

**FACULTY OF COMPUTER SCIENCE AND IT
MASTER OF SCIENCE IN SOFTWARE ENGINEERING**



ADVANCED SOFTWARE ENGINEERING

GROUP PROJECT

PROJECT TITLE

“Inventory System”

Accepted by:

Dr. Igli Hakrama

Presented by:

Xhesjano Halla

Anjeza Xhelilaj

Ema Elezi

Laura Dule

June 2025



TABLE OF CONTENTS

	page
I. INTRODUCTION	1
1.1. Project Overview	1
1.2. Problem Statement	1
1.3. Purpose and Scope of the System	2
II. SYSTEM DESCRIPTION	3
2.1. Product Context	3
2.2. User Characteristics	3
2.3. Assumptions	5
2.3.1. User Assumptions	5
2.3.2. System Assumptions	5
2.4. Constraints	6
2.5. System Dependencies	6
III. REQUIREMENTS	7
3.1. Functional Requirements	7
3.1.1. Item & Product Management	7
3.1.2. Inventory and Stock Management	7
3.1.2.1. Stock Management	7
3.1.2.2. Stock Adjustment	7
3.1.2.3. Stock Valuation	8
3.1.2.4. Multi-Location Support	8
3.1.3. Order & Warehouse Management	8
3.1.3.1. Order Management	8
3.1.3.2. Purchase Order Approval	8
3.1.3.3. Warehouse Management	8
3.1.4. Reporting & Analytics	8
3.1.4.1. Reporting	8
3.1.4.2. Analytics & Business Intelligence	9
3.1.4.3. Customizable Dashboards	9
3.1.5. User & Access Management	9
3.1.5.1. User Management	9
3.1.5.2. Customizable Workflows	9
3.1.6. Mobile and Real – Time Capabilities	9



3.1.6.1. Mobile Accessibility	9
3.1.6.2. Real-Time Updates	9
3.1.7. Barcode, Scanning & Alerts	10
3.1.7.1. Barcode and Scanning	10
3.1.7.2. Automated Alerts	10
3.1.8. Data Integration & Security	10
3.1.8.1. Data Security	10
3.1.8.2. Data Import / Export	10
3.1.9. Microsoft Single Sign-On (SSO) Integration	10
3.2. Non – Functional Requirements	11
3.2.1. Product Requirements	11
3.2.1.1. User Interface Requirements	11
3.2.1.2. Usability	13
3.2.1.3. Efficiency	14
3.2.1.3.1. Performance Requirements	14
3.2.1.3.2. Space Requirements	15
3.2.1.4. Dependability Requirements	15
3.2.1.4.1. Availability	15
3.2.1.4.2. Latency	16
3.2.1.4.3. Monitoring	16
3.2.1.4.4. Maintenance	16
3.2.1.4.5. Integrity	17
3.2.1.5. Security	17
3.2.2. Organizational Requirements	18
3.2.2.1. Environmental Requirements	18
3.2.2.2. Operational Requirements	18
3.2.2.3. Development Requirement	19
3.2.3. External Requirements	20
3.2.3.1. Regulatory Requirement	20
3.2.3.2. Ethical Requirement	20
3.2.3.3. Legislative Requirement	21
3.2.3.3.1. Accounting Requirement	22
3.2.3.3.2. Security Requirement	22
3.3. Domain Requirements	22
IV. SOFTWARE DESIGN	23
4.1. BPMN Diagrams	23
4.1.1. BPMN – Inventory Request and Maintenance Workflow	23



4.1.2. BPMN – Account Provisioning and Login Process	24
4.1.3. BPMN – Location Management Workflow	24
4.2. Use Case Descriptions	25
4.2.1. Use Case – Check Stock	25
4.2.2. Use Case – Edit Location	26
4.2.3. Use Case – Manage Users	29
4.2.4. Use Case – Process Requests	31
4.2.5. Use Case – Process Stock Request (Finance)	34
4.2.6. Use Case – Submit QA Request	36
4.2.7. Use Case – Submit Stock Request	39
4.2.8. Use Case – View and Update Item State	41
4.3. Use Case Diagrams	44
4.3.1. Use Case Diagram – Club Coordinator	44
4.3.2. Use Case Diagram – Finance (Admin)	45
4.3.3. Use Case Diagram – IT	46
4.3.4. Use Case Diagram – Service Department	47
4.3.5. Use Case Diagram – University Staff	48
4.4. Activity Diagrams	49
4.4.1. Activity Diagram for “Login”	49
4.4.2. Activity Diagram for “Change Password”	50
4.4.3. Activity Diagram for “Add or Delete Stock”	51
4.4.4. Activity Diagram for “Request Item from IT”	52
4.4.5. Activity Diagram for “Request Item from Service Department”	52
4.4.6. Activity Diagram for “Quality Check”	53
4.5. State Diagrams	53
4.5.1. State Diagram – Club Coordinator Session Flow	53
4.5.2. State Diagram – Finance Session Flow	55
4.5.3. State Diagram – IT Session Flow	55
4.5.4. State Diagram – Service Department Session Flow	57
4.5.5. State Diagram – University Staff Session Flow	58
4.6. Data Flow Diagrams	59
4.6.1. Level 0	59
4.6.2. Level 1	59
4.6.3. Level 2	60
V. DESIGN PATTERNS	61
5.1. Pattern 1: Model-View-Controller	61
5.2. Pattern 2: Repository Pattern	61
5.3. Pattern 3: Observer Pattern	61
5.4. Pattern 4: Factory Pattern	62



VI. STRUCTURAL AND BEHAVIORAL MODELING	63
6.1. Class Diagram	63
6.2. Object Diagram	63
6.3. Package Diagram	64
6.4. Sequence Diagrams	65
6.4.1. Add and Edit User – Finance	65
6.4.2. Add Item – Finance	65
6.4.3. Delete Item – Finance	66
6.4.4. Delete User – Finance	66
6.4.5. Logout	67
6.4.6. View Inventory – Finance	67
6.4.7. QA Request	68
6.4.8. Stock Request	69
6.4.9. View Inventory - University Staff / Club Coordinator	70
6.4.10. User Login	71
6.5. Collaboration Diagrams	71
6.5.1. Add User – Finance	71
6.5.2. Submit New Stock Request	72
6.5.3. Submit QA Request – University Staff	72
6.5.4. User Role Management and Deletion by Finance	73
6.5.5. User Login	73
6.5.6. Submit New Stock Request – University Staff / Club Coordinator	74
6.5.7. Requesters	74
6.6. Interaction Overview Diagrams (IOD)	74
6.7. Component Diagram	75
6.8. Deployment Diagram	76
VII. DATA DESIGN	77
7.1. Entity Relationship Diagram (ERD)	77
7.2. Relational Schema (RS)	78
Appendix A: User Stories	79
A.1. University Staff	79
A.2 Club Coordinator	81
A.3 IT Department	81
A.4 Finance Department	82



Appendix B: UI Sketches and Concept Designs	83
B.1 Shared Pages	83
B.2 Role-Based UI Sketches	84
B.2.1 Finance Department	84
B.2.2 IT Department	89
B.2.3 Service Department	91
Appendix C: UI Mockups (Figma Screens)	92
C.1 Shared User Interfaces	92
C.2 Finance Role – UI Mockups	94
C.3 IT Department – UI Mockups	103
C.4 Service Department – UI Mockups	107
C.5 University Staff – UI Mockups	110
C.6 Club Coordinator – UI Mockups	112

TABLE OF TABLES

Table 4.2.1: Fully Developed Use Case Description for “Checking Stock”

Table 4.2.2: Fully Developed Use Case Description for “Editing Location”

Table 4.2.3: Fully Developed Use Case Description for “Managing Users”

Table 4.2.4: Fully Developed Use Case Description for “Processing Requests”

Table 4.2.5: Fully Developed Use Case Description for “Processing Stock Request by Finance”

Table 4.2.6: Fully Developed Use Case Description for “Submitting QA Request”

Table 4.2.7: Fully Developed Use Case Description for “Submitting Stock Request”

Table 4.2.8: Fully Developed Use Case Description for “Viewing and updating Item State”

TABLE OF FIGURES

Figure 4.1.1: BPMN for Inventory Request and Resolution across University Departments

Figure 4.1.2: BPMN Diagram for User Credential Request and Login Workflow

Figure 4.1.3: Business Process Model for Managing and Updating Item Location Information

Figure 4.3.1: Use Case Diagram for Club Coordinator Request Handling

Figure 4.3.2: Use Case Diagram for Finance

Figure 4.3.3: Use Case Diagram for IT

Figure 4.3.4: Use Case Diagram for Service Department

Figure 4.3.5: Use Case Diagram for University Staff

Figure 4.4.1: Activity Diagram for “Login”

Figure 4.4.2: Activity Diagram for “Change Password”

Figure 4.4.3: Activity Diagram for “Add or Delete Stock”

Figure 4.4.4: Activity Diagram for “Request Item from IT”

Figure 4.4.5: Activity Diagram for “Request Item from Service Department”

Figure 4.4.6: Activity Diagram for “Quality Check”

Figure 4.5.1: State transitions for the Club Coordinator role

Figure 4.5.2: State transitions for the Finance role

Figure 4.5.3: State Transitions for the IT Role

Figure 4.5.4: State Transitions for the Service Department Role

Figure 4.5.5: State Transitions for the University Staff Role

Figure 4.6.1: Level 0 Data Flow Diagram for UMT Inventory System

Figure 4.6.2: Level 1 Data Flow Diagram for UMT Inventory System

Figure 4.6.3: Level 2 Data Flow Diagram for UMT Inventory System

Figure 6.1 Class Diagram

Figure 6.3 Package Diagram

Figure 6.4.1 Sequence Diagram – Add and Edit User (Finance Role)

Figure 6.4.2 Sequence Diagram – Add Item (Finance Role)

Figure 6.4.3 Sequence Diagram – Delete Item (Finance Role)

Figure 6.4.4 Sequence Diagram – Delete User (Finance Role)

Figure 6.4.5 Sequence Diagram – Logout

Figure 6.4.6 Sequence Diagram – View Inventory (Finance Role)

Figure 6.4.7 Sequence Diagram – QA Request

Figure 6.4.8 Sequence Diagram – Stock Request

Figure 6.4.9 Sequence Diagram – View Inventory (University Staff/Club Coordinator)

Figure 6.4.10 Sequence Diagram – User Login

Figure 6.5.1 Collaboration Diagram – Add User (Finance)

Figure 6.5.2 Collaboration Diagram – Submit Stock Request

Figure 6.5.3 Collaboration Diagram – Submit QA Request (University Staff)

Figure 6.5.4 Collaboration Diagram – Edit Role / Delete User (Finance Role)

Figure 6.5.5 Collaboration Diagram – User Login

Figure 6.5.6 Collaboration Diagram – New Stock Request

Figure 6.5.7 Collaboration Diagram – Requesters

Figure 6.7 Component Diagram

Figure 6.8 Deployment Diagram



Figure 7.1: Entity Relationship Diagram

Figure 7.2: Relational Schema

Figure B.1.1: “Landing Page”

Figure B.1.2: “Login Page”

Figure B.2.1.1: “Finance Homepage – Version A”

Figure B.2.1.2: “Finance Homepage – Version B”

Figure B.2.1.3: “Finance Check Stock Page”

Figure B.2.1.4: “Finance Edit Location Page – Version A”

Figure B.2.1.5: “Finance Edit Location Page – Version B”

Figure B.2.1.6: “Finance Manage Users Page”

Figure B.2.2.1: “IT – Home Page: Pending Requests View”

Figure B.2.2.2: “IT – Home Page: Requests In-Progress View”

Figure B.2.2.3: “IT – Check Stock Page”

Figure B.2.2.4: “IT – Stock Request Page”

Figure B.2.3.1: “Service Department – Home Page: Pending Requests View”

Figure B.2.3.2: “Service Department – Home Page: Request In-Progress View”

Figure C.1.1: “Landing Page”

Figure C.1.2: “Login Page”

Figure C.2.1: “Finance Homepage – Version 1”

Figure C.2.2: “Finance Homepage – Version 2”

Figure C.2.3: “Finance Check Stock Page – Version 1”

Figure C.2.4: “Finance Check Stock Page – Version 2”

Figure C.2.5: “Finance Edit Location Page – Version 1”

Figure C.2.6: “Finance Edit Location Page – Version 2”

Figure C.2.7: “Finance Manage Users Page”

Figure C.3.1: “IT Department – Home Page”

Figure C.3.2: “IT Department – Check Stock Page”

Figure C.3.3: “IT Department – Stock Request Page”

Figure C.4.1: “Service Department – Home Page”

Figure C.4.2: “Service Department – Check Stock Page”

Figure C.4.3: “Service Department – Stock Request Page”

Figure C.5.1: “University Staff – QC Request Modal on Homepage”

Figure C.5.2: “University Staff – New Item Request Modal on Homepage”

Figure C.6.1: “Club Coordinator – Homepage”

Figure C.6.2: “Club Coordinator – Stock Request Page”



I. INTRODUCTION

1.1. Project Overview

Managing university inventory across classrooms, offices, storage rooms, and laboratories is a complex and error-prone process when handled manually. Recording assets on paper can lead to data inconsistency, delays, and misplaced or forgotten stock, ultimately impacting operational efficiency and budgeting accuracy.

To address these challenges, this project proposes the development of a web-based **University Inventory Management System (UIMS)**. This system is designed to track and manage university stock in real time, offering structured categorization of items and enabling automated tracking and reporting. The web application ensures that all inventory data is securely stored on a protected server and allows flexible updates as the institution's needs evolve.

1.2. Problem Statement

Universities manage a wide variety of physical assets, from classroom equipment to IT devices and office supplies. In many cases, this inventory is tracked using outdated manual methods like spreadsheets, printed lists, or verbal communication between departments. These approaches are not only inefficient but also highly prone to errors such as misplaced items, outdated records, and duplicate requests.

At the institutional level, this lack of a centralized system makes it difficult to monitor what is available, what needs to be replaced, or what condition certain assets are in. Delays in restocking, unnoticed equipment failures, and lack of accountability in item handling can result in unnecessary costs and disruption of services.

The problem becomes even more challenging when different departments — such as IT, Service, and Finance — operate independently without a shared platform for tracking and managing inventory. This disconnect leads to miscommunication, redundant processes, and a lack of visibility over shared resources.

Without an integrated system, it is difficult to:

- Keep real-time records of inventory across multiple locations (e.g., classrooms, storage rooms, and offices)
- Ensure that stock requests and approvals follow a clear workflow
- Alert departments when maintenance is needed or inventory is running low
- Maintain a complete audit trail of user actions and inventory status updates

1.3. Purpose and Scope of the System

The main goal of this system is to help the university manage its stock more efficiently. Instead of relying on paperwork and manual updates — which are time-consuming and often lead to mistakes — this system offers a digital solution that simplifies the whole inventory process.

The focus is entirely on the university's stocks, which are grouped into four key areas: **classrooms, offices, storage rooms, and laboratories**. Each of these spaces has different needs, and the system is built to reflect that.

This web-based platform makes it easier for staff to keep everything organized. It saves time, reduces the chance of errors, and ensures that important information about items — like their condition and availability — is always up to date.

Scope:

The University Inventory Management System will offer several important features:

- **Item categorization** based on where things are located (e.g., in a class, an office, or a storage room), so users can easily find and manage inventory.
- **Live tracking of items**, so everyone knows what's available, what's running low, and what condition each item is in.
- **Notifications and reports** to alert users when something needs replacing or when maintenance is required.
- **User access control** that assigns different permissions depending on the user's role — for example, Finance, IT, or Club Coordinator — so that each department can use the system according to its responsibilities.

Overall, the system is meant to make inventory handling easier for everyone involved, from daily users to administrators, while improving accuracy and saving time.



II. System Description

Our proposed platform is designed to streamline the management of inventory within a university, ensuring efficient tracking, organization, and accessibility of resources. Universities often struggle with maintaining a well-structured inventory system, leading to misplaced assets, inefficient resource allocation, and increased operational costs.

Our system aims to provide an efficient and user-friendly solution for managing equipment, supplies, and other university assets across various departments. By integrating real-time tracking, automated restocking alerts, and detailed usage reports, our platform enhances resource management while reducing manual workload.

Furthermore, the platform is designed to be intuitive and adaptable to the daily operations of university staff, ensuring seamless inventory control without requiring extensive training.

2.1. Product Context

Our platform is independent and self-contained. While other university inventory management systems primarily focus on internal operations within specific departments, our platform extends its functionality to provide a centralized solution for managing inventory across the entire institution. It enables universities to efficiently track, allocate, and maintain their assets while ensuring transparency and accessibility for all relevant departments.

All inventory data, including asset records, and stock levels, will be created, stored, and managed exclusively within our platform, without reliance on external systems. This ensures data security, consistency, and ease of use for university staff.

A detailed view of how the system operates can be found in our GitHub repository:

<https://github.com/XhHalla/Inventory-Management-System>.

2.2. User Characteristics

This program will be used by these primary users:

- Finance
- IT
- Service Department
- University Staff
- Club Coordinators

a) Finance

Finance will be an important administrator which will decide which stock will be removed, where the stock will be placed and in which storeroom it will be held. This administrator will manage the inventory system, also will keep track of inventory status to buy new items if needed and “predict” the items that they would need to buy soon. The Finance department will also take care of the user privileges and roles. They can also accept or decline requests from University Staff, IT, Handyman, or Club Coordinators.

b) IT

The IT department will get requests from different user roles upon checking the status of a device and updating it, changing a device if it is not in a working condition. IT will be aware if any stock is needed that is related to them.

This administrator will decide where the stock will be placed, and it will also be updated in the inventory web-application system. If the IT department wants to add a device or needs a device it will make a request which will go to the Finance department.

c) Service Department

The Service Department will get requests from different user roles upon on checking the status of classes and updating it, changing or fixing the materials that are not in working condition. They will be aware if any stock is needed.

The Service Department will decide where the stock that he is managing will be placed and it will also be responsible for updating the inventory. They will also be responsible for forwarding the requests to Finance whenever a new item is needed.

d) University Staff

The University Staff user would be able to request a quality check for their office or class. They can make this request for an upcoming event so the IT department would make sure that the status of devices in that class have not changed and the handyman can make sure that everything else on that environment is in a good condition. University staff would be responsible for editing the status of items in its office and they also can edit the status of items in a class that they are teaching (Professor).

e) Club Coordinators



Club Coordinators are responsible for managing inventory related to student clubs and activities. They can submit stock requests to either the IT Department or the Service Department, depending on whether the item is IT-related or non-IT-related. Once submitted, these requests follow the standard approval workflow through the Finance Department. Club Coordinators can also monitor the availability and status of club-related items through the system's interface, ensuring timely planning and resource allocation for upcoming events or activities.

2.3. Assumptions

2.3.1. User Assumptions

- It is assumed that every user has the appropriate equipment (computer or mobile device, internet access) to use the program.
- It is assumed that the users of this program have general knowledge on how to use the internet and their respective devices effectively.
- It is assumed that every user of the program has his/her university email activated and working properly.
- It is assumed that the users will update the inventory for every object that is missing, damaged or broken.
- It is assumed that every change that happens should be updated in the system for better organization and better usage of this system.

2.3.2. System Assumptions

- It is assumed that the data generated and registered will be fully confidential and it will be available only for the specific departments of the university.
- It is assumed that the stock is already registered by the service department.
- It is assumed that we have all the barcodes for each product registered.



2.4. Constraints

The system may be constrained by:

- Having every user to understand how this system works and making sure they do not do any mistakes.
- Our system is a standalone platform that does not integrate with external inventory management systems.
- Problems with access management or security.
- The system does not allow the user to make customizations regarding the system name components as it may cause anomalies.
- Inventory requests must be approved by designated Inventory Managers before items can be issued.

2.5. System Dependencies

The functionality and performance of the inventory system rely on the following conditions:

User Environment Dependencies

- The user should always have internet access.
- The user should have an electronic device.
- The need for at least moderate internet speed.

System Operational Dependencies

- There should be a user from the Finance or IT department to make changes such as adding or deleting a user, adding or deleting stocks etc.
- The response time depends on the volume of requests and the number of users that are currently using the system.



III. REQUIREMENTS

3.1. Functional Requirements

The system will provide the following core and advanced functional capabilities, organized by functional area. These features are essential to ensuring efficient inventory tracking, user interaction, process automation, and administrative control.

3.1.1. Item & Product Management

The system shall include the following functionalities:

- The system should allow specific users to add, modify or delete items in the inventory database.
- This includes the ability to enter item descriptions, unit of measure (such as pieces, etc.), and stock levels.
- The item management feature should also allow users to categorize items into groups, such as product type which will be separatable in two main groups. One of the groups will be managed by the IT department and the other will be managed by the Service Department.
- The Item Management system will also be able to move the items in different classes, offices or storage rooms.
- The system should provide a way to manage and track product information, such as product descriptions, status, pricing, and images. This helps users keep their product information up-to-date and accurate.

3.1.2. Inventory and Stock Management

3.1.2.1. Stock Management

The inventory management system should track the current stock levels of each item and provide notifications for low stock levels.

This helps ensure that there is enough inventory to fulfill the demands and to prevent stockouts.

The system should also allow users to set reorder points for each item, which triggers a notification when the stock level drops below a certain threshold.

3.1.2.2. Stock Adjustment



The system should allow users to make adjustments to stock levels, such as to correct discrepancies or to account for damaged items.

3.1.2.3. Stock Valuation

The system should provide a way to value inventory, such as using first-in, first-out (FIFO) or last-in, first-out (LIFO) methods.

This helps users determine the value of their inventory for financial reporting purposes.

3.1.2.4. Multi-Location Support

The system should allow users to manage inventory across multiple locations.

This may include the ability to transfer inventory between locations and to view inventory levels for each location.

3.1.3. Order & Warehouse Management

3.1.3.1. Order Management

The system should provide a way to manage and track purchase orders.

This includes the ability to create, approve, and fulfill purchase orders.

The order management feature should also allow users to view order history.

3.1.3.2. Purchase Order Approval

The system should provide a way for users to approve purchase orders, helping ensure that only authorized purchases are made.

3.1.3.3. Warehouse Management

The system should provide a way to manage and track warehouse activities, such as receiving, storing.

This may include features such as barcode scanning, pick and pack functionality.

3.1.4. Reporting & Analytics

3.1.4.1. Reporting

The inventory management system should provide a variety of reports that allow users to view information about their inventory and purchasing activities.

This may include reports on stock levels, purchase history and states of each item.

The system should also allow users to customize reports and export data to other formats, such as Excel or PDF.



3.1.4.2. Analytics & Business Intelligence

The system should provide robust analytics and business intelligence tools to help users gain insights into their inventory and purchasing activities.

The system should support AI-powered demand forecasting based on historical stock usage, seasonal trends, and reorder patterns.

3.1.4.3. Customizable Dashboards

The system should allow users to create custom dashboards that display real-time information about their inventory, their status and purchasing activities.

3.1.5. User & Access Management

3.1.5.1. User Management

The system should allow administrators to manage user accounts, assign roles and permissions, and control access to the system.

This helps ensure that only authorized users have access to sensitive information, such as financial data or storage room locations with their respective items.

3.1.5.2. Customizable Workflows

The system should allow users to customize workflows, such as purchase order approval processes, to meet the unique needs.

3.1.6. Mobile & Real – Time Capabilities

3.1.6.1. Mobile Accessibility

The system should be accessible from mobile devices, allowing users to manage or view their inventory, make requests on the go.

This includes the ability to access inventory levels, add or modify items and view reports.

3.1.6.2. Real-Time Updates

The system should provide real-time updates on inventory levels, status of the items in the inventory and purchasing activities.

This helps users stay up to date with the latest information and make informed decisions.



3.1.7. Barcode, Scanning & Alerts

3.1.7.1. Barcode and Scanning

The system should provide support for barcode scanning and QR code scanning. This makes it easy for users to scan items for accurate and efficient inventory management.

The barcode and scanning feature should also allow users to print barcodes for items that do not already have them.

3.1.7.2. Automated Alerts

The system should provide automated alerts for low stock levels, and other events such as status updates, helping users stay informed and take timely action.

3.1.8. Data Integration & Security

3.1.8.1. Data Security

The system should provide robust data security to protect sensitive information, such as financial data, customer information, and inventory details.

This includes secure storage of data, secure transmission of data, and access controls to prevent unauthorized access.

The system should also have backup and recovery features to ensure that data can be restored in the event of a disaster.

3.1.8.2. Data Import/Export

The system should allow users to import and export data from other systems, such as spreadsheets or databases, to help streamline data entry and reduce manual data entry.

3.1.9. Microsoft Single Sign-On (SSO) Integration

The requirements for an "on-click login" feature using a Microsoft account could include the following:

- **User registration:** The ability for users to create a Microsoft account if they do not already have one.
- **Login button:** A button or link that allows users to initiate the login process with a single click.



- **Microsoft account authentication:** The ability to authenticate a user's Microsoft account credentials and retrieve their profile information.
- **User authorization:** The ability to grant the user's authorization for the application to access their Microsoft account information.
- **Token management:** The ability to securely store and manage the access token received from Microsoft.
- **Single sign-on (SSO):** The ability to recognize a user who has already logged in with their Microsoft account and automatically log them in without requiring additional authentication.
- **User profile information:** The ability to access and display the user's Microsoft account information, such as their name, email, and profile picture.
- **Logout:** The ability for users to log out of their Microsoft account and end the session.
- **Error handling:** The ability to handle and display error messages in the event of authentication or authorization failure.
- **Security:** The ability to securely store and transmit sensitive user information, such as passwords and access tokens, in accordance with industry standards.

3.2. Non - Functional Requirements

3.2.1. Product Requirements

3.2.1.1. User Interface Requirements

General UI Requirements:

- The user interface (UI) must maintain a minimalist, accessible, and role-specific layout, ensuring efficiency and clarity for all user types.
- Secure login via encrypted credentials will gate access to personalized dashboards, ensuring role-based access control.
- A consistent horizontal navigation bar (one-third width) and a responsive content dashboard (two-thirds width) will structure the UI across all devices.
- The UI must be fully responsive, ensuring smooth functionality across desktops, tablets, and smartphones, with adaptive UI/UX patterns.

Role-Based UI Specifications:

Finance (Admin):

- The navigation bar will contain: Homepage, Check Stock, Edit Location, Manage Users.
- Homepage will display stock requests from IT and Service Departments in a card layout with buttons: Approve, Pending, and Decline.



- An automated notification will be sent based on the selected action.
- Check Stock will show cards representing university buildings and clubs. Upon selecting a building, users can choose to view Class, Office, or Storage Room and see item status via detailed cards or a table format.
- Edit Location page will allow Finance to add, edit, or delete buildings, classes, and offices using an intuitive card-based UI. Clicking a "+" card will open a form for adding new locations.
- The Manage Users page will display all users with filters by role. Finance can add, edit roles, or delete users. User info stored includes: name, email, encrypted password, and assigned role.

IT:

- The navigation bar will contain: Homepage, Check Stock, Stock Request.
- Homepage shows incoming requests from University Staff or Club Coordinators (e.g., QA or stock requests) as cards with buttons: Accept, Forward, Decline.
- Check Stock displays only IT-related equipment for classes, offices, and storages. IT can view, add, edit, or delete items.
- The Stock Request form requires fields like item name, quantity, and reason for request, which will be forwarded to Finance.

University Staff:

- The navigation bar will contain: Homepage, QA Request, Stock Request.
- Homepage will display either:
 - Their assigned office with listed items and editable status fields.
 - Or a building/classroom navigation to find and update item statuses themselves.
- If an item is marked “Not Working,” a request will be auto generated for the relevant department.
- QA Request and Stock Request will share one page with two image-based buttons:
 - Clicking QA opens a form requiring location and reason.
 - Clicking Stock Request opens the request form with a department field.

Club Coordinators:

- The navigation bar will contain: Homepage, Stock Request.
- The homepage displays items assigned to their club, with options to update item status.
- Stock Request will use the same form format, allowing coordinators to select the target department (IT or Service).

Service Department:

- The navigation bar will contain: Homepage, Check Stock, Stock Request.
- The homepage displays requests similar to IT's, including QA or maintenance tasks from Staff or Coordinators.
- Check Stock allows the Service Department to view and manage only service-related items, with full CRUD operations.
- The Stock Request page will follow the same format as IT's, including a department-specific request form.

3.2.1.2. Usability

1. **Learnability:** New users (e.g., university staff or club coordinators) should be able to learn how to use the system within 15 minutes of first interaction, without the need for formal training, as a short onboarding tooltip or guide will be available to assist them during their first use.
2. **Navigation & Interface Consistency:** All users will see a similar navigation bar with options based on their role, making it easy to navigate the system. The overall design and layout of pages such as forms, cards, and tables will stay consistent for everyone, with only the content changing depending on the user. This consistency helps users complete their main tasks, like checking stock or making a request, in just a few clicks from the homepage.
3. **Feedback & Error Handling:** The system provides instant visual feedback whenever a user takes an action, such as submitting a request or updating an item's status. If a user input is invalid, like leaving a required field empty, the system highlights the problem and shows a clear message explaining what needs to be fixed. After successfully submitting or updating a form, a confirmation message appears to let the user know the action was completed.
4. **Accessibility:** All features in the system should be fully usable using keyboard navigation, allowing users to move through elements with keys like Tab and Enter. To ensure accessibility, all button and label text must be clear and easy to read, with proper contrast between text and background.
5. **Task Efficiency:** Users should be able to filter and sort data (like stock items or users) quickly with dropdowns or search bars.
6. **Mobile Responsiveness (Optional/Stretch Goal):** If the system will be used on tablets or mobile, all pages should be responsive and maintain usability on smaller screens.



7. **Satisfaction:** During testing, at least 80% of users should rate the system as "easy to use" or better on usability surveys.

3.2.1.3. Efficiency

3.2.1.3.1. Performance Requirements

The application will be a web-based platform hosted on a web server. Its performance will depend on:

- The speed and stability of the internet connection
- The efficiency of data retrieval from the database
- The number of users actively accessing the application simultaneously

In addition, performance depends on the following key aspects:

1. Response Time

The system should respond to user interactions (such as submitting a request or viewing stock) within 2 seconds under normal network conditions.

2. Throughput

The system should be able to manage large volumes of transactions and data while maintaining fast and efficient processing speeds. This will help ensure high throughput and smooth performance, even when multiple users are interacting with the system at the same time.

3. Performance Optimization

The system should be designed with performance in mind, using efficient algorithms and well-structured data handling to reduce latency and ensure quick response times. This will help provide a smooth and responsive experience for all users.

4. Scalability

The application should be designed to scale horizontally, allowing additional server resources to be added when usage increases, especially during peak periods like semester starts or inventory updates.

5. Load Testing

The system should undergo load testing to assess its performance under varying levels of user demand and to identify potential performance bottlenecks.

6. Data Retrieval

Queries fetching stock or user data should return results within 1–3 seconds, even when accessing large datasets (e.g., 5000+ items).

7. Monitoring and Analysis

The system should include tools for monitoring and analysis, like real-time tracking and logging, to help detect performance issues and improve response times.

8. Caching

The system should use caching methods, like keeping frequently accessed data in memory, to speed up response times and lower latency.



9. Distributed Systems

If the system runs on multiple servers, it should apply distributed system techniques like load balancing and data replication to boost performance and maintain fast, responsive operation.

3.2.1.3.2. Space Requirements

To ensure the system operates reliably over time and remains scalable within the university's infrastructure, specific space-related non-functional requirements have been defined regarding resource utilization and data storage.

1. Efficient Resource Utilization

The system should be optimized to use a reasonable amount of RAM and storage, ensuring smooth operation on standard university hardware and web servers.

2. Optimized Data Storage and Management

The database must efficiently store inventory data, including item records, location details, user accounts, and request logs, while minimizing redundancy.

3. Storage Capacity Planning

Our system will primarily store structured inventory records—item details, location data, user accounts, and request logs—which are very lightweight.

Assuming we support **1,000 active users** generating about **5 MB of text-based data per month**, that amounts to:

5 MB × 12 months × 10 years = 600 MB of logs over 10 years.

In addition, if we include up to **5,000 item photos at 0.2 MB each**, that adds another **1 GB of storage**.

Even allowing for a comfortable growth buffer, **all of this fits well within a 1 TB disk**, ensuring ample room for backups and future expansion.

3.2.1.4. Dependability Requirements

3.2.1.4.1. Availability

- Uptime:** The system should have a high uptime, with minimal downtime, to ensure that users have continuous access to the system and its data.
- Load Balancing:** The system should provide load balancing, such as through a load balancer, to distribute traffic across multiple servers and ensure that the system remains available even under heavy load.



3. **Disaster Recovery:** The system should provide a disaster recovery mechanism, such as a secondary backup server, to ensure that the system can recover from failures and continue functioning even in the event of a disaster.
4. **Backups:** The system should provide regular backups, such as daily or weekly backups, to ensure that data can be recovered in the event of failures or outages.
5. **Error Handling and Notifications:** If an issue occurs or an action results in an error, the system will display clear and relevant messages to inform the user about the problem.

3.2.1.4.2. Latency

Latency may be influenced by several factors:

- Database size, which can influence how quickly data is accessed and processed.
- Network bandwidth and internet connection speed, impacting data transfer rates.
- Efficiency of database queries and data retrieval processes.
- The number of concurrent users interacting with the application at a given time.

In addition, the following requirements support low latency:

1. **Latency Targets:** The system must achieve defined latency goals, delivering responses within a specified number of milliseconds to maintain a fast and responsive user experience.
2. **Real-Time Updates:** The system should deliver live updates, ensuring that users can access the most current information instantly, without noticeable delays.
3. **Network Optimization:** The system should be designed to use network bandwidth efficiently and minimize network latency, ensuring a fast and responsive experience for users.

3.2.1.4.3. Monitoring

- When an error occurs, a message will appear to notify the user about the issue and its possible causes.
- Users must enter the correct and corresponding email and password to access the application. If the login credentials are invalid, an error message will notify the users.
- The application will be designed with reliability in mind, ensuring that all user-entered information is properly validated.
- A user hierarchy will be established as previously outlined by the HR department.

3.2.1.4.4. Maintenance



- If the application crashes, the system will automatically restart and redirect the user to the homepage, preserving any actions previously performed. In such cases, it is safe for users to refresh the webpage. However, if the issue persists and the application crashes again, a maintenance break will be triggered, and users will be notified accordingly.
- The system will undergo scheduled maintenance once a week, typically during midnight on weekends, for approximately 30 to 60 minutes when necessary.
- The system will be maintained by the development team, particularly when updates are needed or issues arise. New features will be introduced based on user needs and feedback.
- Ongoing support for the system will always be available. Developers can be contacted through messages sent by the platform administrator.

3.2.1.4.5. Integrity

- The integrity of the university inventory system is maintained through detailed system logs and routine daily reviews of these logs to identify and understand any issues that may arise during system operation.

3.2.1.5. Security

1. **Authentication and Authorization:** Only registered users (e.g., staff, coordinators, IT, admin) can log in. Role-based access control (RBAC) ensures users only access features relevant to their roles.
2. **Data Protection:** Sensitive data (e.g., login credentials, user info) must be encrypted in transit (using HTTPS) and at rest (e.g., hashed passwords in the database).
3. **Audit Logging:** All critical actions (e.g., adding items, approving requests, editing inventory) should be logged with timestamps and user information. Logs should be protected from tampering and regularly reviewed by administrators.
4. **Input Validation:** The system must validate all user inputs to prevent SQL injection, cross-site scripting (XSS), and other common attacks.
5. **Access Control:** All critical actions (item updates, request approvals, login attempts) should be logged for tracking and accountability.
6. **Backup and Recovery:** Regular backups of inventory data should be maintained to recover from system failure or data loss. Backup data should also be secured and accessible only to authorized personnel.
7. **Security Testing:** The system should allow for quick patching and updates to fix vulnerabilities as they are discovered. Only authorized developers or admins should be able to perform system maintenance or updates.



3.2.2. Organizational Requirements

3.2.2.1. Environmental Requirements

1. **Operating Environment:** The system should be compatible with a range of operating systems, including Windows, macOS, Linux, and Unix.
2. **Server and Hosting:** The application shall be deployed on the university's internal web server or a secure university-managed cloud environment that supports high availability and data integrity.
3. **Hardware Compatibility:** The system shall be optimized to run efficiently on typical university hardware configurations, requiring a minimum of 4 GB RAM, 2.0 GHz processor, and at least 1 GB of available storage for system components.
4. **Network Requirements:** The system shall function reliably within the university's internal network (LAN/Wi-Fi), maintaining stable performance over typical bandwidth speeds provided on campus.
5. **Database Environment:** The system shall integrate with a university-approved relational database system (e.g., MySQL) and support existing database access policies and backup protocols.
6. **Power and Stability:** The system shall remain stable and recoverable in the event of temporary power outages, supporting auto-restart and session recovery features where feasible.
7. **Physical Environment:** The system shall perform in secure office and lab conditions typical of university departments, with no special temperature, humidity requirements and appropriate measures in place to protect the hardware, software, and data from theft, tampering, or damage.

3.2.2.2. Operational Requirements

Our software is a web-based inventory management application designed to support university operations by tracking and managing assets across various departments. It enables efficient handling of items, requests, and updates in real time, ensuring transparency and accountability.

- The system must be deployed on the university's secure web server and accessible 24/7 via standard web browsers



- Daily automated backups of all inventory and user data will be performed at midnight
- System maintenance will occur once a week during low-usage hours (e.g., Sunday 12:00 AM to 1:00 AM)
- If the system encounters a crash, it will automatically restart and redirect users to the homepage. All unsaved data will be preserved where possible
- The university IT department will be responsible for basic technical maintenance, while the development team will handle updates and feature requests

3.2.2.3. Development Requirement

1. Technology Stack

- The frontend must be developed using HTML, CSS, and JavaScript to ensure a responsive and user-friendly interface
- The backend logic and APIs will be implemented using .NET framework (preferably ASP.NET Core for scalability and modern support)
- MySQL will serve as the relational database for storing all inventory records, user accounts, requests, and logs

2. Code Organization

- The application code should follow the WEB API architectural pattern for maintainability and separation of concerns
- The backend should follow a clean architecture or layered architecture pattern, separating concerns across different layers such as Controllers, Services, Repositories, and Models
- Controllers will handle HTTP requests and route them to the appropriate service methods.

3. Security Practices

- Passwords must be hashed using secure algorithms (e.g., bcrypt or SHA-256) before being stored in the database
- Role-based access control (RBAC) must be implemented to restrict access to features based on user roles (Finance, IT, Staff, Coordinators, etc.)

4. Version Control

- All source code must be maintained in a Git repository (e.g., GitHub, GitLab)
- Development will follow a structured branching model (e.g., main, dev, feature branches)



5. Deployment

- The final system will be deployed to a university-hosted web server or cloud server (if applicable)
- Environment variables or config files should be used to manage sensitive data such as database credentials

3.2.3. External Requirements

3.2.3.1. Regulatory Requirement

The University Inventory System will adhere to relevant data protection and privacy policies, in alignment with institutional regulations and applicable national laws concerning the protection of personal information. In particular, the system will comply with the principles outlined in the Albanian Law No. 9887, dated 10.03.2008, "On the Protection of Personal Data", as well as supporting sub-legal acts.

Regulatory requirements also depend on:

1. Legal Compliance

- The system must adhere to national or international laws (e.g., GDPR)
- When handling user data, the system must comply with data protection regulations

2. Industry Standards

- The system must be compliant with specific standards to education, and finance

3. Data Protection and Privacy

- The system must protect sensitive or personal data and give users control over it

4. Copyright and Licensing

- Use of third-party libraries, fonts, images, or frameworks must respect licensing agreements

3.2.3.2. Ethical Requirement

The university inventory system will be designed and implemented with strong ethical considerations to ensure responsible and fair use by all stakeholders. The following principles will guide its development and usage:



1. Data Privacy and Confidentiality:

- The system will safeguard all user data, including login credentials, inventory records, and request logs. Access to sensitive data will be limited to authorized roles such as IT personnel or administrators, ensuring that staff and club coordinators can only view information relevant to their responsibilities.

2. Transparency:

- Users will be informed about how their data is stored, accessed, and used within the system. System actions such as item updates, approvals, and requests will be logged, allowing for transparency without violating user privacy.

3. Accountability:

- All user activities will be traceable through system logs, ensuring that any misuse or errors can be investigated and resolved. The development team and administrators will be responsible for maintaining a secure and reliable platform.

4. Non-Discrimination and Fair Access:

- The system will provide equal access to features for all designated user roles, without favoritism or bias. User permissions will be based strictly on institutional roles and predefined responsibilities.

5. Preventing Misuse:

- To maintain ethical standards, the system will include security and validation mechanisms to prevent unauthorized access, data tampering, or fraudulent requests. Any suspicious activity will be flagged and reviewed.

6. Professional Integrity:

- Developers working on the system will adhere to professional and academic standards, including respecting intellectual property rights, using open-source libraries responsibly, and maintaining clean, well-documented code.

Moreover, Sure, user data from Finance, IT, University Staff, and Club Coordinators will be used solely for verification, communication, and internal system operations. Consent is required to ensure proper tracking and accountability within the inventory system. The personal information that workers will share is full name and email.

3.2.3.3. Legislative Requirement

The system will fully comply with the laws of Albania and follow all relevant standards and regulations established by Albanian government authorities and law enforcement agencies.

Additionally, these regulations require users to consider the safety risks to users and to carry out a risk assessment to protect users from exposure to reasonably foreseeable risks. Those risks include work-related violence. A risk assessment is an examination to:

- Comply with data protection and privacy laws.
- Establish the significance of the risk.
- Identify and implement prevention and control measures.

3.2.3.3.1. Accounting Requirement

The personal information of all users in the university inventory system, including staff, IT personnel, finance, and club coordinators, will be protected in accordance with Albanian legislation. The system will comply with Law Number 9887 dated 10 March 2008, as amended by Law Number 48 of 2012, titled On the Protection of Personal Data. Access to sensitive data will be restricted only to authorized users as defined by institutional roles and responsibilities.

3.2.3.3.2. Security Requirement

This web application will also comply with the most recent updates of the General Data Protection Regulation (GDPR), which became enforceable on May 25, 2018, across European Union (EU) and European Economic Area (EEA) member states.

3.3. Domain Requirements

The university inventory system must follow internal policies and procedures, with role-based access for departments like IT, Finance, and Club Coordinators. Inventory items will be categorized and tagged for easy tracking, and all changes must follow approval workflows. The system should support regular audits, maintain a full history log, and generate reports to help departments manage assets. It will also comply with local standards, including Albanian language support and regional formats.

IV. SOFTWARE DESIGN

4.1. BPMN Diagrams

4.1.1. BPMN – Inventory Request and Maintenance Workflow

The following BPMN (Business Process Model and Notation) diagram illustrates the end-to-end workflow of a typical inventory request within the University Inventory Management System. This process involves all actors: University Staff/Club Coordinators, the IT/Service Department, and the Finance. This BPMN model helps standardize and visualize cross-departmental coordination while ensuring accountability, proper escalation, and timely communication between all involved roles.

