

Software Design Document

for

Department Stock Management System

Prepared by

Adwaith Jayan

Anand Sankar P T

Arjun Sabu

Niranjan M Varma

Department of Computer Science and Engineering

Rajiv Gandhi Institute of Technology,Kottayam

Contents

1 System Design	3
1.1 Module 1: Front End	3
1.2 Module 2: Back End	3
1.2.1 Module 2.1: Connector	4
1.2.2 Module 2.2: Database	4
2 Detailed Design	5
2.1 Module 1: Front End	5
2.2 Module 2: Back End	22
2.2.1 Module 2.1: Connector	22
2.2.2 Module 2.2 : Database	30

Design

The design phase in software development is a critical step that follows the requirements gathering and analysis phase. It involves transforming the high level requirements into a detailed blueprint for the actual implementation of the software system. The primary goal of the design phase is to create a comprehensive plan that guides developers in building software product that meets the specified requirements and is efficient, maintainable, and scalable. The design phase serves as a bridge between the requirements phase and the implementation phase. A well-executed design phase lays the groundwork for efficient development, reduces the likelihood of misunderstandings, and contributes to the successful delivery of a high-quality software product. The design activity is divided into two separate phases namely :

- System Design
- Detailed Design

1 System Design

The System Design Document for the Department Stock Management System provides a comprehensive overview of the technical and architectural aspects of the project. It outlines the system's core components, interactions, and the design decisions made to meet functional and non-functional requirements. The system is built to facilitate efficient inventory management, allowing users such as the Principal, HODs, Faculty-in-Charge, and Custodians to manage, transfer, verify, and maintain stock in a streamlined manner. This document serves as a blueprint for the development team, ensuring clarity in the implementation of features such as stock allocation, stock transfer, and report generation. It includes diagrams, workflows, and data models that illustrate the system's structure and behavior. Additionally, the document highlights the tools, technologies, and methodologies used to create a robust, scalable, and user-friendly application.

The system design is broken down into various modules, each of which addresses a different segment of the system. They are

- Module 1: Front End
- Module 2: Back End
 - Module 2.1: Connector
 - Module 2.2: Database

1.1 Module 1: Front End

The front end of the Department Stock Management System serves as the user facing interface that enables users to efficiently manage stock details. It provides a user-friendly environment for interacting with the system, including accessing stock information, adding new items, updating quantities, and generating reports. The front end includes visual elements such as buttons, forms, and menus, along with the underlying logic that ensures a smooth user experience, like dynamic page updates and real-time data display. By facilitating intuitive interactions and seamless communication with the backend, the front end plays a crucial role in helping users manage stock effectively and accurately.

1.2 Module 2: Back End

The backend of the Department Stock Management System handles the behind-the-scenes processes that support the front end. It is responsible for managing the business logic, database interactions, and data processing. The backend ensures the efficient storage, retrieval, and updating of stock information, handling tasks such as adding new stock items, updating quantities, and generating reports. It communicates with the front end through APIs to provide real-time data and ensure smooth interactions. The backend also enforces security, data validation, and ensures the integrity and consistency of the stock data, playing a vital role in the overall functionality and performance of the system.

1.2.1 Module 2.1: Connector

The connector in the Department Stock Management System acts as the bridge between the front end and the backend, enabling seamless communication between the two layers. It facilitates data exchange by handling requests from the front end, forwarding them to the backend, and then delivering the appropriate responses back to the user interface. The connector ensures that the data flow is smooth and efficient, allowing users to interact with the system in real time. By using APIs or other communication protocols, the connector ensures that stock information is accurately transmitted and that the system operates with minimal delay, playing a key role in maintaining the overall functionality and responsiveness of the system.

1.2.2 Module 2.2: Database

The database in the Department Stock Management System serves as the central repository for all stock-related data, ensuring efficient storage, retrieval, and management of information. It stores critical details such as stock items, quantities, prices, and transaction history, allowing the system to maintain data consistency and integrity. The database works closely with the backend to process and update stock information, ensuring that changes made through the front end are accurately reflected. By using efficient query processing and data indexing, the database ensures fast access to stock data, enabling real-time updates and smooth system performance.

