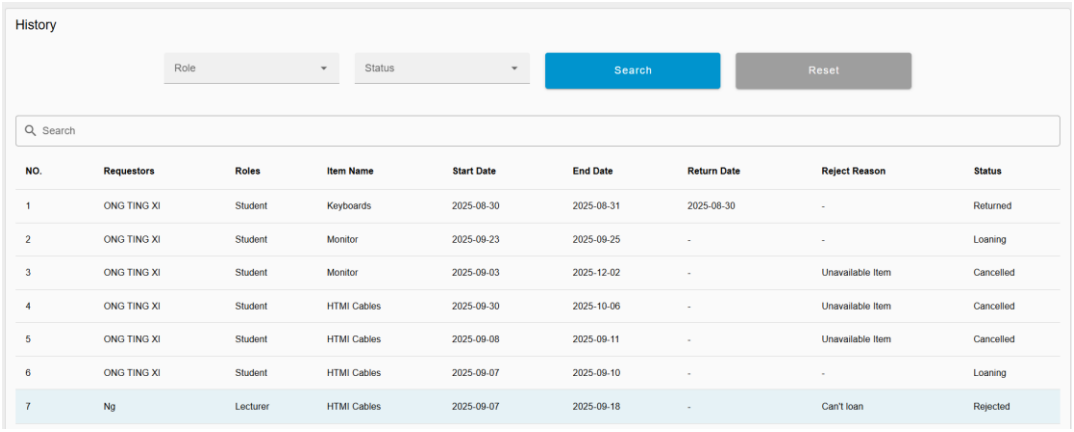


Figure 5.4.155: Updated applications history table (Administrator)

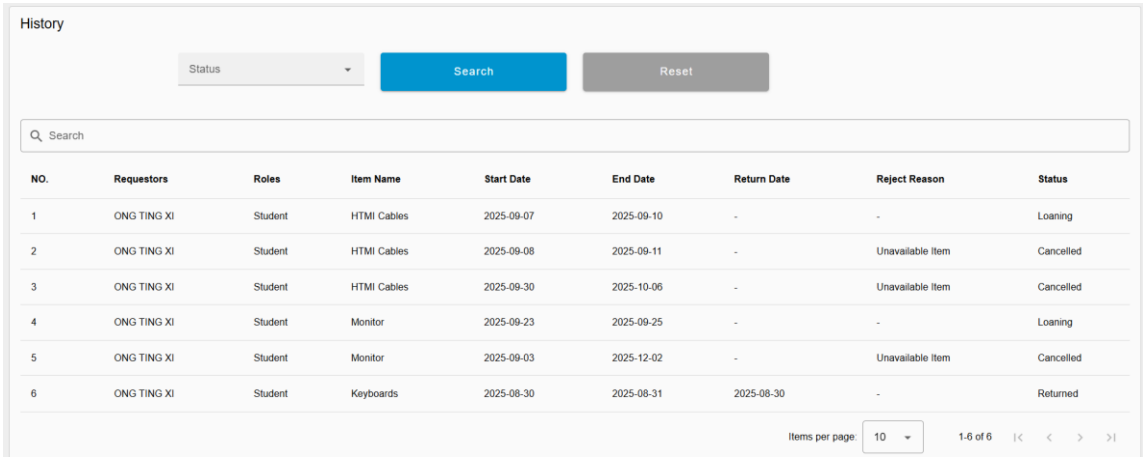


Figure 5.4.156: Updated applications history table (Supervisor)

For renewing loan applications, it includes the rules same as applying loan applications like the maximum loan period and the new end date cannot be passed. Then, it also includes more rules for ensuring the renewing process is following the requirements, which involves can't renew application whose status is other than loaning or overdue, the new end date cannot be earlier than original end date, and each approved application can only be applied renew for maximum 3 times. The limited chances to apply renewing loan period is to prevent the spam attack from the requestors. Moreover, sometimes requestor may select wrong new end date for renewing. Thus, the limited chances allow them to correct the end date and apply for renewing again. Before correcting and applying again, they must let administrator or supervisor to reject their renew loan applications first.

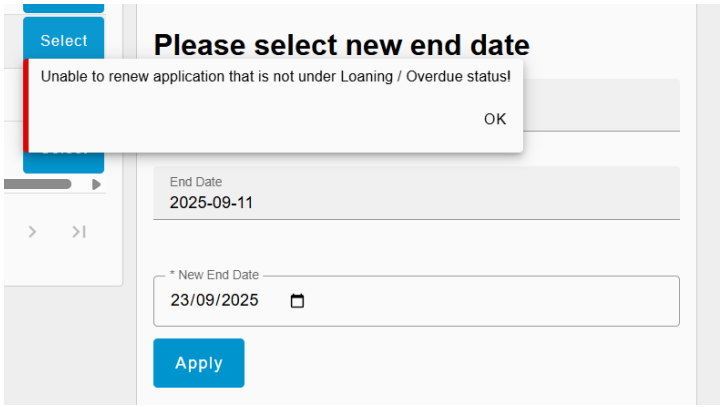


Figure 5.4.157: Renew application that is not under loaning or overdue status

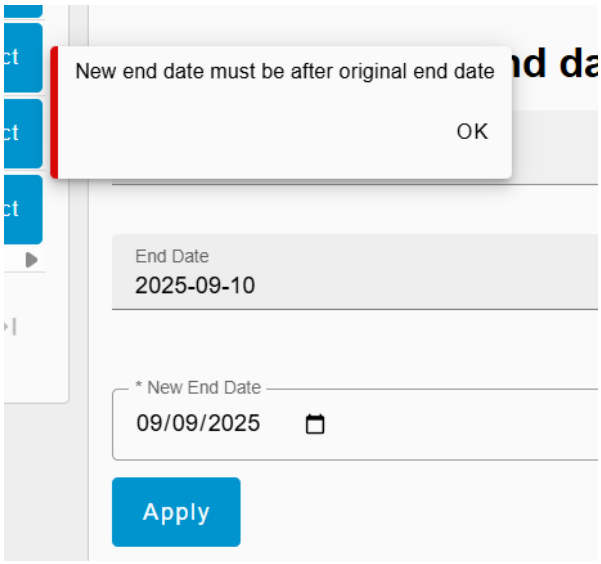


Figure 5.4.158: Select new end date that earlier than original end date

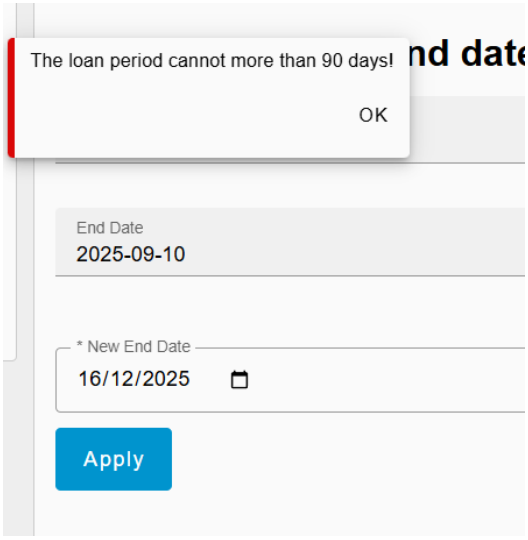


Figure 5.4.159: Select new end date that loan period will more than 90 days

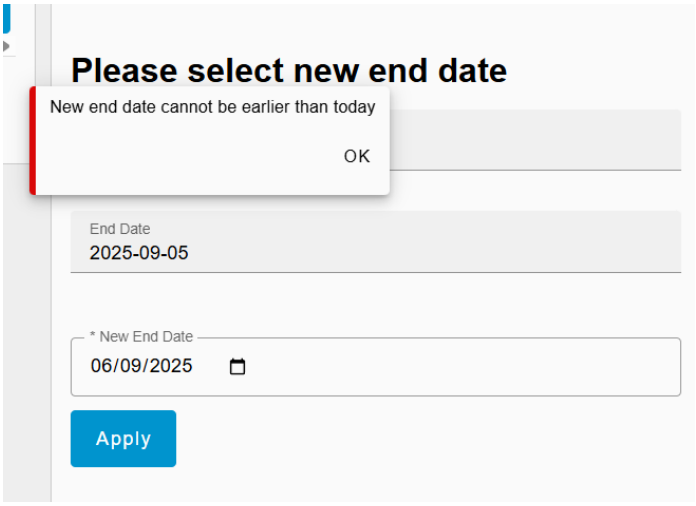


Figure 5.4.160: Select new end date that earlier than today (even though late than original end date)

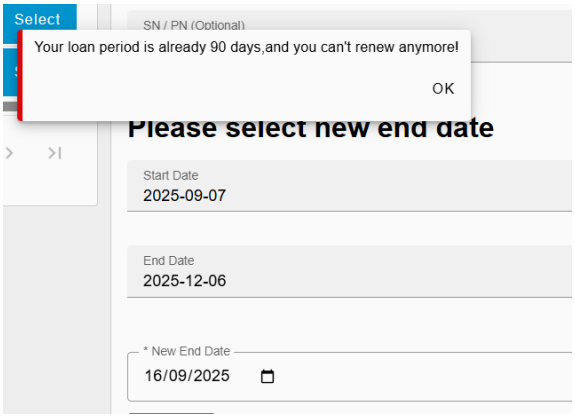


Figure 5.4.161: Renew application which loan period has already achieved maximum loan period