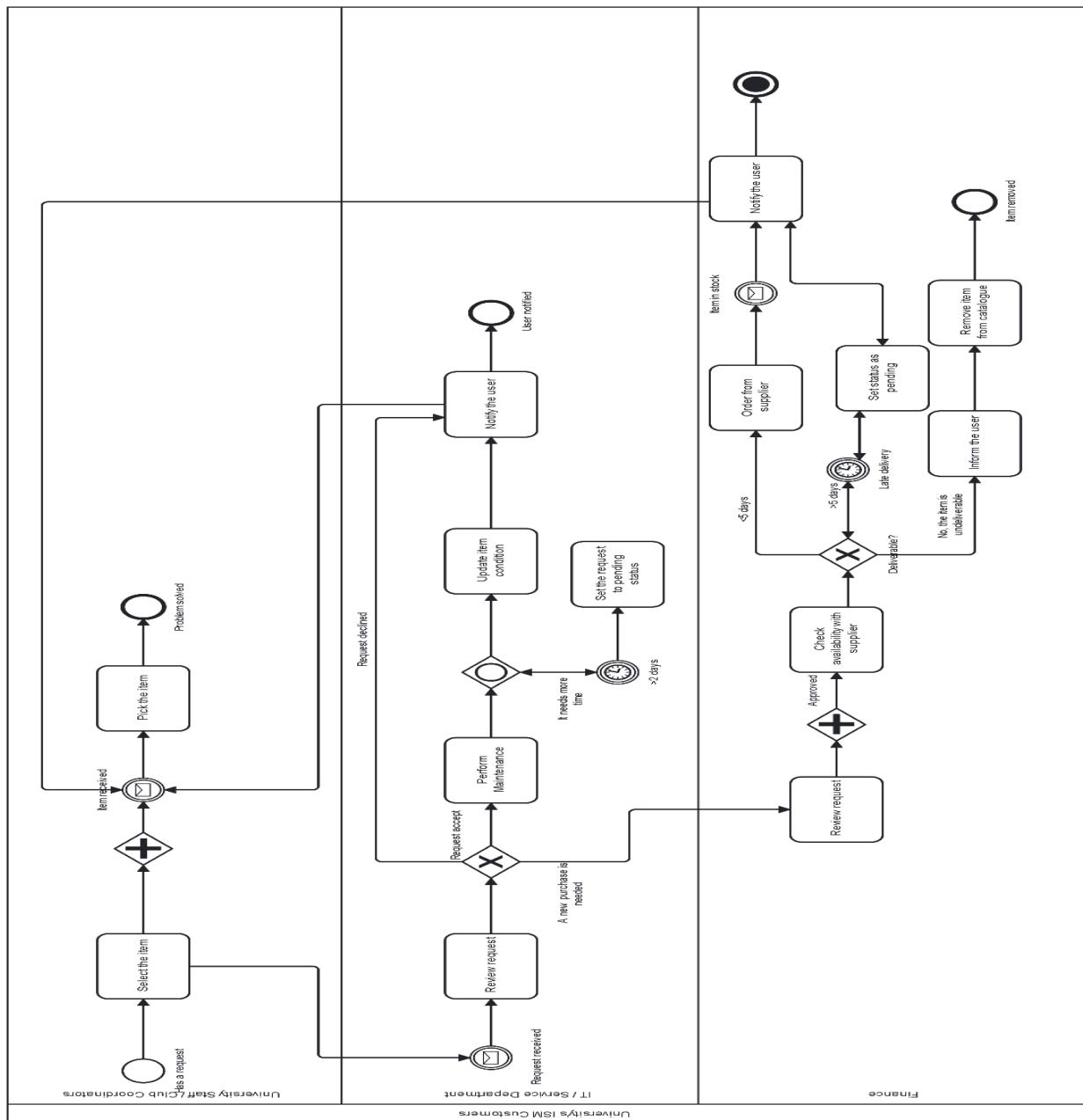


Figure 4.1.1 –BPMN for Inventory Request and Resolution across University Departments

4.1.2. BPMN – Account Provisioning and Login Process

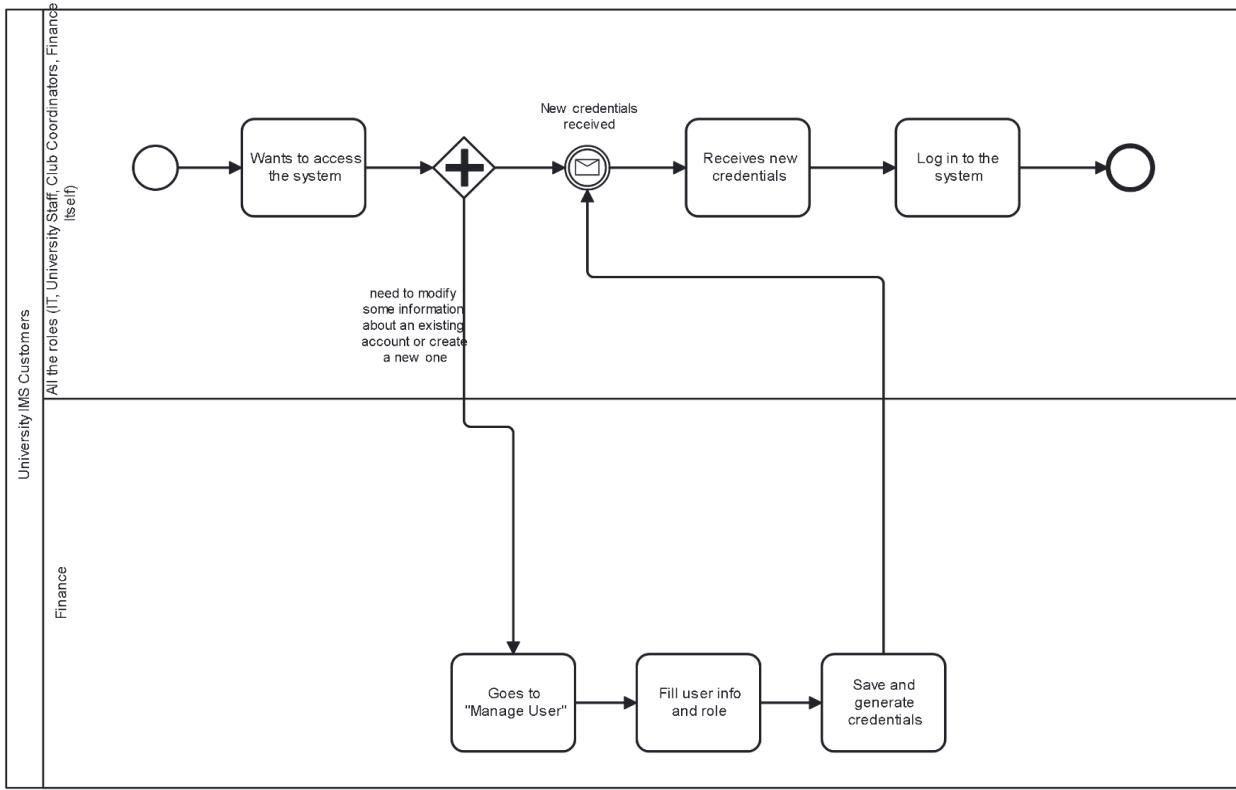


Figure 4.1.2 – BPMN Diagram for User Credential Request and Login Workflow

4.1.3. BPMN – Location Management Workflow

This BPMN diagram illustrates the location management process performed by the Finance Department. The workflow begins when an item is identified as being incorrectly located.

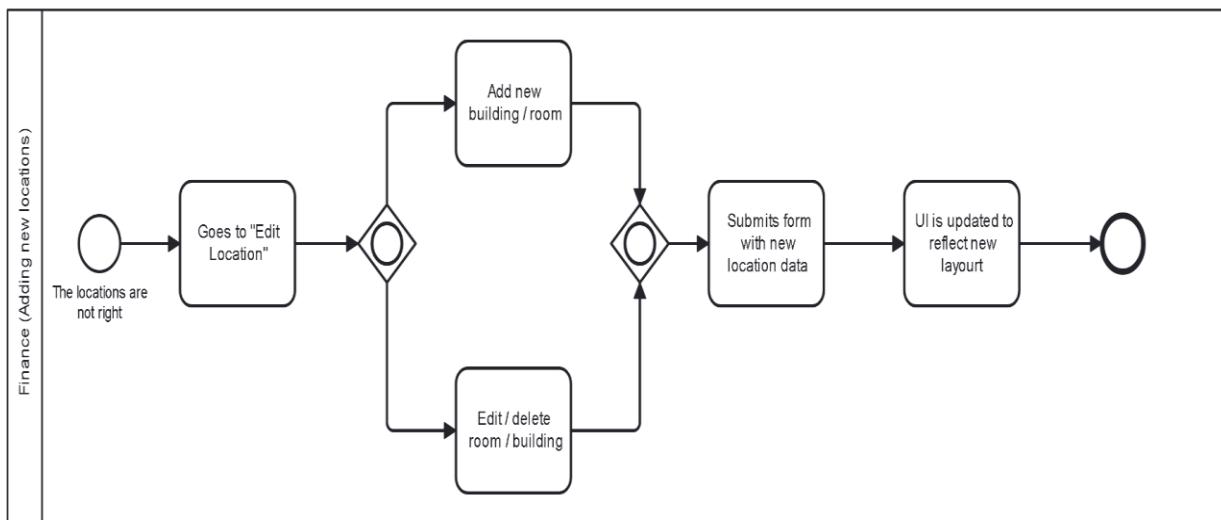


Figure 4.1.3 - Business Process Model for Managing and Updating Item Location Information

4.2. Use Case Descriptions

4.2.1. Use Case – Check Stock

Table 4.2.1: Fully Developed Use Case Description for “Checking Stock”

Use case name:	Check Stock					
Scenario:	It , Service or Finance user browses current inventory by university building or club, drills down to specific rooms (classroom, office, storage), and inspects item details and condition.					
Triggering event:	Clicks on “Check Stock” from the main menu.					
Brief description:	The user views a high-level dashboard: cards representing each of the three main university buildings plus one card for “Clubs.” Upon selecting a building card, Finance can choose to view inventory by Room Type (Class, Office, or Storage). Within a room type, location names are displayed; clicking a location shows a listing of items—with their quantity and condition (New, Good, Barely Working, Not Working). Optionally, each item row can be displayed as a card with a photo.					
Actors:	Primary Actor: Finance Primary Actor: It Primary Actor: Service					
Related use cases:						
Stakeholders:	User(monitors overall inventory) IT & Service Departments (coordinate maintenance tasks when items are flagged as “Barely Working” or “Not Working”) University Staff (relies on accurate inventory records) Club Coordinators (their items appear under “Clubs”)					
Preconditions:	User is authenticated with role assigned as It or Service Department. Inventory records for all buildings, rooms, and clubs are already in the database.					
Postconditions:	User has viewed the up-to-date listing of all items, their quantities, and conditions for the selected building/room/club. No data is modified (read-only scenario).					
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> User clicks “Check Stock” in the horizontal navigation bar. </td> <td> <ul style="list-style-type: none"> System retrieves the three building records plus the “Clubs” category and displays four cards (e.g., </td> </tr> </tbody> </table>	Actor	System	<ul style="list-style-type: none"> User clicks “Check Stock” in the horizontal navigation bar. 	<ul style="list-style-type: none"> System retrieves the three building records plus the “Clubs” category and displays four cards (e.g., 	
Actor	System					
<ul style="list-style-type: none"> User clicks “Check Stock” in the horizontal navigation bar. 	<ul style="list-style-type: none"> System retrieves the three building records plus the “Clubs” category and displays four cards (e.g., 					

	<ul style="list-style-type: none"> User clicks on, say, the “Building A” card. User clicks the “Classrooms” sub-card. User clicks on “Class 201.” User scrolls or searches within that table to inspect individual item details. 	<ul style="list-style-type: none"> “Building A,” “Building B,” “Building C,” “Clubs”). System shows three sub-cards or buttons: “Classrooms,” “Offices,” “Storage Rooms.” System fetches and displays a list of classroom names in Building A (e.g., “Class 201,” “Class 202,” etc.), each with a brief description (e.g., “1 PC, 1 monitor, 20 tables”). System displays a tabular listing (or optional card-style listing) of each item in “Class 201,” showing columns: Item Name, Quantity, Condition, Photo Thumbnail (if enabled).
Exception conditions:	<p>No Inventory Data: If no classrooms exist under that building—for example, if “Classrooms” is empty—system displays “No locations found. Please add new location under ‘Edit Location.’”</p> <p>Database Error: If the inventory fetch fails, the system shows an error message (“Unable to retrieve stock data. Try again later.”)</p>	

4.2.2. Use Case – Edit Location

Table 4.2.2: Fully Developed Use Case Description for “Editing Location”

Use case name:	Edit Location
Scenario:	Finance browses current inventory by university building or club, drills down to specific rooms (classroom, office, storage), and inspects item details and condition.
Triggering event:	Finance clicks “Edit Location” from the main menu
Brief description:	The Finance user sees the same UI layout as “Check Stock” (cards for buildings and clubs), but instead of showing item details, each location card includes an “Edit” icon at its top right. Finance can click an existing location’s edit icon to modify its name/description or delete it, or click a “+” (Add) card to open a form to create a new building, classroom,



	office, or storage room. After submitting, updates are saved to the database and reflected in “Check Stock.”	
Actors:	Primary Actor: Finance (System Administrator)	
Related use cases:		
Stakeholders:	Finance Department (manages location hierarchy) IT & Service Departments (need correct location mappings for their items) University Staff and Club Coordinators (items are tied to these locations)	
Preconditions:	Finance is authenticated. Existing location hierarchy (buildings → room types → specific rooms) is present in the system.	
Postconditions:	If a new location is added, a new record is created in the database under the appropriate parent (e.g., new “Building D,” or new “Office 105” inside Building A). If an existing location is edited, its name/description fields are updated. If an existing location is deleted, the record (and any child sub-records) are removed or flagged for archival (depending on implementation). The “Check Stock” view now reflects the new/modified hierarchy.	
Flow of activities:	Actor	System
	<ul style="list-style-type: none">• Finance clicks “Edit Location” in the navigation bar.• Finance clicks the “+ Add Location” card.• Finance fills out required fields (e.g., “Building D,” description “New Engineering Building”) and clicks “Save.”• Finance clicks the “Edit” icon on “Class 201.”• Finance modifies fields (e.g., rename to “Class 201A”) and clicks “Save.”	<ul style="list-style-type: none">• System displays cards for all buildings and “Clubs,” each with an “Edit” icon, plus an extra “+ Add Location” card.• System opens a modal or separate page containing a form with fields:<ul style="list-style-type: none">Location Type (Building / Classroom / Office / Storage Room)Parent Location (if applicable)Location NameDescriptionPhoto (optional)Save and Cancel buttons

	<ul style="list-style-type: none"> Finance clicks the “Delete” icon on “Office 305.” 	<ul style="list-style-type: none"> System validates input (e.g., non-empty name). If validation passes, system creates a new location record with: <p>Type = Building</p> <p>Name = “Building D”</p> <p>Description = [provided]</p> <p>And persists it to the database.</p> <ul style="list-style-type: none"> System returns to “Edit Location” view and shows the updated set of cards (including the new building card). System loads the “Edit Location” form populated with the current data for “Class 201” (e.g., location type = Classroom, parent = Building A, name = “Class 201,” description). System validates the new data. If valid, updates the “Class 201” record in the database to reflect the new name. System returns to “Edit Location” view, reflecting the updated name. System prompts: “Are you sure you want to delete Office 305 and all associated inventory records?” <p>If Finance clicks “Confirm,” go to next step.</p> <p>If “Cancel,” abort deletion.</p> <ul style="list-style-type: none"> System deletes (or flags as inactive) the “Office 305” record and cascades or flags any children (if they exist), then refreshes the view.
Exception conditions:	Validation Failure on Add/Edit: If “Location Name” is blank or duplicates an existing location at the same level, the system displays “Invalid name—please ensure it’s nonempty and unique.”	



	<p>Database Save Error: If saving to the database fails (e.g., connectivity issue), the system shows “Unable to save changes. Try again later.” Changes are not applied.</p> <p>Delete with Dependent Inventory: If Finance tries to delete a location that still has items assigned, the system warns “Cannot delete location—items still assigned. Please move or delete items first.”</p>
--	--

4.2.3. Use Case – Manage Users

Table 4.2.3: Fully Developed Use Case Description for “Managing Users”

Use case name:	Manage Users	
Scenario:	Finance (as System Admin) views all user accounts, filters by role, adds new users, edits or deletes existing users, and assigns roles.	
Triggering event:	Finance clicks “Manage Users” in the main menu.	
Brief description:	The Finance user accesses a paginated list of all system users (IT, Service, University Staff, Club Coordinators, and other Finance users). Finance can filter by role, search by name or email, click “Edit” to change a user’s role (or reset password), click “Delete” to remove a user, or click “Add New User” to open a form that captures Name, Email, Password, and Role. Upon submission, the system encrypts the password and creates (or updates) the user record.	
Actors:	Primary Actor: Finance (System Administrator)	
Related use cases:		
Stakeholders:	All Departments (user accounts must be accurate so that requests and permissions flow correctly) IT Department (may rely on Finance to create their accounts) University Staff, Club Coordinators (their access depends on correct roles)	
Preconditions:	Finance is authenticated with “Manage Users” permissions. The underlying User table exists, with fields: Name, Email (unique), Encrypted Password, Role.	
Postconditions:	If adding a user: A new user record is created, password is stored as an encrypted hash, and an activation email is sent. If editing a user: That user’s role (or password) is updated, and the change is logged. If deleting a user: That user’s record is removed or flagged as inactive; any open requests or assignments are re-assigned or flagged.	
Flow of activities:	Actor	System

	<ul style="list-style-type: none"> Finance clicks “Manage Users” from the navigation bar. Finance optionally selects a role filter (e.g., “IT”). Finance clicks “Add New User.” Finance fills in the user’s details, selects Role = “Service,” and clicks “Create.” Finance clicks the “Edit” icon next to user “jane.doe@example.edu.” Finance changes “Role” from “Club Coordinator” to “University Staff” and clicks “Save.” Finance clicks the “Delete” icon next to user “john.smith@it.univ.edu.” 	<ul style="list-style-type: none"> System fetches and displays a table of all users, showing columns: Name, Email, Role, Status (Active/Inactive), and action icons (Edit/Delete). System applies filter, re-queries the Users table for Role = “IT,” and updates the table display. System opens a modal or page with a form: <p>Name</p> <p>Email</p> <p>Password</p> <p>Confirm Password</p> <p>Role (dropdown)</p> <p>Buttons: “Create” and “Cancel”</p> <ul style="list-style-type: none"> System validates all fields (e.g., unique email, password match). If validation passes: <p>Encrypts the password</p> <p>Creates new user record in the database</p> <p>Sends activation/notification email to new user</p> <p>Refreshes the user list to include the newly added user</p> <ul style="list-style-type: none"> If validation fails (e.g., email already exists): <p>System displays an appropriate error</p> <ul style="list-style-type: none"> System loads that user’s existing data into an editable form (Name, Email read-only or editable, Role dropdown, optional “Reset Password”). System updates the user’s Role in the database, logs the change, and refreshes the user list.
--	--	---

		<ul style="list-style-type: none"> System prompts: "Are you sure you want to delete john.smith@it.univ.edu? This action cannot be undone." <p>If Finance confirms, proceed to next step</p> <p>If cancels, abort deletion</p> <ul style="list-style-type: none"> System either removes or flags that user as Inactive in the database; any open assignments or requests are reassigned or flagged. Then refresh the list.
Exception conditions:	<p>Duplicate Email on Create/Edit: If Finance enters an email already in use, the system rejects and displays "Email already exists. Choose a different email."</p> <p>Weak Password: If password does not meet complexity rules, the system shows "Password must be at least 8 characters, containing uppercase, lowercase, and a digit."</p> <p>Deletion with Active Dependencies: If the user being deleted has pending requests or open approvals, the system warns "User has pending requests—cannot delete until reassigned."</p>	

4.2.4. Use Case – Process Requests

Table 4.2.4: Fully Developed Use Case Description for “Processing Requests”

Use case name:	Process Requests (IT/Service)
Scenario:	An IT or Service Department user reviews requests (either stock requests or quality-assurance requests) made by University Staff or Club Coordinators, and chooses to Accept, Forward (to Finance), or Decline each.
Triggering event:	A new request (stock or QA) arrives from University Staff or Club Coordinators.
Brief description:	The IT (or Service) user logs in, sees a list of pending requests (displayed on their Homepage as request cards). For each request, the user can “Accept” (meaning the department will fulfill the request), “Forward” (escalate to Finance for budget approval), or “Decline.” Once an action is chosen, the system updates the request status accordingly and notifies the original requester.
Actors:	Primary Actor: IT Department User or Service Department User (depends on which department the request applies to)

Related use cases:					
Stakeholders:	IT Department (responds to IT-related QA/stock requests) Service Department (responds to Service-related QA/stock requests) University Staff and Club Coordinators (requesters) Finance Department (if forwarded)				
Preconditions:	The IT/Service user is authenticated. There is at least one pending request (status = “Pending”) targeted at IT or Service.				
Postconditions:	The request status is updated to “Accepted,” “Forwarded,” or “Declined.” If “Accepted”: The system decrements relevant in-department inventory (or flags item for repair in case of QA). A notification is sent to the requester: “Your request has been accepted by IT (or Service).” If “Forwarded”: The request is re-assigned to Finance with status = “Pending Finance Approval.” A notification is sent to Finance for review. If “Declined”: The request is closed; requester notified of decline.				
Flow of activities:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • IT/Service user clicks “Homepage” in the navigation. • IT/Service user clicks on a request card submitted by “Prof. X.” </td> <td> <ul style="list-style-type: none"> • System retrieves all requests where (Department = “IT” or “Service”) AND status = “Pending,” displaying each as a card with summary details. • System displays full request details (what is being requested, quantity, justification, location, </td> </tr> </tbody> </table>	Actor	System	<ul style="list-style-type: none"> • IT/Service user clicks “Homepage” in the navigation. • IT/Service user clicks on a request card submitted by “Prof. X.” 	<ul style="list-style-type: none"> • System retrieves all requests where (Department = “IT” or “Service”) AND status = “Pending,” displaying each as a card with summary details. • System displays full request details (what is being requested, quantity, justification, location,
Actor	System				
<ul style="list-style-type: none"> • IT/Service user clicks “Homepage” in the navigation. • IT/Service user clicks on a request card submitted by “Prof. X.” 	<ul style="list-style-type: none"> • System retrieves all requests where (Department = “IT” or “Service”) AND status = “Pending,” displaying each as a card with summary details. • System displays full request details (what is being requested, quantity, justification, location, 				

	<ul style="list-style-type: none"> • IT/Service user clicks “Accept.” • OR IT/Service user clicks “Forward.” • OR IT/Service user clicks “Decline.” 	<p>date) and shows three buttons: Accept, Forward, Decline.</p> <ul style="list-style-type: none"> • If “Accept” is clicked: <p>System updates request.status = “Accepted by IT” (or “Accepted by Service”)</p> <p>If stock request: system locates relevant inventory items and reduces available quantity (or flags for reorder if insufficient)</p> <p>If QA request: system flags the item for repair and notifies the maintenance queue</p> <p>System sends notification to the requester: “Your request has been accepted by [Department].”</p> <ul style="list-style-type: none"> • If “Forward” is clicked: <p>System updates request.status = “Forwarded to Finance”</p> <p>Creates a new pending record in Finance’s queue</p> <p>Sends notification to Finance user(s)</p> <p>Notifies original requester: “Your request has been forwarded to Finance for approval.”</p> <ul style="list-style-type: none"> • If “Decline” is clicked: <p>System updates request.status = “Declined by [Department]”</p> <p>Logs reason if entered</p> <p>Sends notification to the requester</p>
Exception conditions:	Insufficient Inventory (upon Accept): If department inventory cannot fulfill the requested quantity, the system:	

	<p>Displays an alert: “Insufficient stock to fulfill request. Consider forwarding to Finance or declining.”</p> <p>Does not let the user confirm “Accept” until they choose to reorder via Finance.</p> <p>Concurrent Processing Conflict: If another IT/Service user already processed the request, the system warns “Request already handled—status is [Accepted/Declined/Forwarded].”</p> <p>Notification Error: If the email server is down, the system logs the failure and queues the notification for retry, showing a message “Unable to send notification now—will retry.”</p>
--	---

4.2.5. Use Case – Process Stock Request (Finance)

Table 4.2.5: Fully Developed Use Case Description for “Processing Stock Request by Finance”

Use case name:	Process Stock Request (For Finance)
Scenario:	Finance reviews stock requests submitted by IT and Service Departments and decides whether to approve, pend, or decline each request.
Triggering event:	A new stock request is submitted by an IT or Service-Department user.
Brief description:	The Finance user (acting as system administrator) logs into the system, views incoming stock requests, and for each request chooses one of three actions—Approve, Pending, or Decline. Once an action is chosen, the system updates the request’s status, adjusts any related inventory or budget records if approved, and sends an automated notification (email or in-system message) back to the requester.
Actors:	Primary Actor: Finance (System Administrator)
Related use cases:	
Stakeholders:	Finance Department (needs to track and authorize purchases) IT Department (requests new stock) Service Department (requests new stock) University Accounting (needs accurate budget and inventory records)
Preconditions:	The Finance user is authenticated and has “System Admin” privileges. There is at least one pending stock request submitted by either IT or Service.



	Inventory catalog and budget information are up to date.	
Postconditions:	<p>The selected request's status is updated to Approved, Pending, or Declined.</p> <p>If Approved:</p> <p>The requester's department's "pending procurement" record is updated.</p> <p>The inventory re-ordering workflow is triggered.</p> <p>If Pending:</p> <p>The request remains open and flagged for follow-up (e.g., awaiting funds or supplier availability).</p> <p>If Declined:</p> <p>The request is closed without procurement.</p> <p>An automatic notification is sent to the original requester indicating the new status (Approved, Pending, or Declined).</p>	
Flow of activities:	Actor	System
	<ul style="list-style-type: none">Finance navigates to the "Homepage" page where pending requests are listed.Finance clicks on a specific request card.Finance reviews the details and clicks "Approve," "Pending," or "Decline."(If "Approve") Finance confirms procurement details (e.g., budget code).	<ul style="list-style-type: none">System retrieves all stock-request records with status = "Pending" and displays them as cards (one per request), showing requester name, department, item details, and justification.System opens a detailed view of the request, showing full item list, quantities, reason, date submitted, and requester contact.System updates the request record in the database with the chosen status.System: <p>Marks request as "Approved."</p> <p>Automatically deducts estimated cost from Finance's available budget.</p>

	<p>Flags items for purchase order generation.</p> <p> 5. (If “Pending”) Finance enters a comment or note (e.g., “Awaiting supplier quote”). 5.1 System saves the “Pending” status and the comment, leaving the request in pending state for later action.</p> <p> 6. (If “Decline”) Finance may optionally enter a reason for decline.</p> <p> 6.1 System sets status = “Declined” and records the reason.</p> <p> 7. Finance clicks “Send Notification.” 7.1 System sends an automated email (and/or in-system notification) to the original requester with their new request status and any comments.</p>
Exception conditions:	<p>Invalid Request ID: If the request ID is missing or malformed, the system displays an error: “Request not found.” Finance must choose another request to process.</p> <p>Insufficient Budget (upon Approve): If the Finance user’s budget is insufficient to cover the total, the system:</p> <p>Rejects the “Approve” action, displays an error message (“Insufficient funds—please adjust amount or choose Pending”) and suggests marking the request as “Pending.”</p> <p>Notification Failure: If the email server is unreachable, the system queues the notification; Finance is shown a warning (“Notification pending—will retry automatically”).</p> <p>Concurrent Update Conflict: If another admin already processed the request moments ago, the system informs Finance that the request’s status has changed; Finance must refresh the list.</p>

4.2.6. Use Case – Submit QA Request

Table 4.2.6: Fully Developed Use Case Description for “Submitting QA Request”

Use case name:	Submit QA Request
Scenario:	University Staff or Club Coordinator identifies malfunctioning or substandard equipment in their assigned locations and submit a QA request to the IT or Service Department.

Triggering event:	Clicks the “QA Request” icon (from their homepage or the combined page).	
Brief description:	A University Staff or Club Coordinator user chooses “QA Request,” selects the location where the QA is needed (building, room), specifies the item(s) or equipment requiring QA, enters a justification, and clicks “Submit.” The system creates a new QA request record addressed to the appropriate department (either IT or Service, based on the item category) with status = “Pending.” The request appears in the IT/Service queue, and the requester is notified.	
Actors:	Primary Actor: University Staff and Club Coordinator	
Related use cases:		
Stakeholders:	University Staff /Club Coordinator(needs malfunctioning equipment fixed) IT/Service Departments (will inspect/repair the item) Finance Department (if a purchase or budget is needed post-QA)	
Preconditions:	University Staff/Club Coordinator is authenticated. The staff has assigned locations (e.g., an office or classroom) where they can identify items to report. The system’s inventory is up to date.	
Postconditions:	A new QA request record is created with fields: Requester ID, Location (e.g., “Office 212”), Item ID (or description), Justification, Target Department (“IT” or “Service”), Status = “Pending,” Timestamp. The request appears in the IT/Service queue. An automatic notification is sent to the IT/Service department user(s).	
Flow of activities:	Actor	System
	<ul style="list-style-type: none"> Actor clicks “QA Request” on their homepage. 	<ul style="list-style-type: none"> System displays a “QA Request” form with fields: <p>Location (dropdown populated with that staff’s assigned locations, e.g., Office 212)</p> <p>Item (dropdown or text; if a dropdown, lists all items in that location)</p> <p>Justification/Description (text area)</p>



		<p>Target Department (automatically set based on item category but editable)</p> <p>Submit/Cancel buttons.</p> <ul style="list-style-type: none">• Staff selects “Office 212,” chooses “Projector Model X,” enters “No power” as justification, and clicks “Submit.” 2.1 System validates all required fields. <p>- If validation passes:</p> <p>Creates a QA Request record with status = “Pending,” linking to the item, location, requester, and timestamp.</p> <p>Notifies the designated department user(s) (IT or Service) via email/in-system.</p> <p>Shows confirmation: “QA Request submitted successfully. You will be notified when the department responds.”</p> <p>- If validation fails, system shows errors (e.g., “Please select an item”).</p> <ul style="list-style-type: none">• Staff logs out or returns to Homepage. 3.1 System retains QA request for action by IT/Service.
Exception conditions:	<p>Validation Failure: If Staff tries to submit without selecting an item or location, system displays “All fields are required.”</p> <p>Invalid Item: If Staff types a free-text item not found in inventory, system prompts “Item not recognized—please choose from list or contact Service.”</p> <p>Notification Failure: If the department notification fails, system queues it and shows “Request saved; notification will be sent when the system is available.”</p>	



4.2.7. Use Case – Submit Stock Request

Table 4.2.7: Fully Developed Use Case Description for “Submitting Stock Request”

Use case name:	Submit Stock Request	
Scenario:	Users who need new items (IT, Service, University Staff, or Club Coordinator) fill out and submit a stock request form.	
Triggering event:	A user clicks “Stock Request” from their homepage or from the combined QA/Stock page.	
Brief description:	The actor (role varies) opens a form to specify the item(s) they require, quantity, justification, and which department should receive the request (IT or Service). After completing the form and clicking “Submit,” the system creates a new request record, assigns it the status “Pending,” and routes it to the appropriate department’s queue. An email/in-system notification is sent to that department.	
Actors:	Primary Actor: IT Department User or Service Department User (depends on which department the request applies to)	
Related use cases:		
Stakeholders:	IT Department (processes or forwards the request) Service Department (same) Finance Department (if forwarded for budget approval) Requesting actor’s department (needs items)	
Preconditions:	The user is authenticated (any of the four roles). At least one valid “Department” option is available (IT or Service).	
Postconditions:	A new stock request record is created in the database with fields: Requester ID, Department (target), Item Description, Quantity, Justification, Date Submitted, and Status = “Pending.” The request appears on the target department’s Homepage under pending requests. A notification (email or in-system) is sent to the target department user(s).	
Flow of activities:	Actor	System
	<ul style="list-style-type: none">Actor clicks “Stock Request” (or, for University Staff, selects the stock icon on the combined page).	<ul style="list-style-type: none">System displays a blank “Stock Request” form with fields: Item Name / Description (text)

	<p>Quantity (numeric)</p> <p>Justification (text area)</p> <p>Target Department (dropdown: “IT” or “Service”)</p> <p>Submit and Cancel buttons.</p> <ul style="list-style-type: none"> Actor enters “Wireless Keyboard,” Quantity = 10, Justification = “Lab upgrades,” selects Department = “IT,” and clicks “Submit.” <p>System validates that all mandatory fields are filled and quantity is > 0.</p> <p>If validation passes:</p> <p>System creates a new Request record in the database with status = “Pending,” storing all fields plus timestamp and ActorID.</p> <p>System sends a notification to the IT department queue and to any designated “IT Approver” role.</p> <p>System displays a confirmation message: “Request submitted successfully. You will be notified when the status changes.”</p> <p>- If validation fails: displays appropriate error (e.g., “Quantity must be at least 1”).</p> <ul style="list-style-type: none"> Actor logs out or navigates away. <p>System retains the request for later processing by IT/Service.</p>
Exception conditions:	<p>Validation Error: If “Quantity” is missing or ≤ 0, system shows “Quantity must be a positive number.” If “Item Name” is blank, “Please describe the item.”</p> <p>System Error on Save: If database insert fails, system displays “Unable to submit request—please try again.”</p>

	Notification Failure: If email/in-system notification to the target department is down, system queues the notification and shows “Request saved, but notification will retry shortly.”
--	--

4.2.8. Use Case – View and Update Item State

Table 4.2.8 : Fully Developed Use Case Description for “Viewing and updating Item State”

Use case name:	View and Update Item State
Scenario:	A University Staff or Club Coordinator browses their assigned location(s), sees a list of items, and updates the condition/status of any item (e.g., marking it “Barely Working” or “Not Working”). If an item’s updated state equals “Not Working,” the system automatically generates a new QA request. Triggering Event: The user (University Staff or Club Coordinator) logs in and clicks their “Homepage,” then selects their assigned building/location.
Triggering event:	The user (University Staff or Club Coordinator) logs in and clicks their “Homepage,” then selects their assigned building/location/Item.
Brief description:	On their personalized homepage, University Staff either sees a direct listing of items for their assigned office, or they first select their building, then room. Club Coordinators see their club’s inventory. In either case, a table or card view shows each item’s name, quantity, and current state. For each item, the user can click an “Edit” or “Update State” icon, select a new state, and click “Save.” If the new state is “Not Working,” the system auto-creates a QA Request to the appropriate department. Otherwise, the system simply updates the record.
Actors:	Primary Actor: University Staff and Club Coordinator
Related use cases:	QA Request
Stakeholders:	University Staff (ensures equipment in their office/class is accurately flagged) Club Coordinators (manage club assets) IT/Service Departments (handle QA requests when items are broken) Finance Department (needs accurate condition data for budgeting/replacement)
Preconditions:	User (University Staff or Club Coordinator) is authenticated. The system knows the user’s assigned location(s) (e.g., Staff A is assigned to Office 212; Club B is assigned “Club Room 5”).

	Inventory records for those locations exist.					
Postconditions:	<p>The selected item's "Condition" field is updated in the database.</p> <p>If condition = "Not Working," a QA Request is automatically generated (in back end) assigned to the correct department (IT or Service).</p> <p>A notification ("You flagged [Item] as 'Not Working.' A QA request has been created.") appears to confirm.</p>					
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • University Staff/Club Coordinator clicks "Homepage." • Actor selects "Office 212" (if needed). • Actor clicks the "Update State" button for "Desk PC #3." • Actor changes state to "Not Working" and clicks "Save." </td> <td> <ul style="list-style-type: none"> • System determines the actor's role and assigned location(s). <p>For Staff: if multiple, system prompts to select (e.g., Building A → Office 212)</p> <p>For Club Coordinator: goes directly to "Club Room 5"</p> <ul style="list-style-type: none"> • System retrieves all inventory items where Location = "Office 212," and displays them in a table with columns: <p>Item Name</p> <p>Quantity</p> <p>Current State</p> <p>"Update State" action button</p> <ul style="list-style-type: none"> • System opens an inline editor or modal showing current state (e.g., "Good Condition") and a dropdown to change to: <p>{New, Good, Barely Working, Not Working}</p> <ul style="list-style-type: none"> • System updates the "Condition" field for "Desk PC #3" in the inventory table. • System recognizes new state = "Not Working," so: <p>Automatically creates a QA Request record with:</p> </td></tr> </tbody> </table>	Actor	System	<ul style="list-style-type: none"> • University Staff/Club Coordinator clicks "Homepage." • Actor selects "Office 212" (if needed). • Actor clicks the "Update State" button for "Desk PC #3." • Actor changes state to "Not Working" and clicks "Save." 	<ul style="list-style-type: none"> • System determines the actor's role and assigned location(s). <p>For Staff: if multiple, system prompts to select (e.g., Building A → Office 212)</p> <p>For Club Coordinator: goes directly to "Club Room 5"</p> <ul style="list-style-type: none"> • System retrieves all inventory items where Location = "Office 212," and displays them in a table with columns: <p>Item Name</p> <p>Quantity</p> <p>Current State</p> <p>"Update State" action button</p> <ul style="list-style-type: none"> • System opens an inline editor or modal showing current state (e.g., "Good Condition") and a dropdown to change to: <p>{New, Good, Barely Working, Not Working}</p> <ul style="list-style-type: none"> • System updates the "Condition" field for "Desk PC #3" in the inventory table. • System recognizes new state = "Not Working," so: <p>Automatically creates a QA Request record with:</p>	
Actor	System					
<ul style="list-style-type: none"> • University Staff/Club Coordinator clicks "Homepage." • Actor selects "Office 212" (if needed). • Actor clicks the "Update State" button for "Desk PC #3." • Actor changes state to "Not Working" and clicks "Save." 	<ul style="list-style-type: none"> • System determines the actor's role and assigned location(s). <p>For Staff: if multiple, system prompts to select (e.g., Building A → Office 212)</p> <p>For Club Coordinator: goes directly to "Club Room 5"</p> <ul style="list-style-type: none"> • System retrieves all inventory items where Location = "Office 212," and displays them in a table with columns: <p>Item Name</p> <p>Quantity</p> <p>Current State</p> <p>"Update State" action button</p> <ul style="list-style-type: none"> • System opens an inline editor or modal showing current state (e.g., "Good Condition") and a dropdown to change to: <p>{New, Good, Barely Working, Not Working}</p> <ul style="list-style-type: none"> • System updates the "Condition" field for "Desk PC #3" in the inventory table. • System recognizes new state = "Not Working," so: <p>Automatically creates a QA Request record with:</p>					

	<p>Requester = current user</p> <p>Item ID = “Desk PC #3”</p> <p>Location = “Office 212”</p> <p>Department = “IT” (since it’s a computer)</p> <p>Justification = “Marked not working by staff”</p> <p>Status = “Pending”</p> <p>Sends notification to IT queue</p> <ul style="list-style-type: none"> • System updates the inventory view to reflect “Not Working.” • System displays confirmation: <p>“Item state updated to ‘Not Working.’ QA request #12345 has been generated.”</p>
Exception conditions:	<p>Concurrent Update: If another user just updated the same item’s state, the system warns “This item’s condition was just changed by someone else—please refresh.”</p> <p>Invalid State Transition: If user tries to change from “Not Working” directly back to “New,” system allows it but logs a comment (“Possible data inconsistency—please confirm inventory physically”).</p> <p>Automatic QA Failure: If the automatic QA Request creation fails (e.g., database queue is down), system still updates the item state but shows “Item state saved, but unable to generate QA request. Please contact IT manually.”</p>

4.3. Use Case Diagrams

4.3.1. Use Case Diagram – Club Coordinator

The following diagram plays a critical role in the software design phase, as it visualizes the end-to-end interaction between the *Club Coordinator* and other system components within the University Inventory Management System. By clearly mapping out the flow of actions, decisions, and interdepartmental communication, it helps identify system responsibilities, refine functional requirements, and ensure proper role-based access.