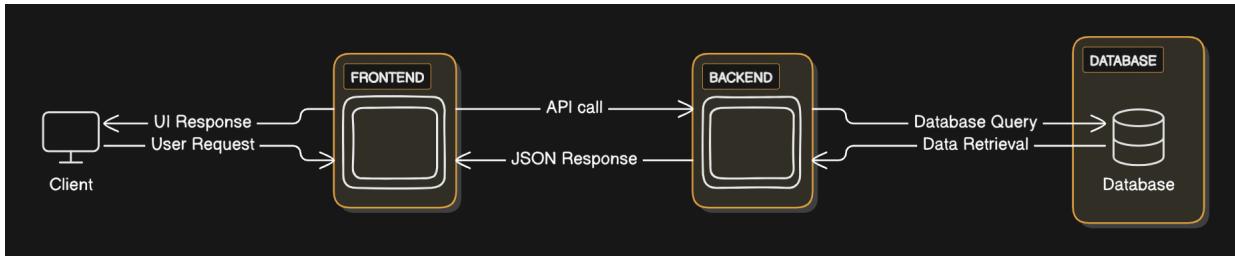


Figure 1: System Overview

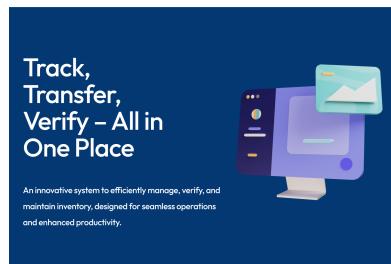
2 Detailed Design

The Detailed Design section of a software design document provides a comprehensive, in-depth explanation of the system's architecture, components, and implementation specifics. It breaks the system into modules detailing their functionality, interactions, data structures, and algorithms. It includes detailed descriptions of each module, comprising interface diagrams, state chart diagrams, and algorithms for each functionality.

2.1 Module 1: Front End

The detailed design of the front end includes designing the user interface, defining the structure and behavior of UI components, and creating interface diagrams to illustrate interactions and data flow between components.