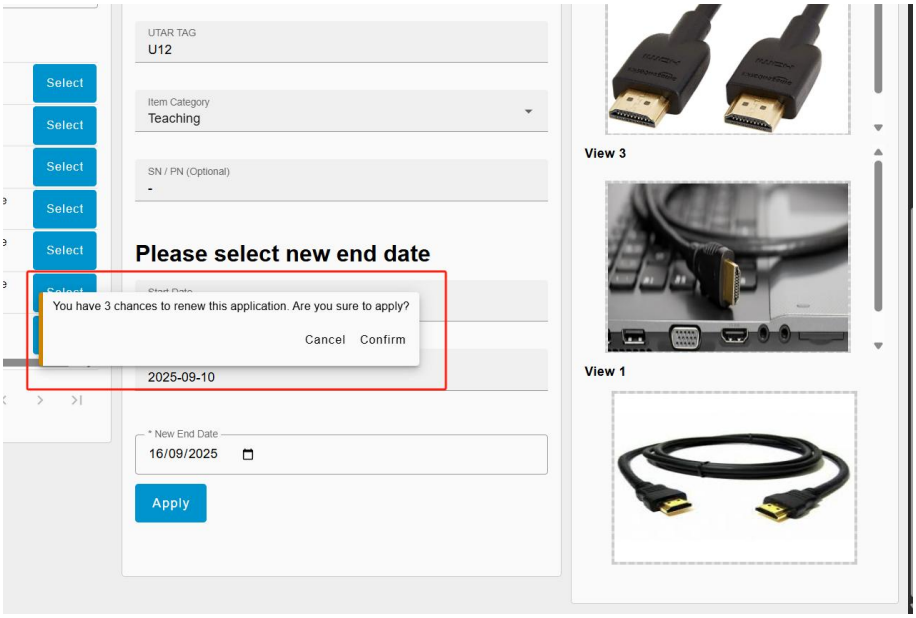


Figure 5.4.162: Select valid new end date and double confirmation (maximum 3 chances to apply renew)

NO.	Item Name	Start Date	End Date	Status	Reject Reason	
1	Keyboards	2025-08-30	2025-08-31	Returned	-	Select
2	HTMI Cables	2025-08-30	2025-09-17	Renewing	-	Select
3	Monitor	2025-09-23	2025-09-25	Loaning	what is 1 + 1?	Select
4	Monitor	2025-09-03	2025-12-02	Cancelled	Unavailable Item	Select
5	HTMI Cables	2025-09-30	2025-10-06	Cancelled	Unavailable Item	Select
6	HTMI Cables	2025-09-08	2025-09-11	Cancelled	Unavailable Item	Select
7	HTMI Cables	2025-09-07	2025-09-10	Renewing	-	Select

Figure 5.4.163: Updated My Application list

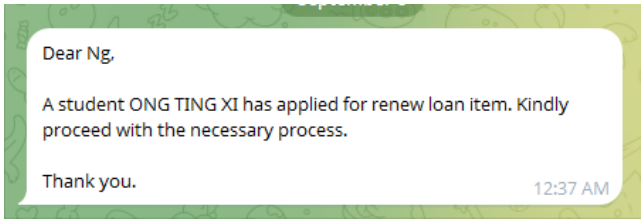


Figure 5.4.164: Notification received by corresponding supervisor (Student's renew application)

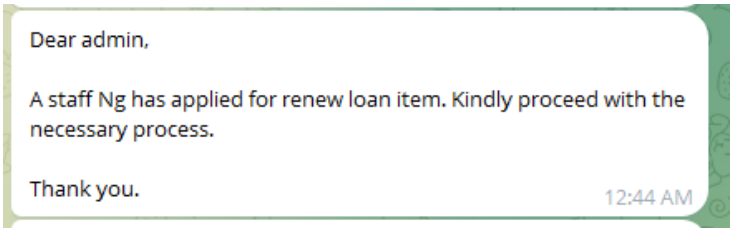


Figure 5.4.165: Notification received by admin (Supervisor's renew application)

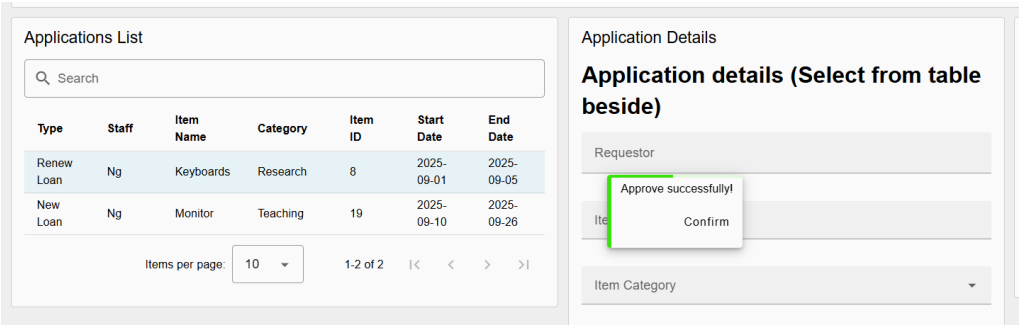


Figure 5.4.166: Approve renew application successfully

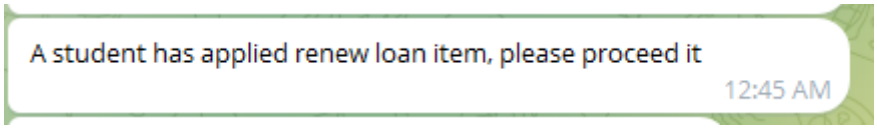


Figure 5.4.167: Notification received by lab staff after approved by supervisor (Student's renew application)

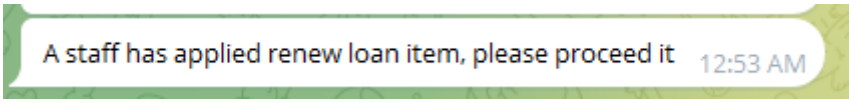


Figure 5.4.168: Notification received by lab staff after approved by administrator (Supervisor's renew application)

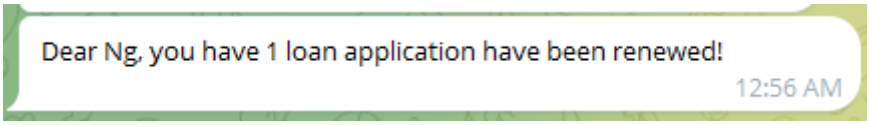


Figure 5.4.169: Notification received by requestor when renew application is approved by lab staff

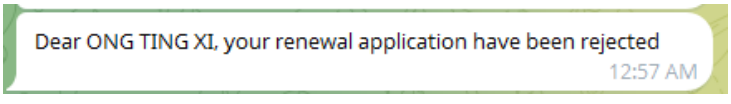


Figure 5.4.170: Notification received by requestor when renew application is rejected by lab staff, administrator, or supervisor

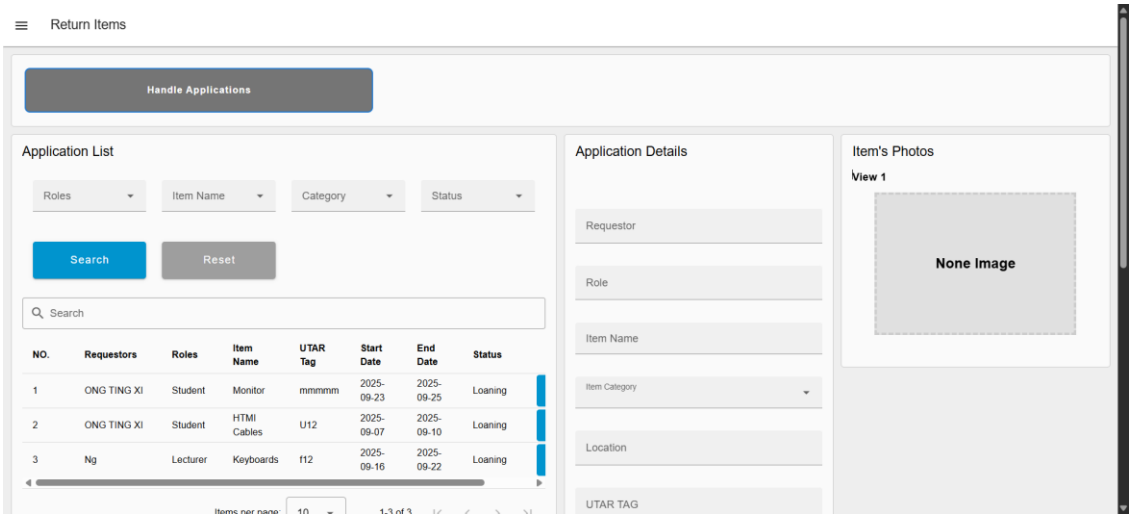


Figure 5.4.171: Interface of return items (Lab Staff only, Part 1)

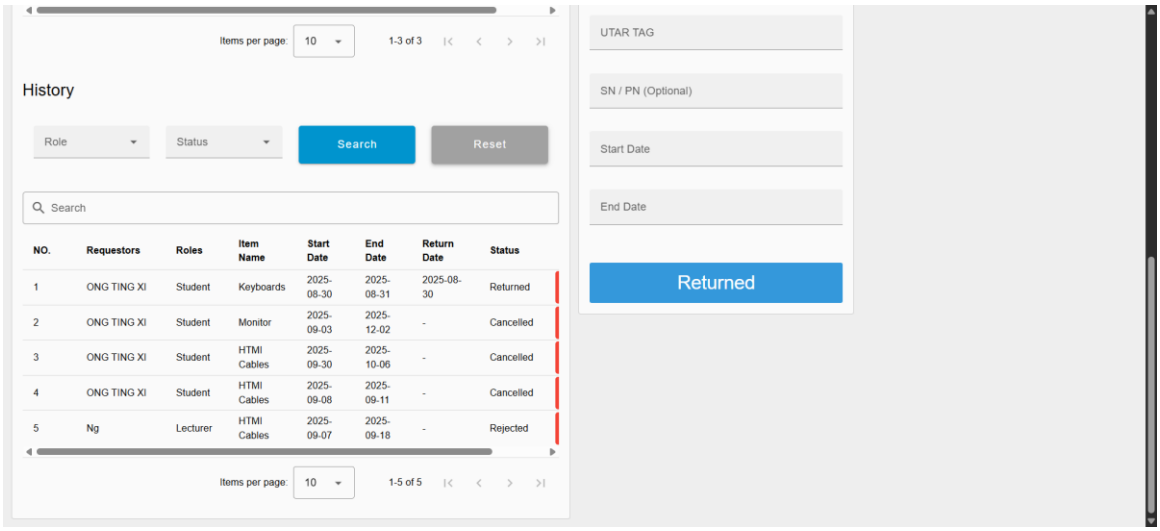


Figure 5.4.172: Interface of return items (Lab Staff only, Part 2)