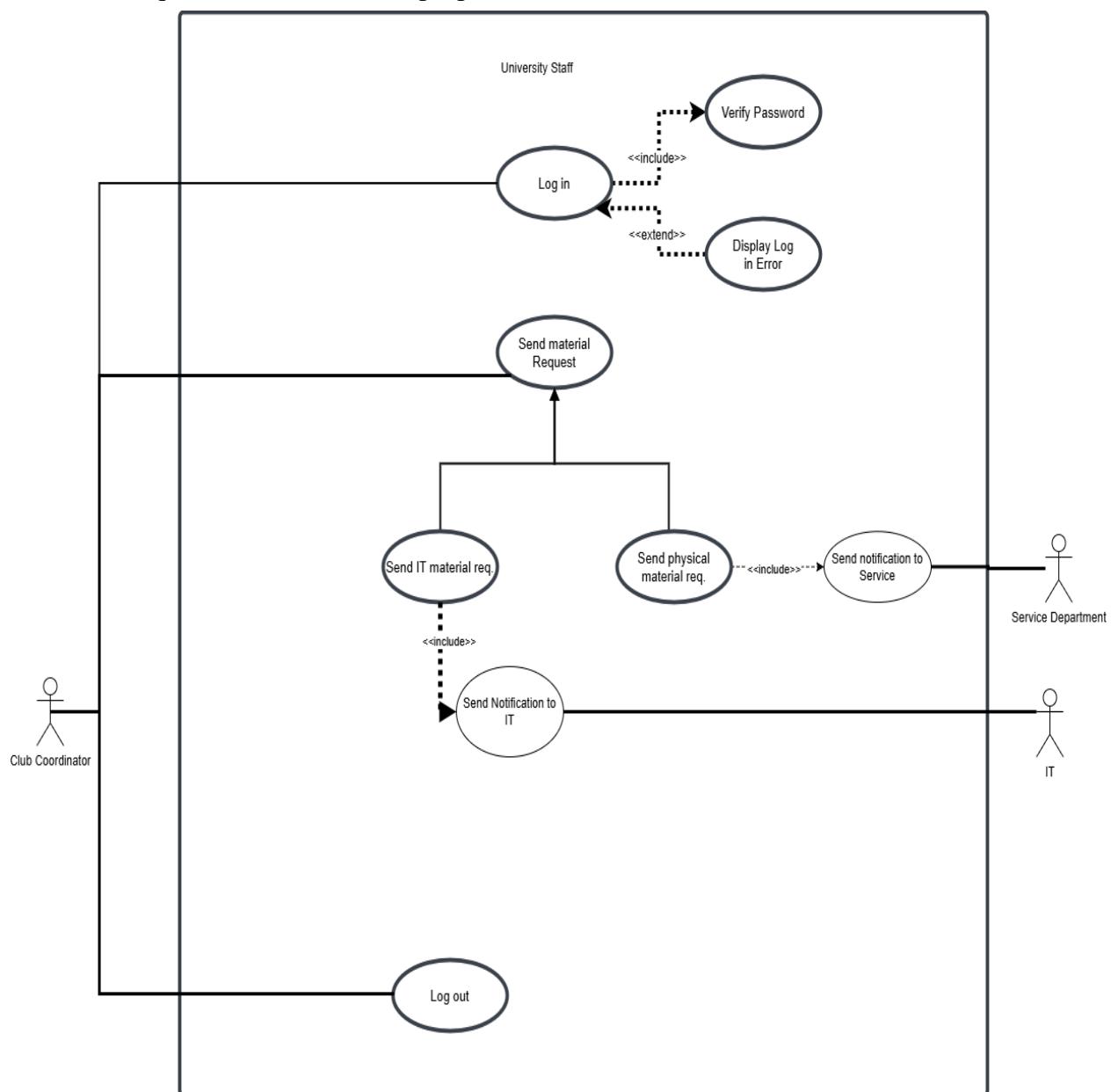


Figure 4.3.1: Use Case Diagram for Club Coordinator Request Handling

4.3.2. Use Case Diagram – Finance (Admin)

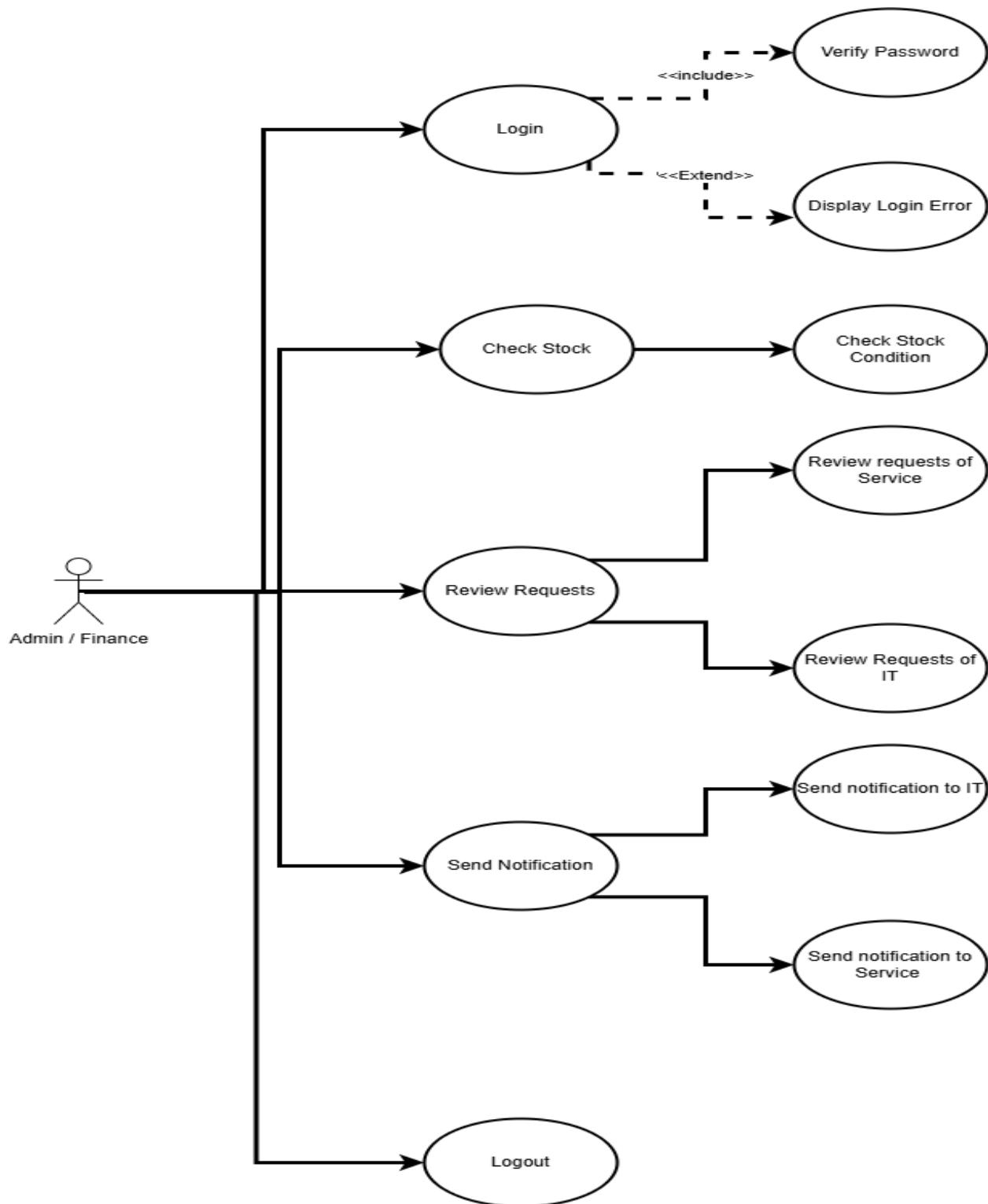


Figure 4.3.2: Use Case Diagram for Finance

4.3.3. Use Case Diagram – IT

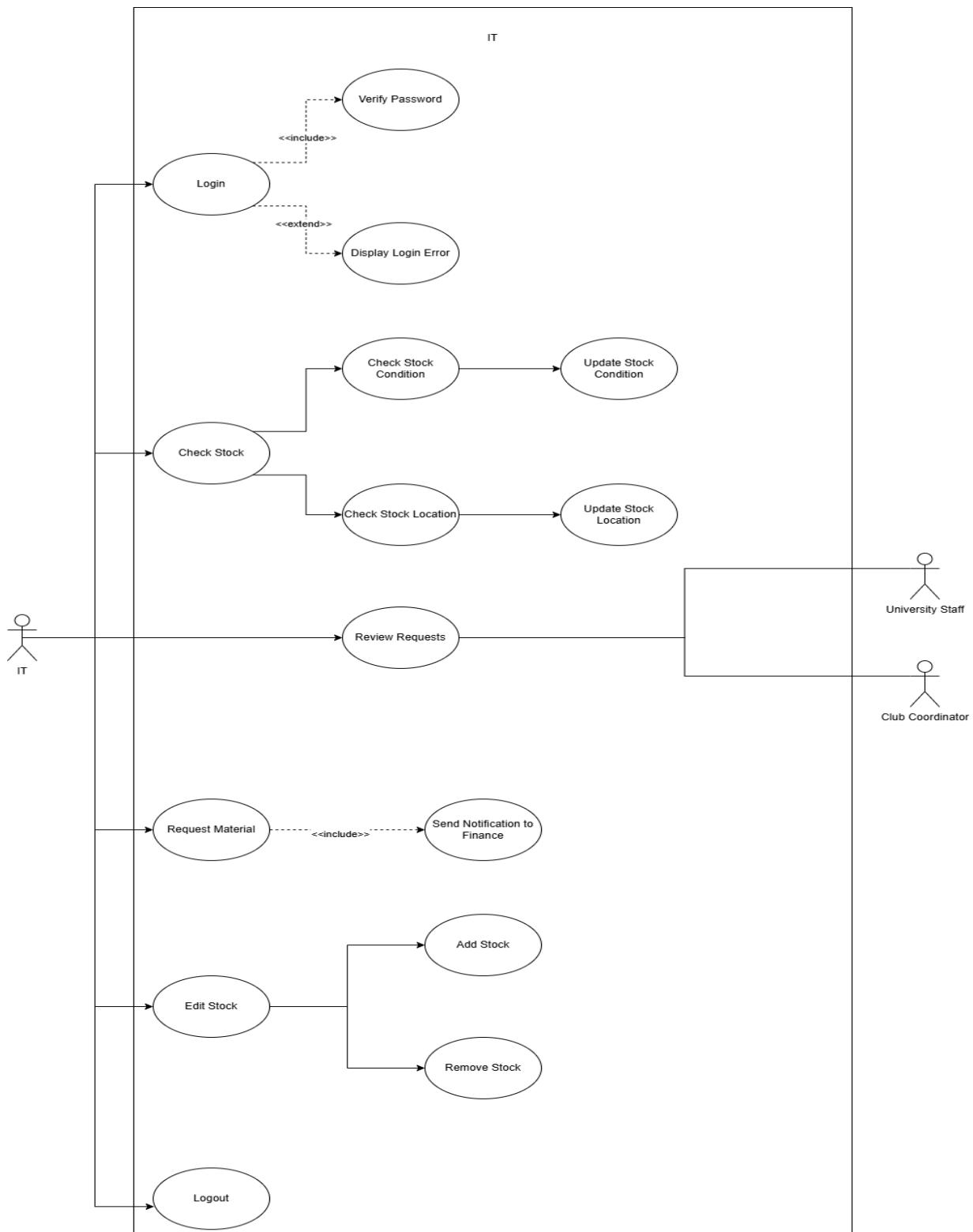


Figure 4.3.3: Use Case Diagram for IT

4.3.4. Use Case Diagram – Service Department

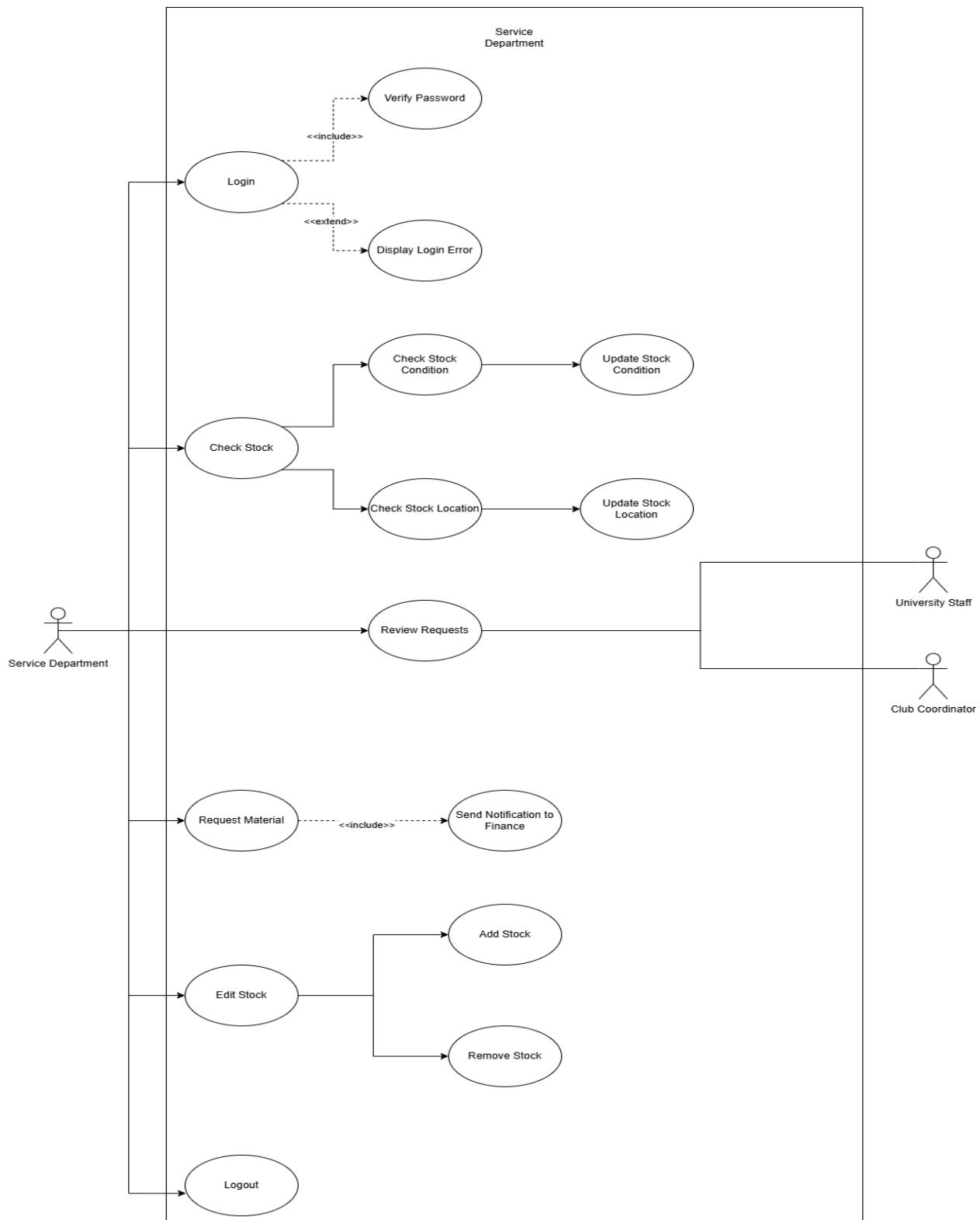


Figure 4.3.4: Use Case Diagram for Service Department

4.3.5. Use Case Diagram – University Staff

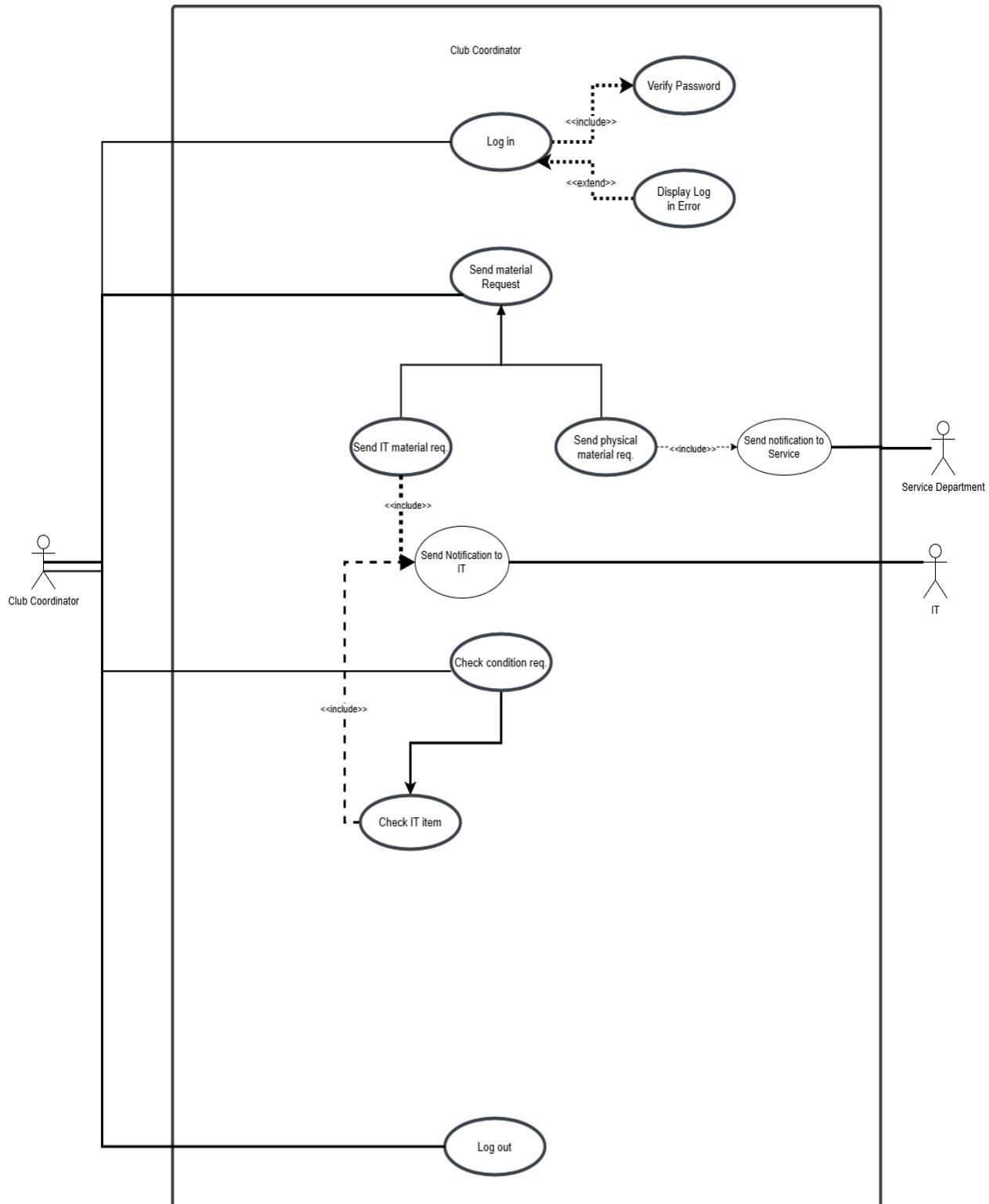


Figure 4.3.5: Use Case Diagram for University Staff

4.4. Activity Diagrams

4.4.1. Activity Diagram for “Login”

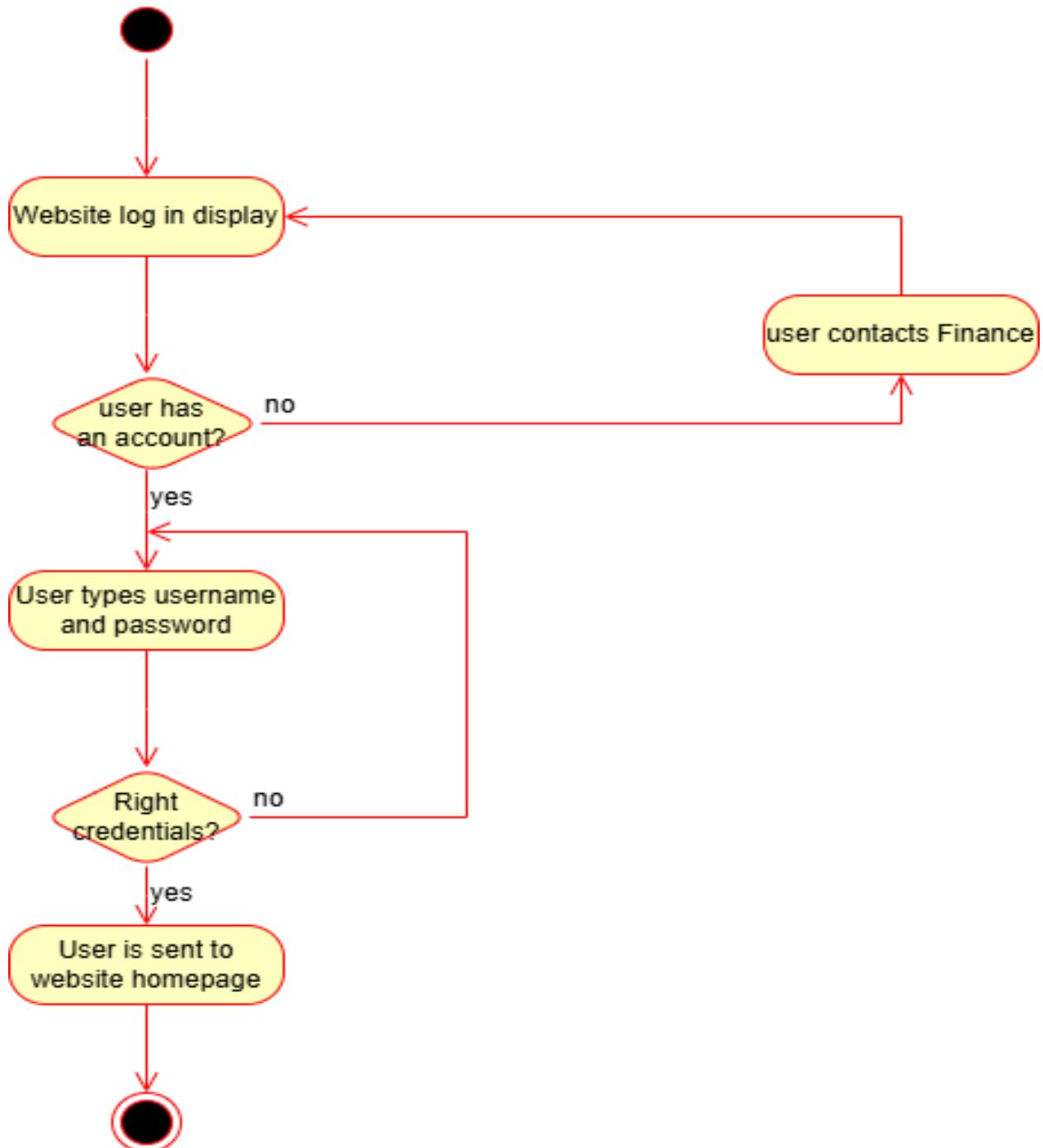


Figure 4.4.1: Activity Diagram for “Login”

4.4.2. Activity Diagram for “Change password”

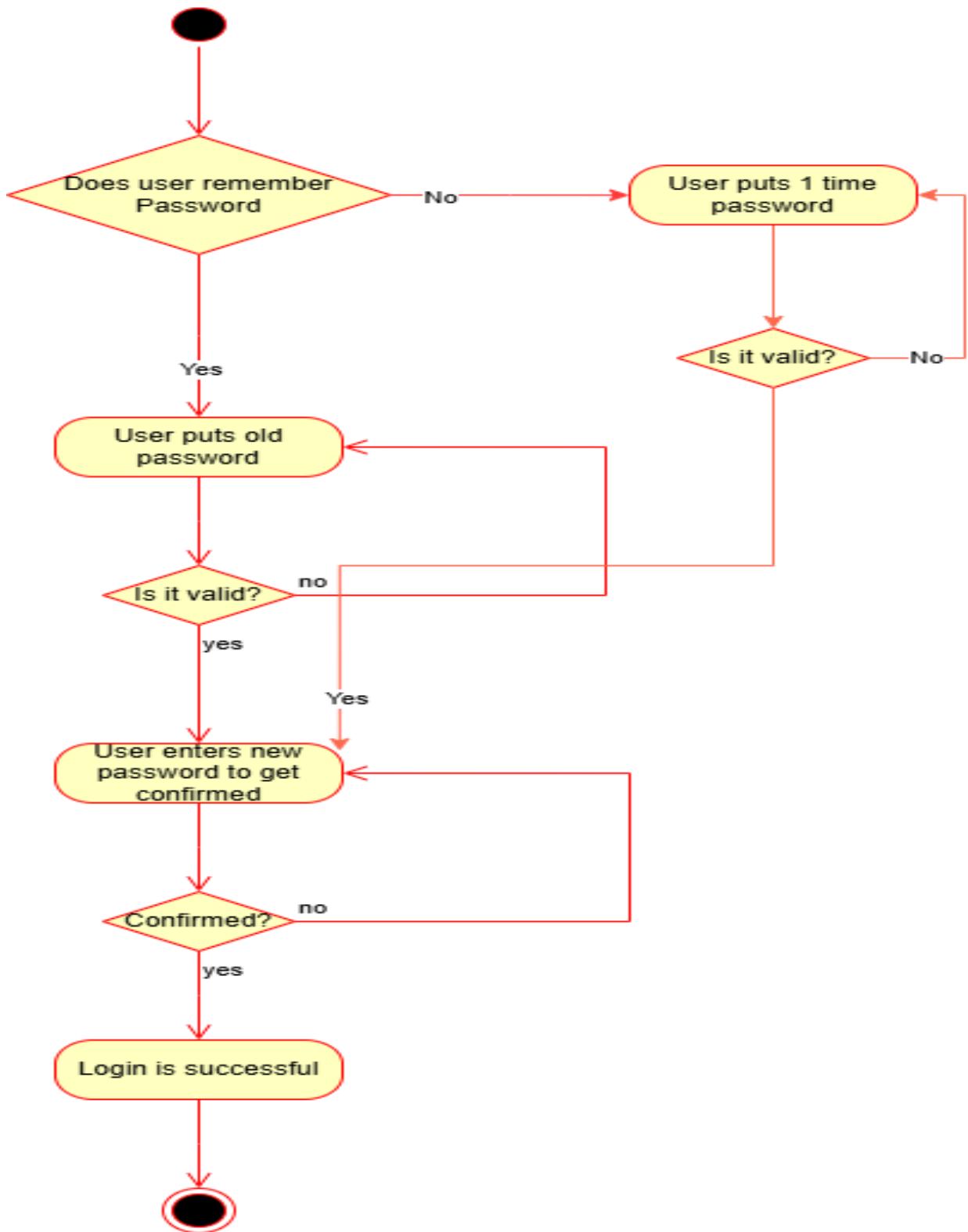


Figure 4.4.2: Activity Diagram for “Change Password”

4.4.3. Activity Diagram for “Add or Delete Stock”

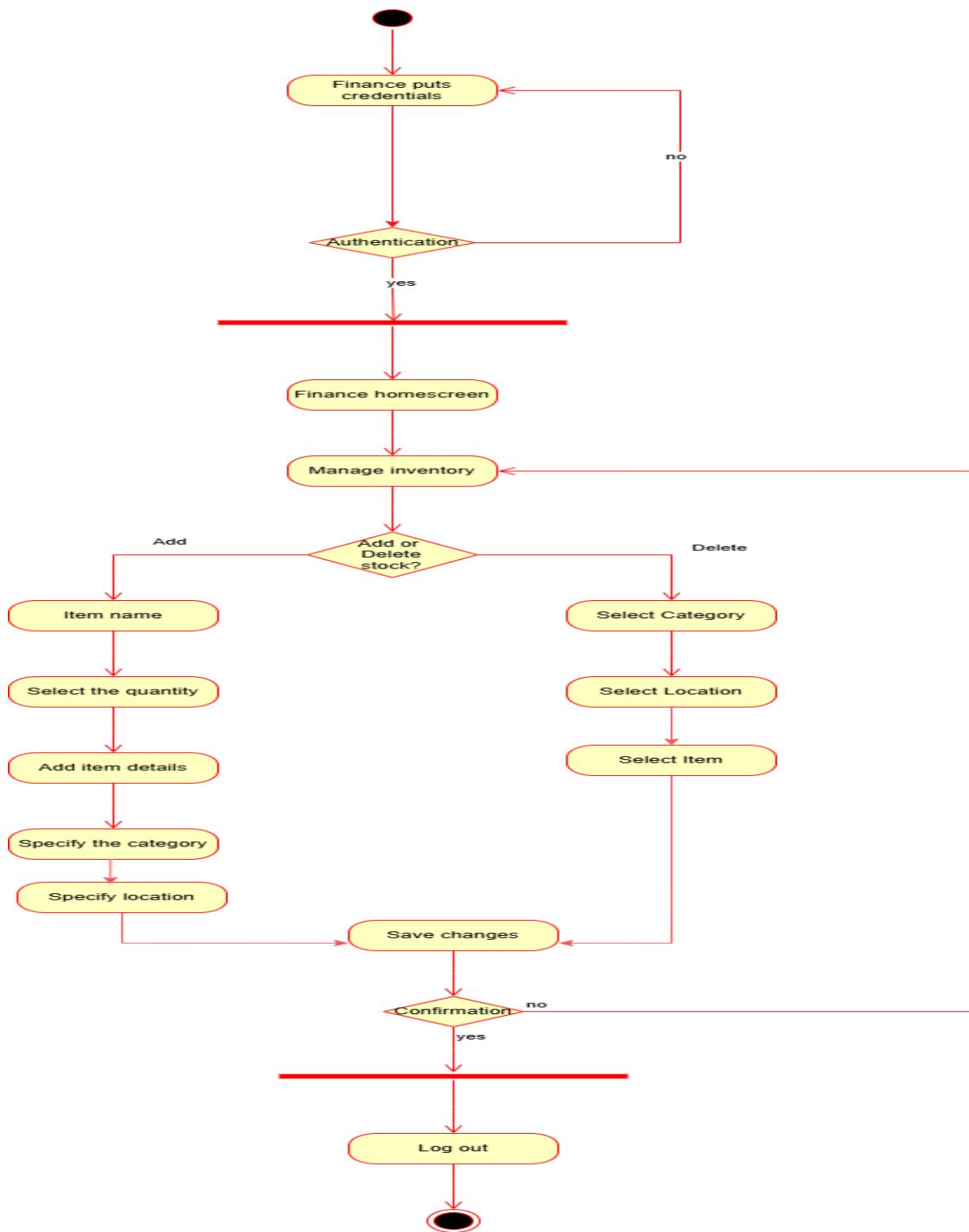


Figure 4.4.3: Activity Diagram for “Add or Delete Stock”

4.4.4. Activity Diagram for “Request Item from IT”

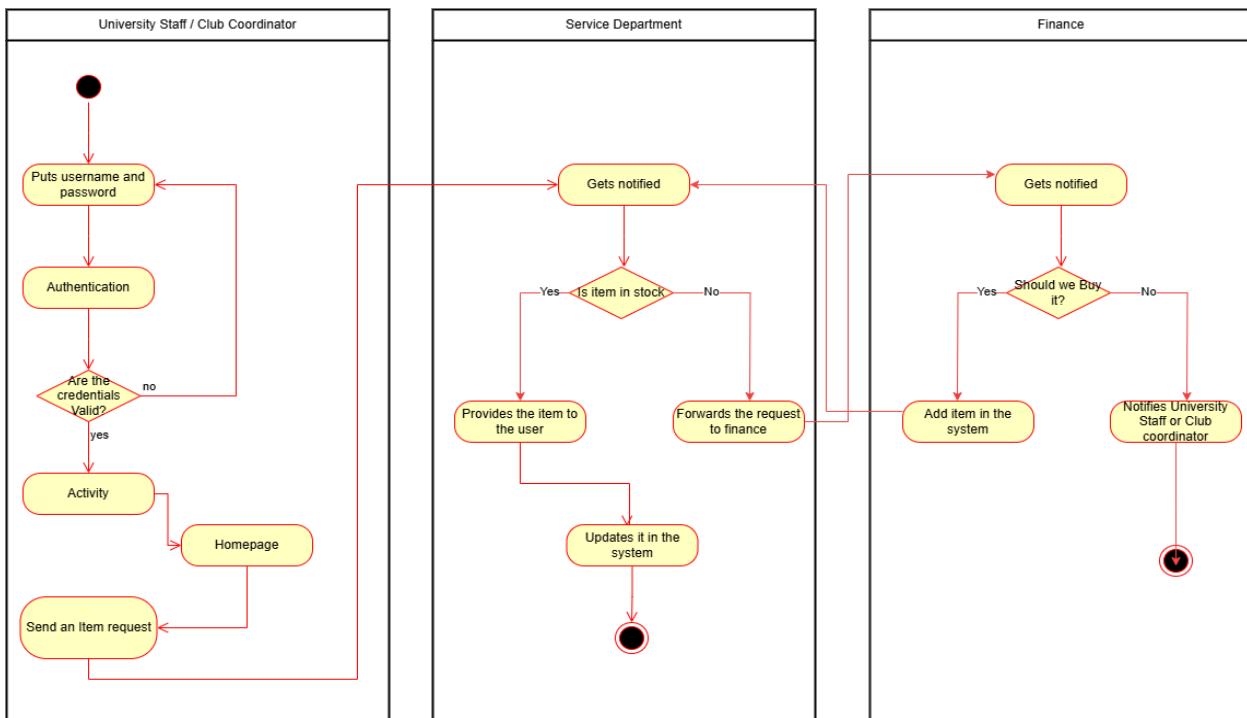


Figure 4.4.4: Activity Diagram for “Request Item from IT”

4.4.5. Activity Diagram for “Request Item from Service Department”

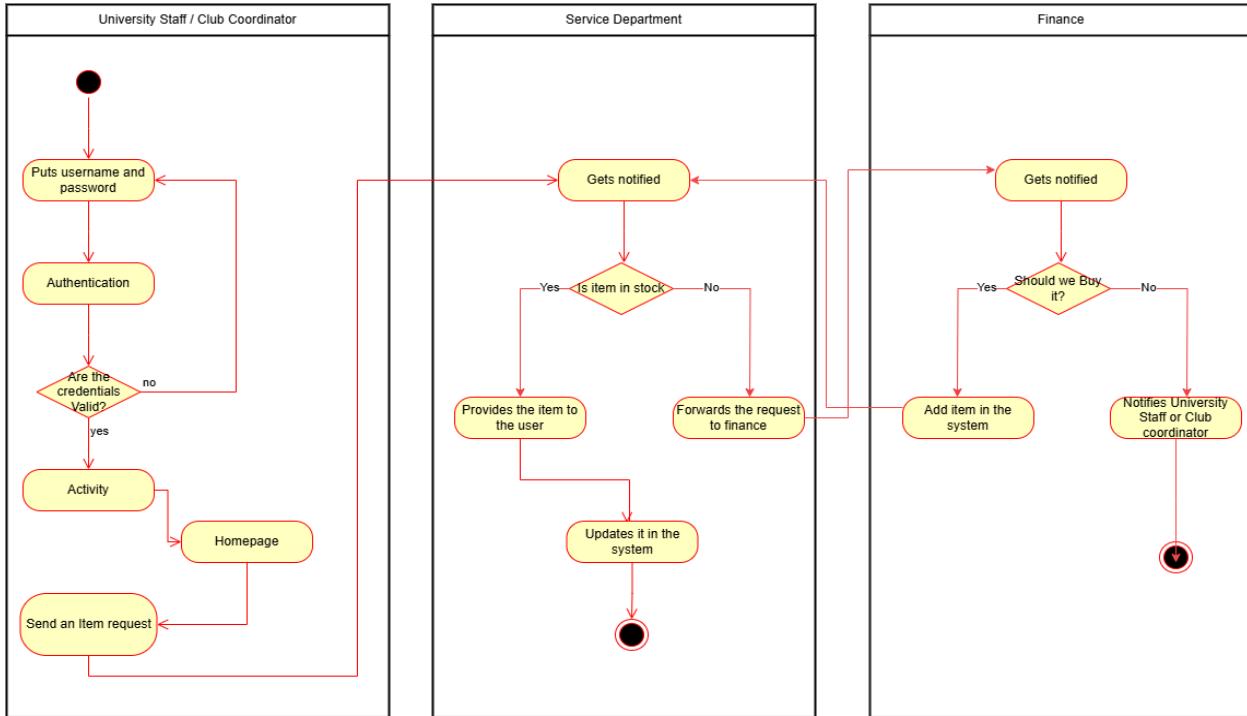


Figure 4.4.5: Activity Diagram for “Request Item from Service Department”

4.4.6. Activity Diagram for “Quality Check”

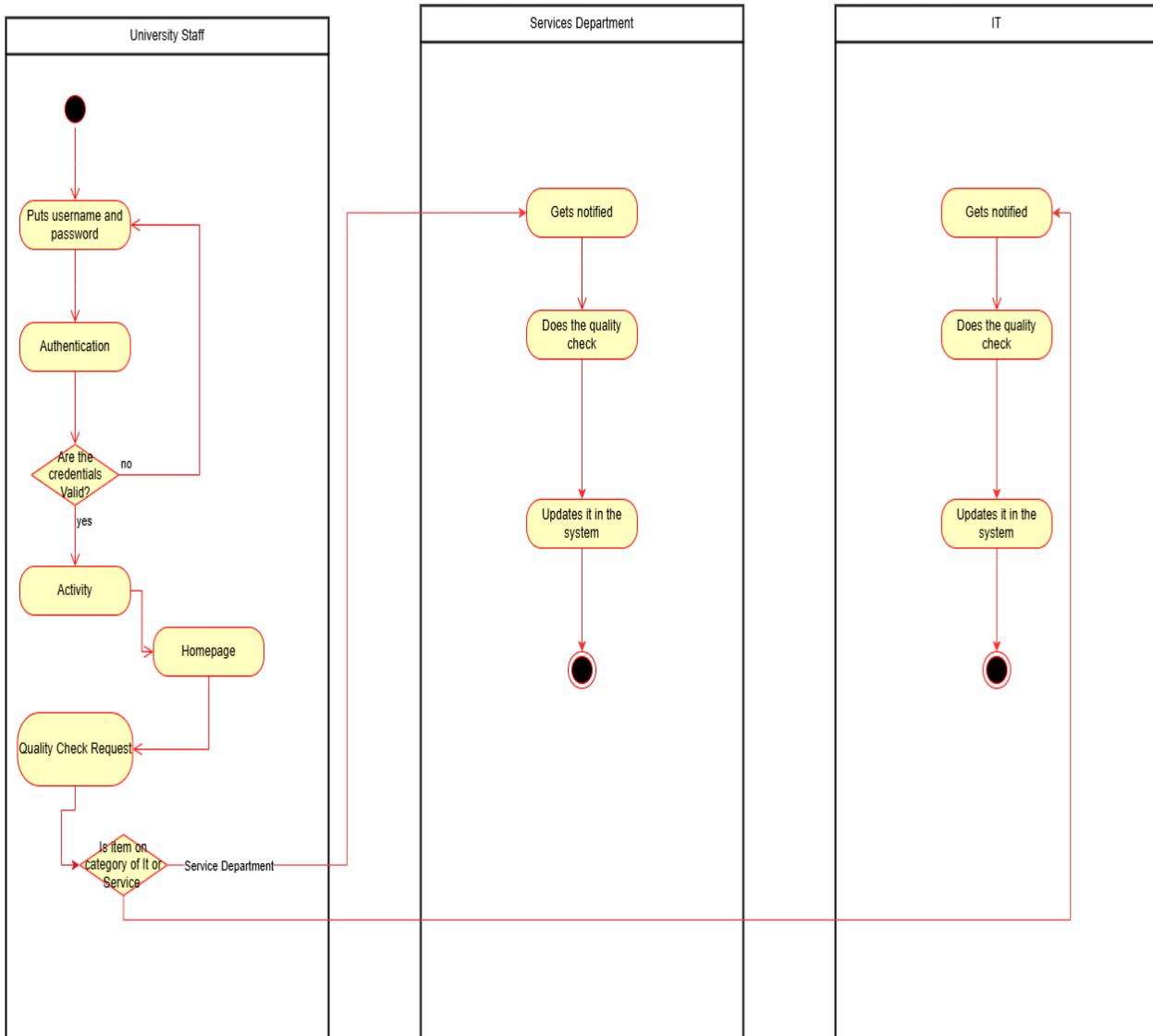


Figure 4.4.6: Activity Diagram for “Quality Check”

4.5. State Diagrams

4.5.1. State Diagram – Club Coordinator Session Flow

The following state diagram models the session flow of the *Club Coordinator* in the University Inventory System. It captures key states such as login, dashboard access, item request submission, and logout. This diagram is essential for visualizing system responses to user actions and ensuring accurate behavior during implementation.