UI Design



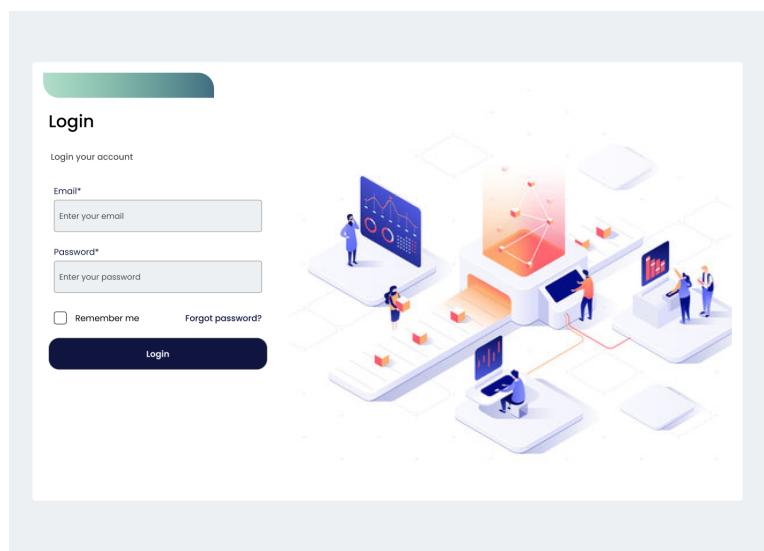
STOCK MANAGEMENT SYSTEM

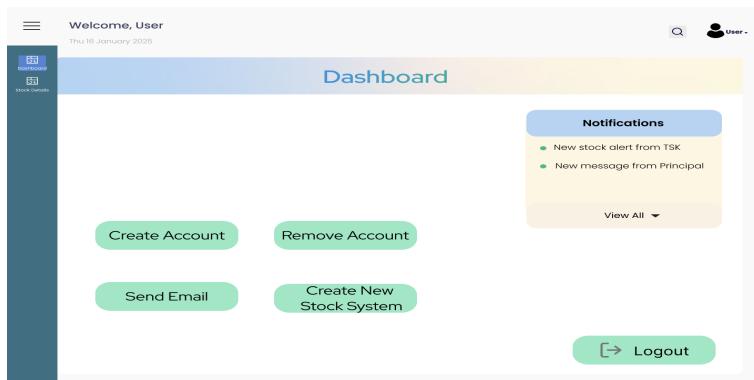
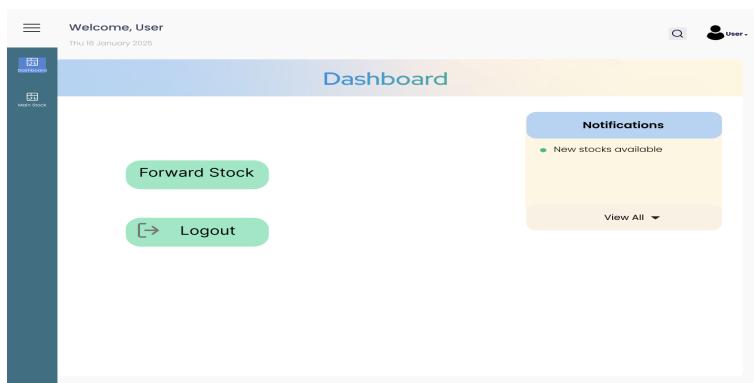
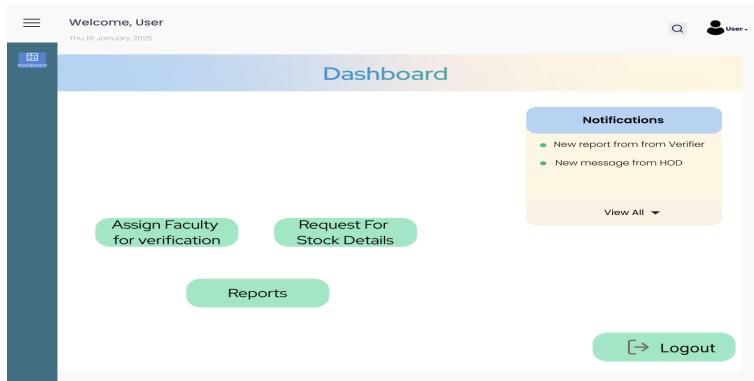
Effortlessly manage and maintain inventory with a streamlined, user-friendly system designed for accuracy and efficiency.

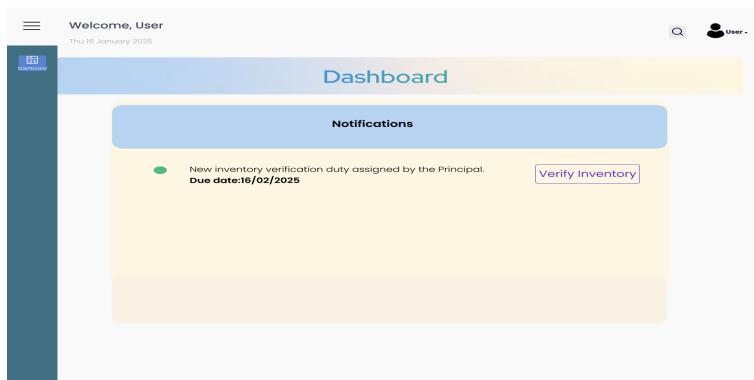
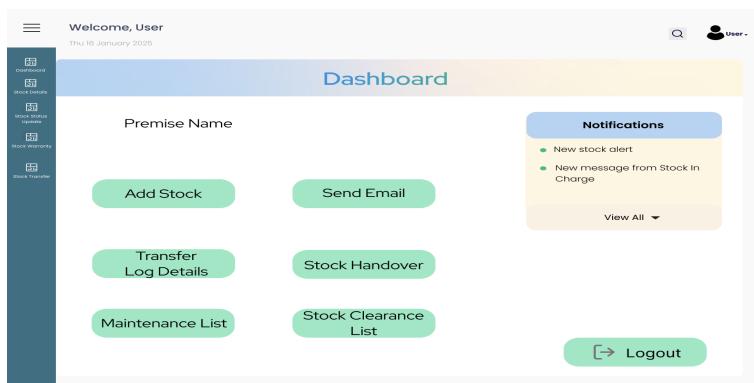
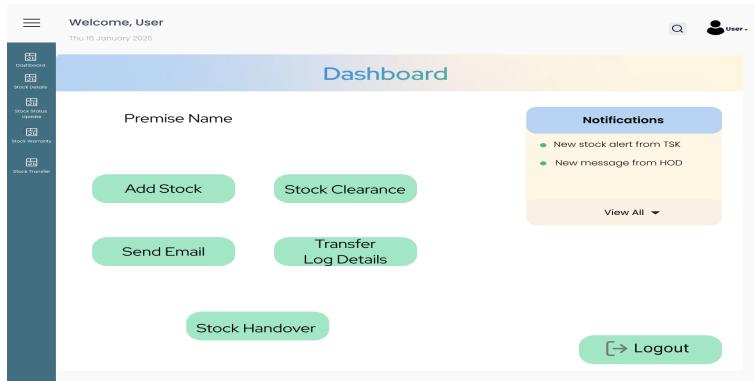