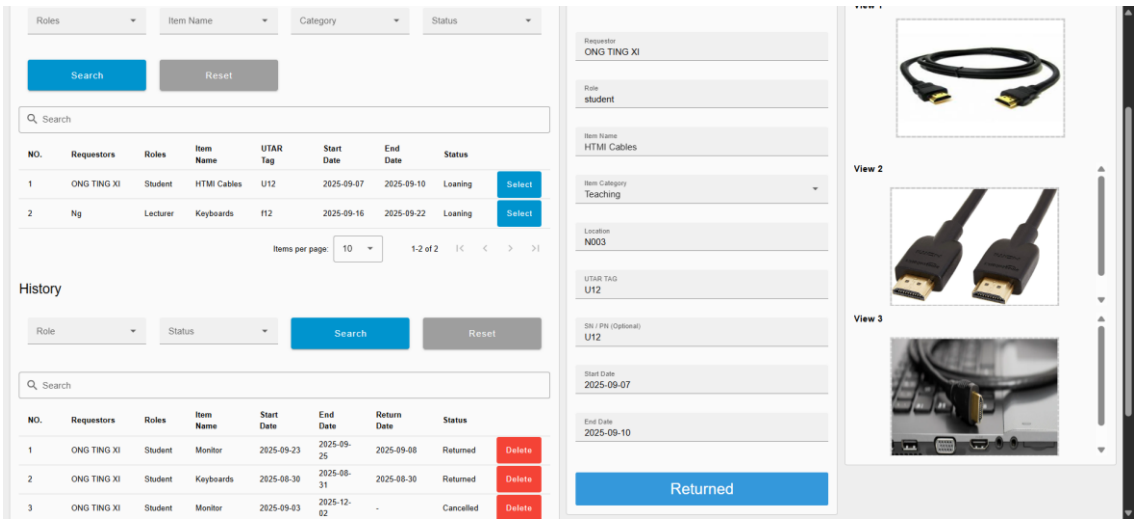


Figure 5.4.173: Select application that has returned item

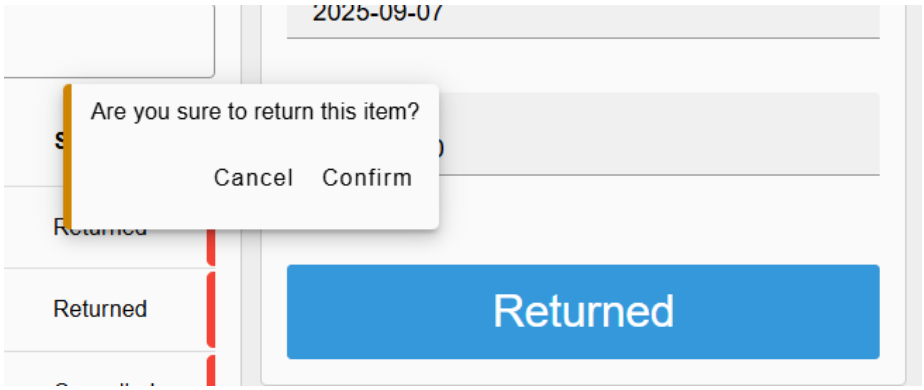
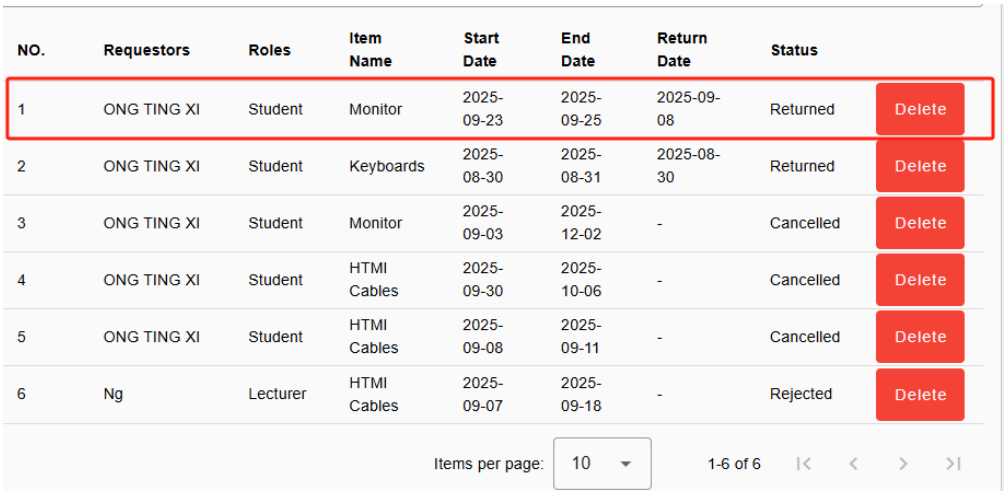


Figure 5.4.174: Double confirm on return item



NO.	Requestors	Roles	Item Name	Start Date	End Date	Return Date	Status	
1	ONG TING XI	Student	Monitor	2025-09-23	2025-09-25	2025-09-08	Returned	Delete
2	ONG TING XI	Student	Keyboards	2025-08-30	2025-08-31	2025-08-30	Returned	Delete
3	ONG TING XI	Student	Monitor	2025-09-03	2025-12-02	-	Cancelled	Delete
4	ONG TING XI	Student	HTMI Cables	2025-09-30	2025-10-06	-	Cancelled	Delete
5	ONG TING XI	Student	HTMI Cables	2025-09-08	2025-09-11	-	Cancelled	Delete
6	Ng	Lecturer	HTMI Cables	2025-09-07	2025-09-18	-	Rejected	Delete

Items per page: 10 1-6 of 6

Figure 5.4.175: Updated applications history list

Lab staff can delete the old application records that are under returned and cancelled status as they are useless. Then, the rejected applications can't be removed because they include the reasons of rejection to the requestors. The rejection reasons can be the lessons to let the requestors to perform valid applications in the future.