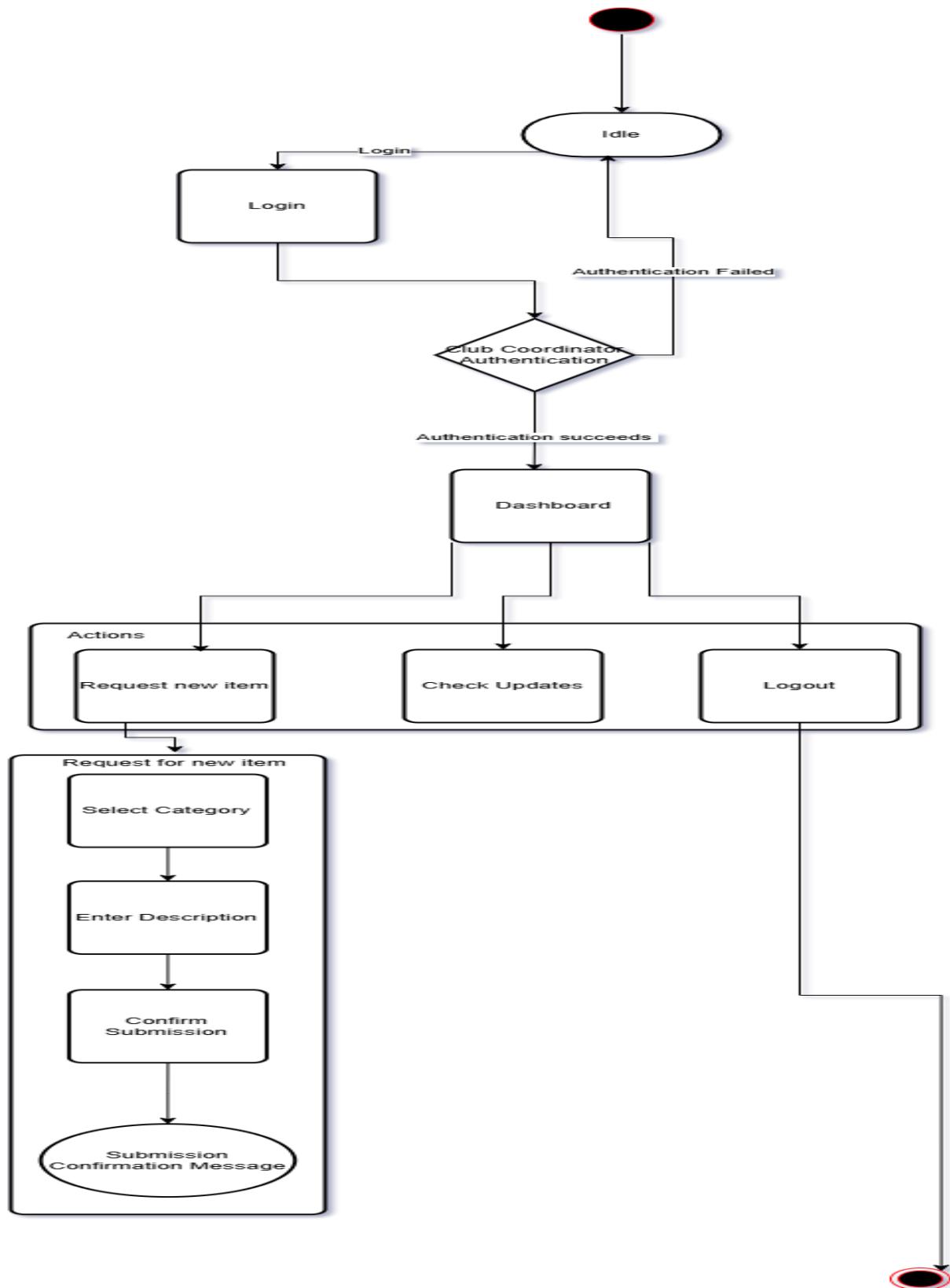


Figure 4.5.1 – State transitions for the Club Coordinator role

4.5.2. State Diagram – Finance Session Flow

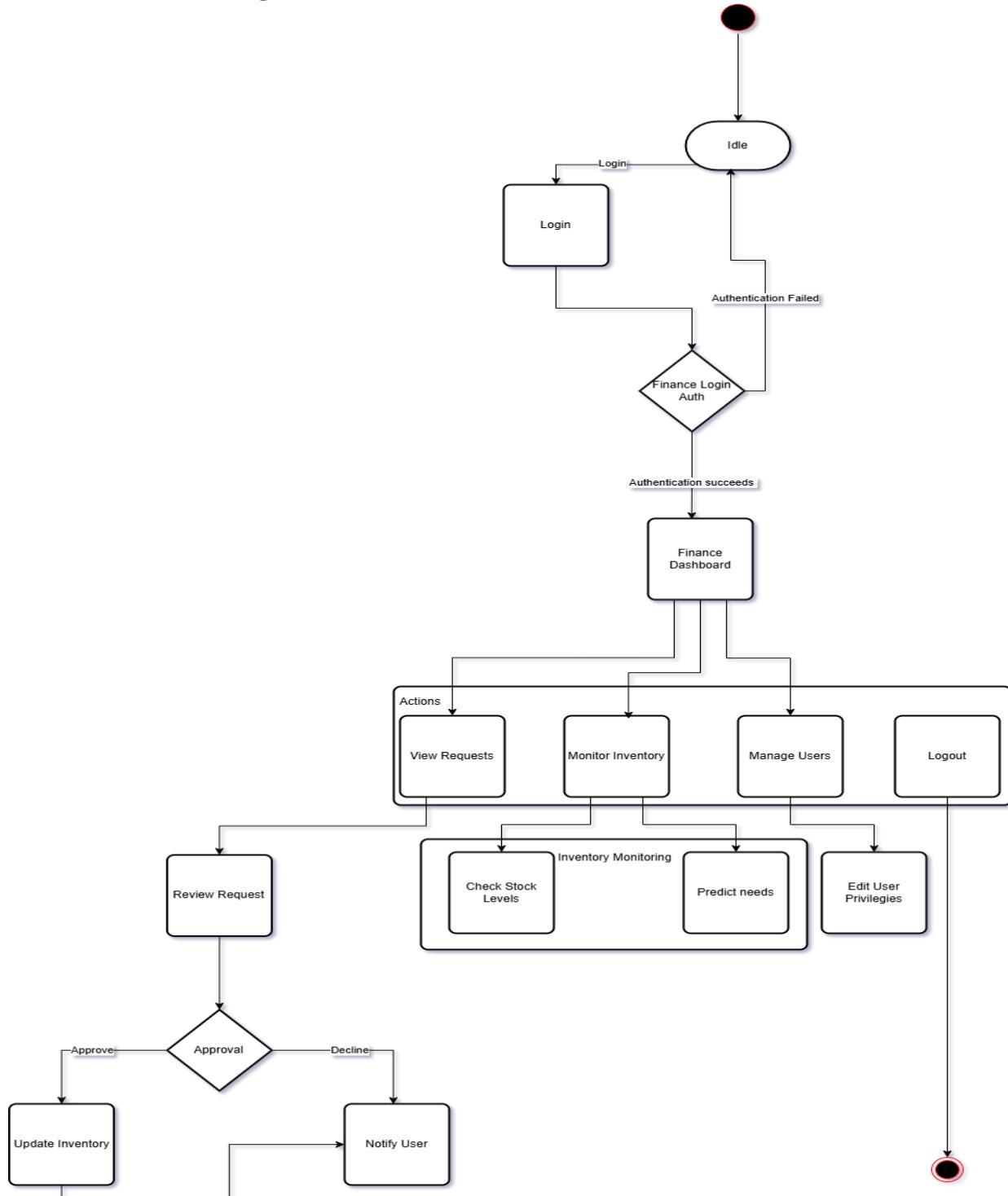


Figure 4.5.2 – State transitions for the Finance role

4.5.3. State Diagram – IT Session Flow

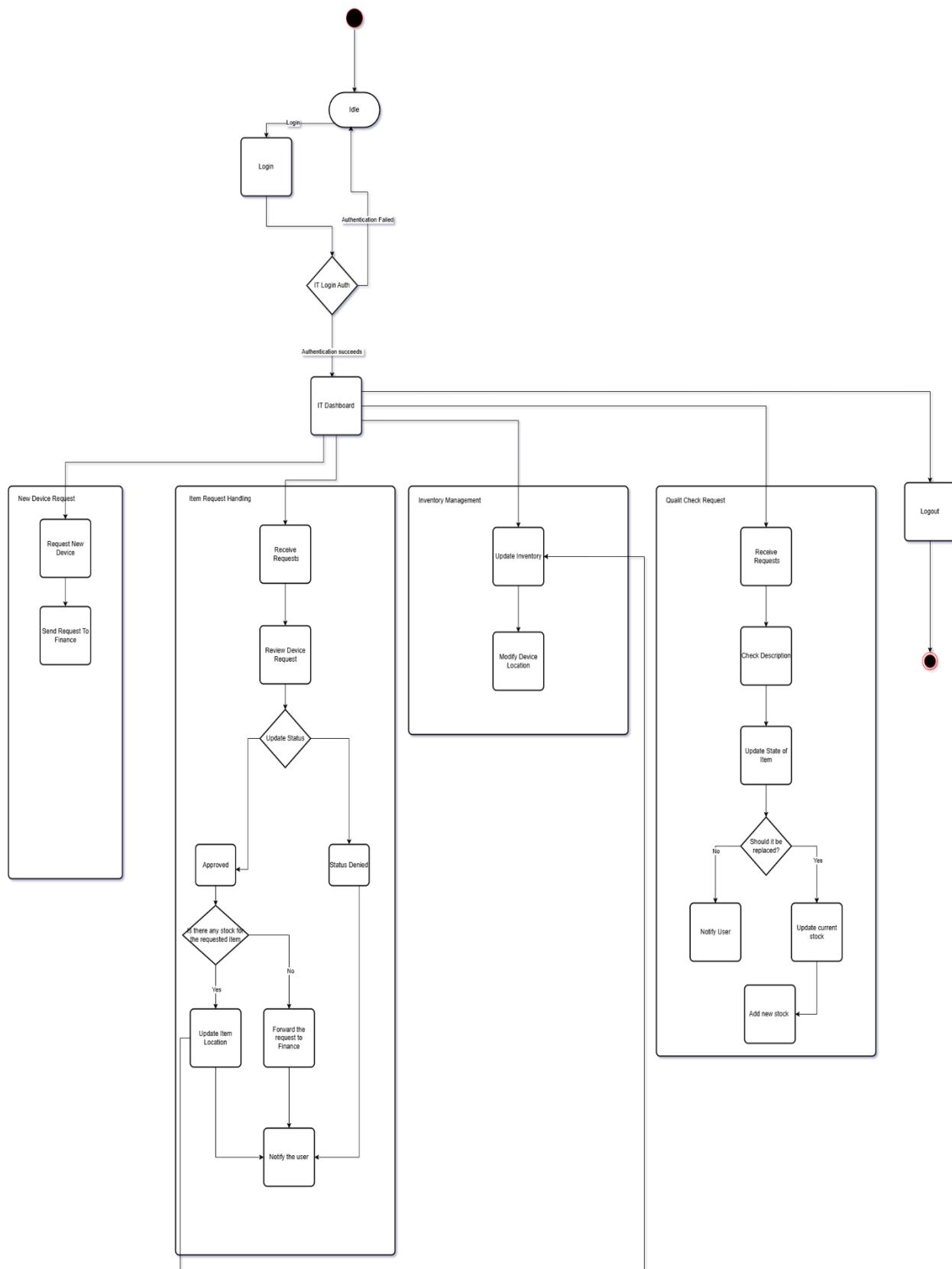


Figure 4.5.3: State Transitions for the IT Role

4.5.4. State Diagram – Service Department Session Flow

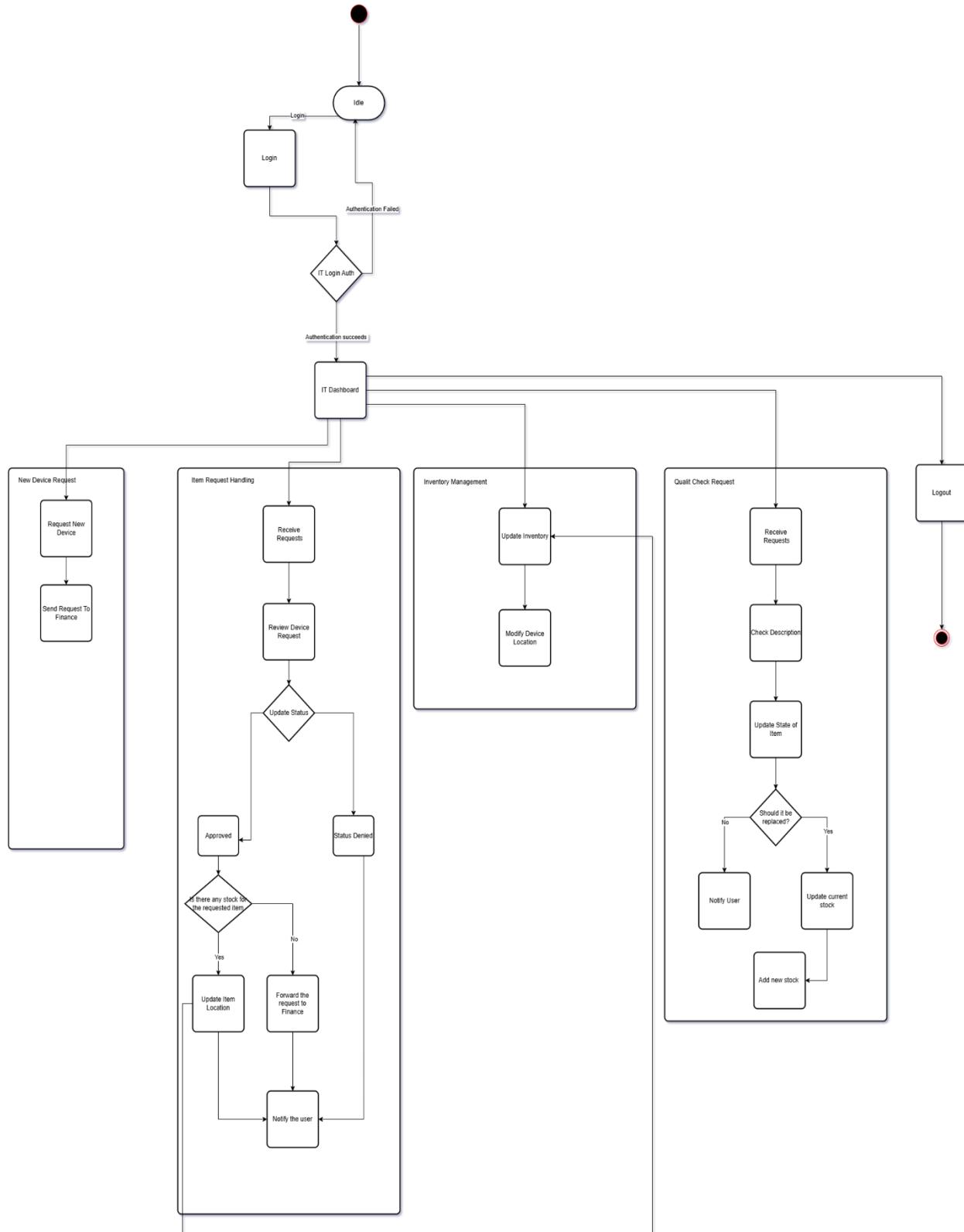


Figure 4.5.4: State Transitions for the Service Department Role

4.5.5. State Diagram – University Staff Session Flow

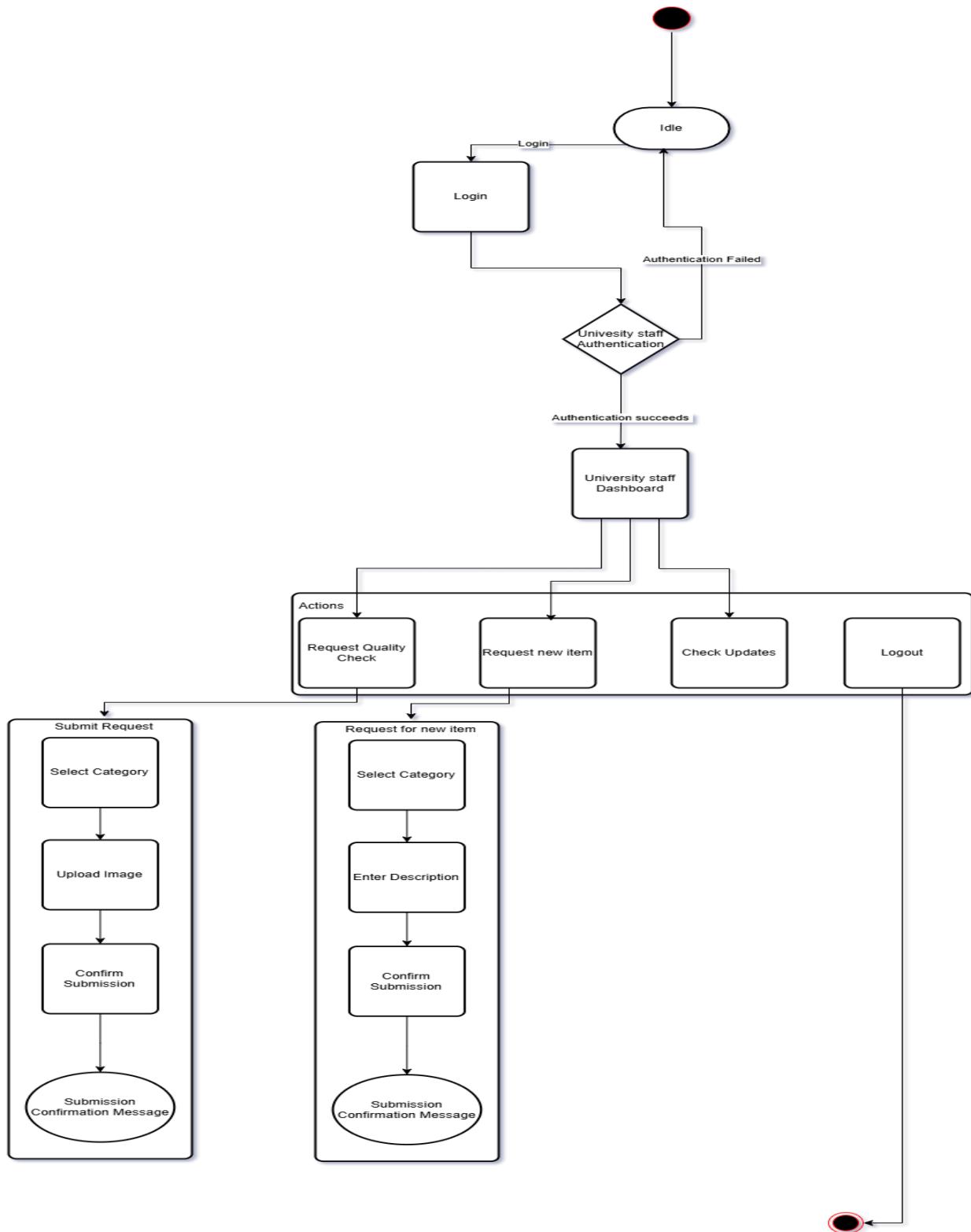


Figure 4.5.5: State Transitions for the University Staff Role

4.6. Data Flow Diagram (DFD)

4.6.1. Level 0

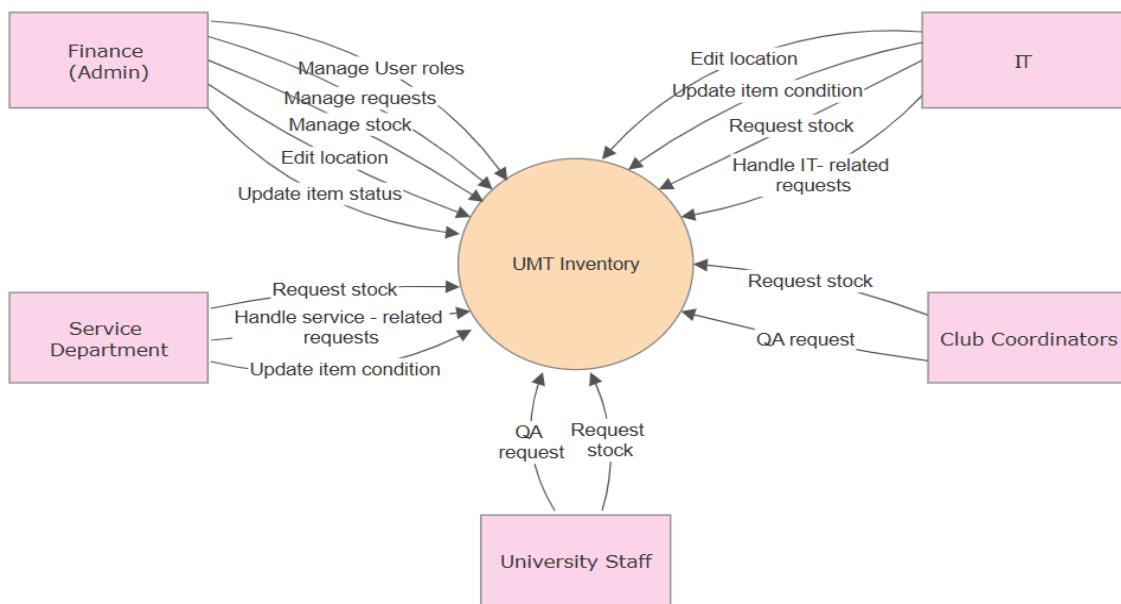


Figure 4.6.1: Level 0 Data Flow Diagram for UMT Inventory System

4.6.2. Level 1

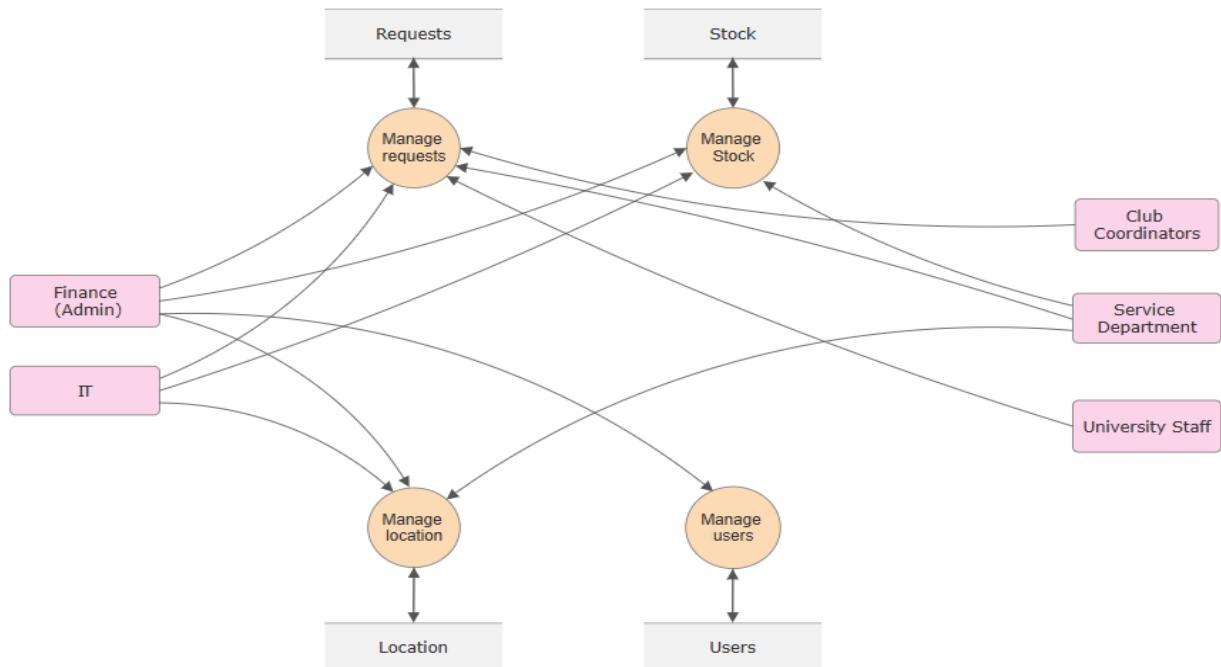


Figure 4.6.2: Level 1 Data Flow Diagram for UMT Inventory System

4.6.3. Level 2

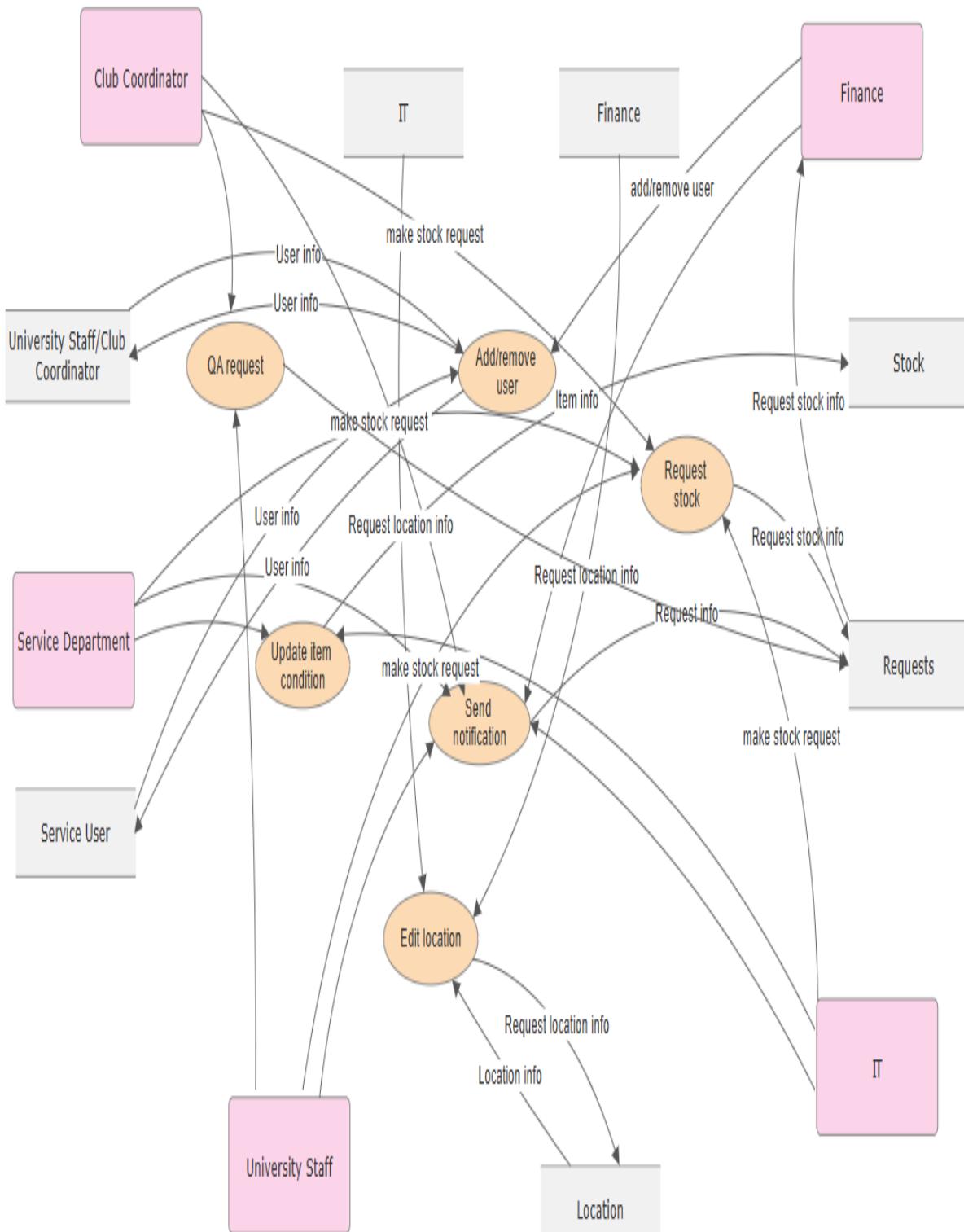


Figure 4.6.3: Level 2 Data Flow Diagram for UMT Inventory System



V. DESIGN PATTERNS

To ensure a robust, maintainable, and scalable architecture, the UMT Inventory System adopts several established software design patterns. These patterns support modularity, simplify complex role-based workflows, and enable seamless communication between components. The primary design patterns implemented are the **Model-View-Controller (MVC)**, **Repository**, **Observer**, and **Factory** patterns.

5.1. Pattern 1: Model-View-Controller (MVC)

Usage Scenario

The entire system is structured using the MVC pattern to separate business logic from user interface and control flow. This supports role-based dashboards (e.g., Finance, IT, Staff) while maintaining clean code organization.

Application Reference

- **Controllers:** Handle role-specific actions like FinanceController, ITController.
- **Views:** Razor pages displaying dashboards and request forms.
- **Models:** Represent database entities like InventoryItem, User, StockRequest.

5.2. Pattern 2: Repository Pattern

Usage Scenario

To abstract data access logic from the business logic layer. Each entity like User, Item, Request uses a dedicated repository to encapsulate database operations.

Application Reference

- IUserRepository for user CRUD operations
- IItemRepository for handling inventory queries
- Makes unit testing and dependency injection easier.

5.3. Pattern 3: Observer Pattern

Usage Scenario

Used for the notification system. When a request is updated (e.g., marked “Approved” or “Declined”), observers (users or departments) are notified automatically.

Application Reference

- When a StockRequest is updated, the system triggers a notification to the relevant department.
- Notifications are sent via email and system alerts.

5.4. Pattern 4: Factory Pattern

Usage Scenario

Used to create request forms dynamically based on user roles (IT, Service, Staff, Coordinator), especially since forms vary by role.

Application Reference

- The UI dynamically renders appropriate fields.
- Backend uses factory logic to instantiate role-based request handlers.

Reusability and Benefits

Pattern	Benefits
MVC	Clean separation of concerns, maintainable structure, UI flexibility
Repository	Database abstraction, easier testing, and logic reuse
Observer	Real-time communication, improves responsiveness
Factory	Simplifies UI/backend logic for multiple roles, improves code reuse

These patterns collectively ensure that the system is modular, easier to test and extend, and resilient to future changes; such as adding new roles or adjusting approval flows, while keeping code complexity manageable.

VI. STRUCTURAL AND BEHAVIORAL MODELING

6.1. Class Diagram

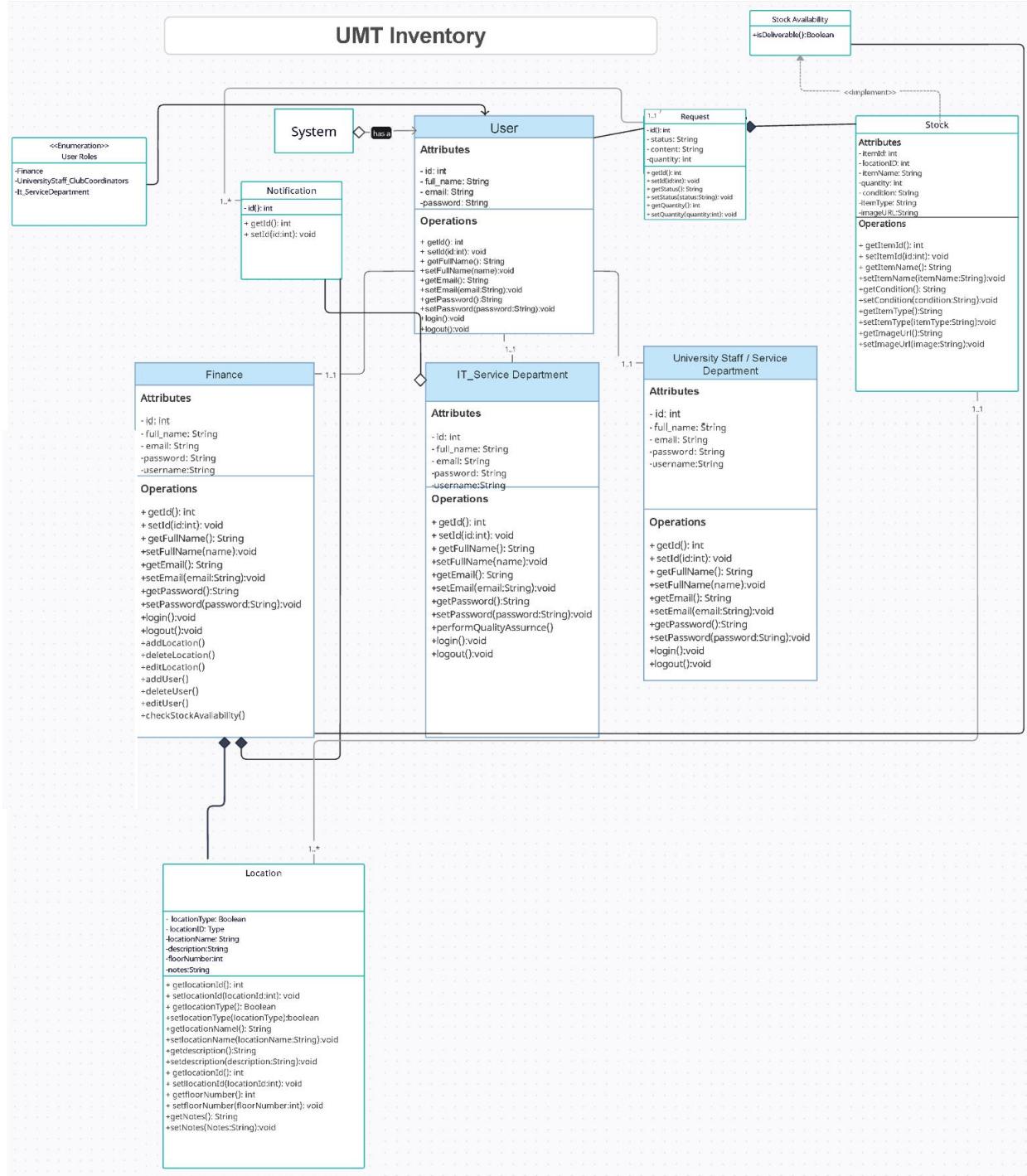


Figure 6.1: Class Diagram

6.2. Object Diagram

6.3. Package Diagram

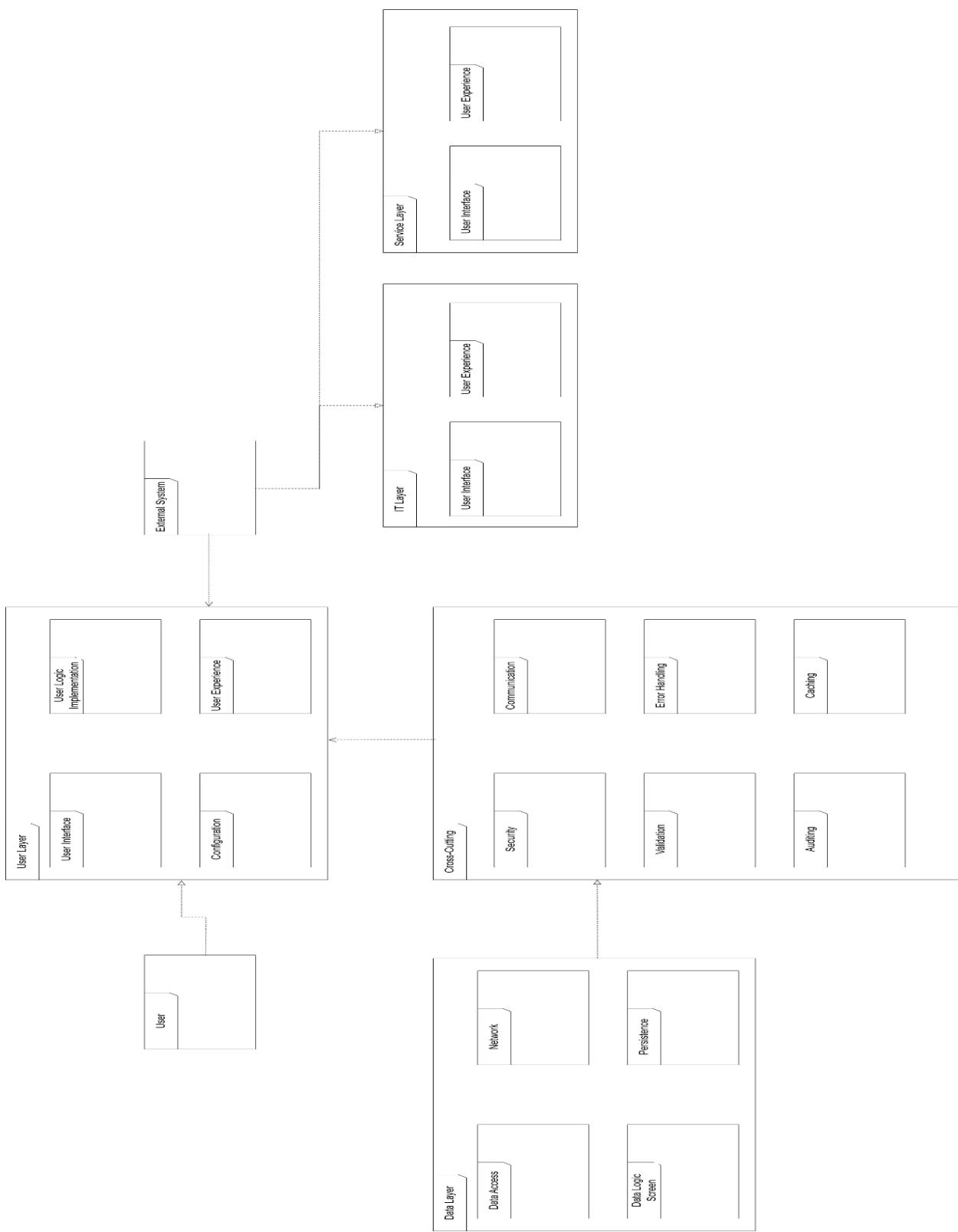


Figure 6.3: Package Diagram

6.4. Sequence Diagrams

6.4.1. Add and Edit User - Finance

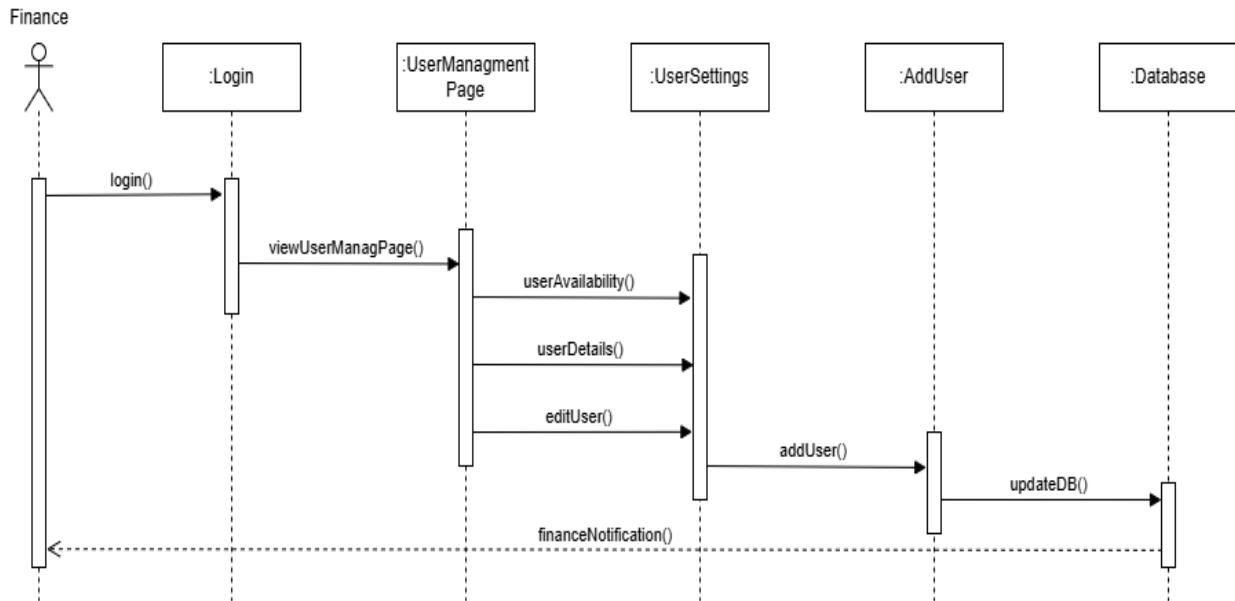


Figure 6.4.1: Sequence Diagram – Add and Edit User (Finance Role)

6.4.2. Add Item - Finance

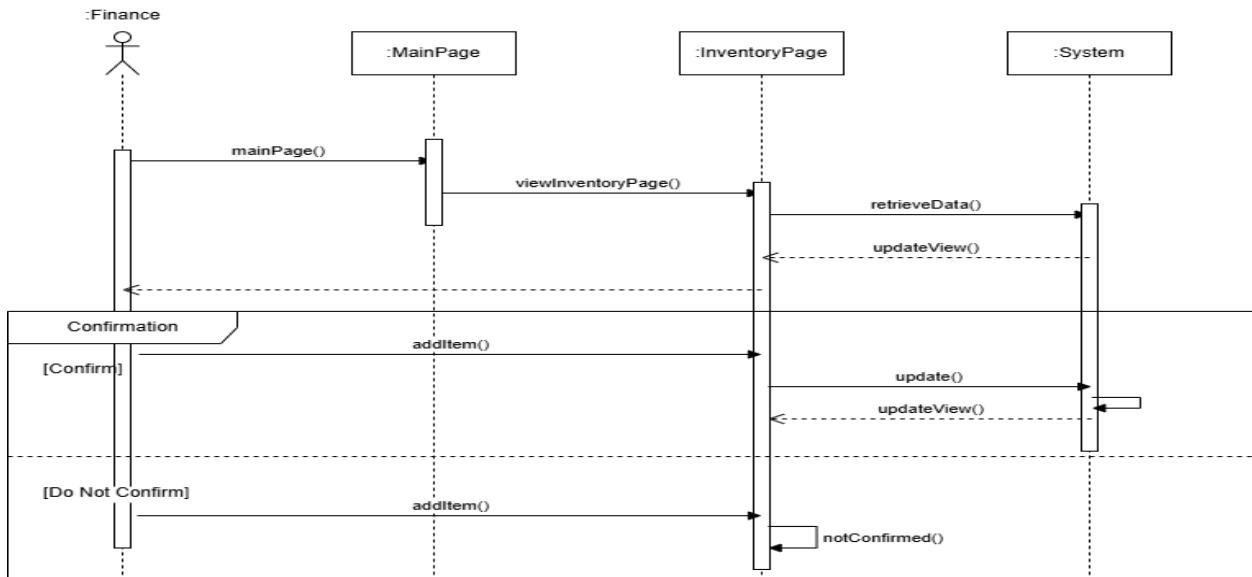


Figure 6.4.2: Sequence Diagram – Add Item(Finance Role)

6.4.3. Delete Item - Finance

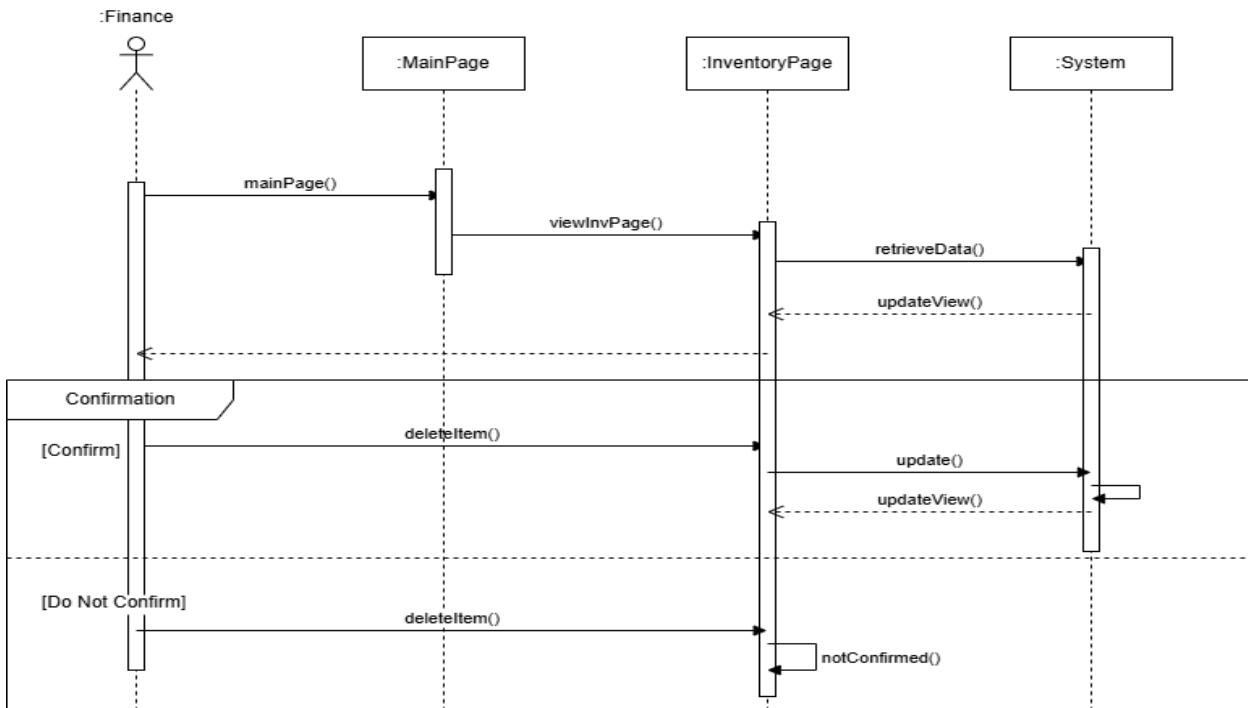


Figure 6.4.3: Sequence Diagram – Delete Item (Finance Role)

6.4.4. Delete User - Finance

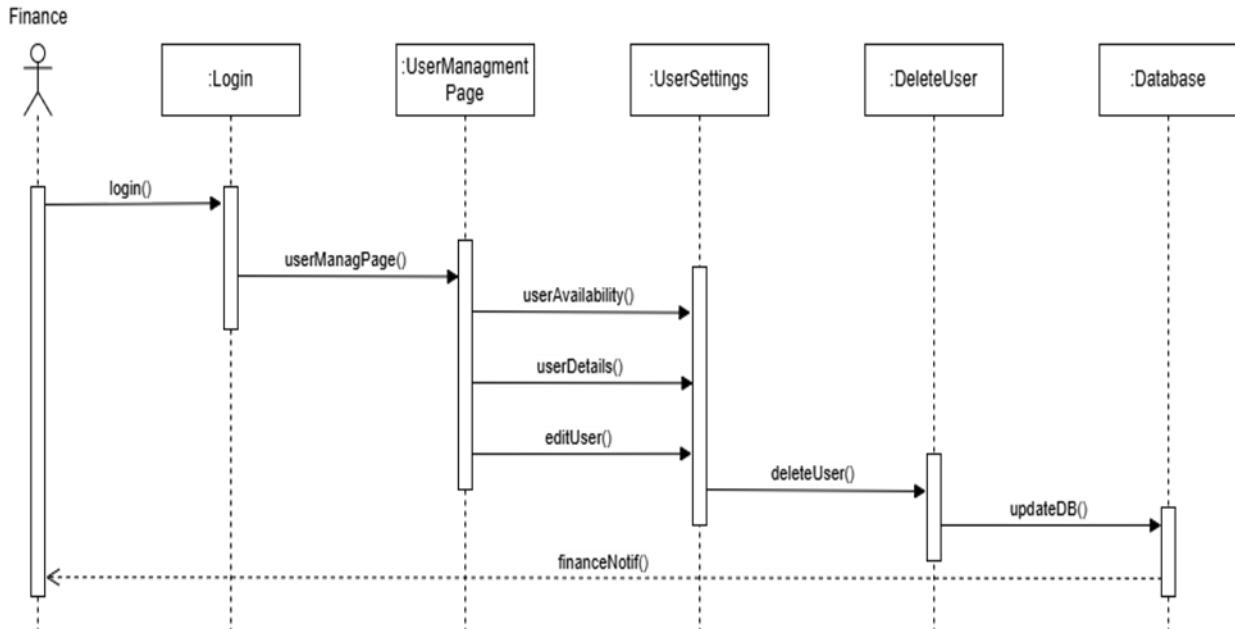
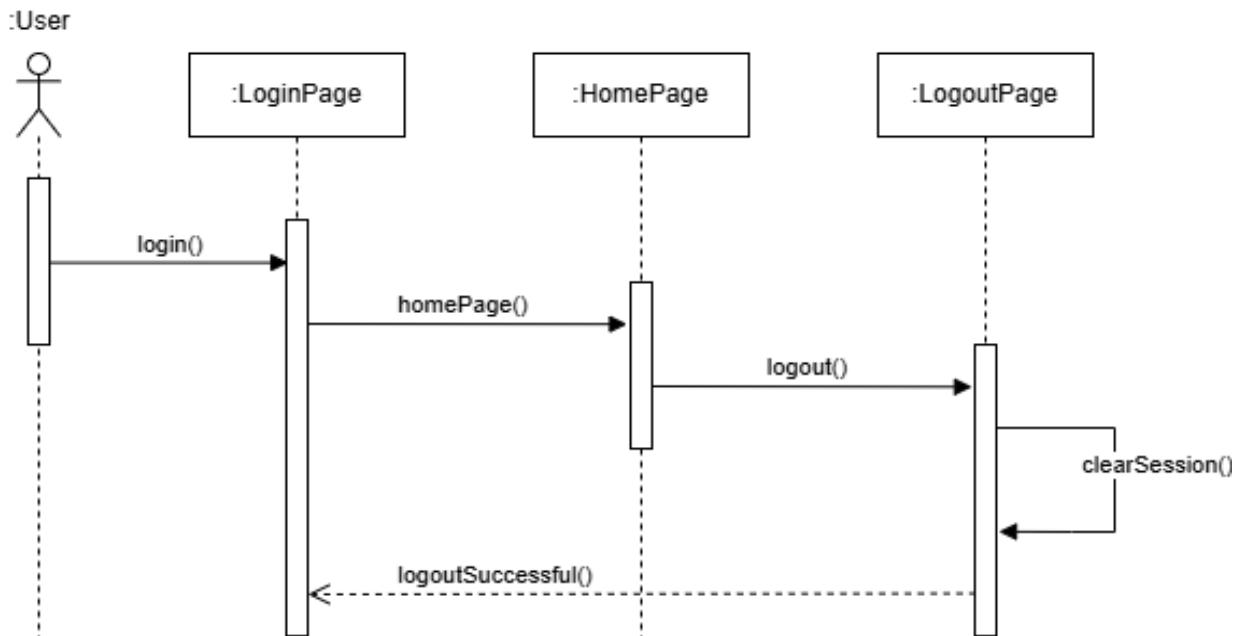
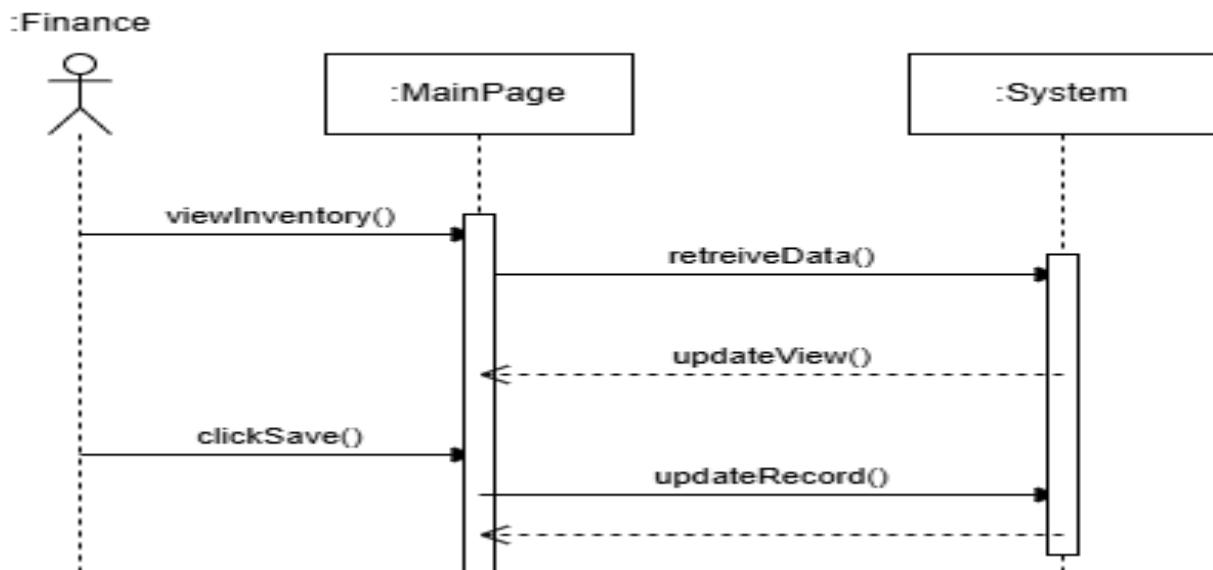


Figure 6.4.4: Sequence Diagram – Delete User (Finance Role)

6.4.5. Logout



6.4.6. View Inventory - Finance



6.4.7. QA Request

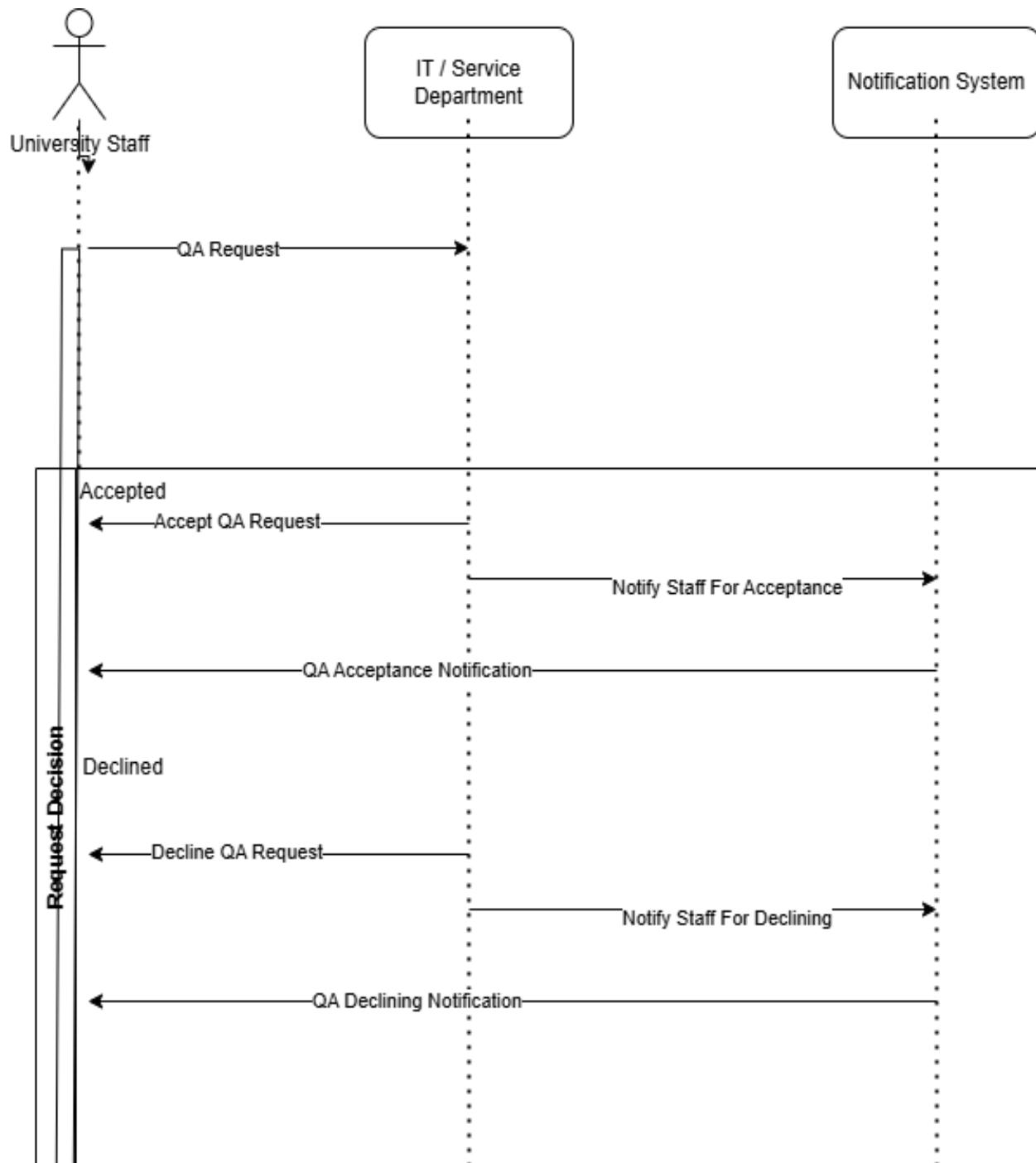


Figure 6.4.7: Sequence Diagram – QA Request

6.4.8. Stock Request

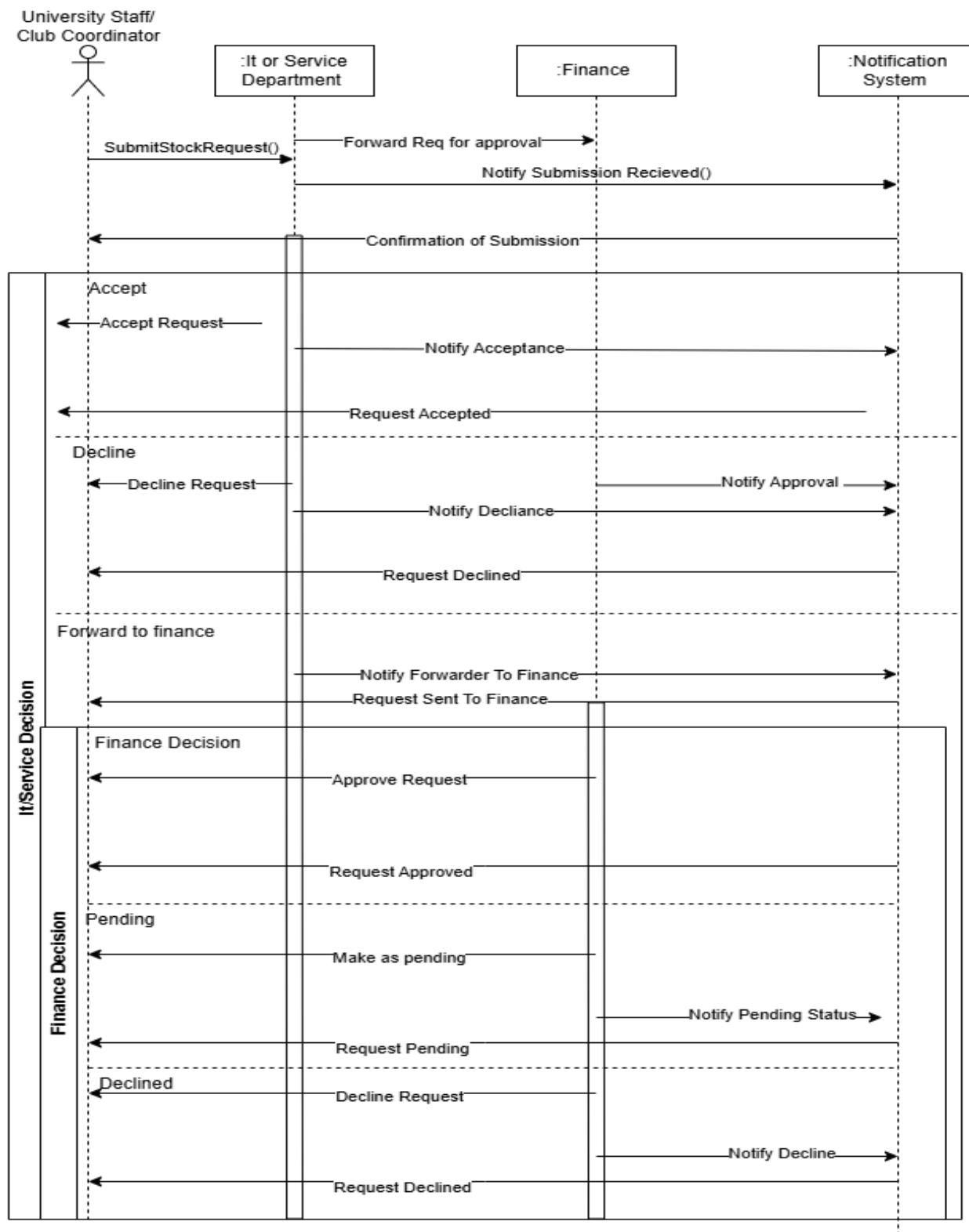


Figure 6.4.8: Sequence Diagram –Stock Request

6.4.9. View Inventory - University Staff/ Club Coordinator

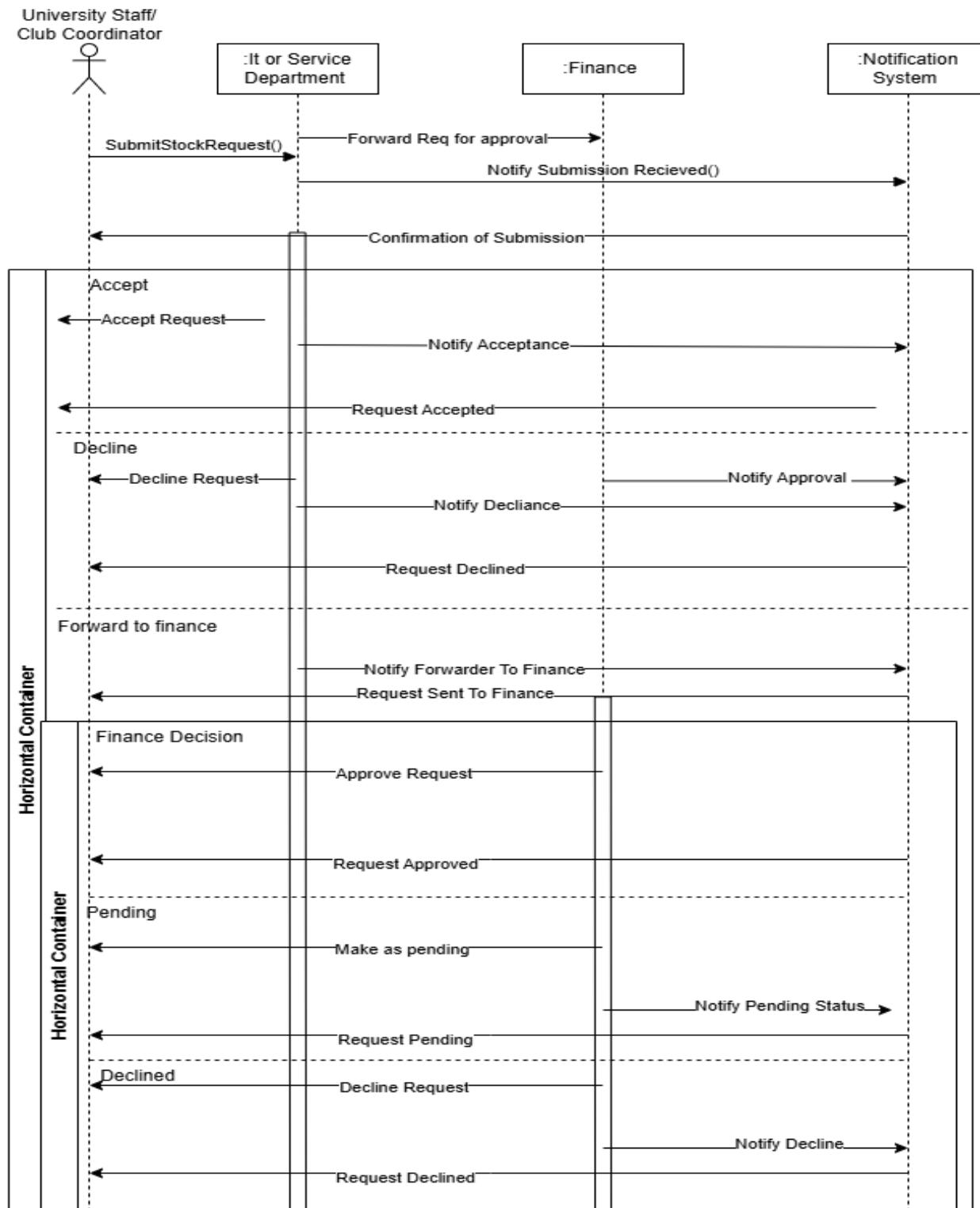


Figure 6.4.9. Sequence Diagram - View Inventory (University Staff/ Club Coordinator)