[GET STARTED](#)

[Login](#)







Welcome, User

Thu 16 January 2025

 Dashboard

Assign Faculty

Enter name of faculty

Enter Email of faculty

Enter department

Enter premise

Enter last date 

Assign

 Dashboard

Report

Item ID	Date of Invoice	Item Name	Description	Remarks	Status
#7676	30/06/2024	CPU	Intel i5 12th gen		 Working
#7677	30/06/2024	CPU	Intel i5 12th gen	CPU not functioning; requires repair	 Not Working
#7678	28/08/2024	Monitor	Monitor DELL		 Working
#7679	28/08/2024	Monitor	Monitor DELL		 Working
#7680	28/08/2024	Monitor	Monitor DELL		 Working

Approve ✓



User

	Dashboard
	Stock Details
	Stock Status Update
	Stock Warranty
	Stock Transfer

Request Stock Report

+ Request

Department	Premise
<input type="button" value="Department"/>	<input type="button" value="Premise"/>



Welcome, User

Thu 16 January 2025



User

--

Reports

Report Submitted

Department:

Premise:

By:

Date:



Forward Items

➡️ Forward

Item Name

Item Name

Item Quantity

+ Quantity -

Enter Department

Enter Department ▾

Enter Date of Purchase

Date of purchase



Register

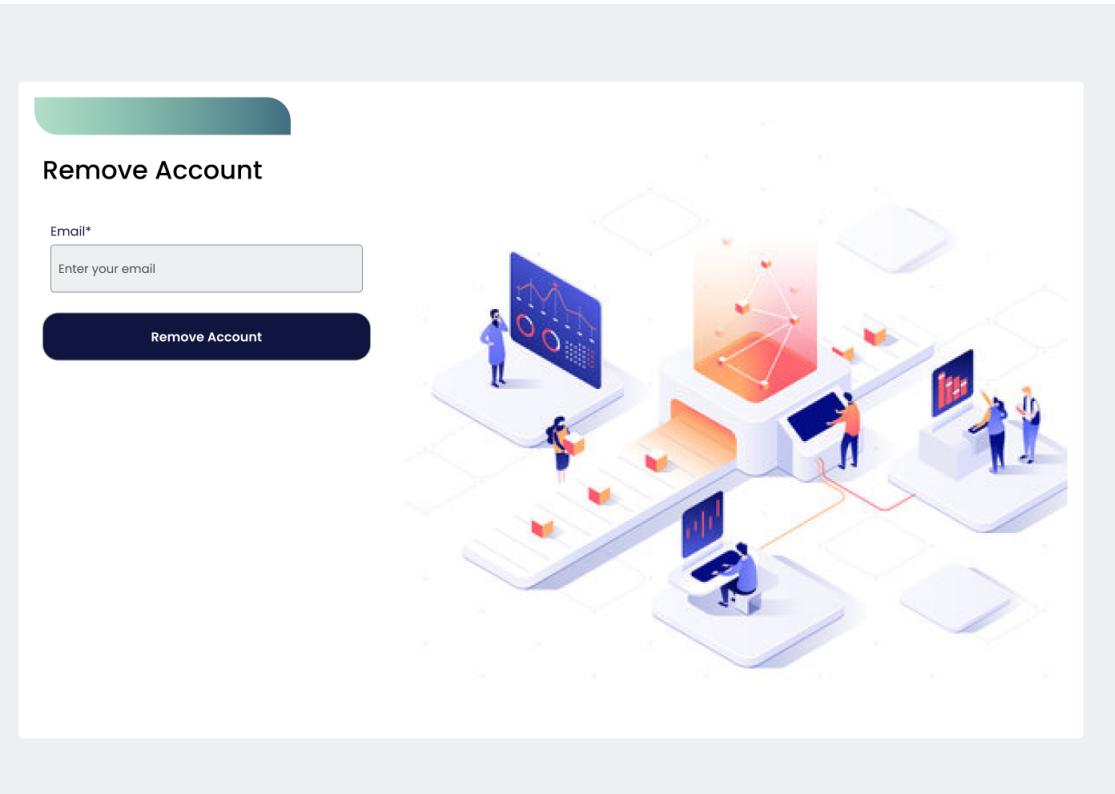
Manage all your inventory efficiently
Get started by creating your account and setting up your work profile to access all features seamlessly.