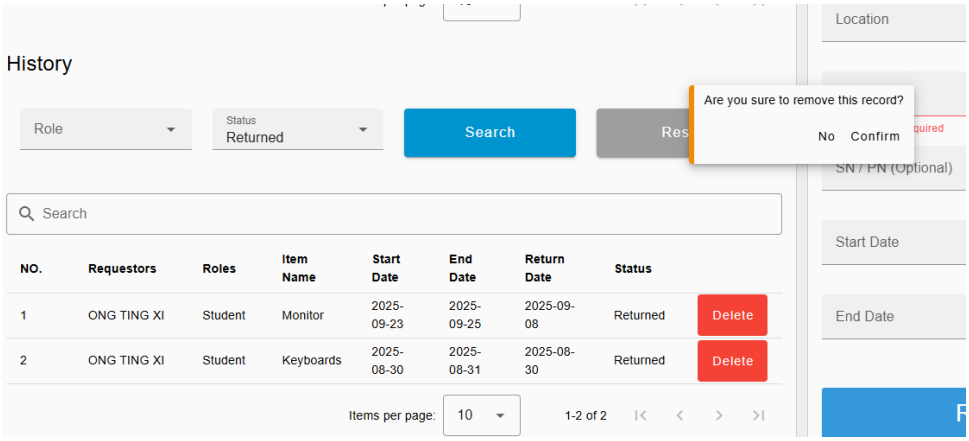


Figure 5.4.176: Delete old record about returned item

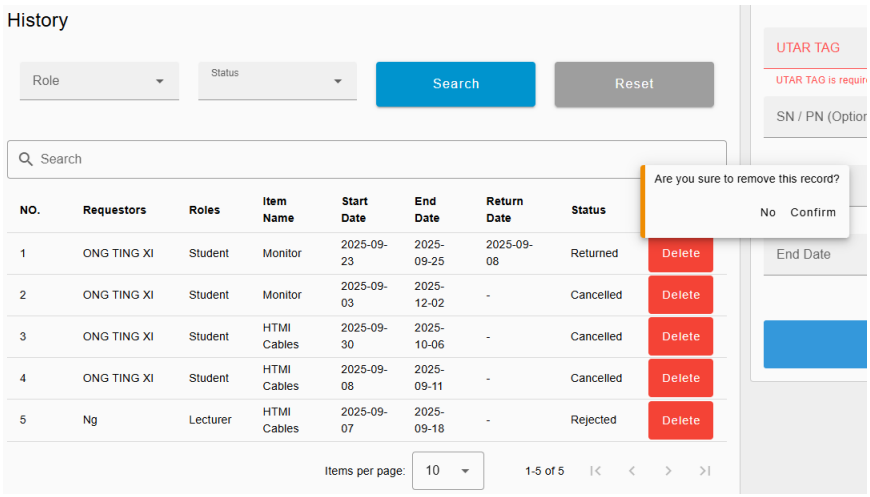


Figure 5.4.177: Delete old record about cancelled application

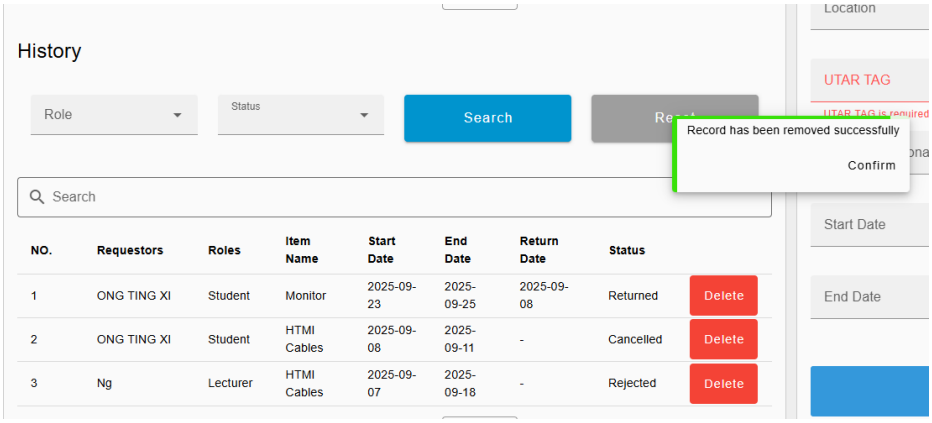


Figure 5.4.178: Delete old record successfully

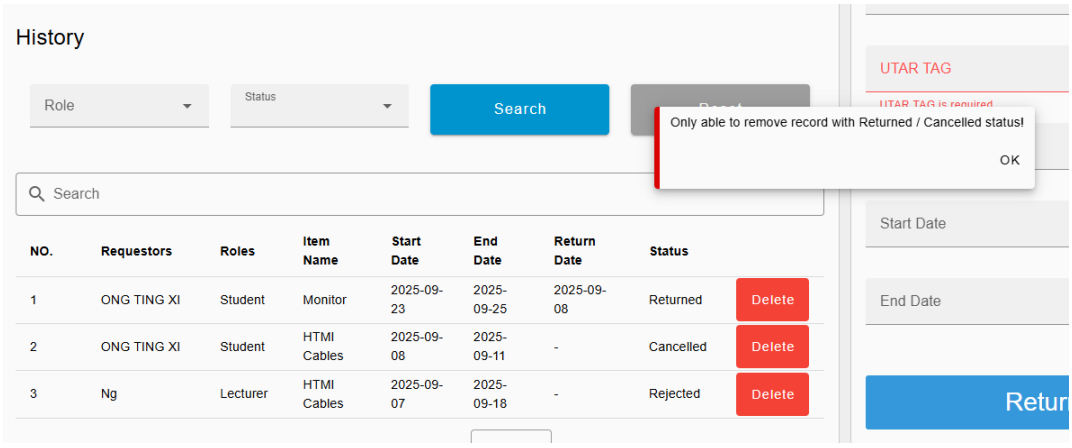


Figure 5.4.179: Delete old record which is other than Returned or Cancelled status

Lastly, the system performs checking on the due date of each application at 8:00 A.M. everyday. Then, it sends notifications to the requestors who have the overdue applications and the applications whose remaining loan periods are less than 4 days.

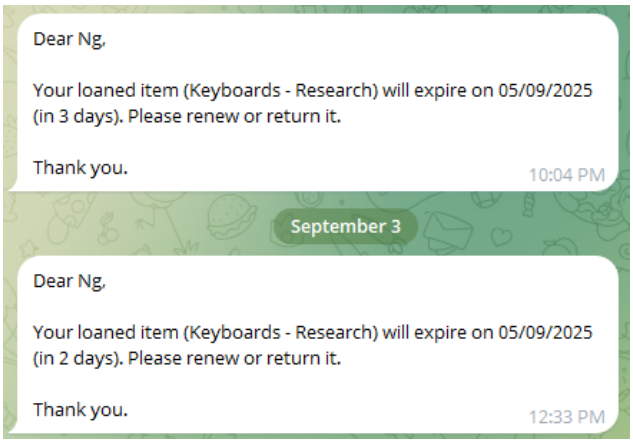


Figure 5.4.180: Automated notifications received by user when the remaining loan period is less than 4 days

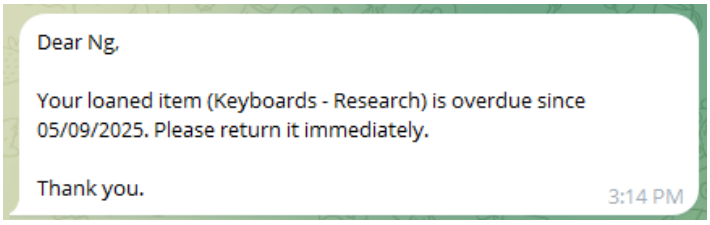


Figure 5.4.181: Automated notification received by users about overdue loaning

5.5 Concluding Remark

This chapter presents the ways to set up hardware and software. Besides that, this chapter also involves how each module performs tasks according to the workflow, prevents invalid input, and sends notification based on specified event. The modules of this system ensure smooth operations from creating users to completing the applications.

Chapter 6 - System Evaluation and Discussion

6.1 System Testing and Performance Metrics

The system testing is carried out through black-box testing methodology where the testers don't know any coding structure and they only know the workflows about managing the items and applications. There are 6 modules within this system being tested with several test cases. The text cases are mainly focusing on input format, completeness of inputs, workflows, access rights, and automated notifications.

Then, the performance of the system is quantified by the test case coverage and pass rate of each module. Each test case was executed and measured by comparing the actual result with the expected result.

6.1.1 User Management Module

Table 6.1.1: Performance Metrics of User Management Module

Metrics	**Description**
Testing method	Black-box testing on: • Sign up application (student) • Log in • Log out • Account registration • Encryption of passwords and Telegram Chat IDs
Number of test cases	26
Past Criteria	Actual results match expected results
Test Case Coverage	Percentage of executed test cases against planned test cases
Pass Rate	Percentage of passed test cases

6.1.2 Inventory Management Module

Table 6.1.2: Performance Metrics of Inventory Management Module

Metrics	**Description**
Testing method	Black-box testing on: • Register items • Update items • Report and remove items

Number of test cases	13
Past Criteria	Actual results match expected results
Test Case Coverage	Percentage of executed test cases against planned test cases
Pass Rate	Percentage of passed test cases

6.1.3 Item Types Management Module

Table 6.1.3: Performance Metrics of Item Types Management Module

Metrics	**Description**
Testing method	Black-box testing on: • Add item types • Edit item types • Delete item types
Number of test cases	8
Past Criteria	Actual results match expected results
Test Case Coverage	Percentage of executed test cases against planned test cases
Pass Rate	Percentage of passed test cases

6.1.4 Item Categories Management Module

Table 6.1.4: Performance Metrics of Item Categories Management Module

Metrics	**Description**
Testing method	Black-box testing on: • Add item categories • Edit item categories • Delete item categories
Number of test cases	8
Past Criteria	Actual results match expected results
Test Case Coverage	Percentage of executed test cases against planned test cases
Pass Rate	Percentage of passed test cases

6.1.5 Location Management Module

Table 6.1.5: Performance Metrics of Location Management Module

Metrics	**Description**
Testing method	Black-box testing on: • Add locations • Edit locations

	• Delete locations
Number of test cases	8
Past Criteria	Actual results match expected results
Test Case Coverage	Percentage of executed test cases against planned test cases
Pass Rate	Percentage of passed test cases

6.1.6 Applications Management Module

Table 6.1.6: Performance Metrics of Applications Management Module

Metrics	**Description**
Testing method	Black-box testing on: • Apply loan item • Renew loan period • Handle applications • Return items • Delete application history records • Automated notifications
Number of test cases	29
Past Criteria	Actual results match expected results
Test Case Coverage	Percentage of executed test cases against planned test cases
Pass Rate	Percentage of passed test cases

6.2 Testing Setup and Result

Each module of this system has several use cases that contain several planned test cases. The actual result of each test case is compared to the expected result for getting the percentage of passed test cases and test case coverages.

The final percentages of passed test cases and test case coverages for each module would be used for analyzation of system performance. The table below shows the use cases that are included in each module:

Table 6.2.1: Use cases of each module

Modules	**Use Cases**	**Total Test Cases**
User Management Module	• 6.2.1 Student Sign Up Application Use Case • 6.2.2 Log In and Log Out Use Case. • 6.2.3 Account Registration Use Case • 6.2.4 Account Deactivation Use Case	26
Inventory Management Module	• 6.2.5 Register Items Use Case • 6.2.6 Update Items Use Case • 6.2.7 Report and Remove Items Use Case	13
Item Types Management Module	• 6.2.8 Manage Item Types Use Case	8
Item Categories Management Module	• 6.2.9 Manage Item Categories Use Case	8
Location Management Module	• 6.2.10 Manage Locations Use Case	8
Applications Management Module	• 6.2.11 Loan Item Applications Use Case • 6.2.12 Student and Supervisor Check and Renew Loan Period Use Case • 6.2.13 Supervisor Handles Student's Applications Use Case • 6.2.14 Admin Handles Student's Applications Use Case • 6.2.15 Lab Staff Handles Loan Applications and Renew Loan Applications Use Case • 6.2.16 Lab Staff Record Returned Items Use Case • 6.2.17 Lab Staff Delete Application History Records Use Case • 6.2.18 Automated Notifications Use Case	29