6.4.10. User Login

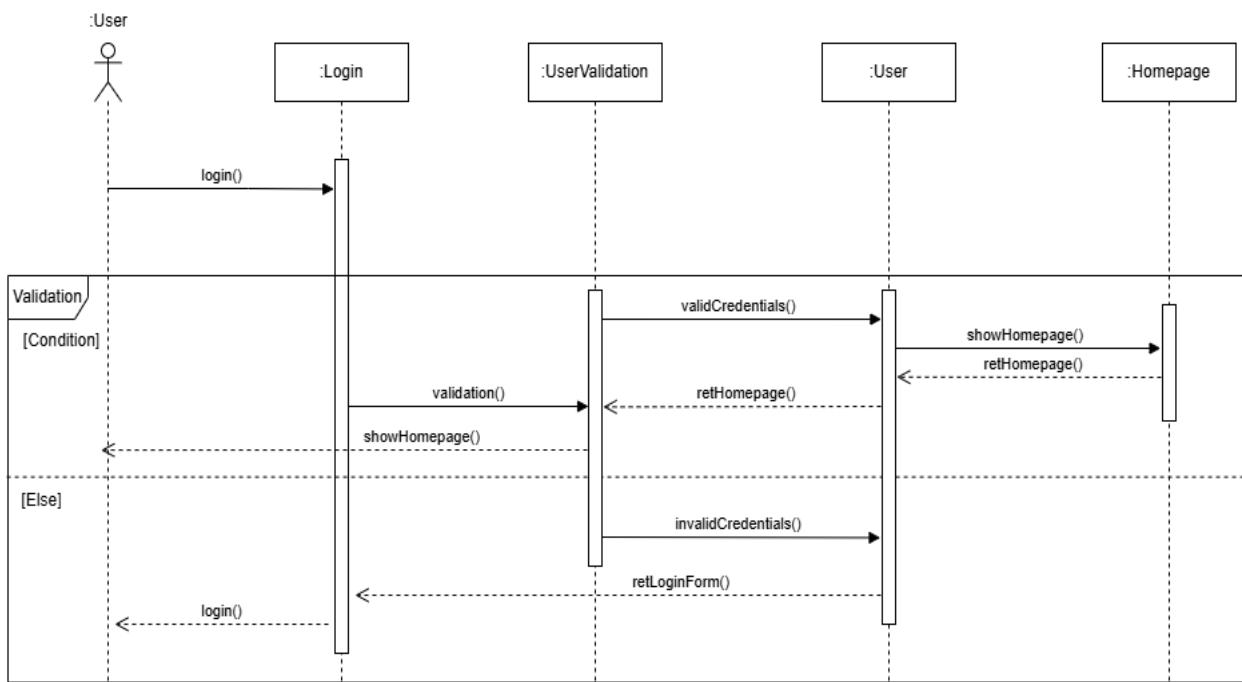


Figure 6.4.10: Sequence Diagram – User Login

6.5. Collaboration Diagrams

6.5.1. Add User - Finance

This UML collaboration diagram illustrates how the Finance role interacts with system components to add a new user. It focuses on message exchanges between objects such as the user interface, database, and notification service to complete the *Add User* process.

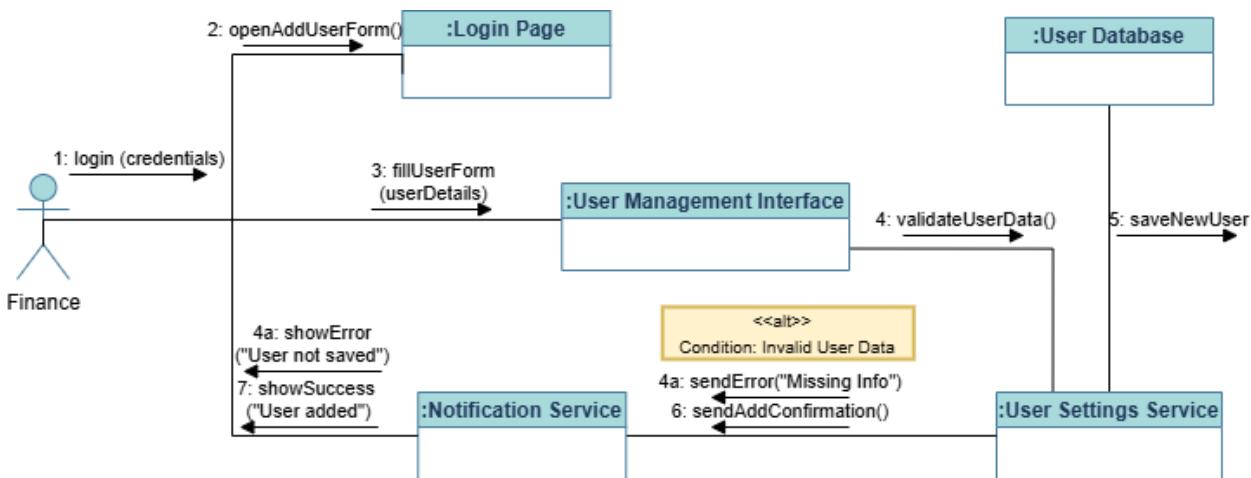


Figure 6.5.1: Collaboration Diagram – Add User(Finance)

6.5.2. Submit New Stock Request

This collaboration diagram models the communication between system components during a new stock request submission. It outlines how roles like University Staff, Finance, IT, Service Department, and Vendor interact to validate, process, and finalize the request using coordinated message flows.

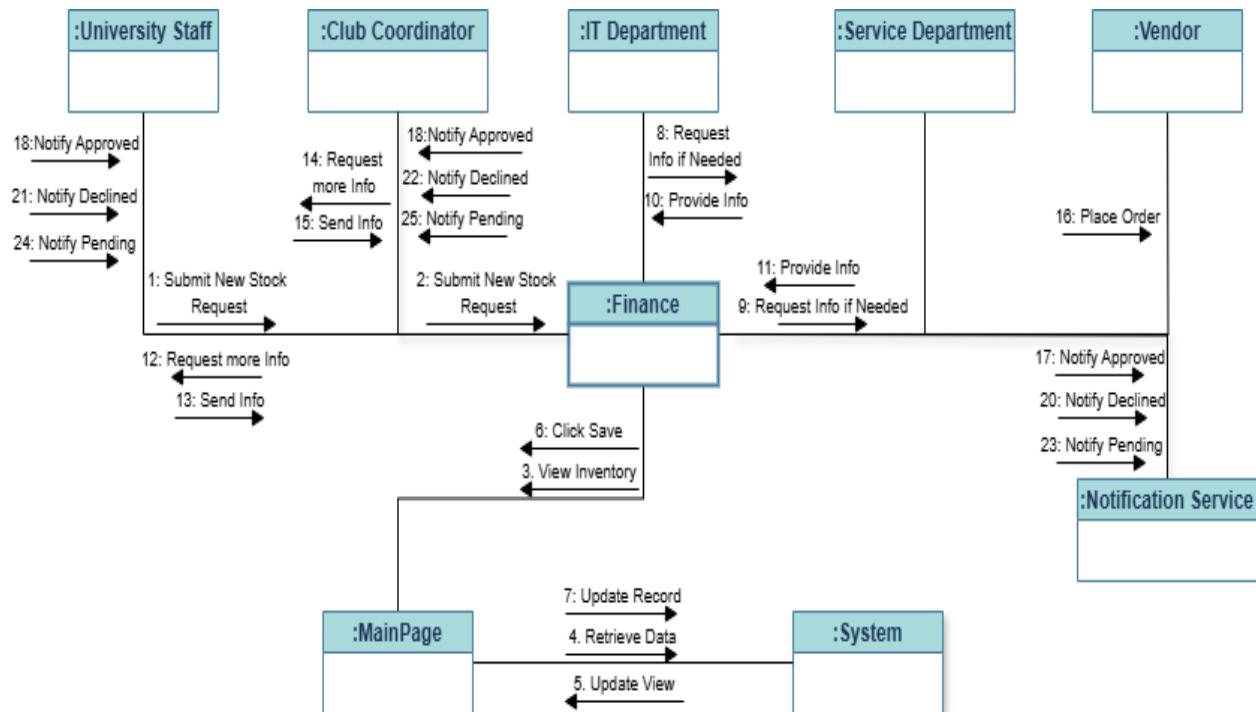


Figure 6.5.2: Collaboration Diagram – Submit Stock Request

6.5.3. Submit QA Request – University Staff

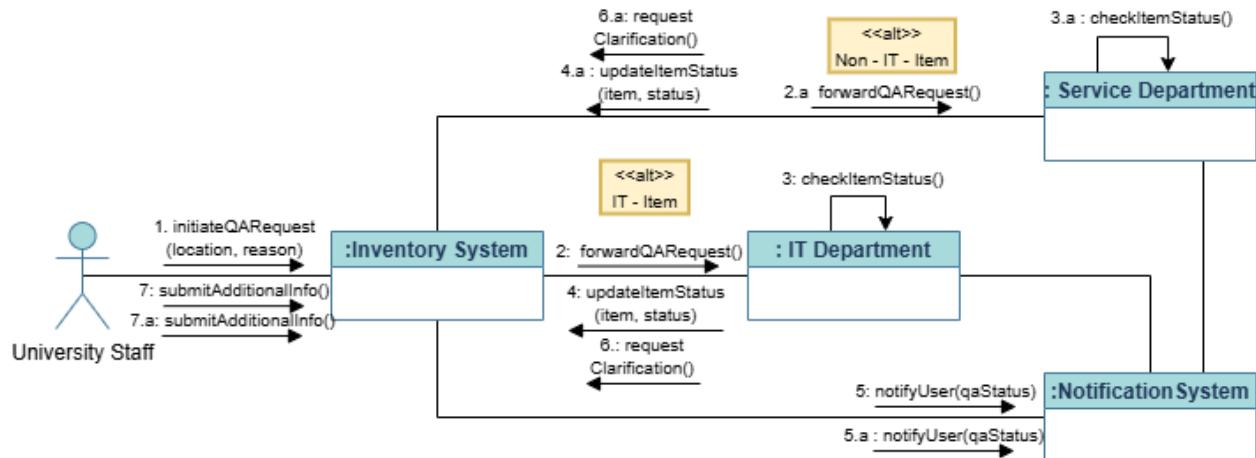


Figure 6.5.3: Collaboration Diagram – Submit QA Request (University Staff)

6.5.4. User Role Management and Deletion by Finance

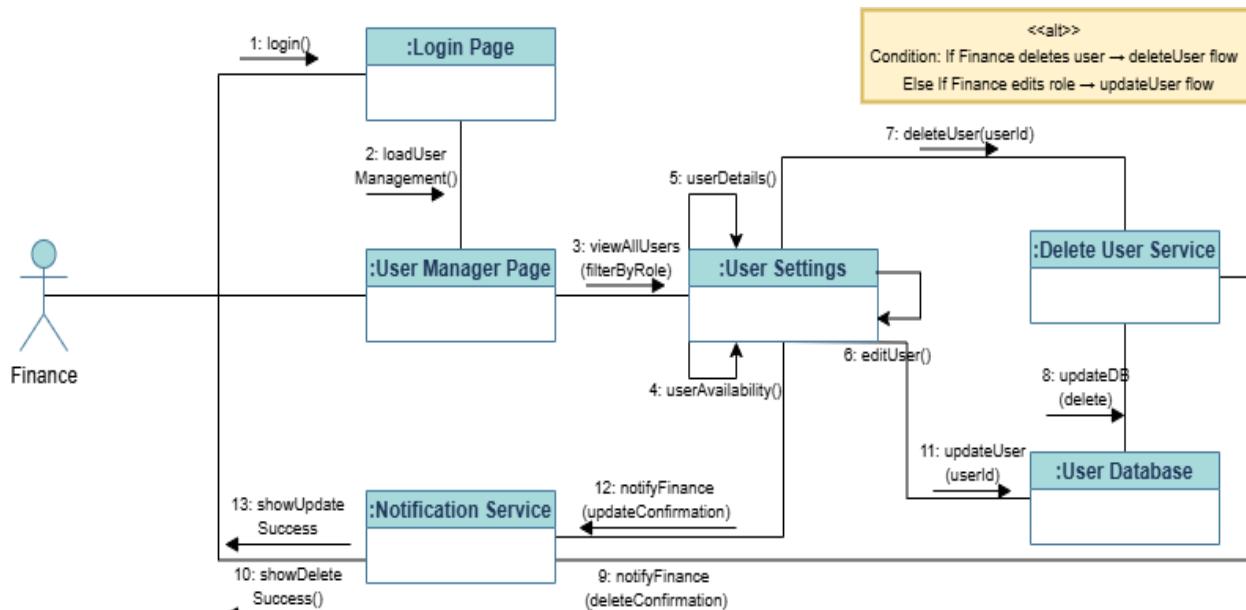


Figure 6.5.4: Collaboration Diagram – Edit Role / Delete User (Finance Role)

6.5.5. User Login

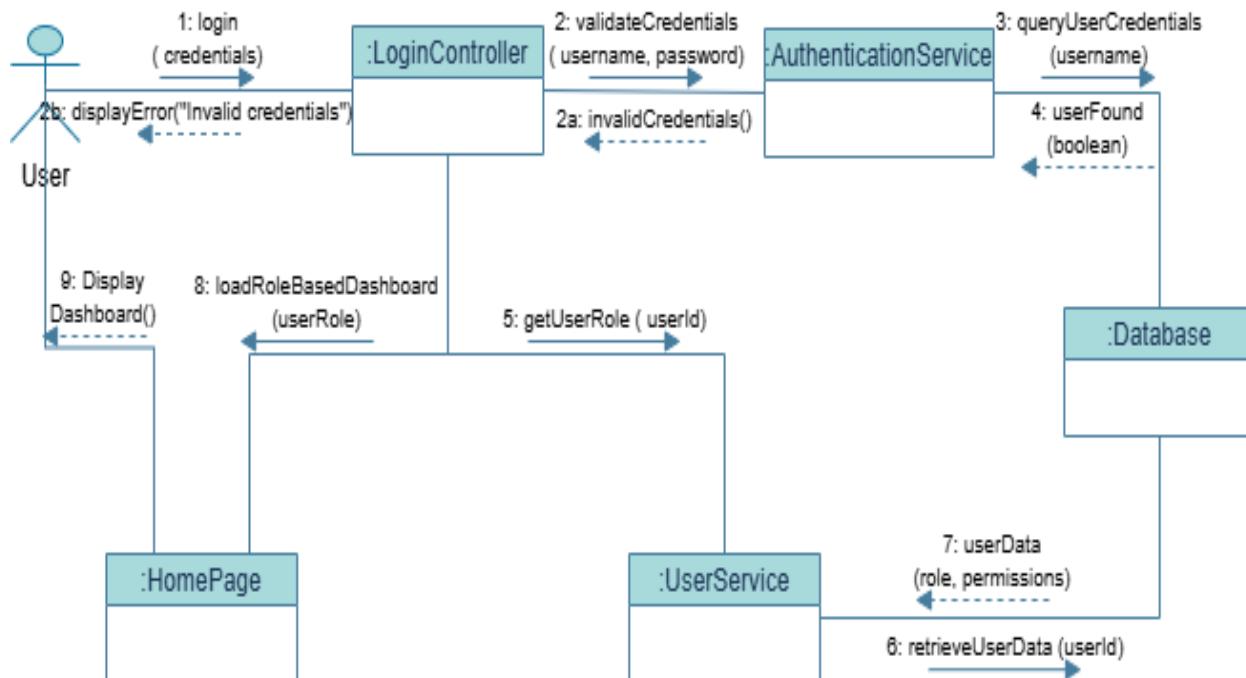


Figure 6.5.5: Collaboration Diagram – User Login

6.5.6. Submit New Stock Request – University Staff / Club Coordinator

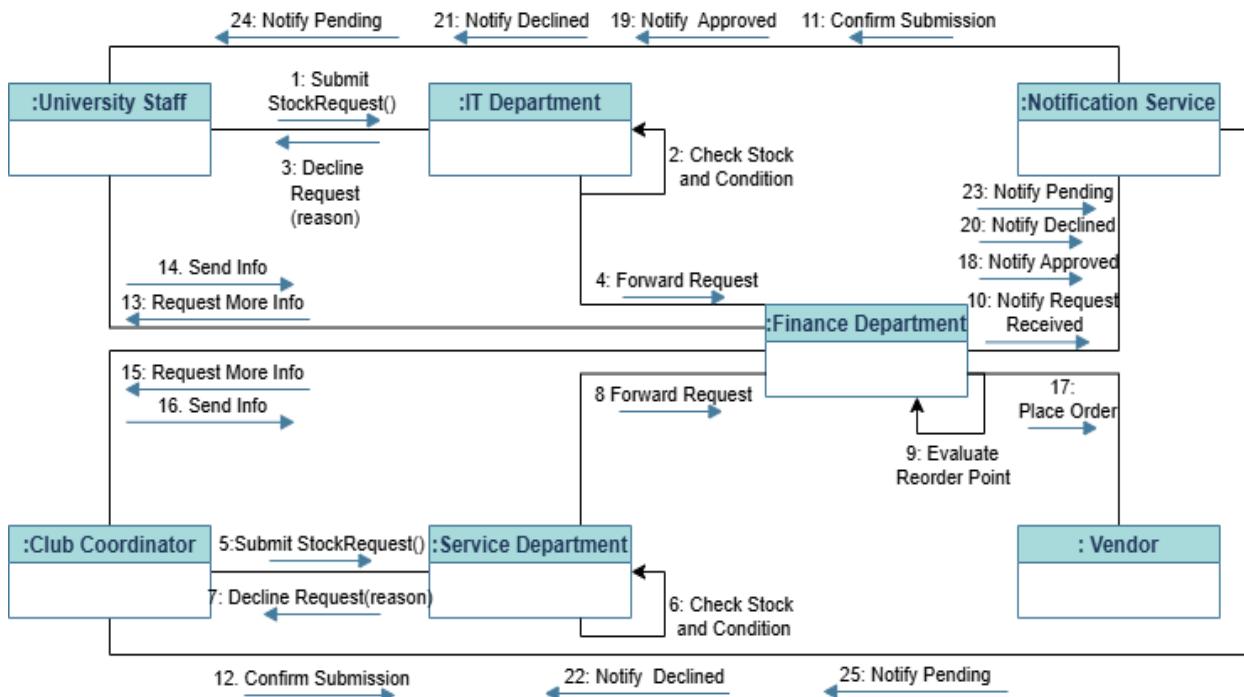


Figure 6.5.6: Collaboration Diagram – New Stock Request

6.5.7. Requesters

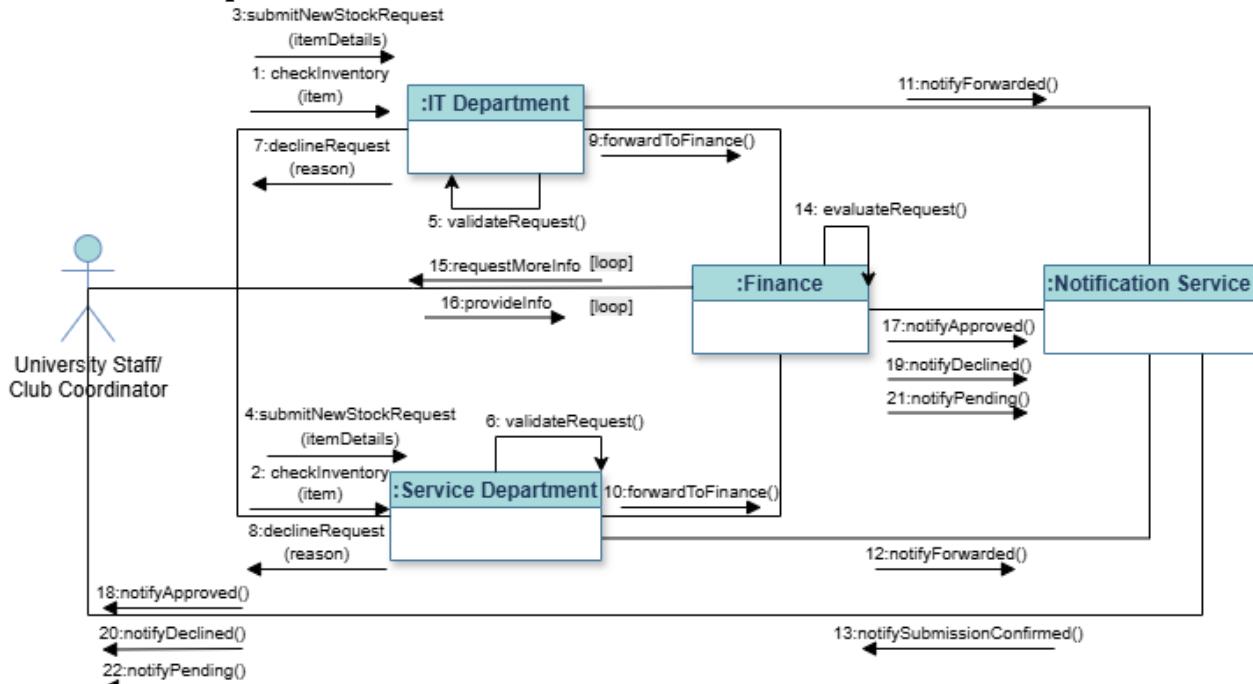


Figure 6.5.7: Collaboration Diagram – Requesters

6.6. Interaction Overview Diagrams (IOD)

6.7. Component Diagram

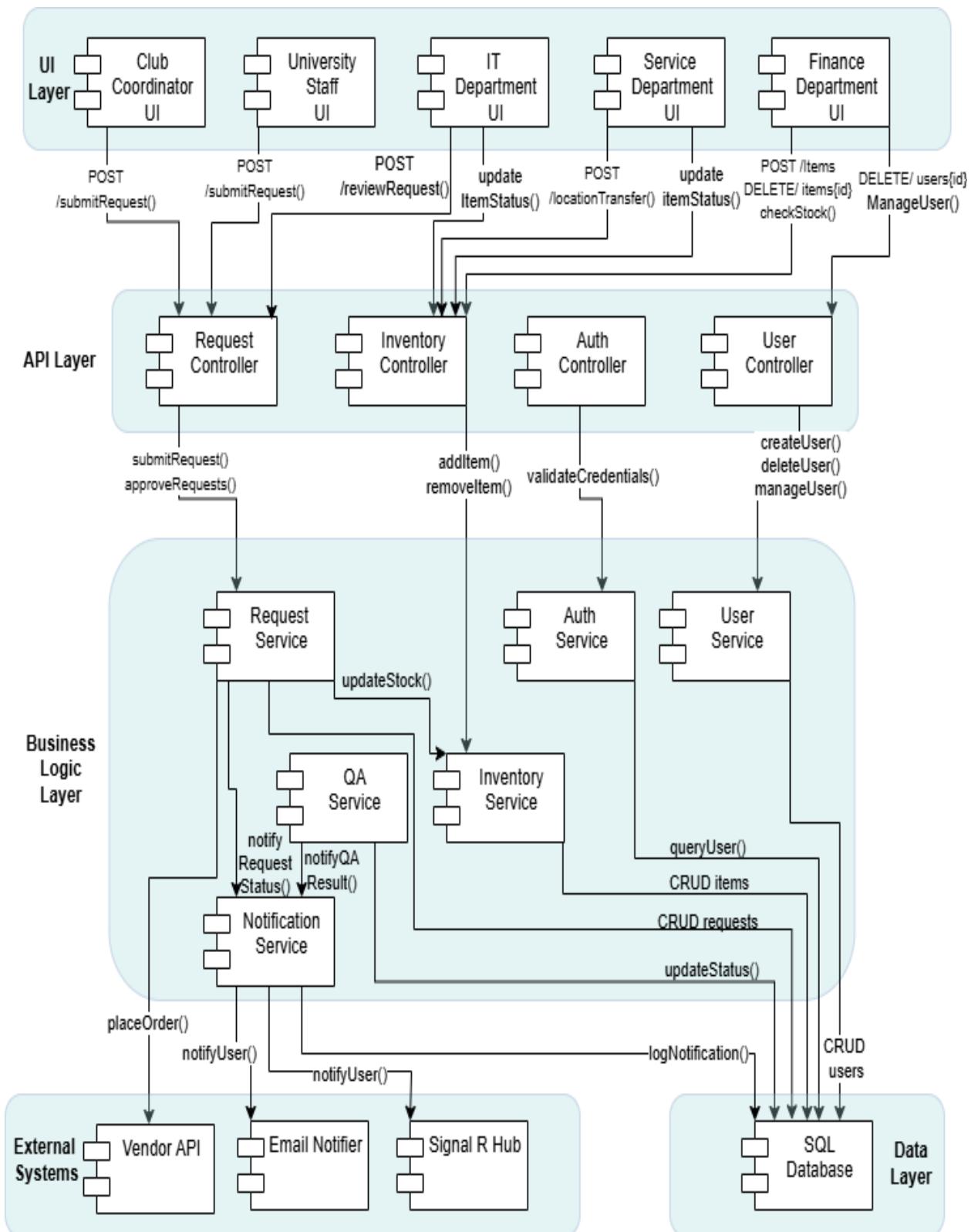


Figure 6.7: Component Diagram

The above UML component diagram illustrates a clear, layered architecture for the University Inventory Management System, dividing responsibilities across UI, API, Business Logic, Data, and External Systems layers. This separation promotes modularity, scalability, and maintainability. Leveraging technologies like .NET Core, Entity Framework, REST APIs, and asynchronous notifications via Email and SignalR, the system ensures real-time updates and smooth interaction for various user roles. The design supports efficient communication between components and is well-suited for future extensions and continuous evolution.

6.8. Deployment Diagram

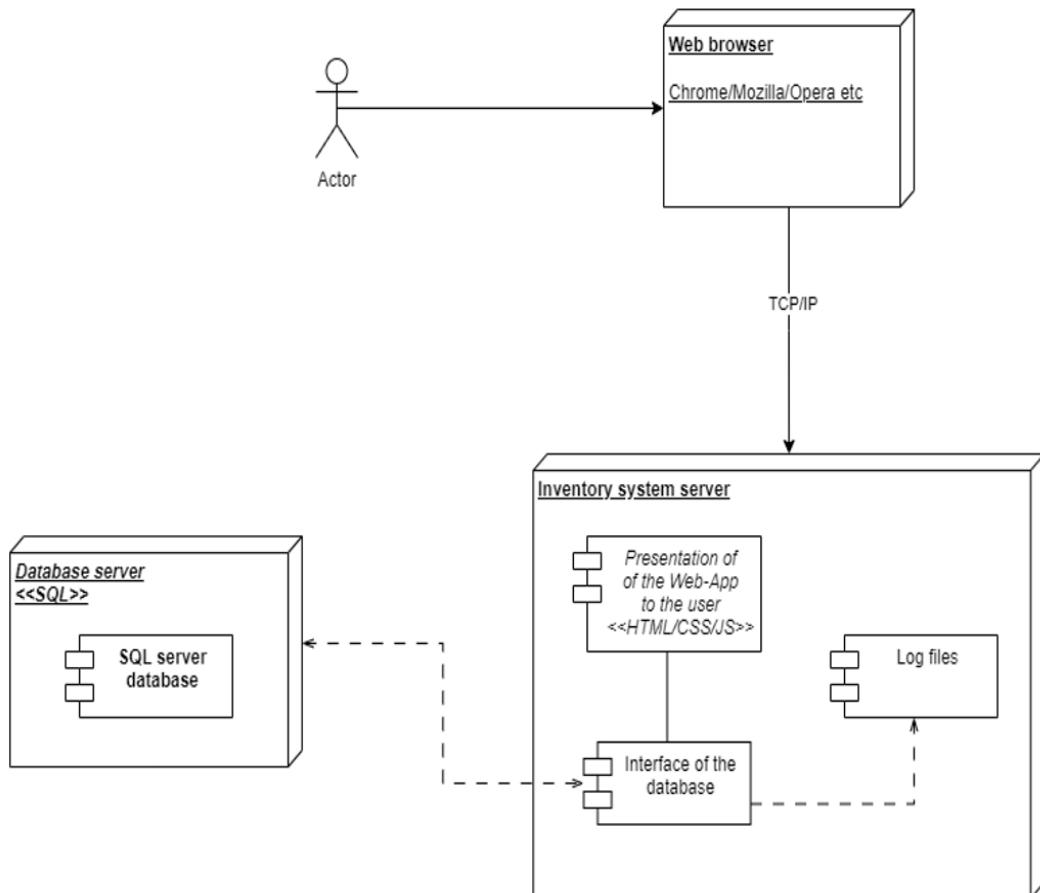


Figure 6.8: Deployment Diagram

VII. DATA DESIGN

7.1. Entity Relationship Diagram (ERD)

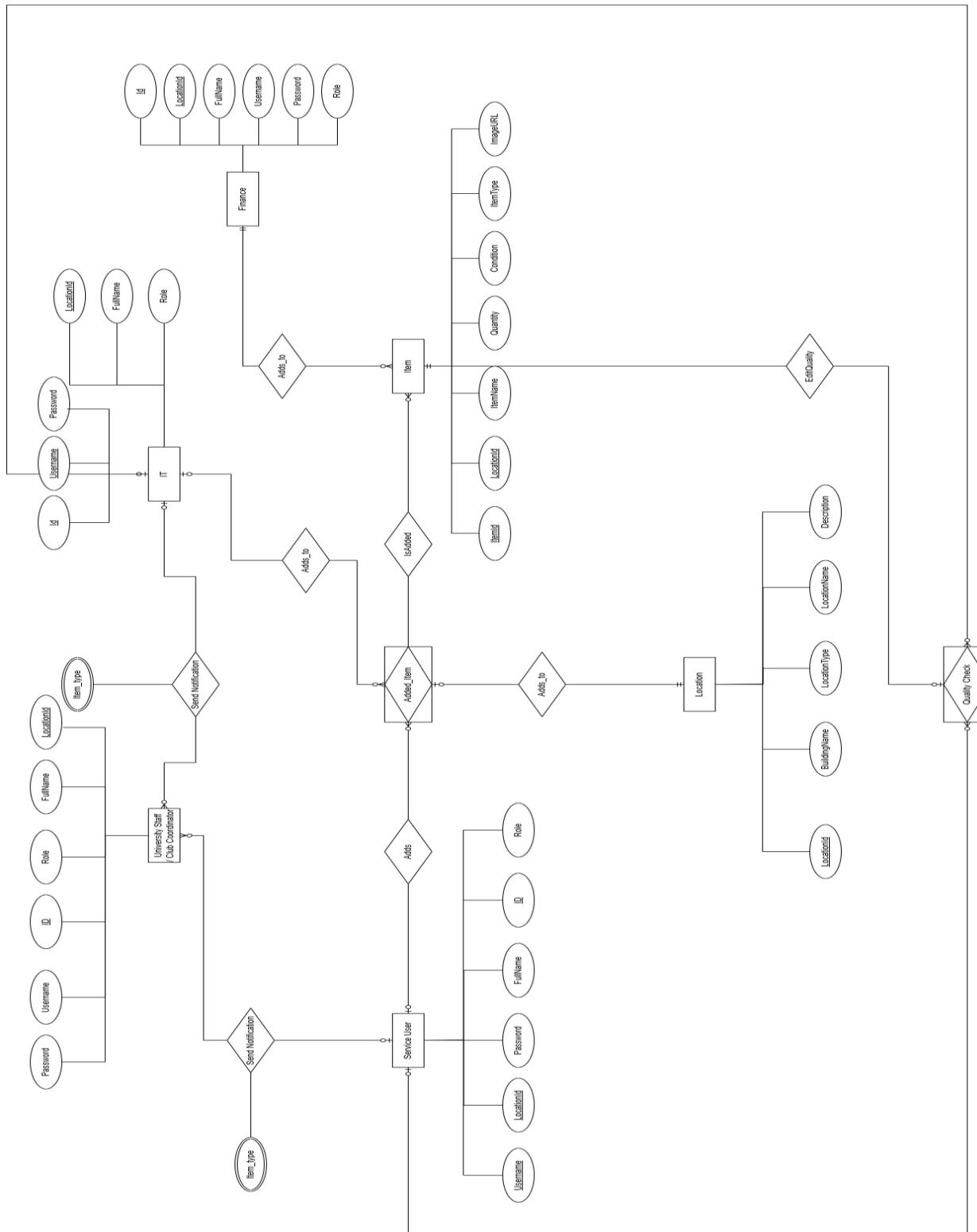


Figure 7.1: Entity Relationship Diagram

7.2. Relational Schema (RS)

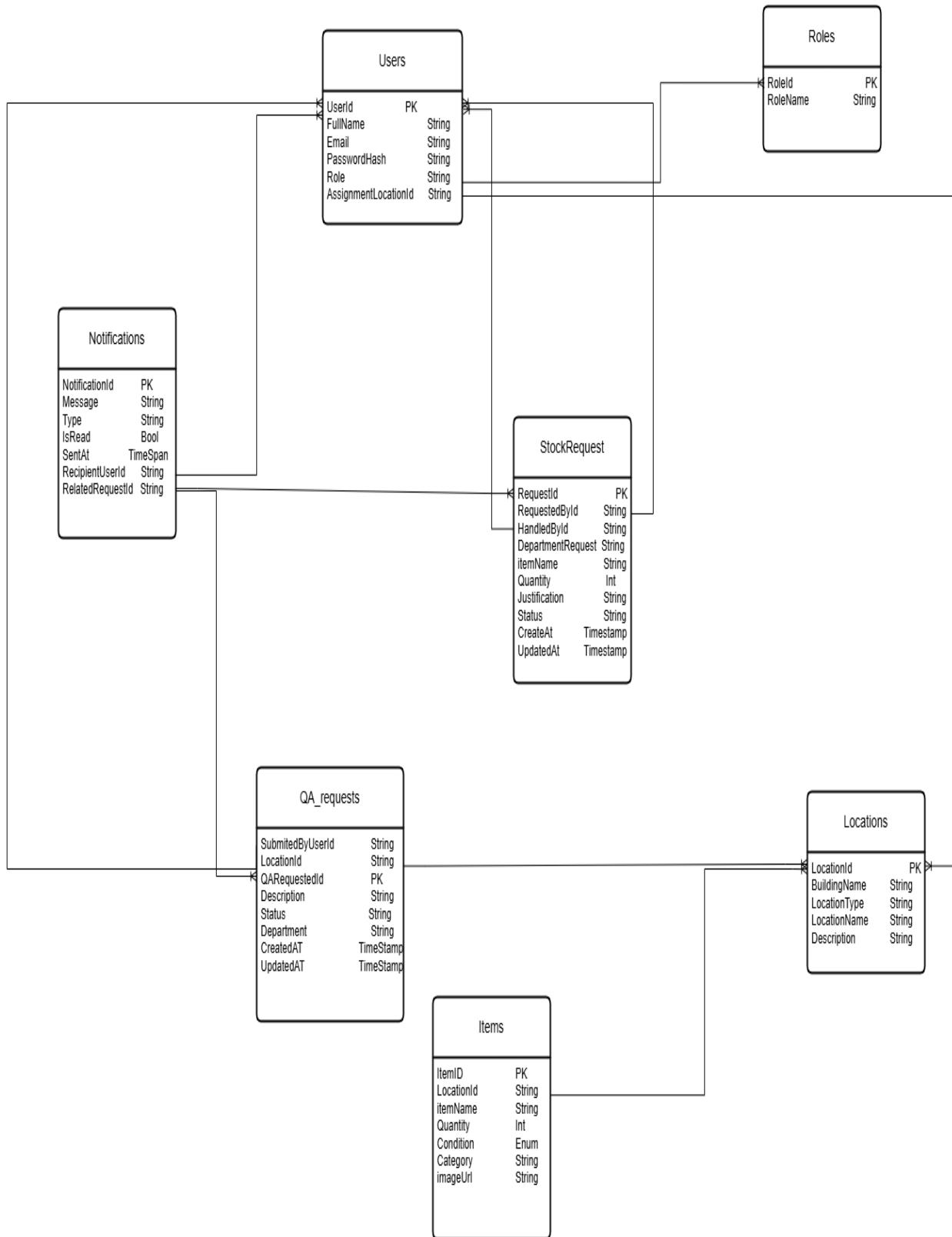


Figure 7.2: Relational Schema



Ø Appendix A: User Stories by System Roles (Handwritten by System Users)

A.1 University Staff

(User stories completed by lecturers, academic secretary, receptionist)

_____'s User Story

Si nje (kush je ti ne lidhje me shkollen?) **Cektor**

Une duar (cfare problem ka qe sistemi i inventarizimit mund ta rreulloj?)
automatizohet kriteri per supozit
ne menyre qe (perse? Cfare do te fitosh nga kjo qie)
to qe me e drejtuar

Detaje te tjera:
(cfare mund te na thuash me shume, ne menyre qe ne te te
kuptojme me mire)

Kriteret e pranimit:
(ju do te ishit dakort qe ky problem eshte zgjidhur nese...)
kofar e zyrtar e ultimata

_____'s User Story

Sekretaria Mësimore

Si nje (kush je ti ne lidhje me shkollen?)
Sekretare fakulteti

Une duar (cfare problem ka qe sistemi i inventarizimit mund ta rreulloj?)
Organizim, Koordinim, shërbim; shpejte
ne menyre qe (perse? Cfare do te fitosh nga kjo qie)
- Lehtësimim e pumës
- Efikasitet

Detaje te tjera:
(cfare mund te na thuash me shume, ne menyre qe ne te te
kuptojme me mire)

Kriteret e pranimit:
(ju do te ishit dakort qe ky problem eshte zgjidhur nese...)
Po patjetër po po, zgjidja e problemit me mënyre
të lehta dhe të shpejte përmes gjallimin e përbashkët.



_____’s User Story

Si nje (kush je ti ne lidhje me shkollen?) Recepsoni G3.

Une dua (cfare problem ke qe sistemi i inventarizimit mund ta rregulloj?)
qe“ te kete“ nje“ sektion ku studentet mund te“
bejn“ pyetjet dhje te shprekin pakenaqesit e tyre.
ne menyre qe (perse? Cfare do te fitosh naa kia aie)

Pyetjet ti dikojn“ personave kompetente“
qe“ te“ bejn“ dika“ te mund dhime dhje te“
mawin Perxhigjen e dukur.

Detaje te tjera:

*(cfare mund te na thuash me shume, ne menyre qe ne te te
kuptojme me mire)*

Kriteret e pranimit:

(ju do te ishit dakort qe ky problem eshte zgjidhur nese...)

Mos te“ marr pyetje te“ cilave ~~nek~~ jone“
jash te“ mundesive te“ mia.

_____’s User Story

Si nje (kush je ti ne lidhje me shkollen?) Recepson

Une dua (cfare problem ke qe sistemi i inventarizimit mund ta rregulloj?) Procedura
je“ me e thyeshite.

ne menyre qe (perse? Cfare do te fitosh naa kia aie) Qe“ te“ zyjdet
me“ shpejt problemi..

Detaje te tjera:

*(cfare mund te na thuash me shume, ne menyre qe ne te te
kuptojme me mire)*

Kriteret e pranimit:

(ju do te ishit dakort qe ky problem eshte zgjidhur nese...)

Kur te“ zyjdet qrohemi“ me“ shpejt



A.2 Club Coordinator

A User Storie completed by Club Coordinator Role:

Club Coordinator's User Story

Si nje (kush je ti ne lidhje me shkollen?) **projekti i** clubi

Une dua (cfare problem ke qe sistemi i inventarizimit mund ta rregulloi?)
gë deret kerxja i projekteve që më bllokk
posisjoni kryjet dre më kontrolli & shume
ne menyre qe (perse? Cfare do te fitosh nga kjo aie) persone.

te kenne i ligjmet më shpesh që dha t'eb kushtem
t'ek punet e tjera.

Detaje te tjera:
*(cfare mund te na thuash me shume, ne menyre qe ne te te
kuptojme me mire)*

A.3 IT Department

A User Storie completed by IT Role:

_____'s User Story

Si nje (kush je ti ne lidhje me shkollen?)
Specialisti i IT

Une dua (cfare problem ke qe sistemi i inventarizimit mund ta rregulloi?)
Menaxhim më të lente të inventarit dhe kërtuesave.

ne menyre qe (perse? Cfare do te fitosh nga kjo aie)
Puna do të bëhet më eficiente. Do të keni
mundësi në kohë reale të shoh kërcësat dhe
vendos prioritetet

Detaje te tjera:
*(cfare mund te na thuash me shume, ne menyre qe ne te te
kuptojme me mire)*



A.4 Finance

(User stories from Finance Department)

_____ 's User Story

Financa

Si nje (kush je ti ne lidhje me shkollen?)

Une dua (cfare problem ke qe sistemi i inventarizimit mund ta rregulloj?)

Dixhitalizimin e inventarit

ne menyre qe (perse? Cfare do te fitosh nga kjo qie)

Të kuptojmë më mirë gjendjen e stokut

Detaje te tjera:
(cfare mund te na thuash me shume, ne menyre qe ne te te kuptojme me mire)

Kriteret e pranimit:
(ju do te ishit dakort qe ky problem eshte zgjidhur nese...)

Kur të krijohet sistemi.

_____ 's User Story

Financa

Si nje (kush je ti ne lidhje me shkollen?)

Une dua (cfare problem ke qe sistemi i inventarizimit mund ta rregulloj?)