First name Role

Email Enter Inventory

Password

Create Account



A screenshot of a dashboard titled "Notifications". The sidebar on the left includes icons for "Dashboard" (selected), "Stock Details", and "Stock Items". The main area shows two notifications: 1) "New stock items from TSK." from user "A" (timestamp 15h ago), with buttons for "Forward" and "Decline", and a "View All" dropdown. 2) "New Assigned Faculty List from Principal" from user "A" (timestamp 1 day ago), with a "View Details" button. At the top right, there are search, notification, and user profile icons, along with a "Mark all as read" button.



Assigned Faculties

Search Item ID

Filter

<input type="checkbox"/>	Name	Premise	Email	Date of Assignment	Last Date to verify
<input type="checkbox"/>	Faculty1	DBMS LAB	faculty1@mail.com	30/01/2025	30/03/2025
<input type="checkbox"/>	Faculty2	OS LAB	faculty2@mail.com	19/01/2025	19/02/2025
<input type="checkbox"/>	Faculty3	Networking LAB	faculty3@mail.com	15/02/2025	15/03/2025



Welcome, User

Thu 16 January 2025

New Stock System

Enter name of Premise

Enter Room no

Enter Type

Create



- Dashboard
- Stock Details
- Stock Status Update
- Stock Warranty
- Stock Transfer

Stocks

[Export](#)[+ New Items](#) Search Item ID

Status ▾

Filter ▾

Item ID	Date of Invoice	Date of Indent	Item Name	Description	Price	Status
#7676	30/06/2024	01/07/2024	CPU	Intel i5 12th gen	20000	Working
#7677	30/06/2024	01/07/2024	CPU	Intel i5 12th gen	21000	Not Working
#7678	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Not Working
#7679	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Working
#7680	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Working



- Dashboard
- Stock Details
- Stock Status Update
- Stock Transfer

Stocks Warranty

[Export](#) Search item ID

Status ▾

Filter ▾

Item ID	Date of Invoice	Warranty Period	Item Name	Description	Warranty Status	Status
#7676	30/06/2024	30/06/2026	CPU	Intel i5 12th gen	In Warranty	Working
#7677	30/06/2024	30/06/2026	CPU	Intel i5 12th gen	In Warranty	Not Working
#7678	28/08/2024	28/08/2026	Monitor	Monitor DELL	In Warranty	Not Working
#7679	28/08/2024	28/08/2026	Monitor	Monitor DELL	In Warranty	Working
#7680	28/08/2024	28/08/2026	Monitor	Monitor DELL	In Warranty	Working



- Dashboard
- Stock Details
- Stock Status Update
- Stock Transfer

Stock Transfer

Transfer Stocks

Search Item ID

Filter ▾

<input type="checkbox"/>	Item ID	Date of Invoice	Date of Indent	Item Name	Description	Price	Transfer To
<input type="checkbox"/>	#7676	30/06/2024	01/07/2024	CPU	Intel i5 12th gen	20000	Premise ▾
<input type="checkbox"/>	#7677	30/06/2024	01/07/2024	CPU	Intel i5 12th gen	21000	Premise ▾
<input type="checkbox"/>	#7678	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Premise ▾
<input type="checkbox"/>	#7679	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Premise ▾
<input type="checkbox"/>	#7680	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Premise ▾