6.2.1 Student Sign Up Application Use Case

Table 6.2.2: Testing Results of Student Sign Up Application Use Case

Test Cases	Expected Result	Actual Result	Status
Incomplete registration form	Prevent user from submit registration form and alert the user to complete form	Prevent user from submit registration form and alert the user to complete form	Pass
Fill in Telegram Chat ID directly without let lab Telegram bot to send chat ID to student	Unsuccessful submission, and alert users about the verification of Telegram Chat ID is failed.	Unsuccessful submission, and alert users about the verification of Telegram Chat ID is failed.	Pass
Fail to follow requested input formats	Unsuccessful submission and the users get alerts	Unsuccessful submission and the users get alerts	Pass
Fill in all fields of form correctly, including filling Telegram Chat ID which get from UTAR Lab Telegram Bot	• Successful submission of account application form • Database stores password and Telegram Chat ID in encrypted form	• Successful submission of account application form • Database stores password and Telegram Chat ID in encrypted form	Pass

6.2.2 Log In and Log Out Use Case

Table 6.2.3: Testing Results of Log In and Log Out Use Case

Test Cases	Expected Result	Actual Result	Status
Log in activated account with valid email and password	• User log in successfully • Direct user to home page	• User log in successfully • Direct user to home page	Pass
Log in activated account with invalid email, password, or both	Unable log in	Unable log in	Pass
Log in deactivated, rejected, or pending	Unable log in	Unable log in	Pass

account with valid email and password			
Log in deactivated, rejected, and pending accounts with invalid email and password	Unable log in	Unable log in	Pass
Log in as student	Navigation list includes: • Home page • My Applications • Log out	Navigation list includes: • Home page • My Applications • Log out	Pass
Log in as supervisor	Navigation list includes: • Home page • My Applications • Handle Applications • Log out	Navigation list includes: • Home page • My Applications • Handle Applications • Log out	Pass
Log in as lab staff	Navigation list includes: • Home page • Items management • Items Types Management • Categories Management • Users Management • Applications Management • Log Out	Navigation list includes: • Home page • Items management • Items Types Management • Categories Management • Users Management • Applications Management • Log Out	Pass
Log in as administrator	Navigation list includes: • Home page • Items management • Locations Management • Users Management • Handle Applications • Log Out	Navigation list includes: • Home page • Items management • Locations Management • Users Management • Handle Applications Log Out	Pass
Log out	Return to Log In page	Return to Log In Page	Pass

Directly connect any page other than log in and sign up pages without login	Direct user back to log in page	Direct user back to log in page	Pass
User attempt to access unauthorized page	Direct user back to home page	Direct user back to home page	Pass
User closes the webpage	System clears the user session after 2 minutes	System clears the user session after 2 minutes	Pass

6.2.3 Account Registration Use Case

Table 6.2.4: Testing Results of Account Registration Use Case

Test Cases	Expected Result	Actual Result	Status
Approve student account application	• Student receives notification about approval of account application • Student able to log in system	• Student receives notification about approval of account application • Student able to log in system	Pass
Reject student account application	• Student receives notification about rejection of account application • Student unable to log in system	• Student receives notification about rejection of account application • Student unable to log in system	Pass
Staff registers new accounts for others	Only able to register account for other lab staff and supervisor	Only able to register account for other lab staff and supervisor	Pass
Adminstrator registers new accounts for others	Able register account for every role except student	Able register account for every role except student	Pass
Incomplete registration form	• Prevent user from submit registration form • Alert the user to complete form	• Prevent user from submit registration form • Alert the user to complete form	Pass
Fill in Telegram Chat ID directly without let lab	• Unsuccessful submission,	• Unsuccessful submission,	Pass

Telegram bot to send chat ID to account requestor	• Alert users about the verification of Telegram Chat ID is failed.	• Alert users about the verification of Telegram Chat ID is failed.	
Fail to follow requested input formats of registration form	• Unsuccessful submission • Users get alerts	• Unsuccessful submission • Users get alerts	Pass
Fill in all fields of form correctly, including filling Telegram Chat ID which get from UTAR Lab Telegram Bot	• Account is registered successfully • Database stores password and Telegram Chat ID in encrypted form	• Account is registered successfully • Database stores password and Telegram Chat ID in encrypted form	Pass

6.2.4 Account Deactivation Use Case

Table 6.2.5: Testing Results of Account Deactivation Use Case

Test Cases	**Expected Result**	**Actual Result**	**Status**
Deactivate student, supervisor, and admins accounts	• Failed deactivation, if the user is holding any uncompleted cases • Success deactivation, if the user doesn't holding any uncompleted cases • Only staff can deactivate student account • Only admin can deactivate account with each role except student	• Failed deactivation, if the user is holding any uncompleted cases • Success deactivation, if the user doesn't holding any uncompleted cases • Only staff can deactivate student account • Only admin can deactivate account with each role except student	Pass
Deactivate lab staff account	Can be directly deactivated after making double confirmation	Can be directly deactivated after making double confirmation	Pass

6.2.5 Register Items Use Case

Table 6.2.6: Testing Results of Register Items Use Case

Test Cases	Expected Result	Actual Result	Status
Register item with unique UTAR Tag and SN/PN	Register item successfully	Register item successfully	Pass
Register item with duplicate UTAR Tag, SN/PN, or both	• Register item unsuccessfully • System alerts user about invalid inputs	• Register item unsuccessfully • System alerts user about invalid inputs	Pass
Register item without attach photo	Register item successfully	Register item successfully	Pass
Register item with maximum 3 photos	Register item successfully	Register item successfully	Pass
Register item with less than 3 photos that are arranged randomly, includes photos attached in View 1 and View 3 or attached in View 2 and View 3	• Register item successfully • Photos are auto sorted by system when storing them in the database	• Register item successfully • Photos are auto sorted by system when storing them in the database	Pass

6.2.6 Update Items Use Case

Table 6.2.7: Testing Results of Update Items Use Case

Test Cases	Expected Result	Actual Result	Status
Submit update form without selecting any item	• Fail update • Alerts user to select item first	• Fail update • Alerts user to select item first	Pass
Send comment without selecting any item	• Fail update • Alerts user to select item first	• Fail update • Alerts user to select item first	Pass
Update item without changing UTAR Tag, SN/PN, or both	Update item successfully	Update item successfully	Pass

Update item with duplicate UTAR Tag, SN/PN, or both	• Unsuccessful update • Alerts user about invalid inputs	• Unsuccessful update • Alerts user about invalid inputs	Pass
Send comment for selected item	• Comment is sent successfully	• Comment is sent successfully	Pass
Update item that has been fixed	• Item status is changed from maintaining to available • All staff and administrators receive notification about the item status	• Item status is changed from maintaining to available • All staff and administrators receive notification about the item status	Pass

6.2.7 Report and Remove Items Use Case

Table 6.2.8: Testing Results of Report and Remove Items Use Case

Test Cases	**Expected Result**	**Actual Result**	**Status**
Report item that is under maintenance	• Cancel and update status as cancelled for any related pending loan applications • Send notification to the requestors of applications if any • All staff and administrators receive notification about the item status • Item status is changed from available or pending to maintaining • Comment of the item will be updated based on the reason entered by user	• Cancel and update status as cancelled for any related pending loan applications • Send notification to the requestors of applications if any • All staff and administrators receive notification about the item status • Item status is changed from available or pending to maintaining • Comment of the item will be updated based on the reason entered by user	Pass

Report item that can't function anymore	• Cancel and update status as cancelled for any related pending loan applications • Send notification to the requestors of applications if any • All staff and administrators receive notification about the item status • Can report item that under maintaining status • Item status is changed from available, pending, or maintaining to removed. • Removed items will not be shown in item list anymore	• Cancel and update status as cancelled for any related pending loan applications • Send notification to the requestors of applications if any • All staff and administrators receive notification about the item status • Can report item that under maintaining status • Item status is changed from available, pending, or maintaining to removed. • Removed items will not be shown in item list anymore	Pass

6.2.8 Manage Item Types Use Case

Table 6.2.9: Testing Results of Manage Item Types Use Case

Test Cases	Expected Result	Actual Result	Status
Add unique item type	Add item type successfully	Add item type successfully	Pass
Add duplicate item type	• Add item type unsuccessfully • Alerts user to correct the input • Options of item type within item registration form and item	• Add item type unsuccessfully • Alerts user to correct the input • Options of item type within item registration form and item	Pass

	modification form are updated	modification form are updated	
Submit update form without selecting item type	• Update item type unsuccessfully • Alerts user to select item type first	• Update item type unsuccessfully • Alerts user to select item type first	Pass
Update item type without any change	Update item type successfully	Update item type successfully	Pass
Update item type with duplicate item type name	• Update item type unsuccessfully • Alerts user to correct the input	• Update item type unsuccessfully • Alerts user to correct the input	Pass
Update item type with unique item type name	• Update successfully • Options of item type within item registration form and item modification form will be updated	• Update successfully • Options of item type within item registration form and item modification form will be updated	Pass
Delete item type that doesn't used by any item	Delete item type successfully	Delete item type successfully	Pass
Delete item type that has been used by items	• Unable delete item type • Alerts user about invalid deletion	• Unable delete item type • Alerts user about invalid deletion	Pass

6.2.9 Manage Item Categories Use Case

Table 6.2.10: Testing Results of Manage Item Categories Use Case

Test Cases	Expected Result	Actual Result	Status
Add unique item category	Add item category successfully	Add item category successfully	Pass