Dixhitalizimin e kerkesave për materiallet që blejme

ne menyre qe (perse? Cfare do te fitosh nga kjo qie)

Të kemi lehtësi në parafisjen e tyre

Detaje te tjera:
(cfare mund te na thuash me shume, ne menyre qe ne te te kuptojme me mire)

Kriteret e pranimit:
(ju do te ishit dakort qe ky problem eshte zgjidhur nese...)

Kur të krijohet sistemi.

Ø Appendix B: UI Sketches and Concept Designs

This appendix presents the early-stage UI sketches and concept designs created for each user role within the system.

B.1. Shared Pages

Landing Page

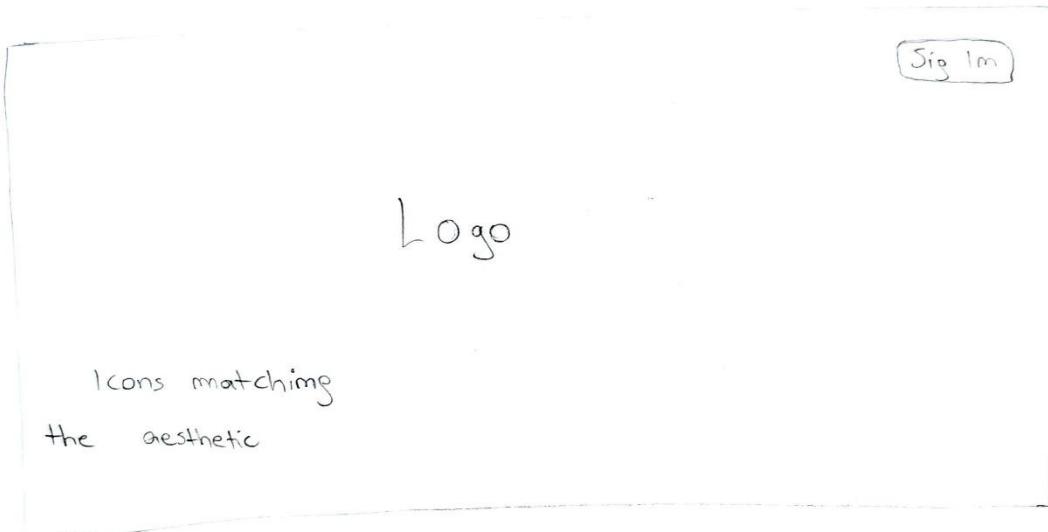


Figure B.1.1: “Landing Page”

Log In Page

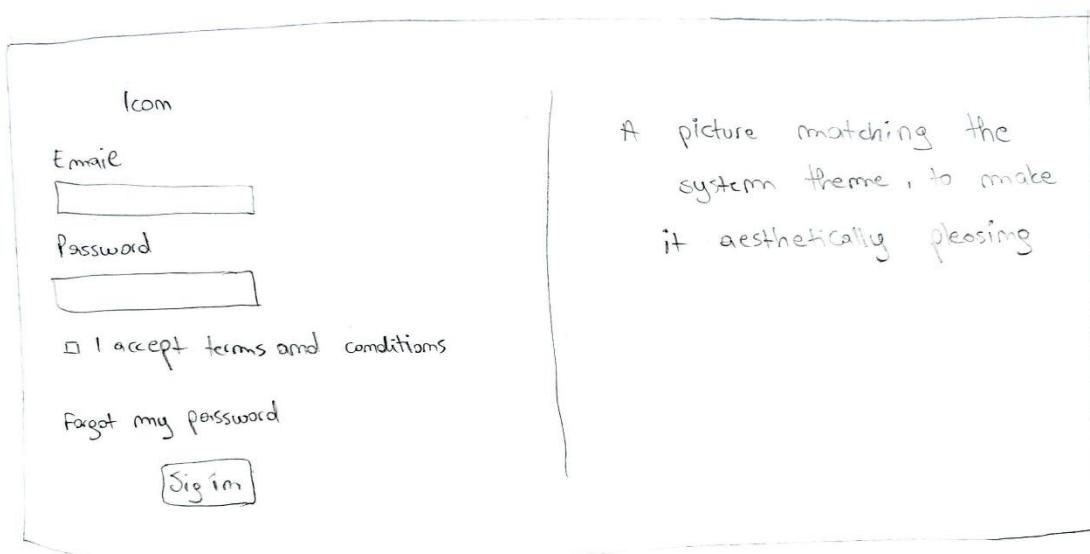


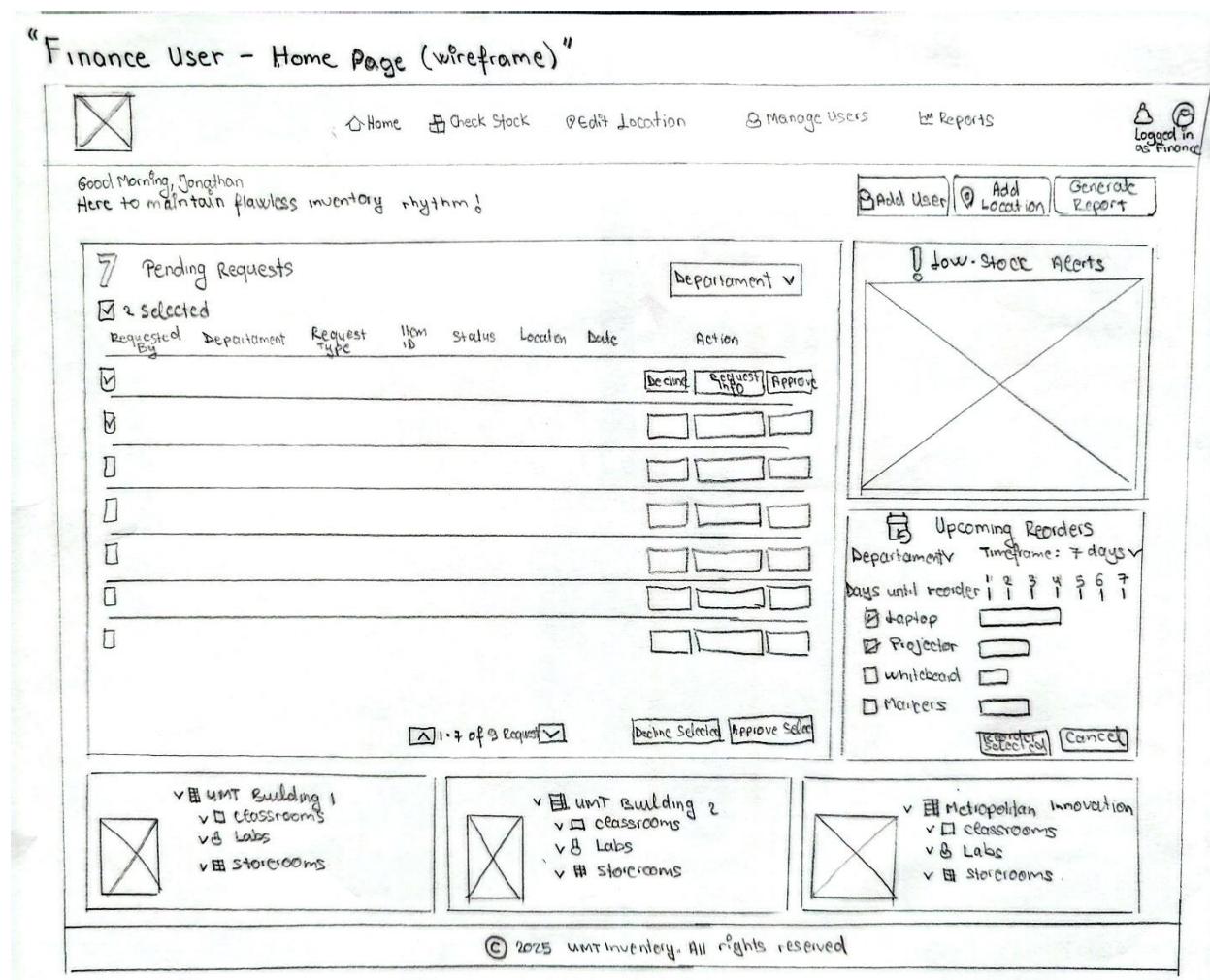
Figure B.1.2: “Login Page”

B.2 Role-Based UI Sketches

B.2.1. Finance Department

Two hand-drawn homepage wireframes were designed for the Finance role. These variations reflect early-stage design ideation and were used to determine the most effective layout before moving to high-fidelity mockups.

"Finance User - Home Page (wireframe)"



Good Morning, Jonathan
Here to maintain flawless inventory rhythm!

Pending Requests

7 Pending Requests
 2 Selected

Requested By	Department	Request Type	Item ID	Status	Location	Date	Action
<input checked="" type="checkbox"/>							<input type="button" value="Decline"/> <input type="button" value="Request PO"/> <input type="button" value="Approve"/>
<input checked="" type="checkbox"/>							<input type="button"/> <input type="button"/> <input type="button"/>
<input type="checkbox"/>							<input type="button"/> <input type="button"/> <input type="button"/>
<input type="checkbox"/>							<input type="button"/> <input type="button"/> <input type="button"/>
<input type="checkbox"/>							<input type="button"/> <input type="button"/> <input type="button"/>
<input type="checkbox"/>							<input type="button"/> <input type="button"/> <input type="button"/>
<input type="checkbox"/>							<input type="button"/> <input type="button"/> <input type="button"/>

1 of 7 Request Decline Selected

Low Stock Alerts

1 Low Stock Alerts

Department: Timeframe: 7 days

days until reorder: 1 2 3 4 5 6 7

- Laptop
- Projector
- Whiteboard
- Markers

Upcoming Reorders

Department: Timeframe: 7 days

days until reorder: 1 2 3 4 5 6 7

Locations

UMT Building 1:

- Classrooms
- Labs
- Storerooms

UMT Building 2:

- Classrooms
- Labs
- Storerooms

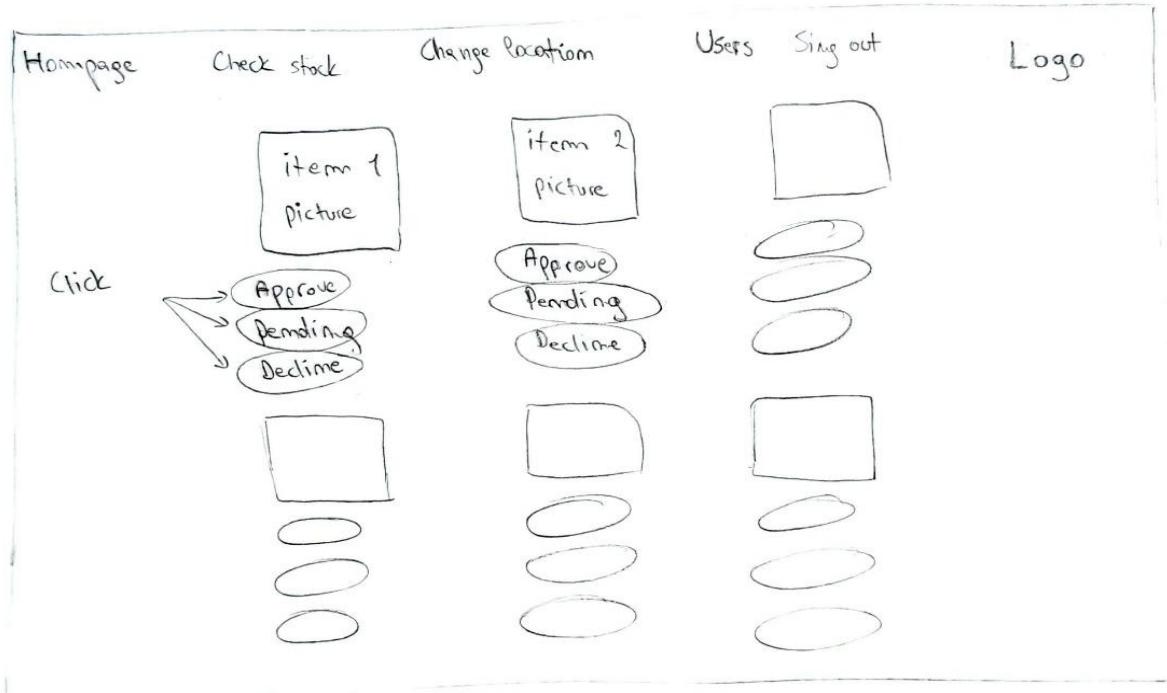
Metropolitan Innovation:

- Classrooms
- Labs
- Storerooms

© 2025 UMT Inventory. All rights reserved

Figure B.2.1.1: "Finance Homepage – Version A"

Home page (Finance)



Homepage (finance)

When we click one of the buttons...

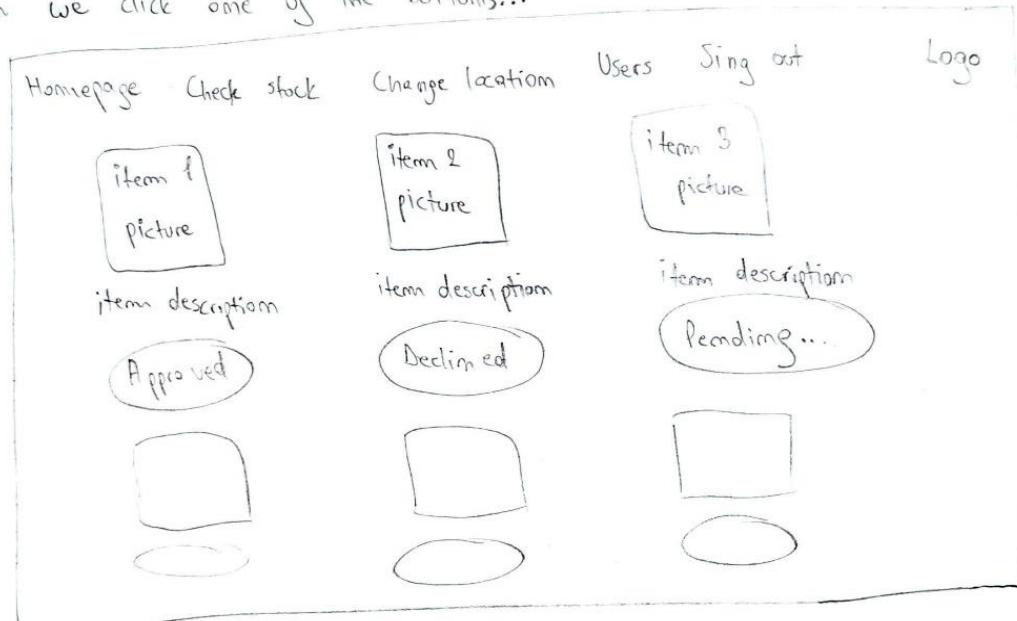
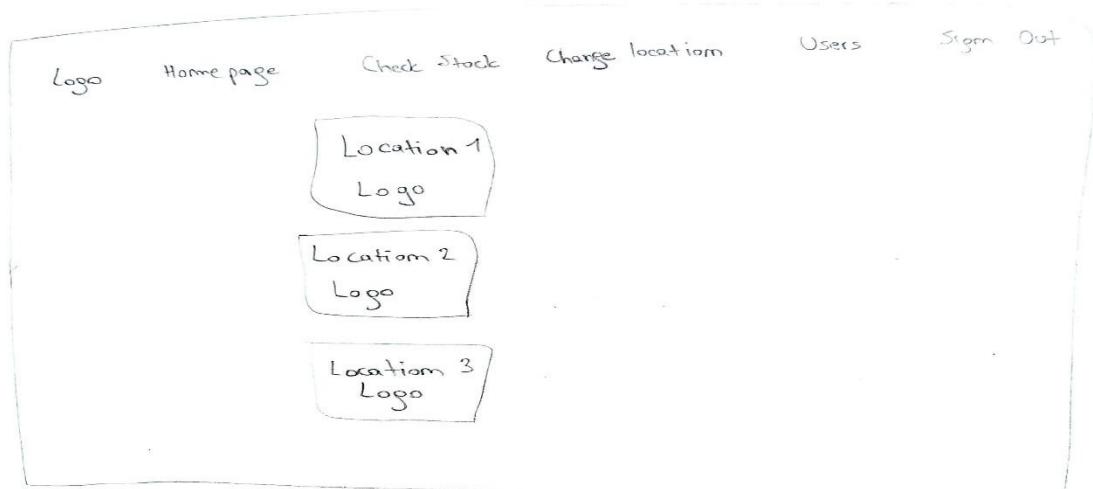


Figure B.2.1.2: “Finance Homepage – Version B”

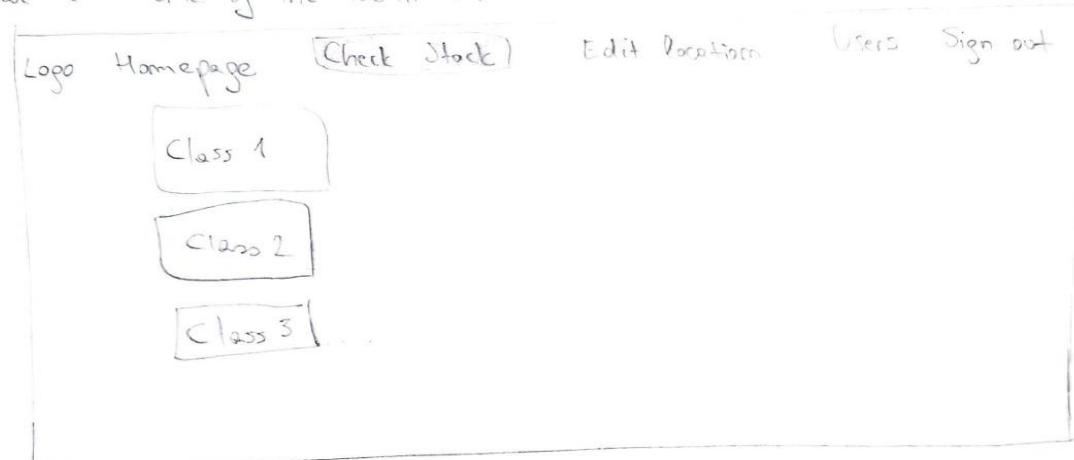


Check Stock (Finance)



Check Stock 1 (finance)

When we click one of the locations...



Check Stock 2 (finance)

When we click one of the classrooms...

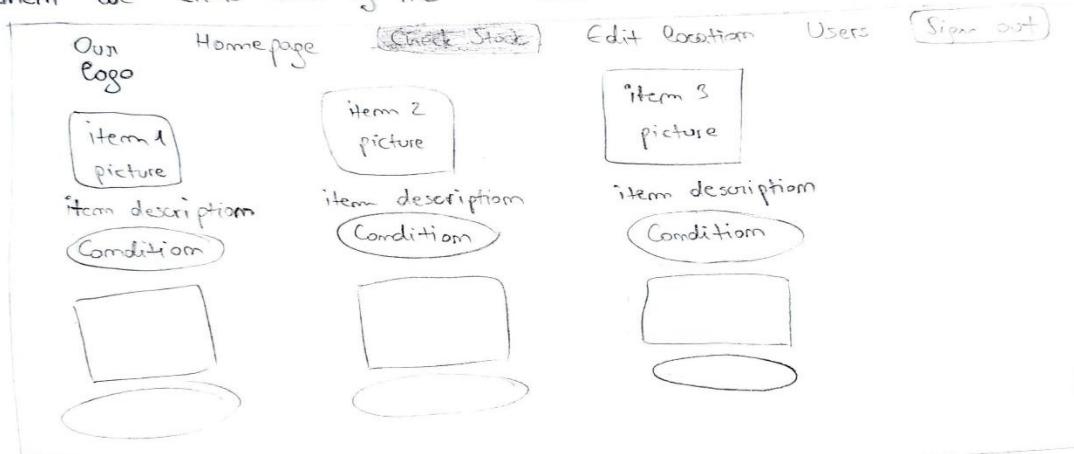
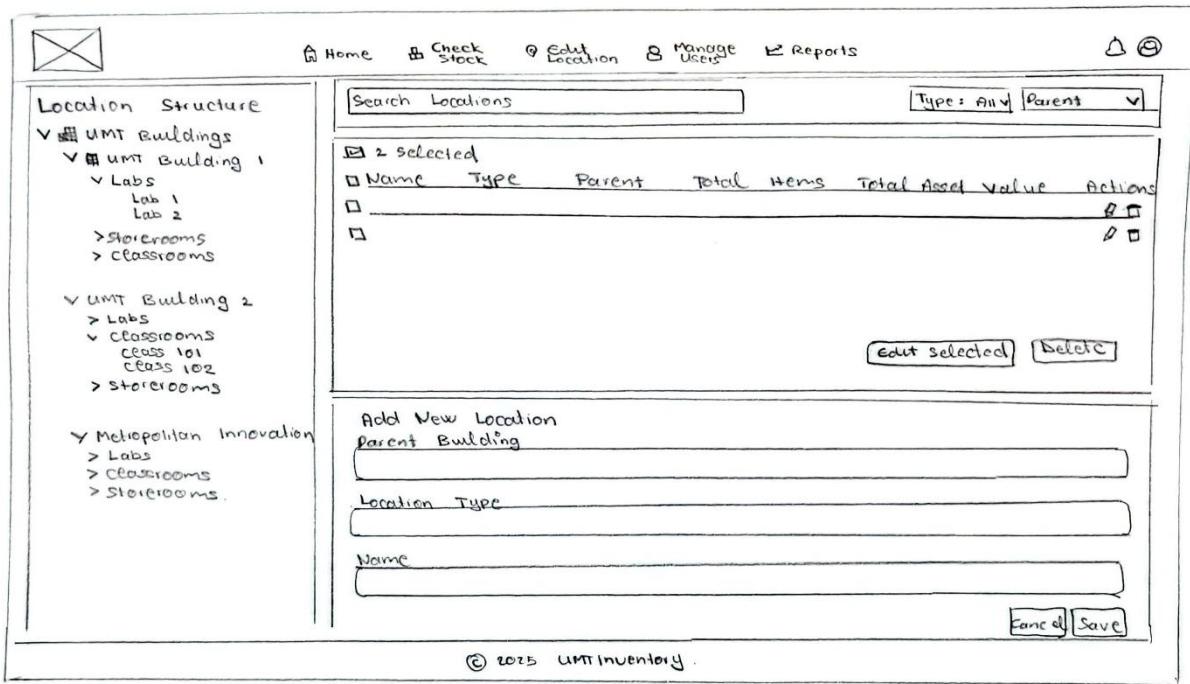


Figure B.2.1.3: "Finance Check Stock Page"

"Finance User - Edit Location Page (wireframe)"



The wireframe shows a navigation bar at the top with links: Home, Check Stock, Edit Location, Manage Users, Reports, and a sign-out icon.

Location Structure:

- UMT Buildings
 - UMT Building 1
 - Labs
 - Lab 1
 - Lab 2
 - Storerooms
 - Classrooms
 - UMT Building 2
 - Labs
 - Classrooms
 - Class 101
 - Class 102
 - Storerooms
 - Metropolitan Innovation
 - Labs
 - Classrooms
 - Storerooms

Search Locations: Search bar with placeholder "Search Locations". A dropdown menu shows "Type: All" and "Parent" with a checked checkbox.

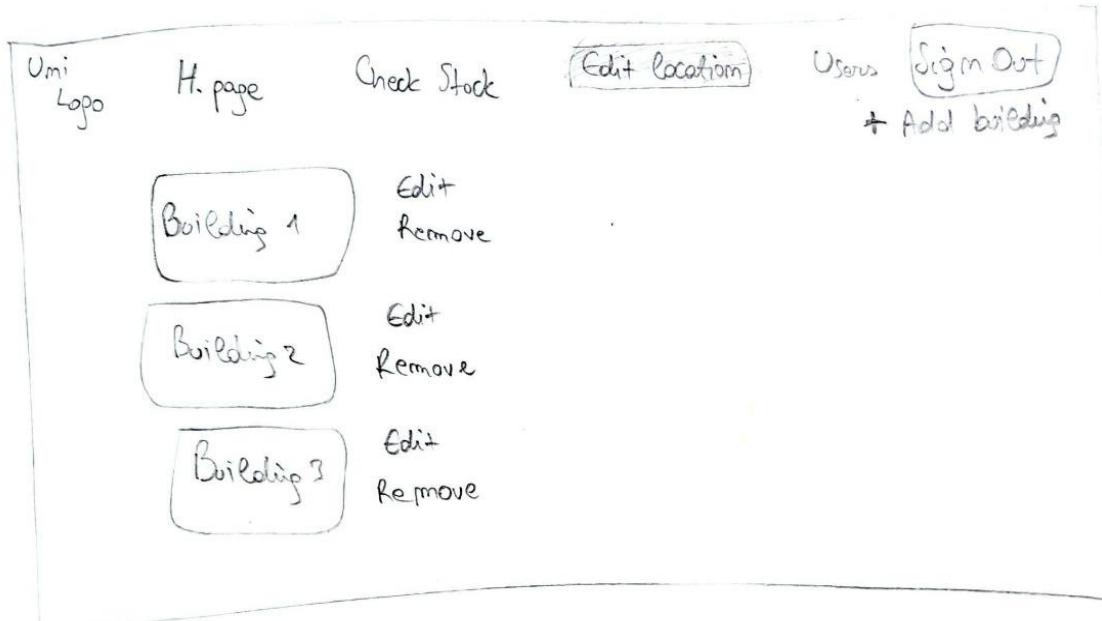
Table: A table titled "2 selected" showing location details. The columns are Name, Type, Parent, Total Items, Total Asset Value, and Actions. There are two rows of data, each with edit and delete icons.

Add New Location: Form fields for "Parent Building" (dropdown), "Location Type" (dropdown), and "Name" (text input). Buttons for "Cancel" and "Save" are at the bottom right.

Footer: © 2025 UMTInventory

Figure B.2.1.4: "Finance Edit Location Page – Version A"

Edit location 1 (Finance)





Edit Location 2 (Finance)

University logo Homepage Check Stock Edit location Users Sign Out

≡

Building 1

Building 2

Building 3

This sketch shows a top navigation bar with links for University logo, Homepage, Check Stock, Edit location, Users, and Sign Out. Below the navigation is a menu icon (≡). Three rounded rectangular boxes labeled Building 1, Building 2, and Building 3 are arranged vertically.

Edit location (finance)
form

Own Logo Homepage Check Stock Edit location Users Sign Out

Location

Type
Building, Class etc.

Name

Class name

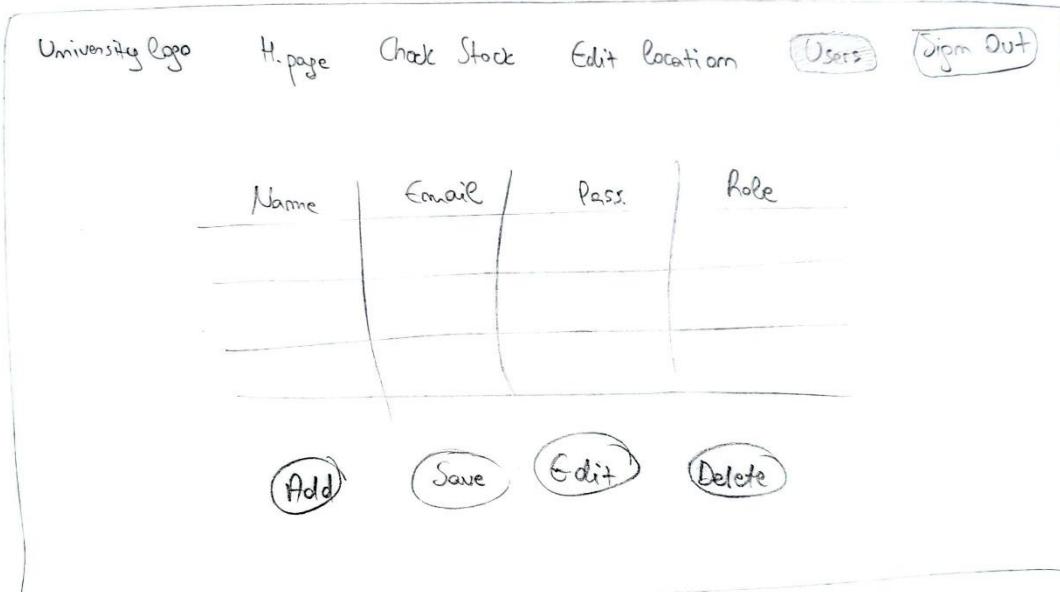
Additional Notes

Submit

This sketch shows a top navigation bar with links for Own Logo, Homepage, Check Stock, Edit location, Users, and Sign Out. Below the navigation is a section labeled Location. It includes fields for Type (with a sub-note: Building, Class etc.), Name, Class name, Additional Notes, and a Submit button at the bottom.

Figure B.2.1.5: “Finance Edit Location Page – Version B”

Manage Users (Finance)

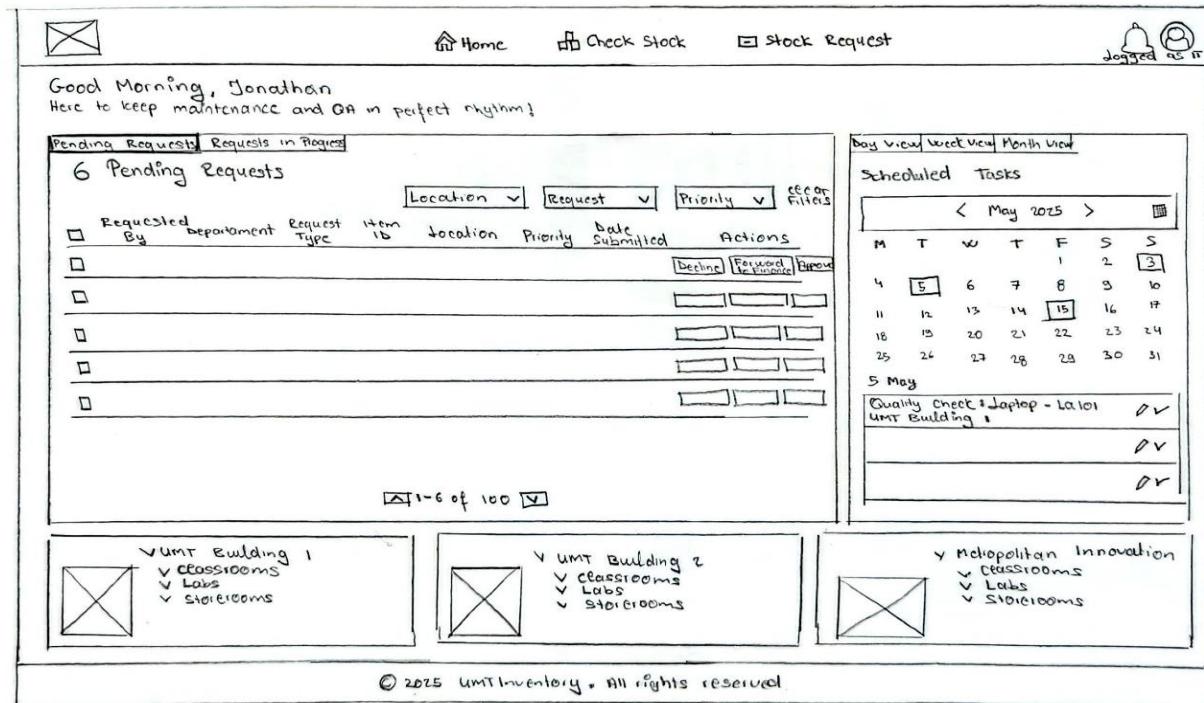


The sketch depicts a user interface for managing users. At the top, there are navigation links: "University Logo", "Home page", "Check Stock", "Edit Locations", "Users" (which is highlighted with a box), and "Sign Out". Below these are four columns labeled "Name", "Email", "Pass.", and "Role". Underneath the columns are four circular buttons labeled "Add", "Save", "Edit", and "Delete".

Figure B.2.1.6: “Finance Manage Users Page”

B.2.2 IT Department

IT User - Home Page, Pending requests view



The sketch shows the IT department's home page with a pending requests view. At the top, there are links for "Home", "Check Stock", "Stock Request", and a user icon indicating "Logged as IT". A message says "Good Morning, Jonathan. Here to keep maintenance and QA in perfect rhythm!". Below this is a section titled "6 Pending Requests" with a table header: "Requested By", "Department", "Request Type", "Item ID", "Location", "Priority", "Date Submitted", and "Actions". The table contains six rows of pending requests. To the right is a calendar for May 2025 showing "Scheduled Tasks" for each day. At the bottom, there are three boxes for "UMT Building 1", "UMT Building 2", and "Metropolitan Innovation", each listing "Classrooms", "Labs", and "Storage". The footer includes a copyright notice: "© 2025 UMT Inventory. All rights reserved."

Figure B.2.2.1: “IT – Home Page: Pending Requests View”

"IT User - Home Page, Requests In-Progress View"

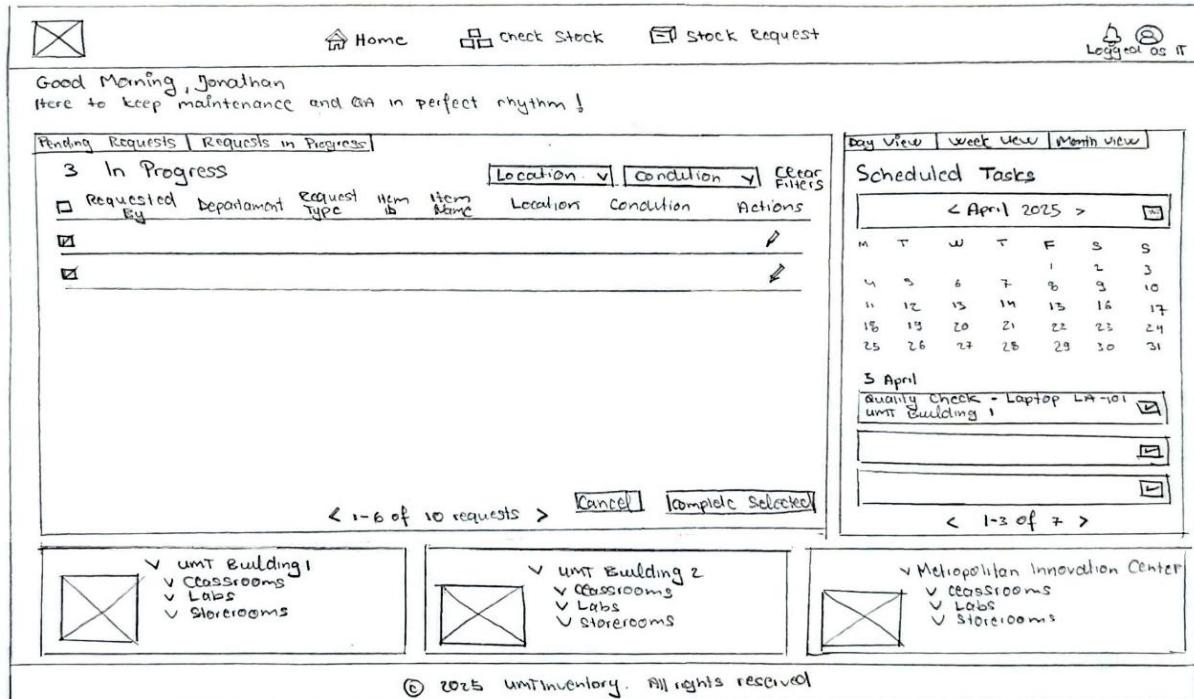


Figure B.2.2.2: "IT – Home Page: Requests In-Progress View"

"IT User - Check Stock Page (wireframe)"

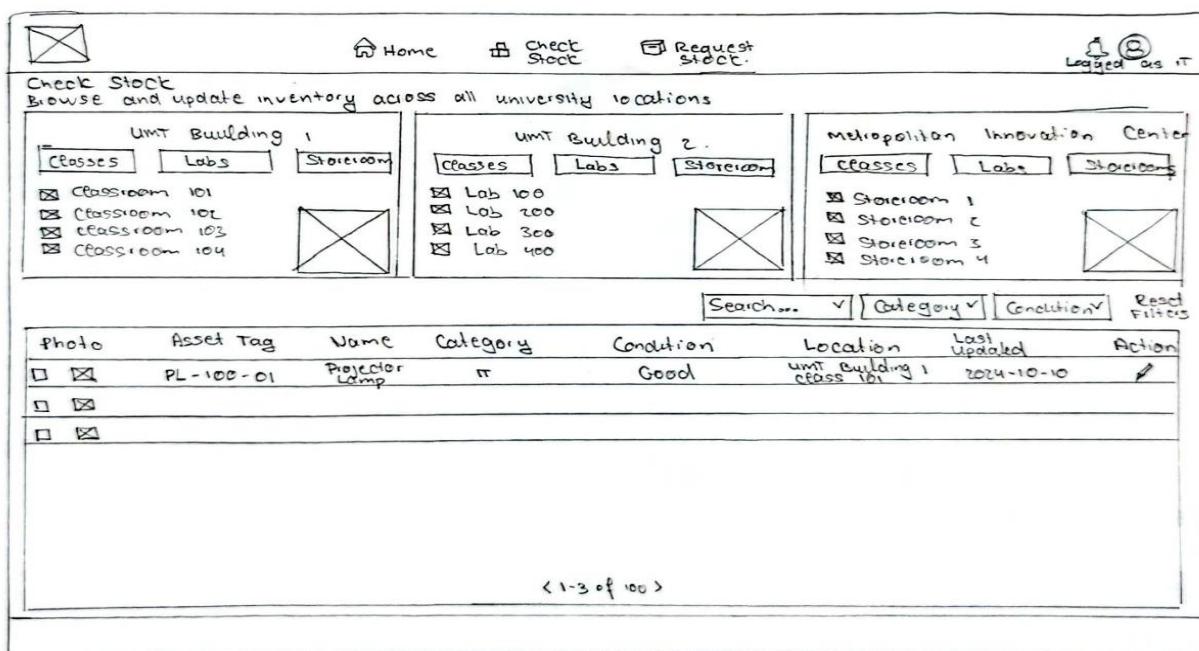


Figure B.2.2.3: "IT – Check Stock Page"



"IT User - Stock Request Page (wireframe)"

Name	Quantity	Proposed Location	Priority	Date	Status	Comments	Action
					Pending		
					Info Needed		
					Approved		
					Declined	X	
					Approved		
					Declined	X	

1-6 of 50

Additional Information Required

From Finance: Anne, Specify the model of laptop: Monday

Enter additional information: Bell XPS, Intel i7, 16 GB RAM, 13.40

Figure B.2.2.4: "IT – Stock Request Page"

B.2.3 Service Department

"Service Department User - Home Page - Pending Requests View."

Requested Department	Request Type	Item	Condition	Location	Priority	Date Submitted	Action
					X	✓	
					X	✓	
					X	✓	
					X	✓	
					X	✓	
					X	✓	

1-6 of 77

Scheduled Tasks

Day	Week	Month
< Feb 2025 >		
1		
2	3	4
3	10	11
4	12	13
5	17	18
6	19	20
7	21	22
8	24	25
9	26	27
10	28	

1-3 of 5

UMT Building 1: Classrooms, Labs, Storerooms

UMT Building 2: Classrooms, Labs, Storerooms

Metropolitan Innovation: Classrooms, Labs, Storerooms

© 2025 UMT Inventory. All rights reserved.

Figure B.2.3.1: "Service Department–Home Page: Pending Requests View"

Service Department User - Home Page - Requests in Progress View.

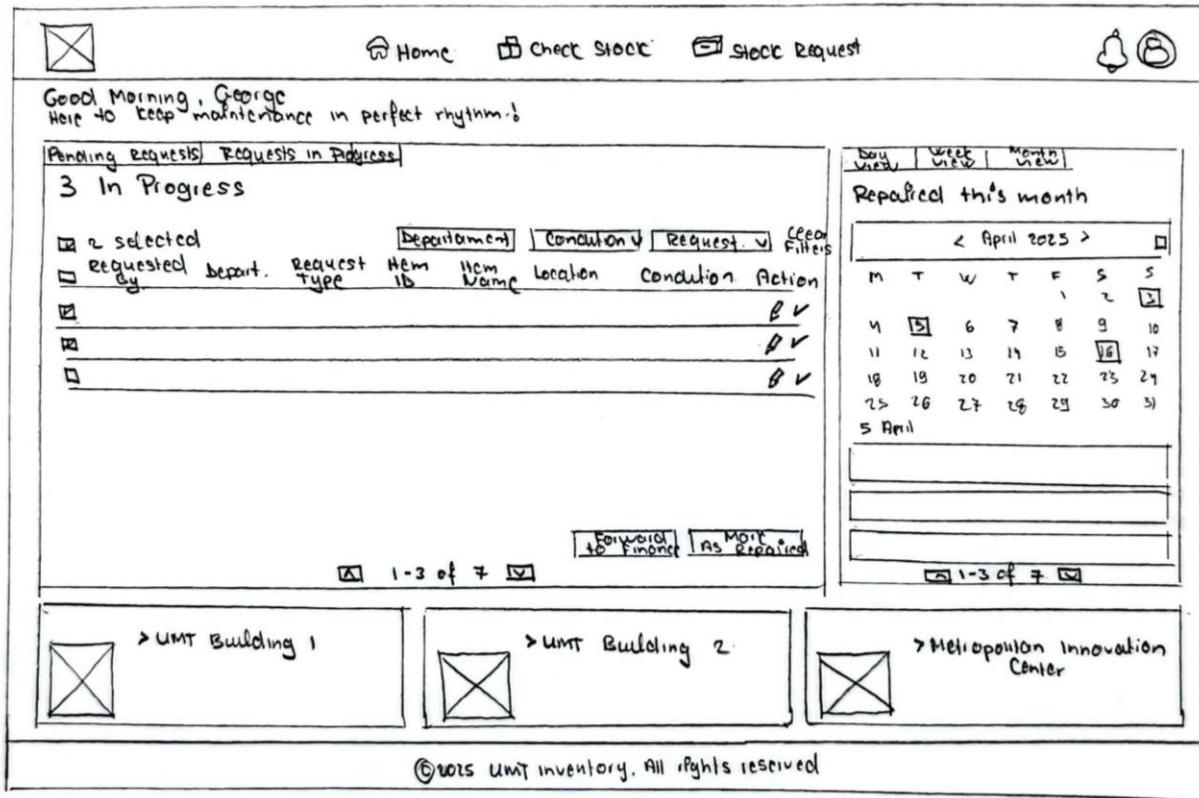


Figure B.2.3.2: “Service Department–Home Page: Request In-Progress View”

⌚ Appendix C: UI Mockups (Figma Screens)

Note: Some user interfaces have multiple design versions to reflect the iterative design process carried out by the team. These alternatives helped us evaluate usability and visual clarity in order to finalize the most effective solution.

C.1 Shared User Interfaces

This section includes the Login and Landing Page designs for each user role in the system.



Sign In

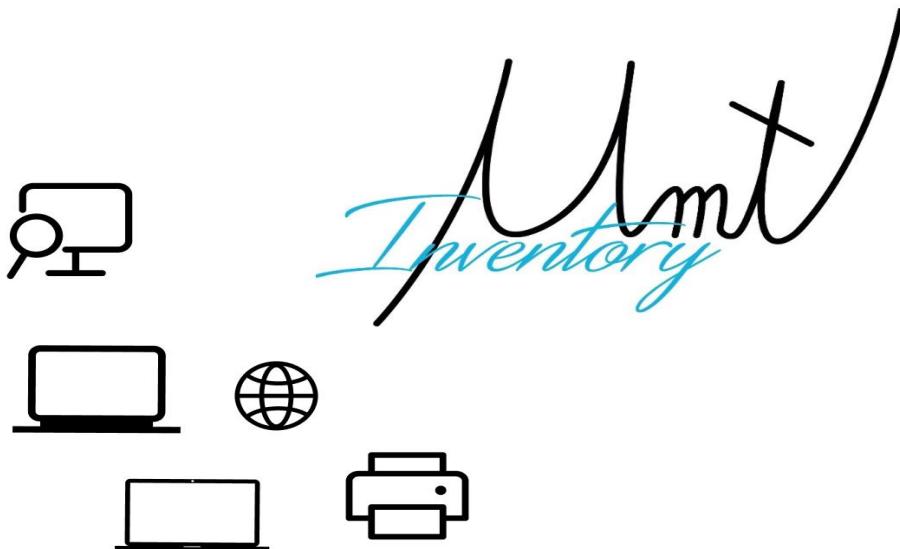


Figure C.1.1: “Landing Page”

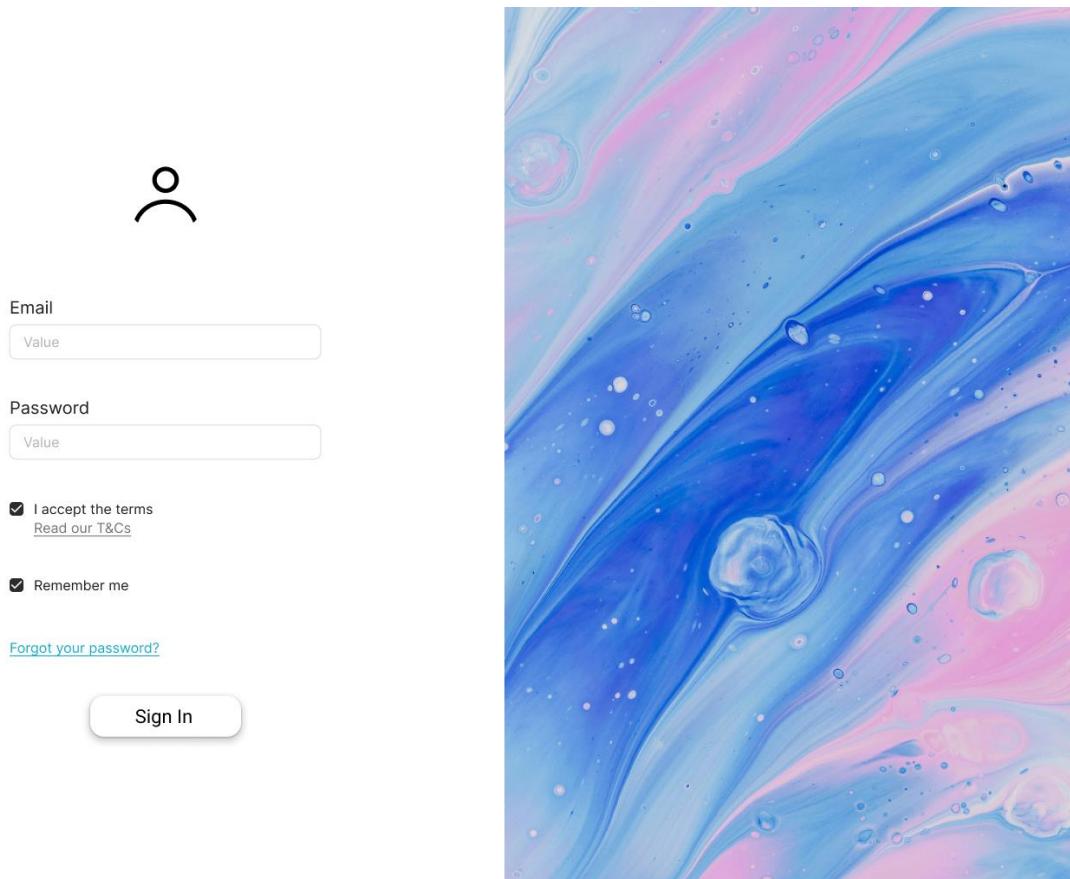


Figure C.1.2: “Login Page”

C.2 Finance Role – UI Mockups

UNIVERSITETI' METROPOLITAN TIRANA
Homepage
Check Stock
Edit Location
Manage Users
Sign Out

Approved (5)
Declined (2)
Pending (1)



Laptop, 13.6", 16GB, 256GB SSD, M3

Approve
Decline
Pending



Kamer Sigurie, Wireless IP Camera P2P

Approve
Decline
Pending



All-In-One Desktop, Windows 11 Home, 4GB RAM, 128GB SSD, Wireless Keyboard & Mouse

Approve
Decline
Pending



Brother 3-in-1 White & Black Inkjet Printer

Approve
Decline
Pending



Monitor 24" 16GB / 512 GB

Approve
Decline
Pending



A4Tech ComfortKey FN Keyboard

Approve
Decline
Pending

When we click one of them



UNIVERSITETI' METROPOLITAN TIRANA
Homepage
Check Stock
Edit Location
Manage Users
Sign Out

Approved (5)
Declined (2)
Pending (1)



Laptop, 13.6", 16GB, 256GB SSD, M3

Declined



Kamer Sigurie, Wireless IP Camera P2P

Approved



All-In-One Desktop, Windows 11 Home, 4GB RAM, 128GB SSD, Wireless Keyboard & Mouse

Pending...