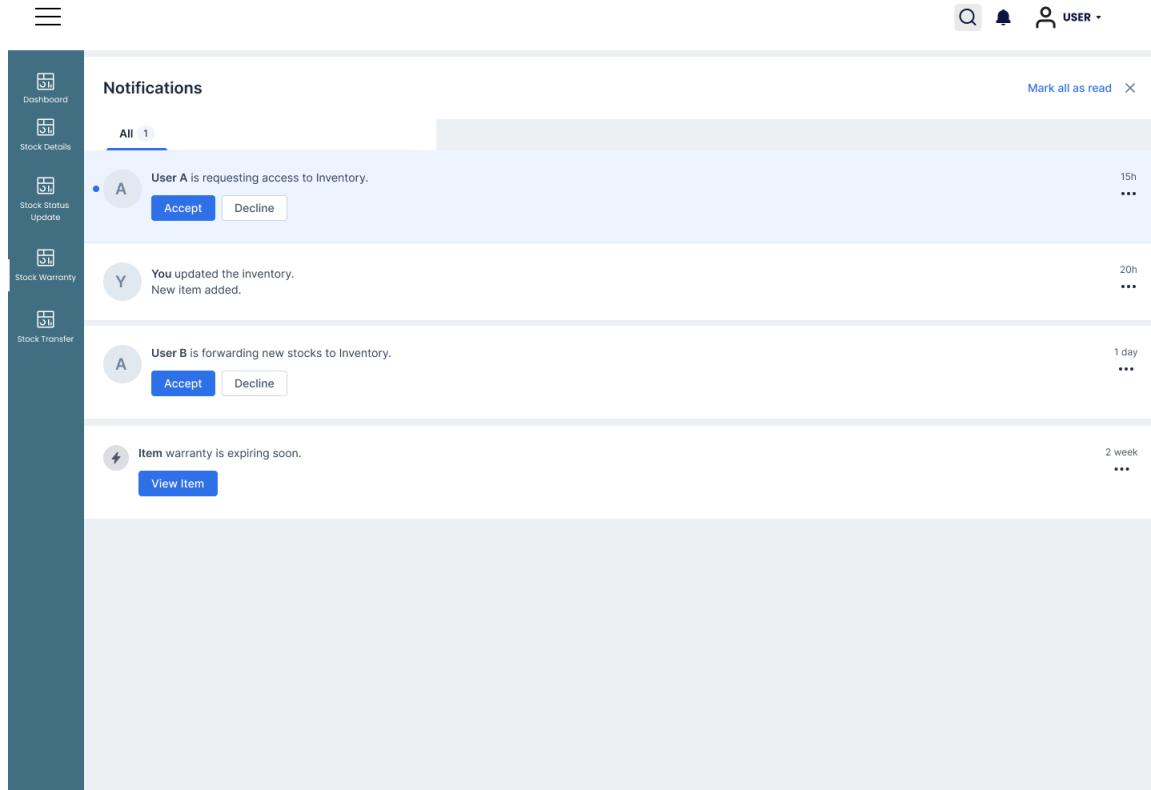
- Dashboard
- Stock Details
- Stock Status Update
- Stock Warranty
- Stock Transfer