Add duplicate item category	• Add item category unsuccessfully • Alerts user to correct the input • Options of item category within item registration form and item modification form are updated	• Add item category unsuccessfully • Alerts user to correct the input • Options of item category within item registration form and item modification form are updated	Pass
Submit update form without selecting item category	• Update item category unsuccessfully • Alerts user to select item category first	• Update item category unsuccessfully • Alerts user to select item category first	Pass
Update item category without any change	Update item category successfully	Update item category successfully	Pass
Update item category with duplicate item category name	• Update item category unsuccessfully • Alerts user to correct the input	• Update item category unsuccessfully • Alerts user to correct the input	Pass
Update item category with unique item category name	• Update successfully • Options of item category within item registration form and item modification form will be updated	• Update successfully • Options of item category within item registration form and item modification form will be updated	Pass
Delete item category that doesn't used by any item	Delete item category successfully	Delete item category successfully	Pass
Delete item category that has been used by items	• Unable delete item category • Alerts user about invalid deletion	• Unable delete item category • Alerts user about invalid deletion	Pass

6.2.10 Manage Locations Use Case

Table 6.2.11: Testing Results of Manage Locations Use Case

Test Cases	Expected Result	Actual Result	Status
Add unique location	Add location successfully	Add location successfully	Pass
Add duplicate location	• Add location unsuccessfully • Alerts user to correct the input • Options of location within item registration form and item modification form are updated	• Add location unsuccessfully • Alerts user to correct the input • Options of location within item registration form and item modification form are updated	Pass
Submit update form without selecting location	• Update item location unsuccessfully • Alerts user to select location first	• Update item location unsuccessfully • Alerts user to select location first	Pass
Update location without any change	Update location successfully	Update location successfully	Pass
Update location with duplicate location name	• Update location unsuccessfully • Alerts user to correct the input	• Update location unsuccessfully • Alerts user to correct the input	Pass
Update location with unique location name	• Update successfully • Options of location within item registration form and item modification form will be updated	• Update successfully • Options of location within item registration form and item modification form will be updated	Pass
Delete location that doesn't used by any item	Delete location successfully	Delete location successfully	Pass

Delete location that has been used by items	• Unable delete location • Alerts user about invalid deletion	• Unable delete location • Alerts user about invalid deletion	Pass

6.2.11 Loan Item Applications Use Case

Table 6.2.12: Testing Results of Loan Item Applications Use Case

Test Cases	**Expected Result**	**Actual Result**	**Status**
Student or Supervisor select item	Corresponding item details and photos are shown	Corresponding item details and photos are shown	Pass
Student applying for loan item	• The corresponding supervisor receives relevant notification • Selected item's status is updated to pending	• The corresponding supervisor receives relevant notification • Selected item's status is updated to pending	Pass
Supervisor applying for loan item	• All admins receive relevant notification • Selected item's status is updated to pending	• All admins receive relevant notification • Selected item's status is updated to pending	Pass
Two users apply loaning the same item simultaneously	• One user applies successfully • Another user applies unsuccessfully	• One user applies successfully • Another user applies unsuccessfully	Pass
Apply loaning item with more than 90 days	Unable to apply	Unable to apply	Pass
Apply loaning item with the start date earlier than current date	Unable to apply	Unable to apply	Pass
Apply loaning item with the end date earlier than start date	Unable to apply	Unable to apply	Pass

6.2.12 Student and Supervisor Check and Renew Loan Period Use Case

Table 6.2.13: Testing Results of Student and Supervisor Check and Renew Loan Period Use Case

Test Cases	Expected Result	Actual Result	Status
View own applications	Display all user's own applications with status	Display all user's own applications with status	Pass
Fill new end date without selecting any item	Alerts user to select item first	Alerts user to select item first	Pass
Apply for renewing the application that is not under loaning or overdue status	• Alerts user about invalid application for renewing • Failure to renew application	• Alerts user about invalid application for renewing • Failure to renew application	Pass
Apply for renewing application that is under loaning or overdue status with valid remaining chances and valid new loaning period	• Apply for renewing application successfully • Application's status is updated as renewing • The corresponding supervisor or all admins receive relevant notification	• Apply for renewing application successfully • Application's status is updated as renewing • The corresponding supervisor or all admins receive relevant notification	Pass
Apply for renewing application that is under loaning or overdue status with invalid remaining chances	• Applying for renewing application unsuccessfully • Alerts user about invalid remaining chance	• Applying for renewing application unsuccessfully • Alerts user about invalid remaining chance	Pass
Apply for renewing application that already has 90 days approved loan period	• Applying for renewing application unsuccessfully • Alerts user about unable to extend loan period	• Applying for renewing application unsuccessfully • Alerts user about unable to extend loan period	Pass

Apply for renewing application with new end date that is earlier than original end date	Unable to apply for renewing	Unable to apply for renewing	Pass
Apply new end date that exceeds 90 days loaning period	Unable to apply for renewing	Unable to apply for renewing	Pass
Apply new end date that earlier than today but later than original end date	Unable to apply for renewing	Unable to apply for renewing	Pass

6.2.13 Supervisor Handles Student's Applications Use Case

Table 6.2.14: Testing Results of Supervisor Handles Student's Applications Use Case

Test Cases	**Expected Result**	**Actual Result**	**Status**
Submit form without select applications	Unable submit	Unable submit	Pass
• Select application • Approve application	• Only able to select application of student • Approve successfully • Lab staff receive notification about perform final confirmation	• Only able to select application of student • Approve successfully • Lab Staff receive notification about perform final confirmation	Pass
• Select application • Reject application • Enter reason	• Only able to select application of student • Reject successfully • Requestor receive notification about rejection of application • Requestor's application status is updated as rejected • Update item status as available	• Only able to select application of student • Reject successfully • Requestor receive notification about rejection of application • Requestor's application status is updated as rejected • Update item status as available	Pass

6.2.14 Administrator Handles Supervisor's Applications Use Case

Table 6.2.15: Testing Results of Administrator Handles Supervisor's Applications Use Case

Test Cases	Expected Result	Actual Result	Status
Submit form without select applications	Unable submit	Unable submit	Pass
• Select application • Approve application	• Only able to select application of supervisor • Approve successfully • Lab staff receive notification about perform final confirmation	• Only able to select application of supervisor • Approve successfully • Lab staff receive notification about perform final confirmation	Pass
• Select application • Reject application • Enter reason	• Only able to select application of supervisor • Reject successfully • Requestor receive notification about rejection of application • Requestor's application status is updated as rejected • Update item status as available	• Only able to select application of supervisor • Reject successfully • Requestor receive notification about rejection of application • Requestor's application status is updated as rejected • Update item status as available	Pass

6.2.15 Lab Staff Handles Loan Applications and Renew Loan Applications Use Case

Table 6.2.16: Testing Results of Lab Staff Handles Loan Applications and Renew Loan Applications Use Case

Test Cases	Expected Result	Actual Result	Status
Submit form without select applications	Unable submit	Unable submit	Pass
• Select application • Approve application	• Requestor receives notification • Requestor's application status is updated to loaning	• Requestor receives notification • Requestor's application status is updated to loaning	Pass
• Select application • Reject application	• Requestor receives notification • Requestor's application status is updated as rejected • Update item status as available	• Requestor receives notification • Requestor's application status is updated as rejected • Update item status as available	Pass

6.2.16 Lab Staff Record Returned Items Use Case

Table 6.2.17: Testing Results of Lab Staff Record Returned Items Use Case

Test Cases	Expected Result	Actual Result	Status
• Select application • Submit form	• Requestor's application status is updated to returned • Item's status is updated as available • Applications history list is updated	• Requestor's application status is updated to returned • Item's status is updated as available • Applications history list is updated	Pass

6.2.17 Lab Staff Delete Application History Records Use Case

Table 6.2.18: Testing Results of Lab Staff Delete Application History Records Use Case

Test Cases	Expected Result	Actual Result	Status
Delete application record that is not under returned or cancelled status	Fail deletion of record	Fail deletion of record	Pass
Delete application record with returned or cancelled status	Successful deletion	Successful deletion	Pass

6.2.18 Automated Notifications Use Case

Table 6.2.19: Testing Results of Automated Notifications Use Case

Test Cases	Expected Result	Actual Result	Status
System performs checking on the due date of each application at 8 A.M. every day	• System sends reminders to the requestors whose remaining loan periods are less than 4 days • System sends reminders to the requestors whose applications are overdue	• System sends reminders to the requestors whose remaining loan periods are less than 4 days • System sends reminders to the requestors whose applications are overdue	Pass

6.2.19 Results of testing

Table 6.2.20: Results of testing

Modules	**Total Planned Test Cases**	**Passed Test Cases**	**Test Case Coverage (%)**	**Pass Rate (%)**
User Management Module	26	26	100	100
Inventory Management Module	13	13	100	100
Item Types Management Module	8	8	100	100
Item Categories Management Module	8	8	100	100
Location Management Module	8	8	100	100
Applications Management Module	29	29	100	100

According to the table above, all modules are evaluated based on the planned test cases, and the results show that both test case coverage and pass rate achieve 100%.

This outcome indicates that the lab inventory system meets all the functional requirements across different modules. Besides that, the results also indicate that the system is stable in handling various scenarios with normal usage conditions. In other words, this system has good performance.

6.3 Project Challenges

6.3.1 Security of the system

This lab inventory monitoring system will store and utilize the sensitive information which includes account details, passwords, telegram account, and items details. Therefore, this system must have a relatively standard level of security to ensure data confidentiality, integrity, and availability. It must use powerful and up-to-date encryption and decryption tools to store and utilize the sensitive information during its operations.

6.3.2 Handling concurrent user access

Node-RED is a low-code programming language that is mainly designed for dashboard purposes, which lets users monitor real time data through graph charts and gauges. Thus, its default setting is syncing the screen across all users' devices during the operation.

For handling multiple users' access, the components and coding within the system must be carefully managed. Each node must be able to send the client's information which contains socket ID and socket IP because Node-RED depends on this information to distinguish and isolate individual user sessions.