Brother 3-in-1 White & Black Inkjet Printer

Approved



Monitor 24" 16GB / 512 GB

Pending...



A4Tech ComfortKey FN Keyboard

Declined

Figure C.2.1: “Finance Homepage – Version 1”

Good Morning, Jonathan
Here to maintain flawless inventory rhythm !

Pending Requests My Decisions

7 Pending Requests

<input checked="" type="checkbox"/> Selected	Item Name	Quantity	Requested By	Category	Needed By	Priority	Location	Actions
<input type="checkbox"/>	Laptop LA01	2	University Staff	IT	2025-04-12	Medium	Building 1 Room 101	X i ✓
<input type="checkbox"/>	Chair C01	2	Service Department	Non - IT	Row	High	Row	X i ✓
<input type="checkbox"/>	Row	Row	IT	IT	Row	Low	Row	X i ✓
<input type="checkbox"/>	Row	Row	Club Coordinator	IT	Row	High	Row	X i ✓
<input type="checkbox"/>	Row	Row	Row item	Non - IT	Row	Low	Row	X i ✓
<input type="checkbox"/>	Row	Row	Row item	IT	Row	Low	Row	X i ✓
<input type="checkbox"/>	Row	Row	Row item	IT	Row	Medium	Row	X i ✓

1-7 of 7 Requests Decline Selected Approve Selected



- UMT Building 1
- Classrooms
- Labs
- Storerooms

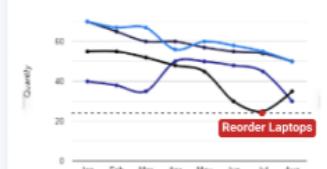


- UMT Building 2
- Classrooms
- Labs
- Storerooms



- Metropolitan Innovation Center
- Classrooms
- Labs
- Storerooms

Low-Stock Alerts



Quantity

Reorder Laptops

Jan Feb Mar Apr May Jun Jul Aug

Upcoming Reorders

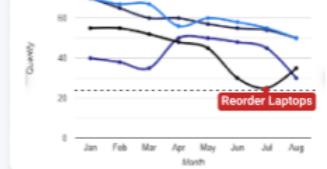
Department: Next 7 days

Days until reorder: 1 2 3 4 5 6 7

- Laptop Model X
- Projector Bulb
- Whiteboard Markers
- Printer Cartridges

Reorder Selected Cancel

Low-Stock Alerts



Quantity

Reorder Laptops

Jan Feb Mar Apr May Jun Jul Aug

Upcoming Reorders

Department: Next 7 days

Days until reorder: 1 2 3 4 5 6 7

- Laptop Model X
- Projector Bulb
- Whiteboard Markers
- Printer Cartridges

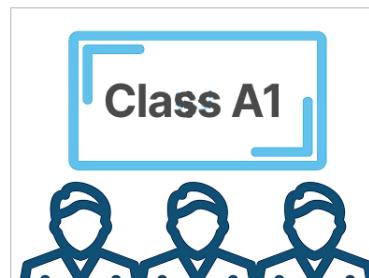
Reorder Selected Cancel

Figure C.2.2: “Finance Homepage – Version 2”

95



When we click
one of them



When we click
one of them



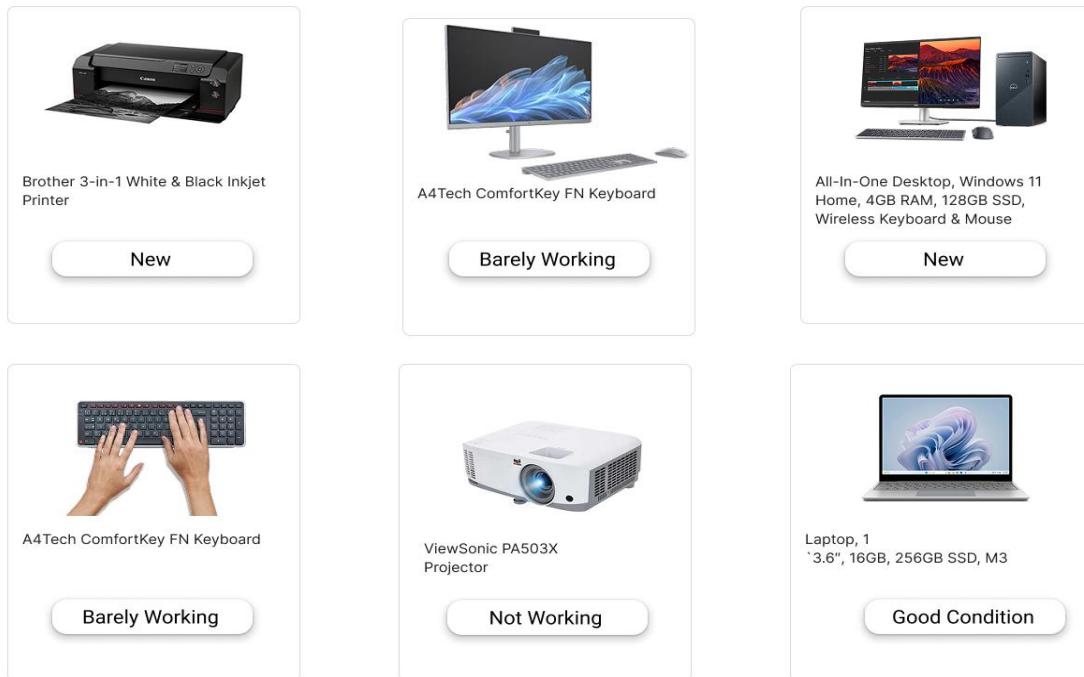
[New \(5\)](#)
[Not Working \(2\)](#)
[Barley Working \(2\)](#)
[Good Condition \(2\)](#)


Figure C.2.3: “Finance Check Stock Page— Version 1”

The screenshot shows the 'Check Stock' section of the UMT Inventory System. It displays three building locations with their respective inventories:

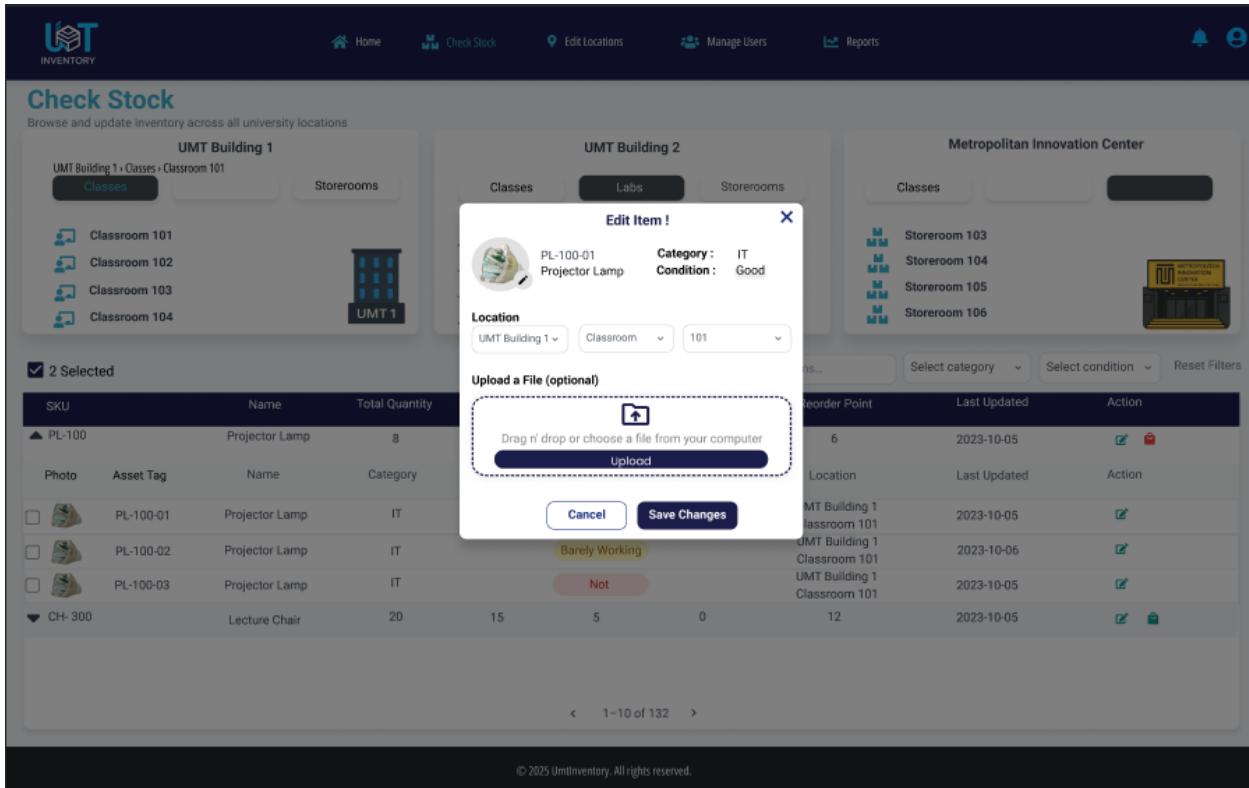
- UMT Building 1**: Shows Classroom 101 with sub-items Classroom 101, 102, 103, and 104. A building icon labeled "UMT 1" is shown.
- UMT Building 2**: Shows Lab 102, 103, 104, and 105. A building icon labeled "UMT 2" is shown.
- Metropolitan Innovation Center**: Shows Storeroom 103, 104, 105, and 106. An icon of a server or storage unit is shown.

Selected Items: 2 Selected

SKU	Name	Total Quantity	Good	Qty by Condition	Reorder Point	Usage Status	Last Updated	Action
PL-100	Projector Lamp	8	4	Barely Working: 2 Not working: 2	6	Low Stock	2023-10-05	
PL-100-01	Projector Lamp	IT		Good			2023-10-05	
PL-100-02	Projector Lamp	IT		Barely Working			2023-10-06	
PL-100-03	Projector Lamp	IT		Not			2023-10-05	
CH-300	Lecture Chair	20	15	5 0	12	Available	2023-10-05	

Search items... Select category Select condition Reset Filters

© 2025 UmtInventory. All rights reserved.



Check Stock

Browse and update inventory across all university locations

UMT Building 1

UMT Building 1 > Classes > Classroom 101

UMT Building 2

Metropolitan Innovation Center

Edit Item !

PL-100-01 Projector Lamp

Category : IT Condition : Good

Location : UMT Building 1 Classroom 101

Upload a File (optional)

Drag n' drop or choose a file from your computer

Upload

Cancel Save Changes

Barely Working Not

SKU	Name	Total Quantity	Photo	Asset Tag	Name	Category	Location	Last Updated	Action
PL-100	Projector Lamp	8		PL-100-01	Projector Lamp	IT	UMT Building 1 Classroom 101	2023-10-05	
				PL-100-02	Projector Lamp	IT	UMT Building 1 Classroom 101	2023-10-06	
				PL-100-03	Projector Lamp	IT	UMT Building 1 Classroom 101	2023-10-05	
CH-300	Lecture Chair	20					UMT Building 1 Classroom 101	2023-10-05	

1–10 of 132

© 2025 UmInventory. All rights reserved.

Figure C.2.4: “Finance Check Stock Page– Version 2”



Add Building

Mabye this button or the other one?

+ Add Building

G1

G2

G3

Delete Edit

Delete Edit

Delete Edit

Add Classroom

+ Add Classroom

Class A1

Delete Edit

Class A2

Delete Edit

Class B1

Delete Edit

Class B2

Delete Edit

New Location Information

Location Type *

Value: Building, Classroom, Office, Storage Room

Location Name *

e.g G4

Floor Number

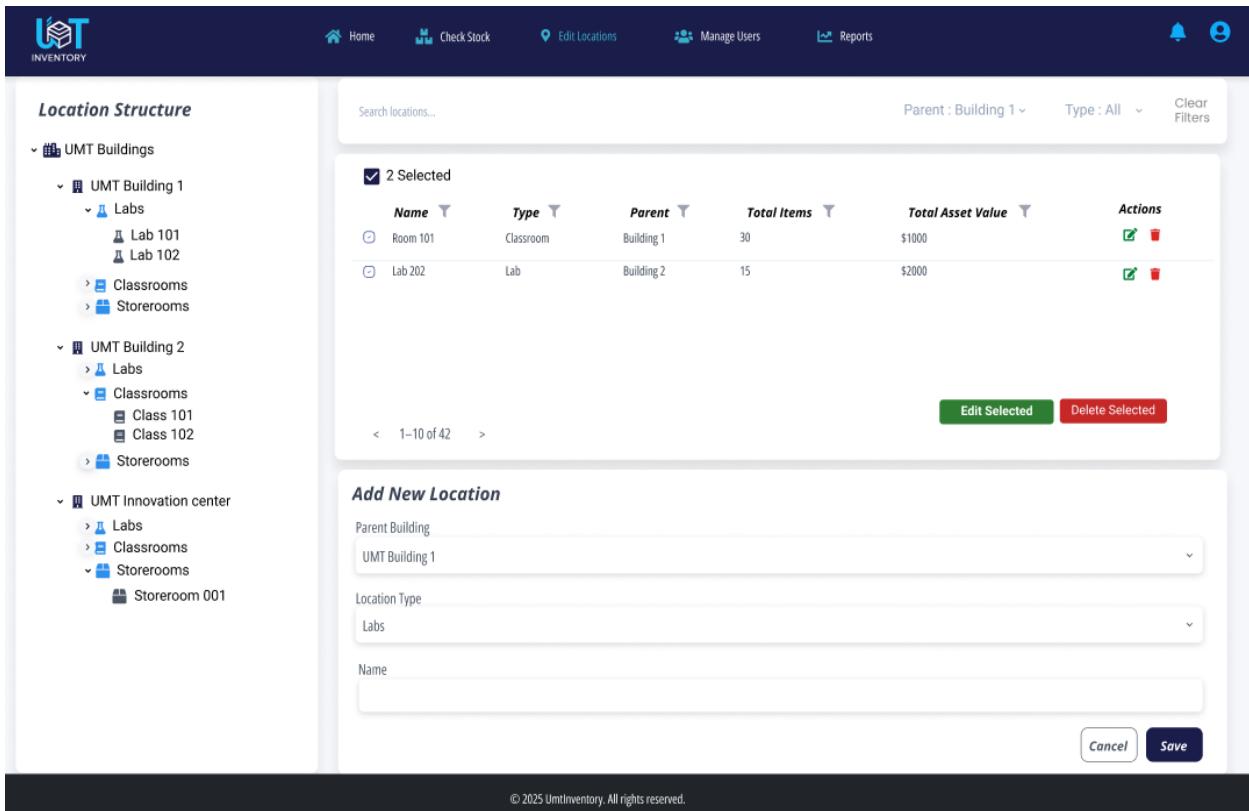
e.g G4

Additional Notes

Loren Ipsum...

Submit

Figure C.2.5: “Finance Edit Location Page– Version 1”



Location Structure

Search locations...

Parent : Building 1 ▾ Type : All ▾ Clear Filters

2 Selected

Name	Type	Parent	Total Items	Total Asset Value	Actions	
Room 101	Classroom	Building 1	30	\$1000		
Lab 202	Lab	Building 2	15	\$2000		

< 1–10 of 42 >

Edit Selected **Delete Selected**

Add New Location

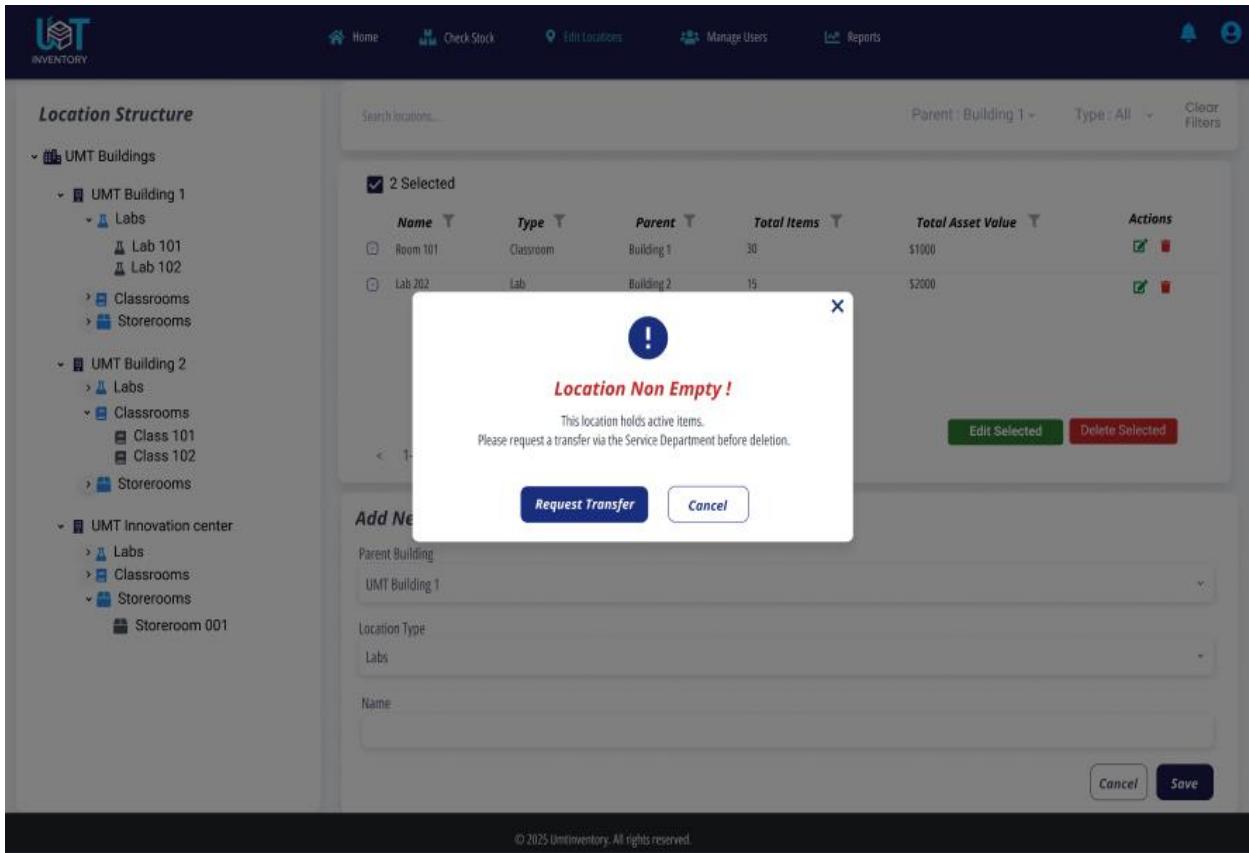
Parent Building: UMT Building 1

Location Type: Labs

Name:

Cancel **Save**

© 2025 Umthinventory. All rights reserved.



Location Structure

Search locations...

Parent : Building 1 ▾ Type : All ▾ Clear Filters

2 Selected

Name	Type	Parent	Total Items	Total Asset Value	Actions	
Room 101	Classroom	Building 1	30	\$1000		
Lab 202	Lab	Building 2	15	\$2000		

X

Location Non Empty!

This location holds active items.
Please request a transfer via the Service Department before deletion.

Request Transfer **Cancel**

Add New Location

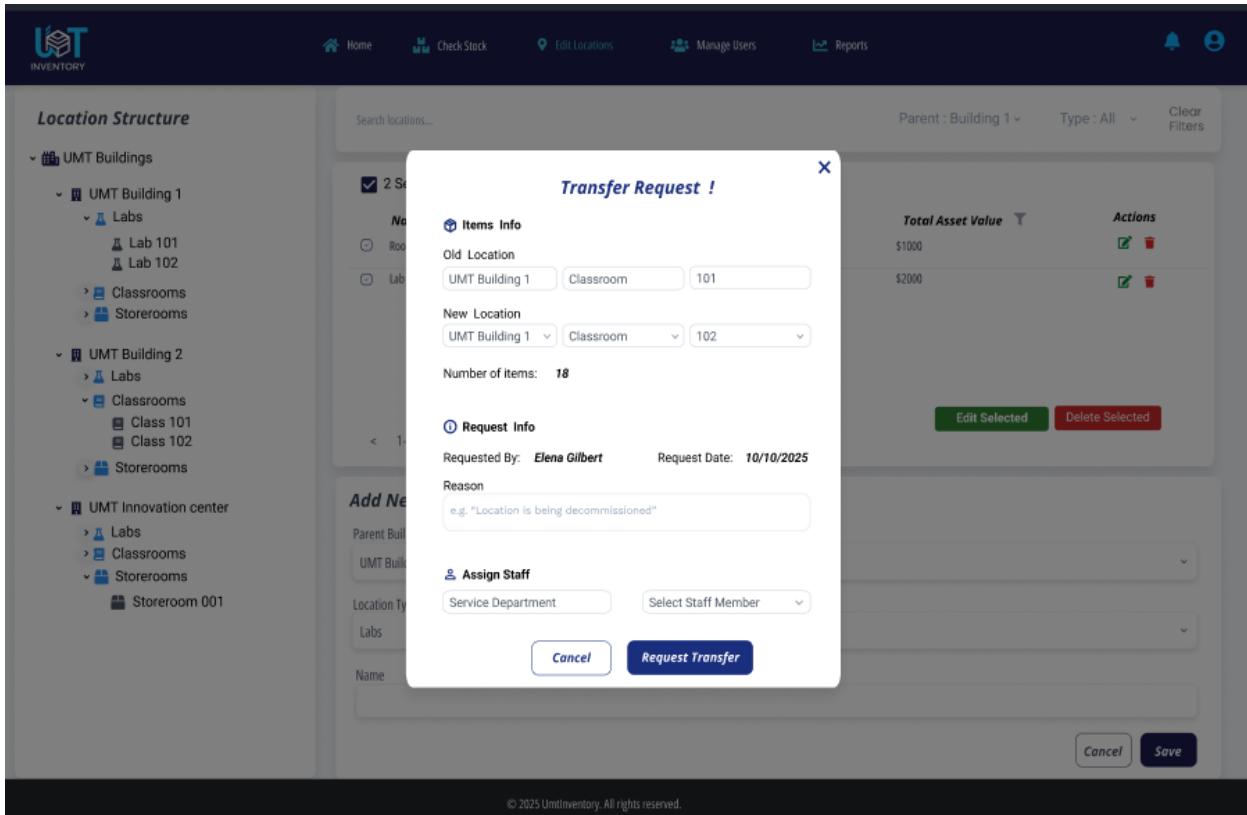
Parent Building: UMT Building 1

Location Type: Labs

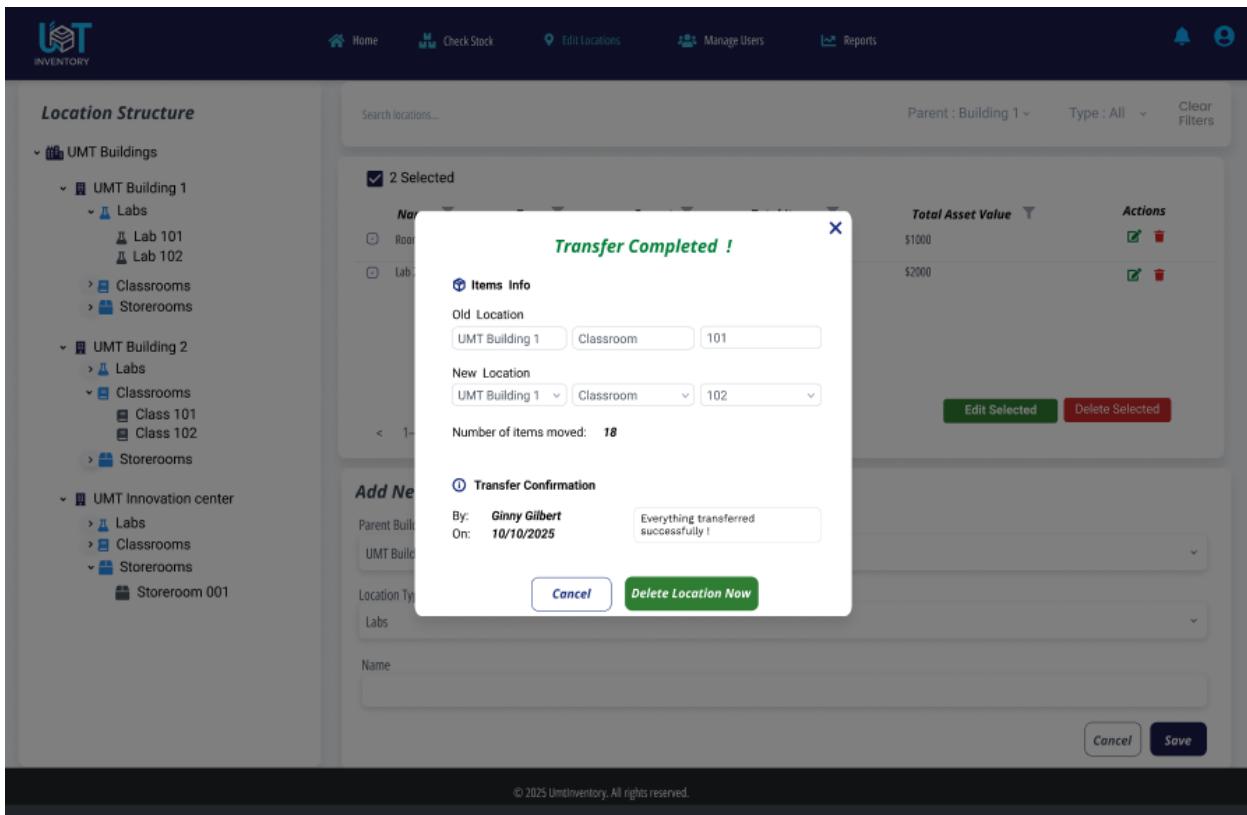
Name:

Cancel **Save**

© 2025 Umthinventory. All rights reserved.



© 2025 UmtInventory. All rights reserved.



© 2025 UmtInventory. All rights reserved.

Figure C.2.6: “Finance Edit Location Page– Version 2”


Users

Select All <input type="checkbox"/>	Name	Email	Password	Role	
<input type="checkbox"/>	Anjeza Xhelilaj	anxhelilaj@umt.edu.al	XXXXXXXXXXXX	Finance	
<input checked="" type="checkbox"/>	Xhesjano Halla	xhhalla@umt.edu.al	XXXXXXXXXXXXXXXXXX	IT	
<input type="checkbox"/>	Ema Elezi	emelezi@umt.edu.al	XXXXXXXXXXXXXXXXXX	Service Department	
<input type="checkbox"/>	Laura Dule	ladule@umt.edu.al	XXXXXXXXXXXX	University Staff	
<input type="checkbox"/>	Filan Fisteku	fifisteku@umt.edu.al	XXXXXXXXXXXXXXXXXX	Club Coordinator	
<input type="checkbox"/>					


Users

Select All <input type="checkbox"/>	Name	Email	Password	Role	
<input type="checkbox"/>	Anjeza Xhelilaj	anxhelilaj@umt.edu.al	XXXXXXXXXXXX	Finance	
<input checked="" type="checkbox"/>	Xhesjano Halla	xhhalla@umt.edu.al	XXXXXXXXXXXXXXXXXX	IT	
<input type="checkbox"/>	Ema Elezi	emelezi@umt.edu.al	XXXXXXXXXXXXXXXXXX	Service Department	
<input type="checkbox"/>	Laura Dule	ladule@umt.edu.al	XXXXXXXXXXXX	University Staff	
<input type="checkbox"/>	Filan Fisteku	fifisteku@umt.edu.al	XXXXXXXXXXXXXXXXXX	Club Coordinator	

Figure C.2.7: “Finance Manage Users Page”



C.3: IT Department – UI Mockups

Pending Requests

Requested By	Department	Request Type	Item ID	Location	Priority	Date Submitted	Actions
John Smith	University Staff	Quality Check	LA001	Building 1 Room 101	Medium	2025-04-12	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
Row	Club Coordinator	New Stock	Row	Row	High	Row	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
Row	Row	Row item	Row	Row	Low	Row	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
Row	Row	Row item	Row	Row	High	Row	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
Row	Row	Row item	Row	Row	Low	Row	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
Row	Row	Row item	Row	Row	Low	Row	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>

1-6 of 77 Requests

Scheduled Tasks

Day View	Week View	Month View				
April 2025						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

5 April

- QC: LA001 – Laptop Building 1, Room 101
- New Stock: LA001 – Laptop Building 1, Room 101
- QC: LA001 – Laptop Building 1, Room 101

1-3 of 7

UMT Building 1: Classrooms, Labs, Storerooms
UMT Building 2: Classrooms, Labs, Storerooms
Metropolitan Innovation Center: Classrooms, Labs, Storerooms

© 2025 UmInventory. All rights reserved.

Requests in Progress

Requested By	Department	Request Type	Item ID	Item Name	Location	Condition	Actions
John Smith	University Staff	Quality Check	LA001	Laptop	Building 1 Room 101	Good	<input type="button" value="Edit"/>
Row	Club Coordinator	New Stock	Row	Row	Row	Barely Working	<input type="button" value="Edit"/>
Row	Row	Row	Row	Row	Row	Not Working	<input type="button" value="Edit"/>

1-6 of 77 Requests

Scheduled Tasks

Day View	Week View	Month View				
April 2025						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

5 April

- QC: LA001 – Laptop Building 1, Room 101
- New Stock: LA001 – Laptop Building 1, Room 101
- QC: LA001 – Laptop Building 1, Room 101

1-3 of 7

UMT Building 1: Classrooms, Labs, Storerooms
UMT Building 2: Classrooms, Labs, Storerooms
Metropolitan Innovation Center: Classrooms, Labs, Storerooms

© 2025 UmInventory. All rights reserved.

Good Morning, Jonathan
Here to keep maintenance and QA in perfect rhythm !

[Pending Requests](#) [Requests In Progress](#)

6 Pending Requests

Requested By	Department	Request Type	Item ID	Location	Priority	Date Submitted	Actions
<input type="checkbox"/> John Smith	University Staff	Quality Check	LA001	Building 1 Room 101	Medium	2025-04-12	X Email Check
<input type="checkbox"/> Row	Club Coordinator	New Stock	Row	Row	High	Row	X Email Check
<input type="checkbox"/> Row	Row	Row item	Row	Row	Low	Row	X Email Check
<input type="checkbox"/> Row	Row	Row item	Row	Row	High	Row	X Email Check
<input type="checkbox"/> Row	Row	Row item	Row	Row	Low	Row	X Email Check
<input type="checkbox"/> Row	Row	Row item	Row	Row	Low	Row	X Email Check

1-6 of 77 Requests

Forward Request to Finance

Item : Laptop - LA001

Requested by : John Smith

Justification : New device needed for incoming staff in Lab 201 of Building 1.

Urgency Level : High **Suggested Budget:** 100.000 ALL

Cancel Forward Request



UMT Building 1

- Classrooms
- Labs
- Storerooms



UMT Building 2

- Classrooms
- Labs
- Storerooms



Metropolitan Innovation Center

- Classrooms
- Labs
- Storerooms

© 2025 UmtInventory. All rights reserved.

Good Morning, Jonathan
Here to keep maintenance and QA in perfect rhythm !

[Pending Requests](#) [Requests In Progress](#)

3 In Progress

Requested By	Department	Request Type	Item ID	Item Name
<input checked="" type="checkbox"/> John Smith	University Staff	Quality Check	LA001	Laptop
<input checked="" type="checkbox"/> Row	Club Coordinator	New Stock	Row	Row
<input type="checkbox"/> Row	Row	Row	Row	Row

Edit Item Details

ID: LA001 Name: Laptop

Condition: Not Working

Consider replacing this device!

Location: Building 1 Room 101

Drag n' drop or choose a file from your computer

Cancel Save Changes

Scheduled Tasks

Day View Week Month

April 2025

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

5 April

QC: LA001 – Laptop Building 1, Room 101

New Stock: LA001 – Laptop Building 1, Room 101

QC: LA001 – Laptop Building 1, Room 101

1-3 of 7



UMT Building 1

- Classrooms
- Labs
- Storerooms



UMT Building 2

- Classrooms
- Labs
- Storerooms

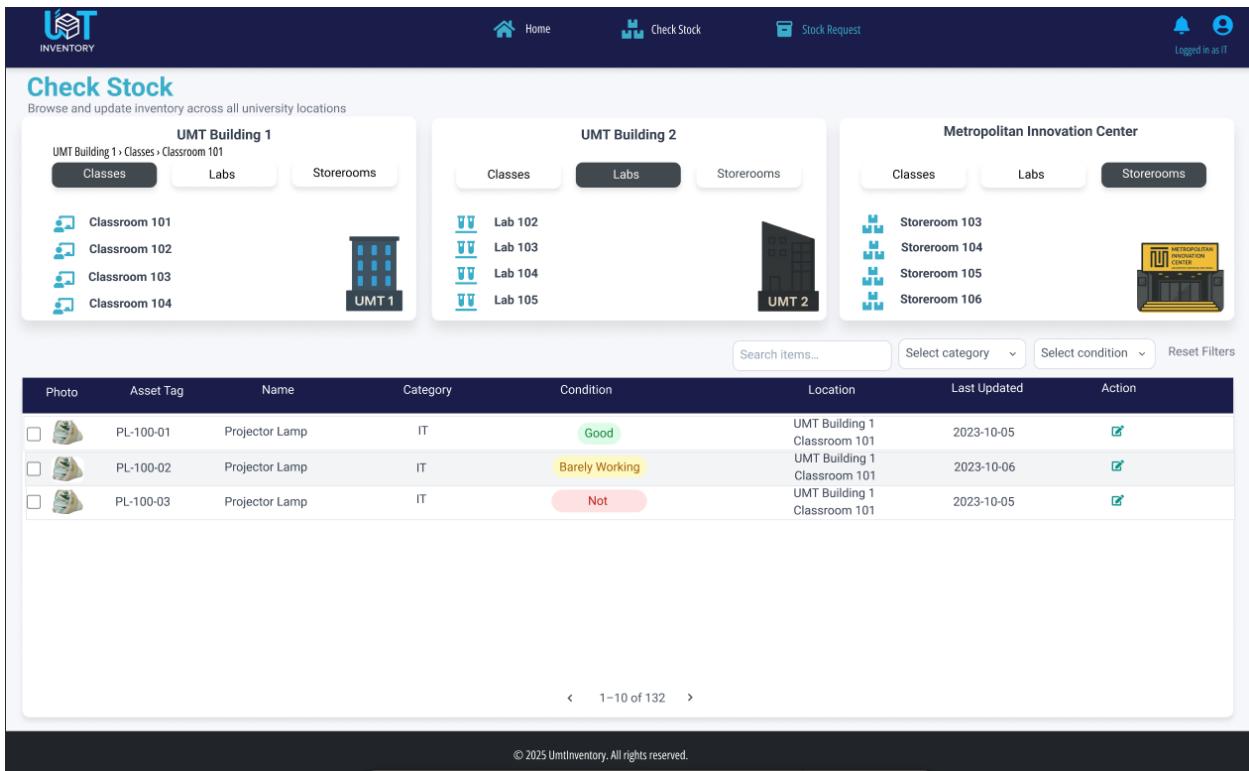


Metropolitan Innovation Center

- Classrooms
- Labs
- Storerooms

© 2025 UmtInventory. All rights reserved.

Figure C.3.1: “IT Department - Home Page”



Check Stock
Browse and update inventory across all university locations

UMT Building 1
UMT Building 1 > Classes > Classroom 101

	Classes	Labs	Storerooms
	Classroom 101		
	Classroom 102		
	Classroom 103		
	Classroom 104		

UMT Building 2

	Classes	Labs	Storerooms
		Lab 102	
		Lab 103	
		Lab 104	
		Lab 105	

Metropolitan Innovation Center

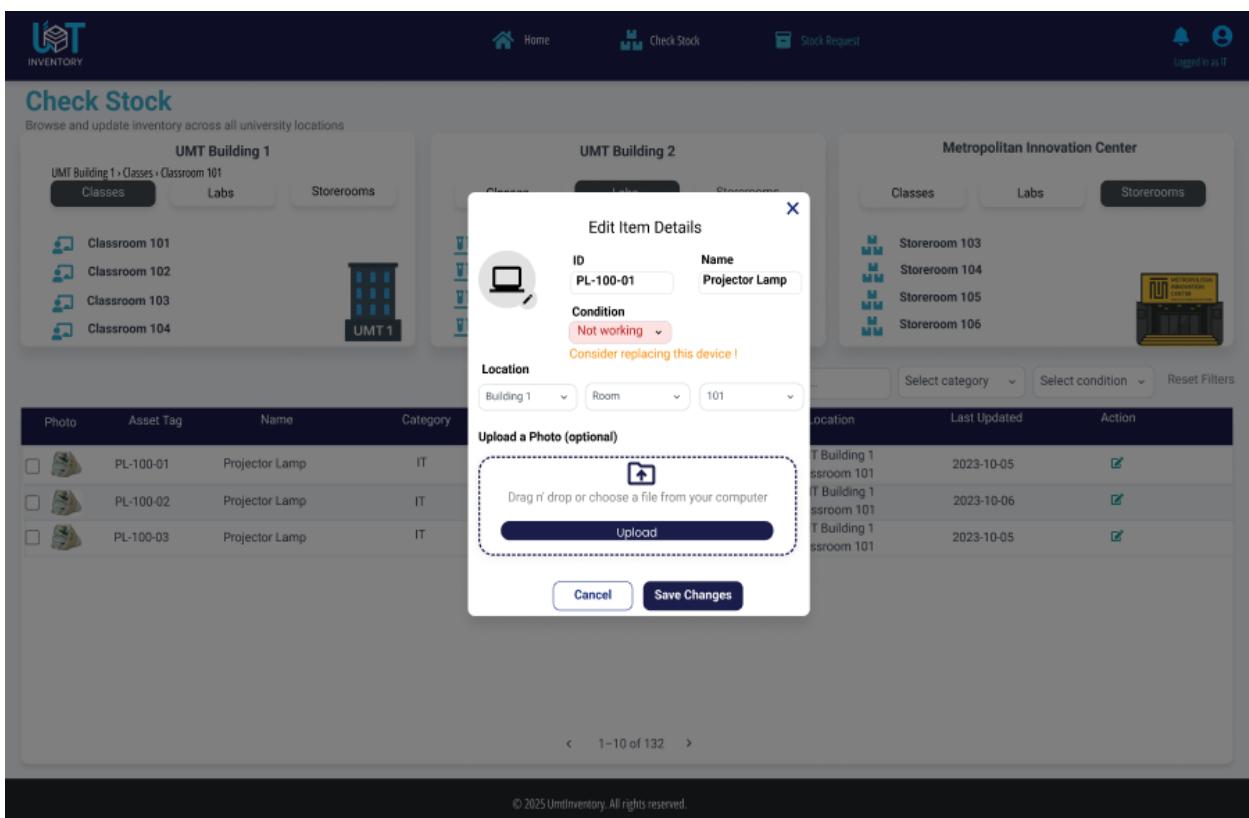
	Classes	Labs	Storerooms
			Storeroom 103
			Storeroom 104
			Storeroom 105
			Storeroom 106

Search items... Select category ▾ Select condition ▾ Reset Filters

Photo	Asset Tag	Name	Category	Condition	Location	Last Updated	Action
	PL-100-01	Projector Lamp	IT	Good	UMT Building 1 Classroom 101	2023-10-05	<input checked="" type="checkbox"/>
	PL-100-02	Projector Lamp	IT	Barely Working	UMT Building 1 Classroom 101	2023-10-06	<input checked="" type="checkbox"/>
	PL-100-03	Projector Lamp	IT	Not	UMT Building 1 Classroom 101	2023-10-05	<input checked="" type="checkbox"/>

1–10 of 132

© 2025 UmtInventory. All rights reserved.



Check Stock
Browse and update inventory across all university locations

UMT Building 1
UMT Building 1 > Classes > Classroom 101

	Classes	Labs	Storerooms
	Classroom 101		
	Classroom 102		
	Classroom 103		
	Classroom 104		

UMT Building 2

	Classes	Labs	Storerooms
		Lab 102	
		Lab 103	
		Lab 104	
		Lab 105	

Metropolitan Innovation Center

	Classes	Labs	Storerooms
			Storeroom 103
			Storeroom 104
			Storeroom 105
			Storeroom 106

Edit Item Details

ID	PL-100-01	Name	Projector Lamp
Condition	Not working	Consider replacing this device!	
Location	Building 1	Room	101

Upload a Photo (optional)

Drag n' drop or choose a file from your computer

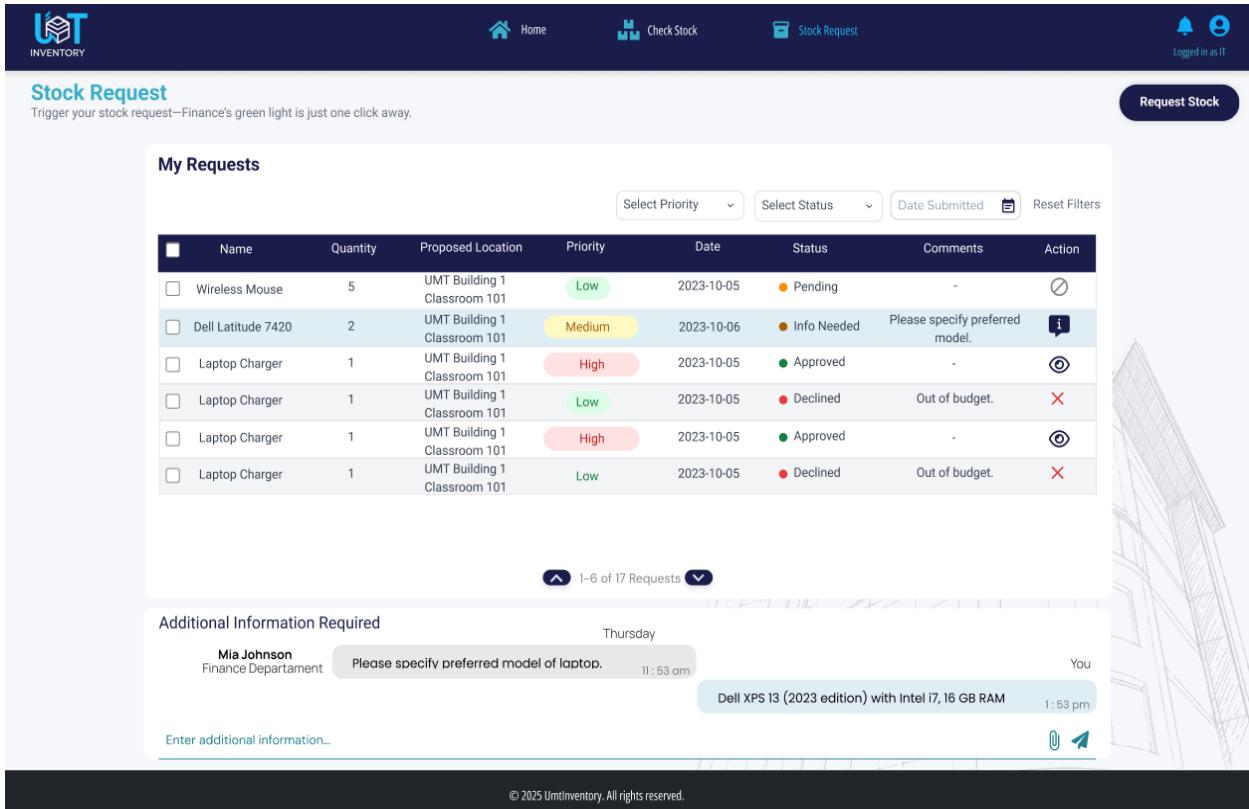
Upload

Cancel Save Changes

1–10 of 132

© 2025 UmtInventory. All rights reserved.