Stock Clearance

[Clear Stocks](#) Search Item ID

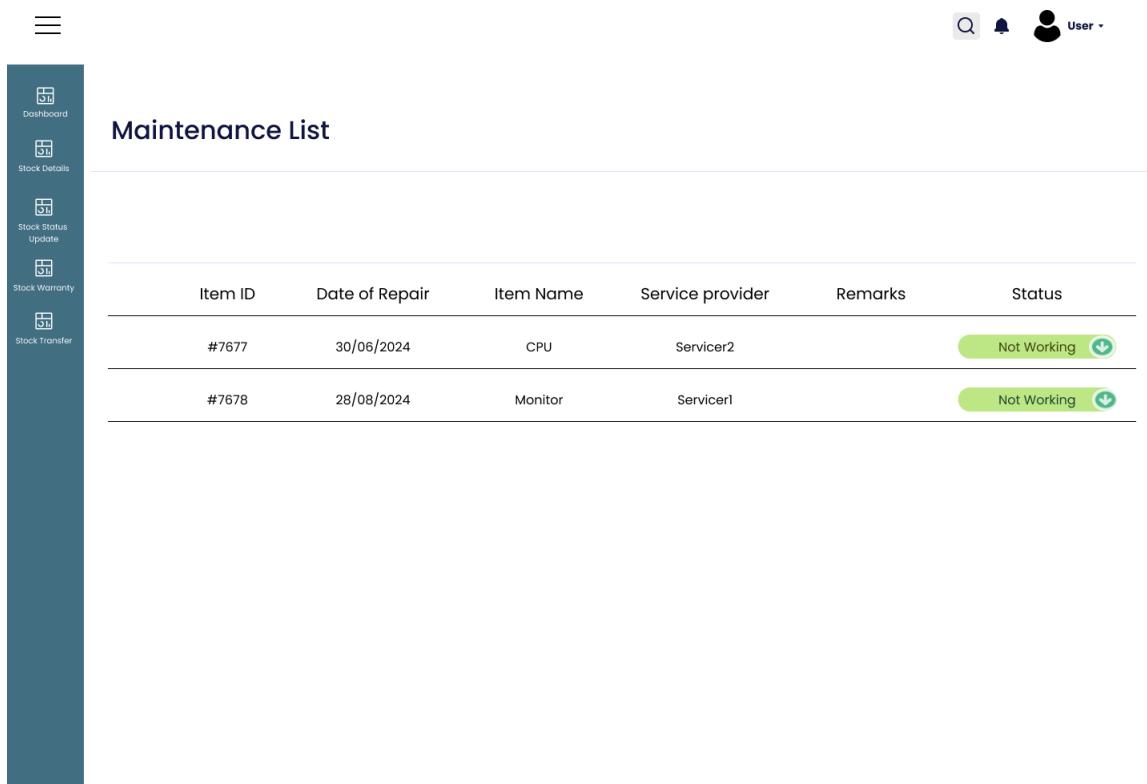
Filter ▾

Item ID	Date of clearance	Warranty Period	Item Name	Description	Status
#7677	03/01/2025	30/06/2023	CPU	i5 12th gen	Not Repairable
#7678	18/01/2025	28/08/2023	Monitor	Monitor DELL	Not Repairable



The Notifications page displays a list of alerts. At the top right are search, notification, and user icons. A 'Mark all as read' button is located in the top right corner. The notifications are listed in descending order of time:

- User A is requesting access to Inventory. (15h ago) - Actions: Accept, Decline
- You updated the inventory. New item added. (20h ago) - Actions: ...
- User B is forwarding new stocks to Inventory. (1 day ago) - Actions: Accept, Decline
- Item warranty is expiring soon. (2 week ago) - Actions: View Item, ...



The Maintenance List page shows a table of items requiring attention. At the top right are search, notification, and user icons. The title 'Maintenance List' is centered above the table.

Item ID	Date of Repair	Item Name	Service provider	Remarks	Status
#7677	30/06/2024	CPU	Servicer2	Not Working	
#7678	28/08/2024	Monitor	Servicer1	Not Working	



- Dashboard
- Stock Details
- Stock Status Update
- Stock Warranty
- Stock Transfer

Log Details

Export

Search Item Name

Filter ▾

Item Name	Quantity	Date of Transfer	Source Premise Name	Destination Premise Name	Status
CPU	4	01/07/2024	Premise1	Premise3	Accepted
CPU	2	02/07/2024	Premise1	Premise3	Not Accepted
Monitor	1	30/08/2024	Premise2	Premise3	Accepted
Monitor	4	03/09/2024	Premise4	Premise3	Accepted
Monitor	2	24/10/2024	Premise1	Premise3	Accepted

New Message

From John Doe <supercoolman@gmail.com>

To John Doe Jane Doe

Cc Bcc

Subject

Send





- Dashboard
- Stock Details
- Stock Status Update
- Stock Warranty
- Stock Transfer

Add Items

+ Add Item

Enter Item Name

Item Name

Enter Item ID

Item ID

Enter Intend Number

Intend Number

Enter Date of Purchase

Date of purchase of intend

Enter Date of Intend

Date of intend

Enter Item Description

Description

Enter Warranty Period

Warranty in years

Enter Room no

Room no



Stocks Status

Export

Search Item ID



Status ▾

Filter ▾

Item ID	Date of Invoice	Date of Indent	Item Name	Description	Price	Status
#7676	30/06/2024	01/07/2024	CPU	Intel i5 12th gen	20000	Working
#7677	30/06/2024	01/07/2024	CPU	Intel i5 12th gen	21000	Not Working
#7678	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Not Working
#7679	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Working
#7680	28/08/2024	30/08/2024	Monitor	Monitor DELL	4000	Working