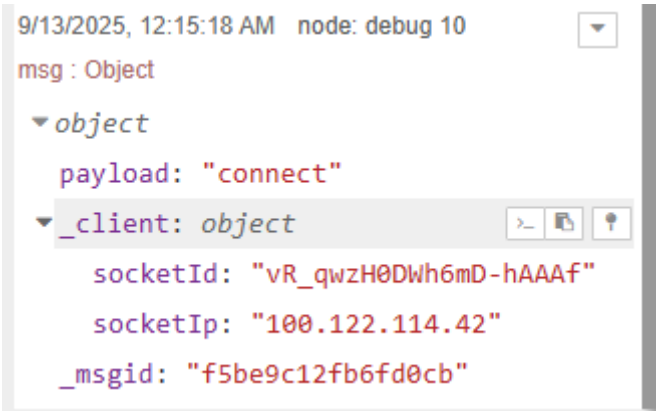


Figure 6.3.1: client's information sent by the node

6.3.3 Exploration of Node-RED

Node-RED provides a wide range of extensions, but it is difficult to find out the extension that suits this system. For achieving this, a lot of effort needs to be taken such as view the associated documentations, visit community, follow up the releases of enhanced versions, view the discussions among the community, and so on.

6.4 Objectives Evaluation

This project includes 3 objectives that are evaluated:

- To provide clear and user-friendly user interfaces and navigations in monitoring inventories
- To minimize the external errors in managing inventory records
- To improve efficiency in handling inventories

6.4.1 Evaluation of objective 1: To provide clear and user-friendly user interfaces and navigations in monitoring inventories

Most of the interfaces are applied with simple designs which include the consistent design on the navigation bar, side bar, and tables. Thus, the users don't need the deep user training to familiarize themselves with this system. During the testing process, the testers were able to perform deep testing without any detailed user manual, and they just need to understand the workflows of inventory management during the system testing. As a result, this objective is considered as achieved.

6.4.2 Evaluation of objective 2: To minimize the external errors in managing inventory records

This system has applied various restrictions, rules, and autocomplete elements for ensuring the users' inputs are valid and not harmful to the system operations and data integrity. For example, the autocomplete fields within the forms, validations of Telegram Chat ID, validations of input formats, validations of registration for items and accounts, validation on loaning items and renewing loan periods, and so on. Moreover, the management of locations, item types, and item categories also ensures that common values for items are entered consistently without external errors. The picture below shows that all the items records are stored in consistent data types.

inv_id	invtype_id	l_id	c_id	inv_utag	inv_spn	inv_status	created_time	updated_time
6	18	63	2	d	d	Available	2025-08-30 18:13:54	2025-09-12 23:09:00
8	10	64	4	xadwq34	3334	Available	2025-08-30 18:18:33	2025-09-12 23:06:09
9	19	64	4	123	4444	Removed	2025-08-30 18:34:11	2025-08-30 19:59:34
10	18	64	2	dwqddddddddd	ddddddddddddddd	Loaning	2025-08-30 18:35:14	2025-09-12 23:31:01
11	18	64	3	aaaaaaaa4	aaaaaaaaa4	Removed	2025-08-30 18:38:14	2025-08-30 20:19:59
12	18	63	2	mmmmm	mmmmm	Available	2025-08-30 18:38:40	2025-09-12 23:04:27
13	18	64	3	111143	11143	Removed	2025-08-30 18:41:59	2025-09-07 21:23:04
14	18	63	4	22224	22224	Removed	2025-08-30 18:44:09	2025-09-07 21:45:50
15	10	63	2	f12	f12	Available	2025-08-30 19:52:06	2025-09-12 23:31:10
17	11	64	2	11a	11a	Available	2025-09-01 21:17:43	2025-09-12 23:05:19
18	10	63	3	22a	22a	Removed	2025-09-01 21:17:43	2025-09-02 14:44:02
19	18	63	2	U123	U123	Available	2025-09-07 20:34:20	2025-09-12 23:35:04
20	11	63	2	U1235	U1235	Removed	2025-09-07 20:38:10	2025-09-07 21:41:48
21	11	63	2	1235a	1235aa	Removed	2025-09-07 20:38:59	2025-09-07 21:44:48
22	11	63	2	U12	U12	Loaning	2025-09-07 20:42:14	2025-09-12 23:20:21

Figure 6.4.1: Items records stored in consistent data types

Therefore, this objective is considered as achieved.

6.4.3 Evaluation of objective 3: To improve efficiency in handling inventories

This system offers functions for reporting, removing, and remarking each item. When the users report or remove the broken item, all lab staff and admins are able to receive the relevant notifications through Telegram. At the same time, they are required to leave a remark on that item to allow other users to track the status of it. The remark section can also be used as a main channel to let users communicate with each other regarding the status of specific item. Additionally, once the item has been repaired, the users are notified accordingly. This ensures convenient reporting and tracking of abnormal items without having to ask others about the processing progress of each abnormal item frequently. Thus, this objective is also considered as achieved.

6.5 Concluding Remark

In conclusion, this system has been evaluated according to the planned test cases and performance metrics, and it has good performance as it has achieved all the functional requirements perfectly. Furthermore, the challenges faced with this project are also stated. Lastly, all the objectives have been achieved at the end of this project.

Chapter 7 - Conclusion and Recommendation

7.1 Conclusion

The utilization of Node-RED for the development of lab inventory monitoring system has demonstrated its effectiveness in providing flexible, scalable, and user-friendly solutions for managing inventory. Through the integration of built-in nodes of Node-Red, various palettes, database, and messaging application, the system successfully addresses the problems that exist in the UTAR FICT lab and other existing inventory management systems. Moreover, this system has fulfilled all the functional requirements through deep testing. Although encountered the challenges during the development, this project has achieved 3 objectives and provides a strong foundation for future improvements.

7.2 Recommendation

For future improvements, the lab inventory management system is recommended to be integrated with AI to enhance automation and decision-making such as schedule job to register new items, delete applications history records, generate report according to the changing user requirements, provide suggestions on the improvement of rules and policies, notify users, and so on.

Moreover, by analyzing the historical data, AI could also be used for predicting the optimal audit frequency and highlighting the items that may require inspection, replacement, or maintenance. This can maximize the span life of all items.

Lastly, this system can also be integrated with an IoT monitoring platform. Through connecting system with IoT sensors, the real-time data can be collected and update the items records automatically which streamline the process of items management.

REFERENCES

[1] S. CNC, V. Sahithi, and S. Yadao, “INVENTORY MANAGEMENT SYSTEM USING MACHINE LEARNING,” *International Journal For Innovative Engineering and Management Research*, vol. 11, no. 06, p. 769, Apr. 2022, doi: https://doi.org/10.48047/ijiemr/v11/i06/51.

[2] Sri Valli Aluguri, Sahithi Maddiveni, Sri Tulasi Gayatri Kandibanda, Bhavana Kotakonda, and M. Anusha, “RFID Based Inventory Management System,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 10, pp. 1363–1369, Oct. 2023, doi: https://doi.org/10.22214/ijraset.2023.56210.

[3] Aditi Kulkarni, *et al.*, “A systematic overview of a inventory processing management developed system (app based),” *international journal of engineering technology and management sciences*, vol. 6, no. 6, pp. 302–305, Nov. 2022, doi: https://doi.org/10.46647/ijetms.2022.v06i06.050.

[4] D. Kurnaedi, E. Oktora, I. Nasrullah, A. Prayoga, and F. A. Pramanto, “Asset and Inventory Data Monitoring Application Design PT. Transmarco,” *bit-Tech : Binary Digital - Technology*, vol. 5, no. 3, pp. 174–181, Apr. 2023, doi: https://doi.org/10.32877/bt.v5i3.731.

[5] S. Umuhani, Solehatin Solehatin, and Sulaibatul Aslamiyah, “Sistem Informasi Inventaris Laboratorium TI SMK Muhammadiyah 6 Rogojampi Berbasis C# dan My Sql,” *Jikom Jurnal Informatika dan Komputer*, vol. 12, no. 2, pp. 11–19, Feb. 2023, doi: https://doi.org/10.55794/jikom.v12i2.83.

[6] K. Talesra and N. G. S., “Low-Code Platform for Application Development,” *International Journal of Applied Engineering Research*, vol. 16, no. 5, p. 346, May 2021, doi: https://doi.org/10.37622/ijaer/16.5.2021.346-351.

[7] G. Juhás, Ľudovít Molnár, A. Juhásová, M. Ondrišová, M. Mladoniczky, and Tomáš Kováčik, “Low-code platforms and languages: the future of software development,” *2022 20th*

International Conference on Emerging eLearning Technologies and Applications (ICETA), pp. 286–293, 2022, doi: https://doi.org/10.1109/iceta57911.2022.9974697.

[8] L. Lyudmila and B. Lily, "DEVELOPING SOFTWARE TO SOLVE CERTAIN PROBLEMS OF INVENTORY MANAGEMENT," *Young Scientist*, vol. 9, no. 109, pp. 133–138, Sep. 2022, doi: https://doi.org/10.32839/2304-5809/2022-9-109-29.

[9] Musafargani Sikkandhar, R. Lim, R. Damalerio, Wei Da Toh, and M.-Y. Cheng, "Conductive Polymer-Based Interactive Shelving System for Real-Time Inventory Management," *Sensors*, vol. 23, no. 21, pp. 8857–8857, Oct. 2023, doi: https://doi.org/10.3390/s23218857.

[10] H. Esrar, H. Zolfaghariania, and H. Yu, "Inventory management practices at a big-box retailer: a case study," *Benchmarking: An International Journal*, vol. 30, no. 7, Jun. 2022, doi: https://doi.org/10.1108/bij-11-2021-0716.

[11] "PostgreSQL: about," *Postgresql.org*. https://www.postgresql.org/about/ (accessed Apr. 28, 2025).