Figure C.3.2: “IT Department – Check Stock Page”



Stock Request
Trigger your stock request—Finance's green light is just one click away.

My Requests

Select Priority ▾ Select Status ▾ Date Submitted ▾ Reset Filters

<input type="checkbox"/>	Name	Quantity	Proposed Location	Priority	Date	Status	Comments	Action
<input type="checkbox"/>	Wireless Mouse	5	UMT Building 1 Classroom 101	Low	2023-10-05	Pending	-	
<input type="checkbox"/>	Dell Latitude 7420	2	UMT Building 1 Classroom 101	Medium	2023-10-06	Info Needed	Please specify preferred model.	
<input type="checkbox"/>	Laptop Charger	1	UMT Building 1 Classroom 101	High	2023-10-05	Approved	-	
<input type="checkbox"/>	Laptop Charger	1	UMT Building 1 Classroom 101	Low	2023-10-05	Declined	Out of budget.	
<input type="checkbox"/>	Laptop Charger	1	UMT Building 1 Classroom 101	High	2023-10-05	Approved	-	
<input type="checkbox"/>	Laptop Charger	1	UMT Building 1 Classroom 101	Low	2023-10-05	Declined	Out of budget.	

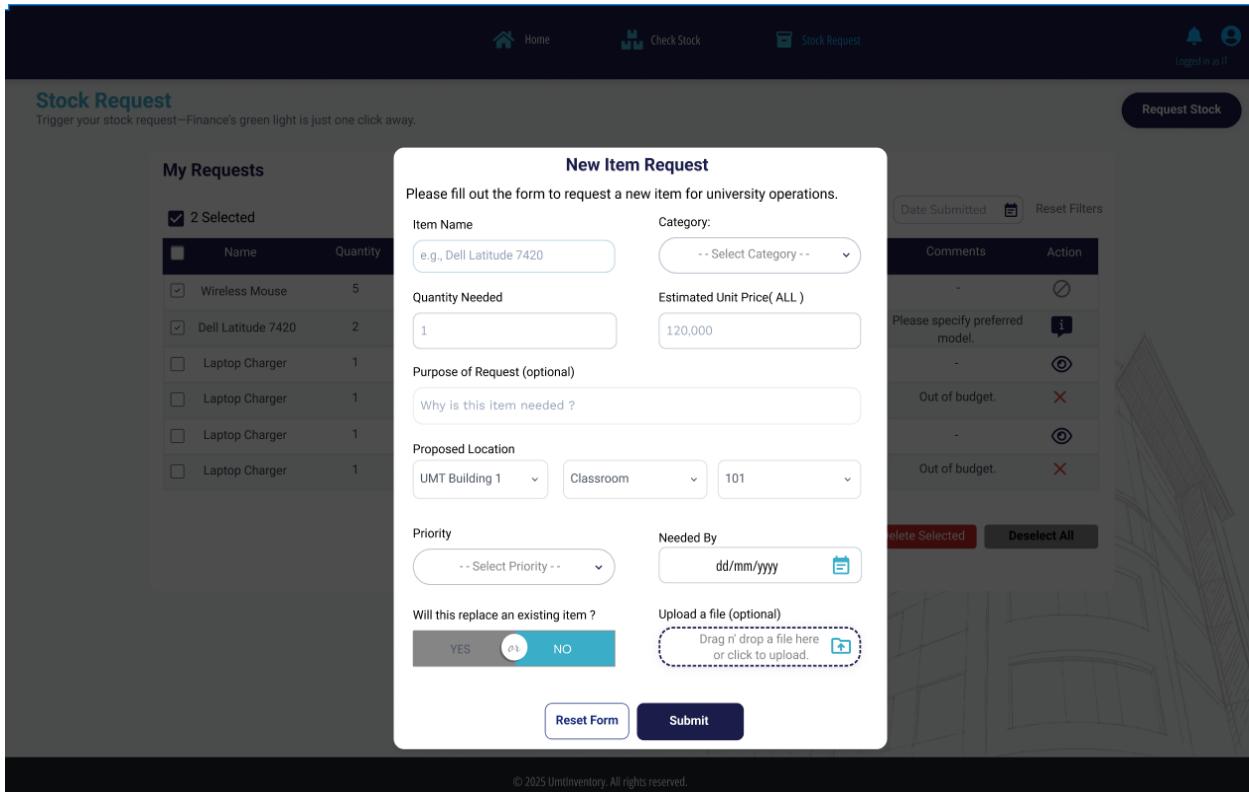
1-6 of 17 Requests

Additional Information Required

Mia Johnson Finance Department Please specify preferred model of laptop. 11:53 am You Dell XPS 13 (2023 edition) with Intel i7, 16 GB RAM 1:53 pm

Enter additional information...

© 2025 UmtInventory. All rights reserved.



Stock Request
Trigger your stock request—Finance's green light is just one click away.

New Item Request

Please fill out the form to request a new item for university operations.

Item Name <input type="text" value="e.g., Dell Latitude 7420"/>	Category: <input type="button" value="-- Select Category --"/>
Quantity Needed <input type="text" value="1"/>	Estimated Unit Price(ALL) <input type="text" value="120,000"/>
Purpose of Request (optional) <input type="text" value="Why is this item needed ?"/>	
Proposed Location <input type="button" value="UMT Building 1"/> <input type="button" value="Classroom"/> <input type="button" value="101"/>	Priority <input type="button" value="-- Select Priority --"/>
Needed By <input type="text" value="dd/mm/yyyy"/>	
Will this replace an existing item ? <input type="button" value="YES"/> <input type="button" value="NO"/>	
Upload a file (optional) <input type="text" value="Drag n' drop a file here or click to upload."/> <input type="button" value="Browse"/>	

© 2025 UmtInventory. All rights reserved.

Figure C.3.3: “IT Department –Stock Request Page”

C.4: Service Department – UI Mockups

Good Morning, Jonathan
Here to keep maintenance and QA in perfect rhythm !

Pending Requests Requests in Progress

Requested By	Department	Request Type	Item	Condition	Location	Priority	Date Submitted	Actions
John Smith	University Staff	Quality Check	Projector P001	Good	Building 1 Lab 101	Medium	2025-04-12	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
Lara Smith	Club Coordinator	Maintenance	Laptop LA001	Barely Working	Building 2 Room 101	High	2025-04-12	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
John Smith	IT	Hardware Fix	Network Switch S5	Not Working	Building 2 Lab 101	Low	2025-04-12	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
John Smith	Club Coordinator	Repair Request	Sound System S3	Good	Building 2 Room 101	Medium	2025-04-12	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
John Smith	University Staff	Replacement	Whiteboard	Not Working	Building 2 Room 101	High	2025-04-12	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
John Smith	Service	Preventive Maintenance	Air-Con Unit AC-7	Barely Working	Building 2 Room 101	Low	2025-04-12	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>

1-6 of 77 Requests

UMT Building 1

- Classrooms
- Labs
- Storerooms

UMT Building 2

- Classrooms
- Labs
- Storerooms

Metropolitan Innovation Center

- Classrooms
- Labs
- Storerooms

Logged in as
Service Department

Scheduled Tasks

April 2025						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

5 April

QC: P001 – Projector
Building 1, Room 101

Maintenance: LA001 – Laptop
Building 2, Room 101

QC: LA001 – Laptop
Building 1, Room 101

1-3 of 7

© 2025 UmtInventory. All rights reserved.

Good Morning, Jonathan
Here to keep maintenance and QA in perfect rhythm !

Pending Requests Requests in Progress

Requested By	Department	Request Type	Item ID	Item Name	Location	Condition	Actions
John Smith	University Staff	Quality Check	P001	Projector	Building 1 Lab 101	Good	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
Lara Smith	Club Coordinator	Maintenance	LA001	Laptop	Building 2 Room 101	Barely Working	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>
Lara Smith	University Staff	Replacement	Wb001	Whiteboard	Building 2 Room 102	Not Working	<input checked="" type="checkbox"/> <input type="button" value="Edit"/> <input checked="" type="checkbox"/>

Forward to Finance

1-6 of 77 Requests

UMT Building 1

- Classrooms
- Labs
- Storerooms

UMT Building 2

- Classrooms
- Labs
- Storerooms

Metropolitan Innovation Center

- Classrooms
- Labs
- Storerooms

Logged in as
Service Department

Repaired this month

April 2025						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

5 April

QC: P001 – Projector
Building 1, Room 101

Maintenance: LA001 – Laptop
Building 2, Room 101

QC: LA001 – Laptop
Building 1, Room 101

1-3 of 7

© 2025 UmtInventory. All rights reserved.

Figure C.4.1: “Service Department – HomePage”

 Check Stock
Home
Check Stock
Stock Request
Logged in as Service Department

Check Stock
Browse and update inventory across all university locations

UMT Building 1
UMT Building 1 > Classes > Classroom 101

- Class Classroom 101
- Class Classroom 102
- Class Classroom 103
- Class Classroom 104



UMT Building 2

- Class Lab 102
- Class Lab 103
- Class Lab 104
- Class Lab 105



Metropolitan Innovation Center

- Storage Storeroom 103
- Storage Storeroom 104
- Storage Storeroom 105
- Storage Storeroom 106



Search Items...
Select category
Select condition
Reset Filters

SKU	Name	Total Quantity	Good	Qty by Condition	Last Updated	Actions
WB-100	Whiteboard	8	4	2 Barely Working Not working	2023-10-05	Low Stock
Photo	Asset Tag	Name	Category	Condition	Location	Last Updated
<input checked="" type="checkbox"/>	PL-100-01	Whiteboard	Furniture	Good	UMT Building 1 Classroom 101	2023-10-05
<input checked="" type="checkbox"/>	PL-100-02	Whiteboard	Furniture	Barely Working	UMT Building 1 Classroom 101	2023-10-06
<input type="checkbox"/>	PL-100-03	Whiteboard	Furniture	Not	UMT Building 1 Classroom 101	2023-10-05
CH-300		Lecture Chair	20	15 5 0	12	Available
1-10 of 132 < >						
Edit Stock Mark as Scrap Deselect All						

© 2025 UmtInventory. All rights reserved.

 Check Stock
Home
Check Stock
Stock Request
Logged in as IT

Check Stock
Browse and update inventory across all university locations

UMT Building 1
UMT Building 1 > Classes > Classroom 101

- Class Classroom 101
- Class Classroom 102
- Class Classroom 103
- Class Classroom 104



UMT Building 2

Edit Item Details

ID: WB001

Name: Whiteboard

Condition: Not Working Consider replacing this item!

Location: Building 2 Room 102

Notes: Enter repair details ...

Upload a Photo (optional):

Drag n' drop or click here to choose a file from your computer

Forward to Finance

Save Changes



Metropolitan Innovation Center

- Storage Storeroom 103
- Storage Storeroom 104
- Storage Storeroom 105
- Storage Storeroom 106



Usage Status
Last Updated
Actions

Low Stock	2023-10-05	Edit Stock Mark as Scrap
Last Updated	Action	
UMT Building 1 Classroom 101	2023-10-05	Edit Stock
UMT Building 1 Classroom 101	2023-10-06	Edit Stock
UMT Building 1 Classroom 101	2023-10-05	Edit Stock Mark as Scrap
Available	2023-10-05	Edit Stock

1-10 of 132 < >

[Edit Stock](#) [Mark as Scrap](#) [Deselect All](#)

© 2025 UmtInventory. All rights reserved.

Figure C.4.2: “Service Department –Check Stock Page”

 Home
 Check Stock
 Stock Request
Logged in as
Service Department

Stock Request

Trigger your stock request—Finance's green light is just one click away.

My Requests

Name	Quantity	Proposed Location	Priority	Date	Status	Comments	Action
Whiteboard	5	UMT Building 1 Classroom 101	Low	2023-10-05	Pending	-	
Projector Mounts	2	UMT Building 1 Classroom 101	Medium	2023-10-06	Info Needed	Specify ceiling/wall preference	
Tool Kit Set	1	UMT Building 1 Classroom 101	High	2023-10-05	Approved	-	
Fire Extinguisher	1	UMT Building 1 Classroom 101	Low	2023-10-05	Declined	Out of budget.	
AC Control Unit	1	UMT Building 1 Classroom 101	High	2023-10-05	Approved	-	
LED Tube Lights	1	UMT Building 1 Classroom 101	Low	2023-10-05	Declined	Already purchased centrally	

23:53

1-6 of 17 Requests

Additional Information Required

Mia Johnson
Finance Department

Please specify wall preference.

Thursday
11:53 am

You

Wall-mounted only. Existing projector wiring supports side placement.
1:53 pm

Enter additional information...

© 2025 UmtInventory. All rights reserved.

Stock Request

Trigger your stock request—Finance's green light is just one click away.

New Item Request

Please fill out the form to request a new item for university operations.

Item Name <input type="text" value="e.g., Fire Extinguisher"/>	Category: <input type="button" value="-- Select Category --"/>
Quantity Needed <input type="text" value="1"/>	Estimated Unit Price(ALL) <input type="text" value="120.000"/>
Purpose of Request (optional) <input type="text" value="Why is this item needed ?"/>	
Proposed Location <input type="button" value="UMT Building 1"/> <input type="button" value="Classroom"/> <input type="button" value="101"/>	Needed By <input type="text" value="dd/mm/yyyy"/>
Priority <input type="button" value="-- Select Priority --"/>	Will this replace an existing item ? <input type="radio" value="YES"/> YES <input type="radio" value="NO"/> NO
Upload a file (optional) <input type="text" value="Drag n' drop a file here or click to upload."/> <input type="button" value="Reset Form"/> <input type="button" value="Submit"/>	

Additional Information Required

Mia Johnson
Finance Department

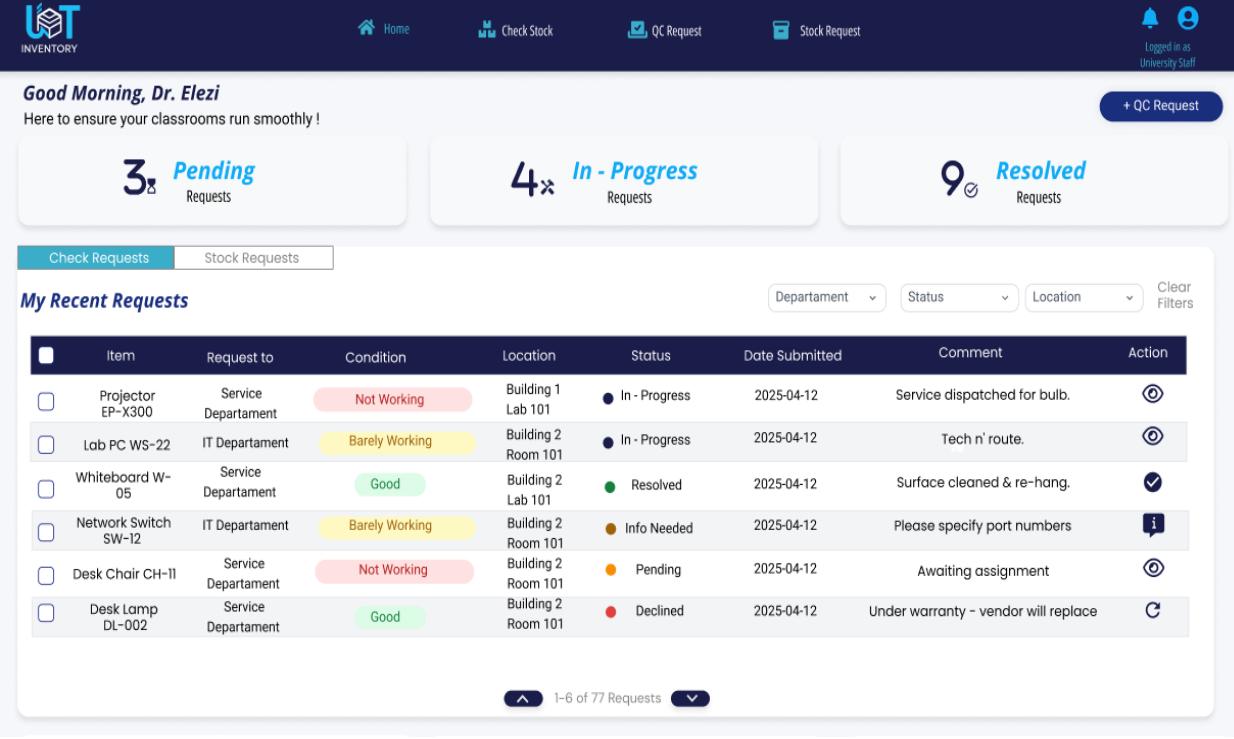
Please specify wall preference.

Enter additional information...

© 2025 UmtInventory. All rights reserved.

Figure C.4.3: “Service Department –Stock Request Page”

C.5: University Staff – UI Mockups

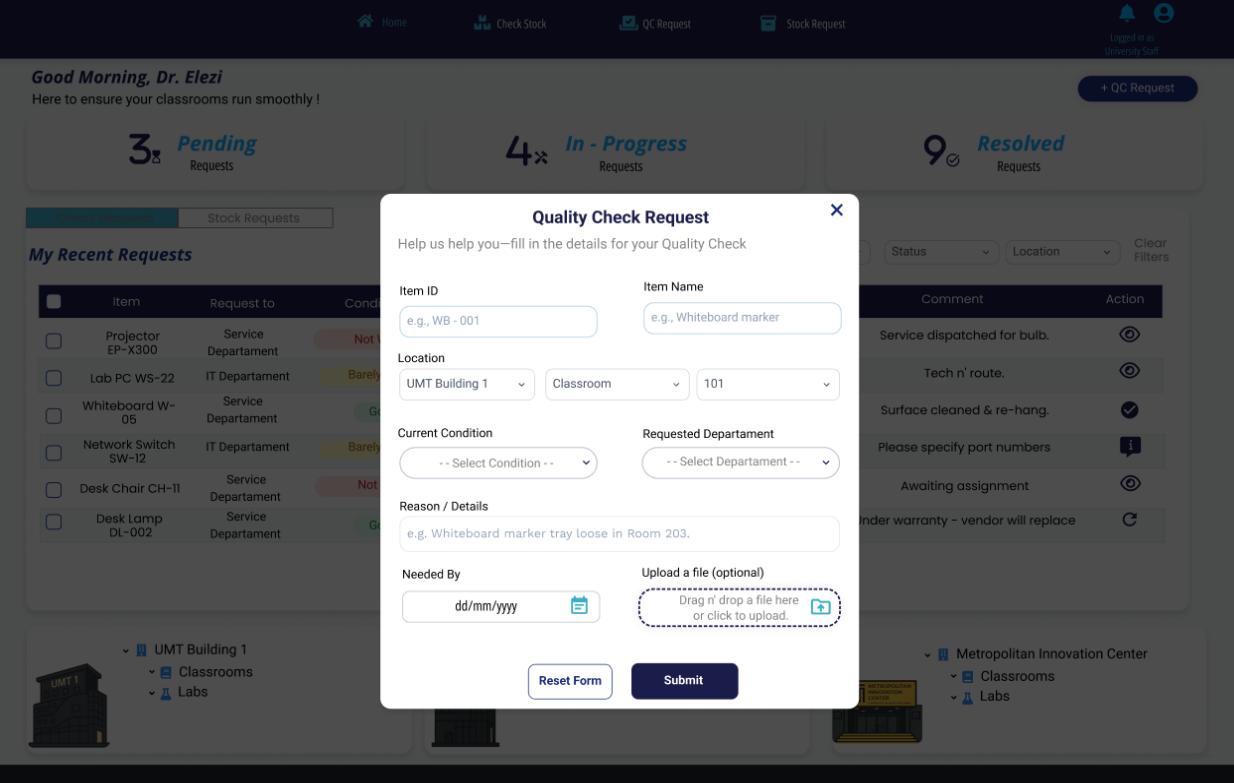


Good Morning, Dr. Elezi
Here to ensure your classrooms run smoothly!

3 Pending Requests

4 In - Progress Requests

9 Resolved Requests



Quality Check Request

Help us help you—fill in the details for your Quality Check

Item ID	Item Name
e.g., WB - 001	e.g., Whiteboard marker

Location

UMT Building 1 Classroom 101

Current Condition

Requested Department

Reason / Details

e.g. Whiteboard marker tray loose in Room 203.

Needed By

dd/mm/yyyy

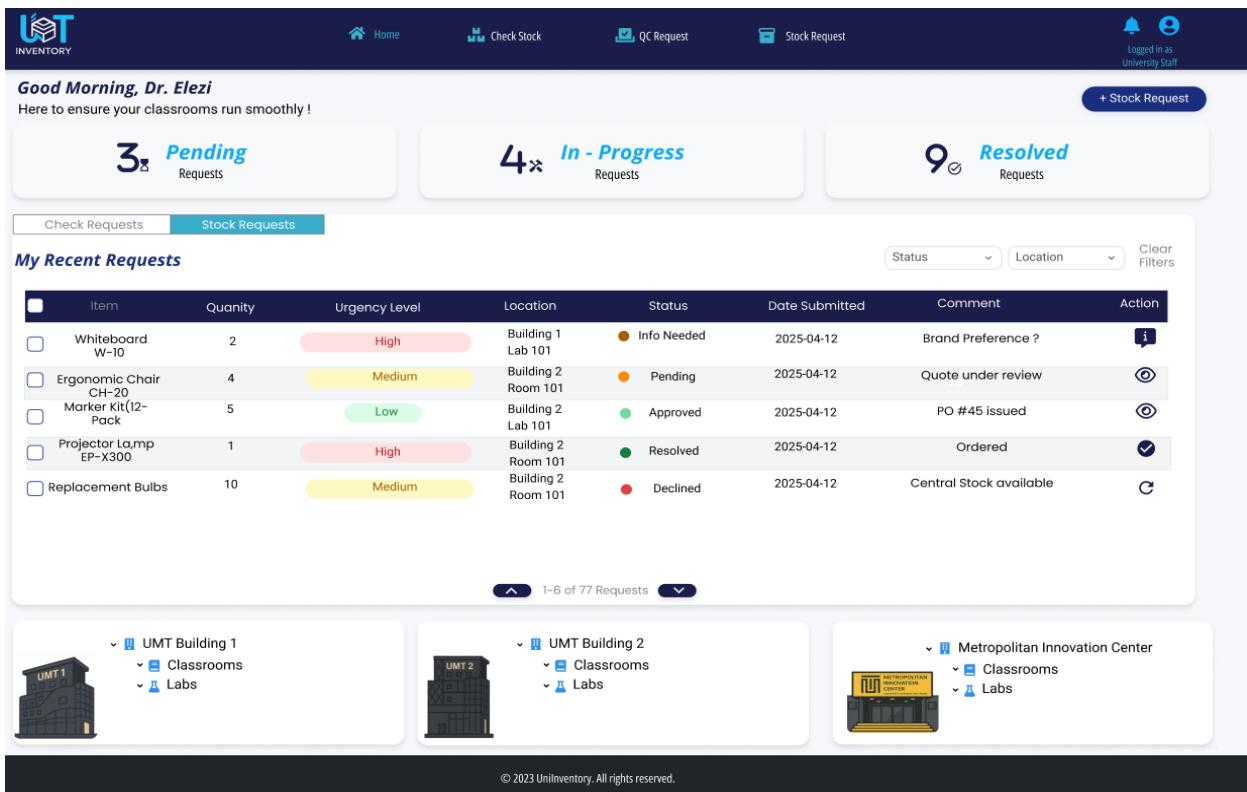
Upload a file (optional)

Drag n' drop a file here or click to upload

Reset Form **Submit**

© 2023 Unlinventory. All rights reserved.

Figure C.5.1: “University Staff – QC Request Modal on Homepage”



Good Morning, Dr. Elezi
Here to ensure your classrooms run smoothly !

3 Pending Requests **4 In - Progress Requests** **9 Resolved Requests**

Check Requests **Stock Requests**

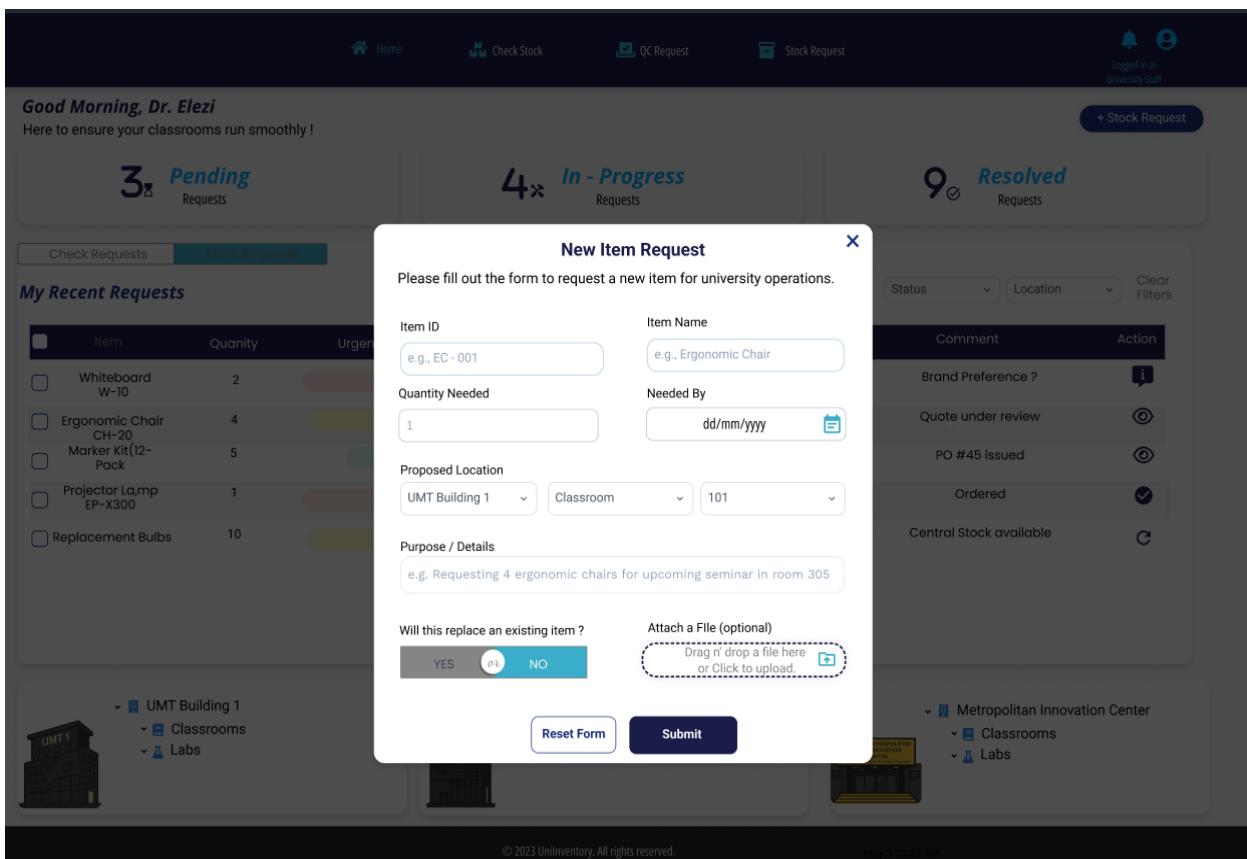
My Recent Requests

Item	Quantity	Urgency Level	Location	Status	Date Submitted	Comment	Action
Whiteboard W-10	2	High	Building 1 Lab 101	Info Needed	2025-04-12	Brand Preference ?	
Ergonomic Chair CH-20	4	Medium	Building 2 Room 101	Pending	2025-04-12	Quote under review	
Marker Kit(12-Pack)	5	Low	Building 2 Lab 101	Approved	2025-04-12	PO #45 issued	
Projector Lamp EP-X300	1	High	Building 2 Room 101	Resolved	2025-04-12	Ordered	
Replacement Bulbs	10	Medium	Building 2 Room 101	Declined	2025-04-12	Central Stock available	

1–6 of 77 Requests

UMT Building 1: Classrooms, Labs
 UMT Building 2: Classrooms, Labs
 Metropolitan Innovation Center: Classrooms, Labs

© 2023 UniInventory. All rights reserved.



Good Morning, Dr. Elezi
Here to ensure your classrooms run smoothly !

3 Pending Requests **4 In - Progress Requests** **9 Resolved Requests**

Check Requests **Stock Requests**

My Recent Requests

Item	Quantity	Urgency Level
Whiteboard W-10	2	High
Ergonomic Chair CH-20	4	Medium
Marker Kit(12-Pack)	5	Low
Projector Lamp EP-X300	1	High
Replacement Bulbs	10	Medium

New Item Request

Please fill out the form to request a new item for university operations.

Item ID: e.g., EC - 001 **Item Name**: e.g., Ergonomic Chair

Quantity Needed: 1 **Needed By**: dd/mm/yyyy

Proposed Location: UMT Building 1, Classroom, 101

Purpose / Details: e.g. Requesting 4 ergonomic chairs for upcoming seminar in room 305

Will this replace an existing item ?: YES NO

Attach a File (optional): Drag n' drop a file here or Click to upload.

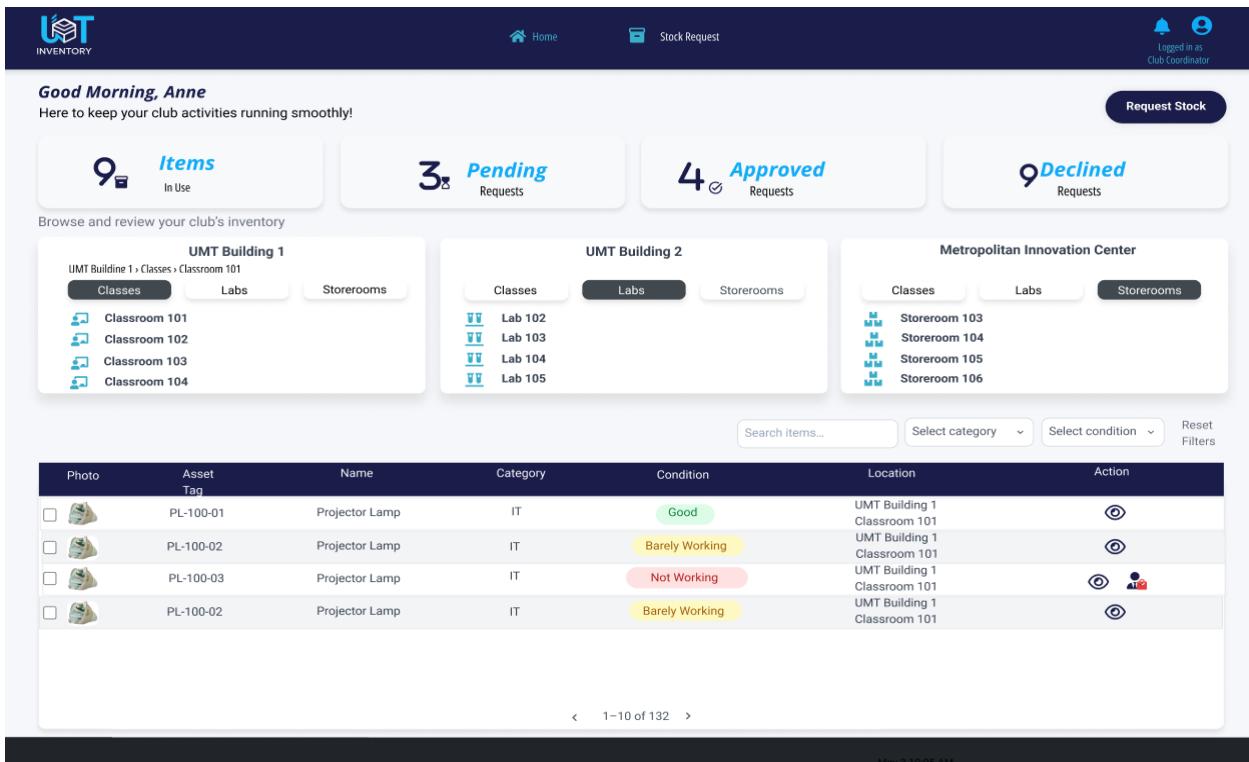
Reset Form **Submit**

UMT Building 1: Classrooms, Labs
 UMT Building 2: Classrooms, Labs
 Metropolitan Innovation Center: Classrooms, Labs

© 2023 UniInventory. All rights reserved.

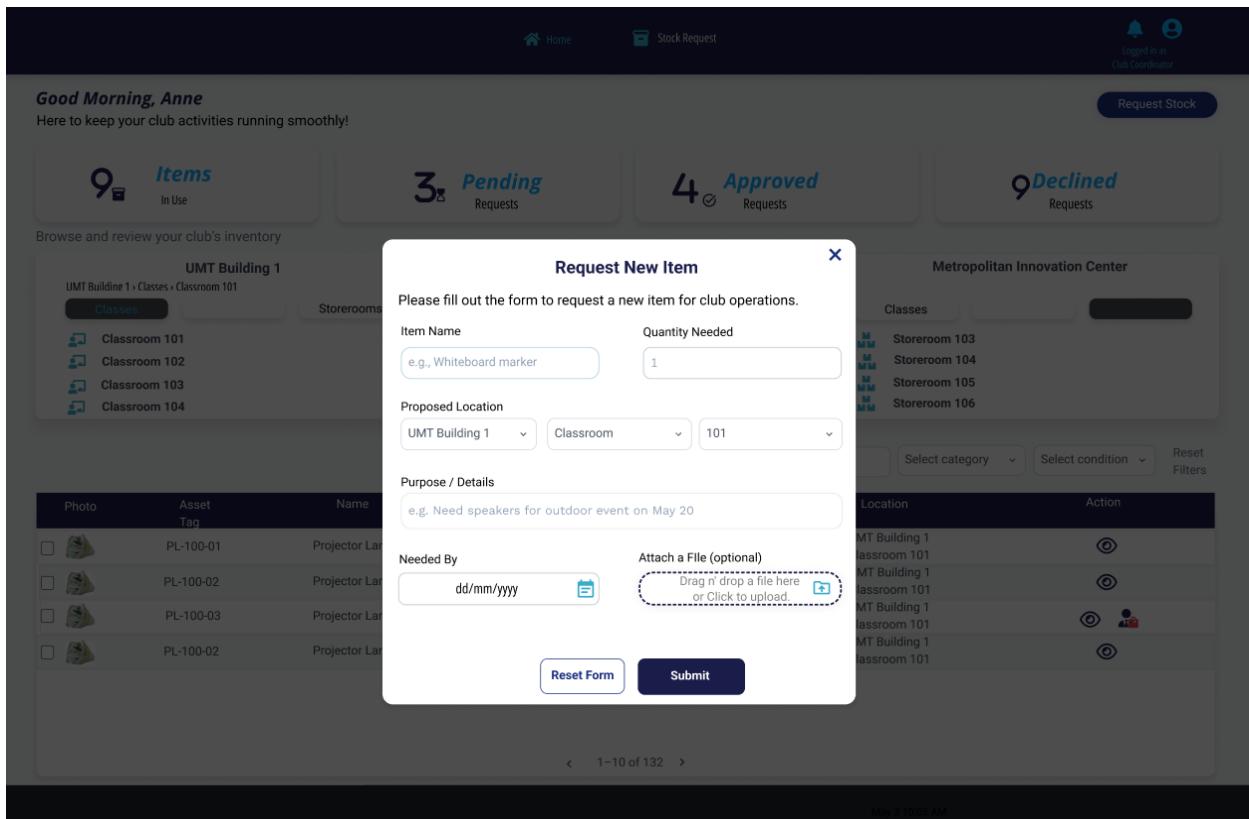
Figure C.5.2: “University Staff –New Item Request Modal on Homepage”

C.6: Club Coordinator– UI Mockups



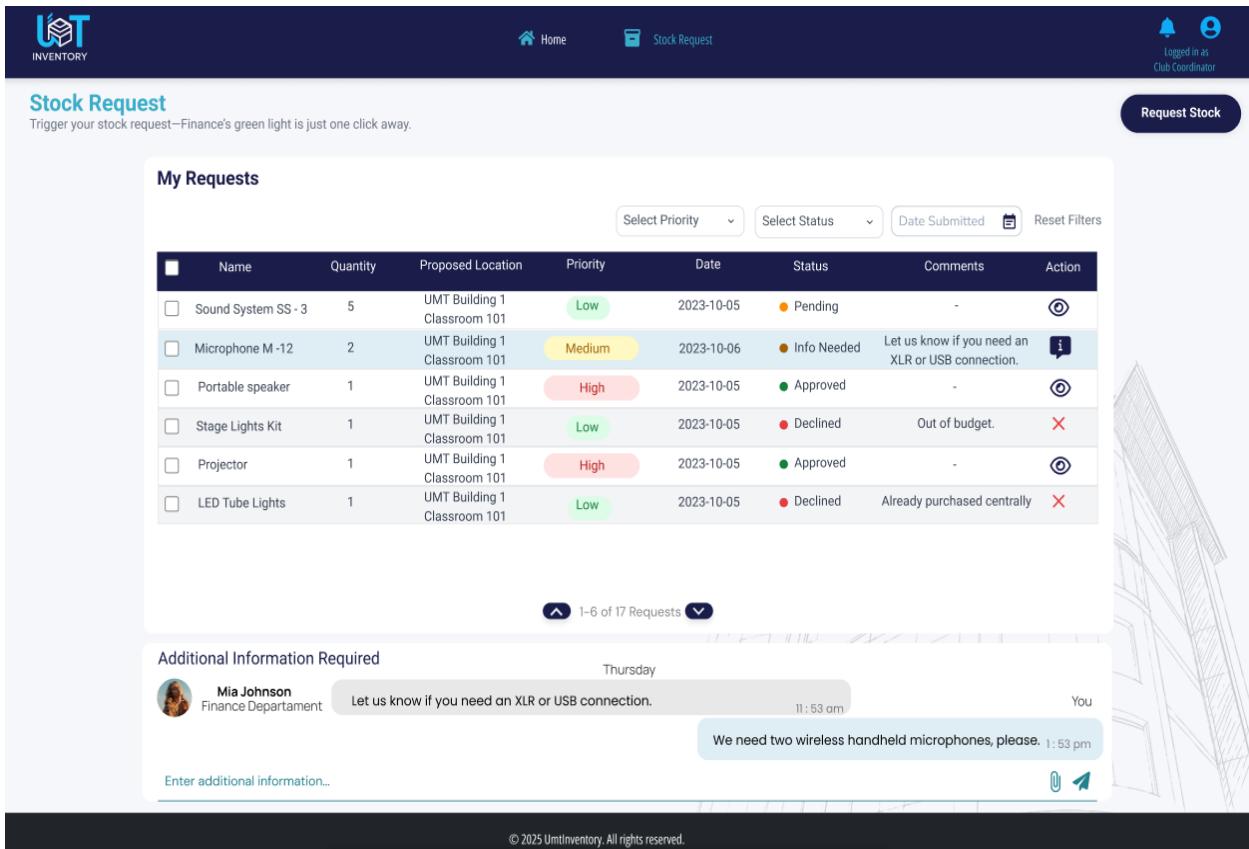
The screenshot shows the Club Coordinator homepage. At the top, there are four summary cards: "Items In Use" (9), "Pending Requests" (3), "Approved Requests" (4), and "Declined Requests" (9). Below these are three sections for "UMT Building 1", "UMT Building 2", and "Metropolitan Innovation Center", each listing asset categories like Classes, Labs, and Storerooms. A main table lists assets with columns for Photo, Asset Tag, Name, Category, Condition, Location, and Action. A modal window titled "Request New Item" is open in the foreground, prompting the user to fill out fields for Item Name, Quantity Needed, Proposed Location, Purpose / Details, and Needed By.

Photo	Asset Tag	Name	Category	Condition	Location	Action
	PL-100-01	Projector Lamp	IT	Good	UMT Building 1 Classroom 101	
	PL-100-02	Projector Lamp	IT	Barely Working	UMT Building 1 Classroom 101	
	PL-100-03	Projector Lamp	IT	Not Working	UMT Building 1 Classroom 101	
	PL-100-02	Projector Lamp	IT	Barely Working	UMT Building 1 Classroom 101	



This screenshot shows the same homepage as above, but with a modal window in the center. The modal is titled "Request New Item" and contains fields for "Item Name" (e.g., Whiteboard marker), "Quantity Needed" (1), "Proposed Location" (UMT Building 1, Classroom, 101), "Purpose / Details" (e.g., Need speakers for outdoor event on May 20), "Needed By" (dd/mm/yyyy), and an "Attach a File (optional)" section with a file upload button. The background table and other interface elements are visible but dimmed.

Figure C.6.1: “Club Coordinator–Homepage”



Stock Request
Trigger your stock request—Finance's green light is just one click away.

My Requests

Select Priority ▾ Select Status ▾ Date Submitted ▾ Reset Filters

Name	Quantity	Proposed Location	Priority	Date	Status	Comments	Action
Sound System SS - 3	5	UMT Building 1 Classroom 101	Low	2023-10-05	Pending	-	🕒
Microphone M-12	2	UMT Building 1 Classroom 101	Medium	2023-10-06	Info Needed	Let us know if you need an XLR or USB connection.	ℹ️
Portable speaker	1	UMT Building 1 Classroom 101	High	2023-10-05	Approved	-	🕒
Stage Lights Kit	1	UMT Building 1 Classroom 101	Low	2023-10-05	Declined	Out of budget.	✗
Projector	1	UMT Building 1 Classroom 101	High	2023-10-05	Approved	-	🕒
LED Tube Lights	1	UMT Building 1 Classroom 101	Low	2023-10-05	Declined	Already purchased centrally	✗

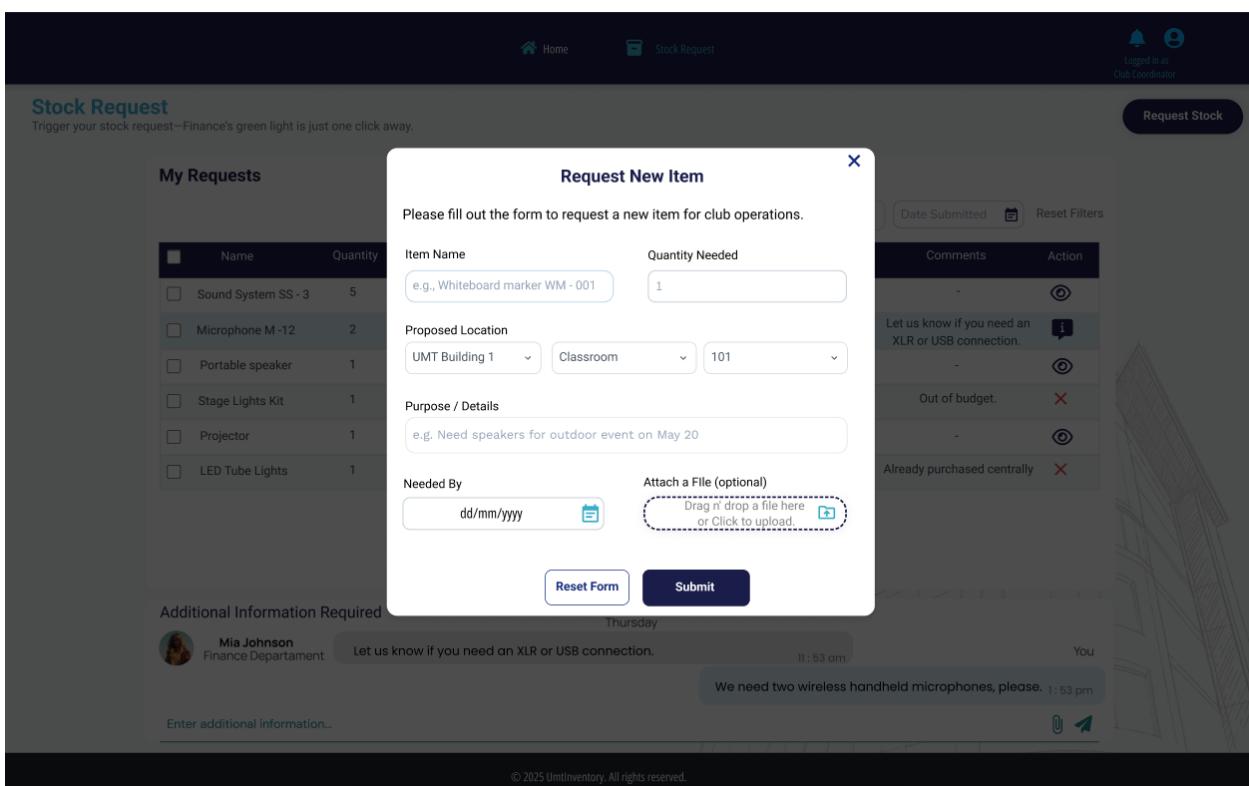
1-6 of 17 Requests

Additional Information Required

Mia Johnson Finance Department Let us know if you need an XLR or USB connection. 11:53 am You We need two wireless handheld microphones, please. 11:53 pm

Enter additional information...

© 2025 Umtnventory. All rights reserved.



Stock Request
Trigger your stock request—Finance's green light is just one click away.

My Requests

Request New Item

Please fill out the form to request a new item for club operations.

Item Name <input type="text" value="e.g., Whiteboard marker WM - 001"/>	Quantity Needed <input type="text" value="1"/>
Proposed Location <input type="text" value="UMT Building 1"/> Classroom <input type="text" value="101"/>	
Purpose / Details <input type="text" value="e.g. Need speakers for outdoor event on May 20"/>	
Needed By <input type="text" value="dd/mm/yyyy"/>	Attach a File (optional) <input type="text" value="Drag n drop a file here or Click to upload"/>

Reset Form Submit

Additional Information Required

Mia Johnson Finance Department Let us know if you need an XLR or USB connection. 11:53 am You We need two wireless handheld microphones, please. 11:53 pm

Enter additional information...

© 2025 Umtnventory. All rights reserved.

Figure C.6.1: “Club Coordinator— Stock Request Page”