[12]IONOS editorial team, "PostgreSQL: a closer look at the object-relational database management system," *IONOS Digitalguide*, Sep. 14, 2022. https://www.ionos.com/digitalguide/server/know-how/postgresql/ (accessed Apr. 28, 2025).

[13] S. Dhruv, "PostgreSQL Advantages and Disadvantages," *Aalpha*, Jul. 12, 2024. https://www.aalpha.net/blog/pros-and-cons-of-using-postgresql-for-application-development/ (accessed Apr. 28, 2025).

[14] A. S. Gillis, "What is MongoDB? A definition from WhatIs.com," *SearchDataManagement*. https://www.techtarget.com/searchdatamanagement/definition/MongoDB (accessed Apr. 28, 2025).

[15]Blue Claw Database Developer Resource, “MySQL Advantages and Disadvantages,” *Blue Claw Database Developer Resource*. https://blueclawdb.com/mysql/advantages-disadvantages-mysql/ (accessed Apr. 28, 2025).

[16] tapanm-MSFT *et al.*, “What is Power Apps? - Power Apps,” *learn.microsoft.com*, Aug. 07, 2024. https://learn.microsoft.com/en-us/power-apps/powerapps-overview (accessed Apr. 28, 2025).

[17] S. Mitrovic, “Power Apps Language: What Programming Language Does Power Apps Use? | CloudJoy,” *CloudJoy*. https://cloudjoy.com.au/power-apps/programming-language/ (accessed Apr. 28, 2025).

[18] T. Inc, “The Pros & Cons of Power Apps,” *TribalScale*, Nov. 28, 2022. https://medium.com/tribalscale/the-pros-cons-of-power-apps-f450543ce3bc (accessed Apr. 28, 2025).

[19] J. Vargas, “What is Glide used for? Top Glide use cases across industries,” *Lowcode.agency*, Apr. 28, 2024. https://www.lowcode.agency/blog/glide-use-cases (accessed Apr. 28, 2025).

[20] J. Vargas, “The Pros and Cons of Glide,” *Lowcode.agency*, Jun. 03, 2024. https://www.lowcode.agency/blog/glide-advantages-disadvantages (accessed Apr. 28, 2025).

[21] “No Code App Builder for Portals,” *Glideapps.com*. https://www.glideapps.com/use-cases/portals (accessed Apr. 28, 2025).

[22] “About,” *nodered.org*. https://nodered.org/about/ (accessed Apr. 28, 2025).

[23] Pratik M., “Node-RED,” *Zeroqode.com*, Nov. 21, 2024. https://zeroqode.com/no-code-tools/nodered-review/ (accessed Apr. 28, 2025).

[24] “What Is RSA Encryption and How Does It Compare to Other Encryption Methods?,” *Cohesity*. https://www.cohesity.com/glossary/rsa-encryption/ (accessed Sep. 14, 2025).

[25] “RSA encryption,” *Itm8*. https://itm8.com/dictionary/rsa-encryption (accessed Sep. 14, 2025).

[26] Aditi Goel, “What is Blowfish in security? Who uses Blowfish?,” *Encryption Consulting*, Mar. 04, 2024. https://www.encryptionconsulting.com/education-center/what-is-blowfish/ (accessed Sep. 14, 2025).

[27] A. Einorytė, “What is Blowfish encryption, and how does it work?,” *NordVPN*, Jul. 26, 2023. https://nordvpn.com/blog/what-is-blowfish-encryption/?srsltid=AfmBOoqrOrteMT2KC4tA-39YNV3CxYn8REUq1vCgFmHq9MJK3V-t_gfp (accessed Sep. 13, 2025).

[28] “Encryption Methods in Programming Languages,” *MojoAuth*. https://mojoauth.com/encryption-decryption/ (accessed Sep. 13, 2025).

[29] “AES: Advantages and Disadvantages of Advanced Encryption Standard,” *RFWirelessWorld*, 2025. https://www.rfwireless-world.com/terminology/aes-advantages-and-disadvantages (accessed Sep. 14, 2025).

[30] Manimit Haldar, “AES | What is AES? | Encryption Consulting,” *Encryption Consulting*, Mar. 04, 2024. https://www.encryptionconsulting.com/education-center/what-is-aes/ (accessed Sep. 14, 2025).

[31] “AES-256 Encryption : NodeJS,” *Mojoauth*. https://mojoauth.com/encryption-decryption/aes-256-encryption--nodejs/ (accessed Sep. 13, 2025).

[32] Code Signing Store, “Hash Algorithm Comparison: MD5, SHA-1, SHA-2 & SHA-3,” *Code Signing Store*. https://codesigningstore.com/hash-algorithm-comparison (accessed Sep. 14, 2025).

[33] Bella, “MD5: A Closer Look at Its Advantages and Why It Still Matters in Tech,” *Medium*, Mar. 10, 2025. https://medium.com/@kosisochukwuchibuike5/md5-a-closer-look-at-its-advantages-and-why-it-still-matters-in-tech-3b36d8568d29

[34] GeeksforGeeks, “What is the MD5 Algorithm?,” *GeeksforGeeks*, Jul. 23, 2022. https://www.geeksforgeeks.org/computer-networks/what-is-the-md5-algorithm/ (accessed Sep. 14, 2025).

[35] “MD5: The Good, The Bad, and The Ugly,” *heycoach*. https://heycoach.in/blog/md5/ (accessed Sep. 13, 2025).

[36] “Comparing SHA-256 and SHA-3: A Deep Dive into Cryptographic Hashing Algorithms,” *hash.tools*, Aug. 05, 2025. https://www.hash.tools/115/cryptographic-algorithms/13339/comparing-sha-256-and-sha-3-a-deep-dive-into-cryptographic-hashing-algorithms?utm_source=chatgpt.com (accessed Sep. 13, 2025).

[37] “Comparing Cryptographic Hash Functions: SHA-256 vs. SHA-3,” *hash.tools*, Aug. 13, 2025. https://www.hash.tools/120/cybersecurity-trends/13490/comparing-cryptographic-hash-functions-sha-256-vs-sha-3?utm_source=chatgpt.com (accessed Sep. 13, 2025).

[38] ec-admin-ps, “What is SHA- 256?,” *Encryption Consulting*, Mar. 04, 2024. https://www.encryptionconsulting.com/education-center/sha-256/ (accessed Sep. 14, 2025).

[39] J. S Pasaribu, “Development of a Web Based Inventory Information System,” *International Journal of Engineering, Science and Information Technology*, vol. 1, no. 2, pp. 24–31, Mar. 2021, doi: https://doi.org/10.52088/ijesty.v1i2.51.

[40] M. T. Tanaman, J. L. A. Baylosis, B. J. A. Abiles, M. L. P. Catungal, and Dr. P. C. Encarnacion, “Web-based Inventory Management System,” *International journal of science and applied information technology*, vol. 12, no. 5, pp. 44–48, Oct. 2023, doi: https://doi.org/10.30534/ijsait/2023/021252023.

[41] MyAssetTag.com, “AssetTiger: Free Online Asset Management Service,” *AssetTiger.com*. https://www.assettiger.com/ (accessed Dec. 05, 2024).

[42] Odoo, “All your business on one platform. Simple, efficient, yet affordable!,” *odoo*. https://www.odoo.com/ (accessed Apr. 28, 2025).

[43] C. Jayathilaka, "Waterfall Methodology," *Medium*, Jul. 30, 2020. https://medium.com/@chathmini96/waterfall-vs-agile-methodology-28001a9ca487 (accessed Apr. 28, 2025).

[44] C. Jayathilaka, "Agile Methodology," *Medium*, Jul. 30, 2020. https://medium.com/@chathmini96/agile-methodology-30ec4cdf3fc (accessed Apr. 28, 2025).

[45] "Agile Development Models – Software Engineering," *GeeksforGeeks*, Apr. 11, 2025. https://www.geeksforgeeks.org/software-engineering-agile-development-models/ (accessed Apr. 28, 2025).

[46] "Rapid Application Development Model (RAD) – Software Engineering," *GeeksforGeeks*, Sep. 22, 2024. https://www.geeksforgeeks.org/software-engineering-rapid-application-development-model-rad/#objectives-of-rapid-application-development-model-rad (accessed Apr. 28, 2025).

[47] "What is Spiral Model in Software Engineering?," *GeeksforGeeks*, Apr. 11, 2025. https://www.geeksforgeeks.org/software-engineering-spiral-model/ (accessed Apr. 28, 2025).

POSTER

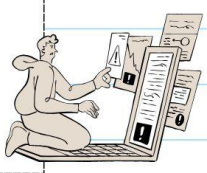
LAB INVENTORY MONITORING SYSTEM USING LOW CODE PROGRAMMING

Introduction

- Utilize low-code programming that are easier to be maintained
- Provide solution for efficient and user-friendly lab inventory monitoring system

Methodology

- Agile Model for frequent feedbacks and updates
- Node-RED for developing web portal
- MySQL for managing data
- Black Box testing for system testing



Results: Objectives Achieved

- ✓ To provide a clear and user-friendly user interfaces and navigations in monitoring inventories
- ✓ To minimize the external errors in managing inventory records
- ✓ To improve efficiency in handling inventories

Conclusion

User Management----------✓
Inventory Management ----------✓
Item Types Management---------✓
Item Categories Management-----✓
Location Management ----------✓
Automated alert and notification --✓

Discussion

This system provides efficient and effective solutions for managing items and loan applications through restrictions, rules, and automated notifications. It also provides high maintainability as it is developed by Node-RED which has less complexity on coding structure.

Ong Ting Xi
Supervisor: Ts Dr Goh Hock Guan

UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information and Communication Technology