Stock Verification

Item ID	Date of Invoice	Item Name	Description	Remarks	Status
#7676	30/06/2024	CPU	Intel i5 12th gen		Working
#7677	30/06/2024	CPU	Intel i5 12th gen	CPU not functioning; requires repair	Not Working
#7678	28/08/2024	Monitor	Monitor DELL		Working
#7679	28/08/2024	Monitor	Monitor DELL		Working
#7680	28/08/2024	Monitor	Monitor DELL		Working

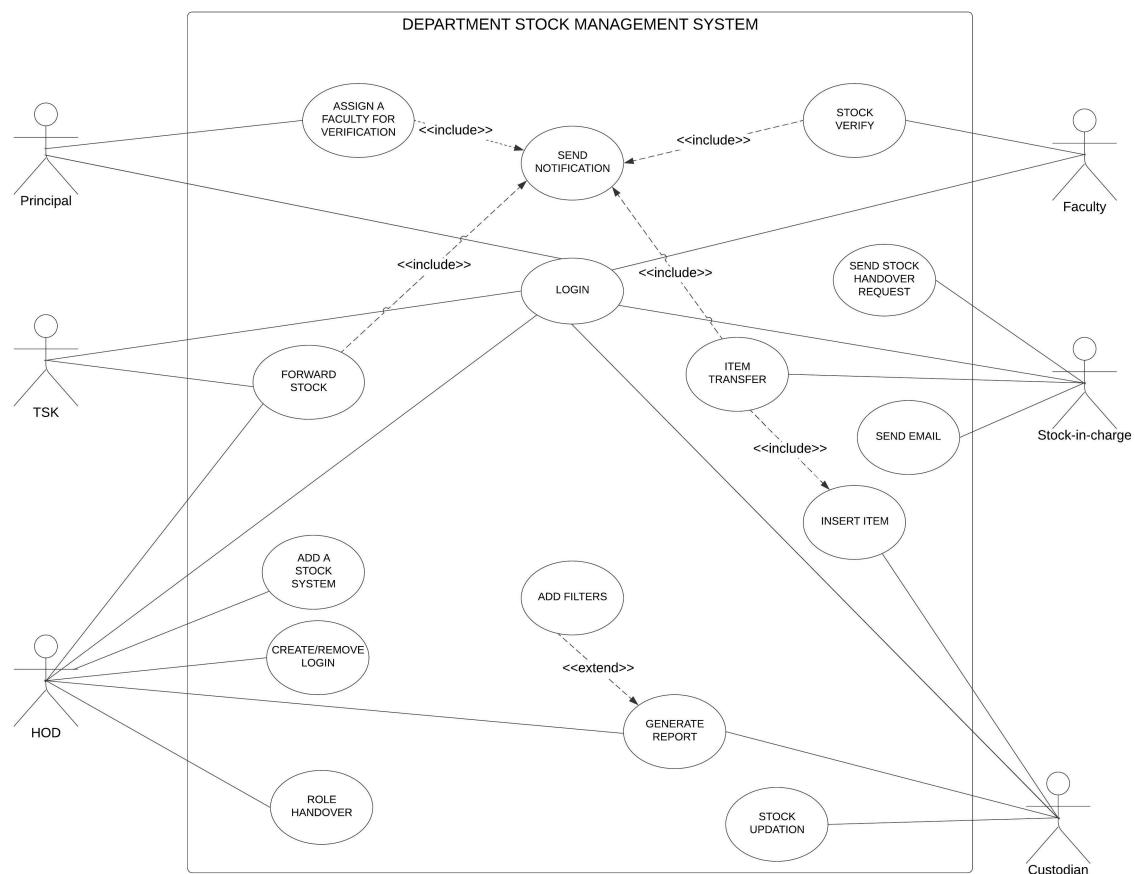
2.2 Module 2: Back End

The detailed design of the back end consists of the detailed design of the connector module and the detailed design of the database module.

2.2.1 Module 2.1: Connector

The detailed design of the connector module involves representing the system using Use Case Diagrams and developing algorithms for each functionality. State chart diagrams are also included along with the algorithms.

Use Case Diagram



Algorithm Design

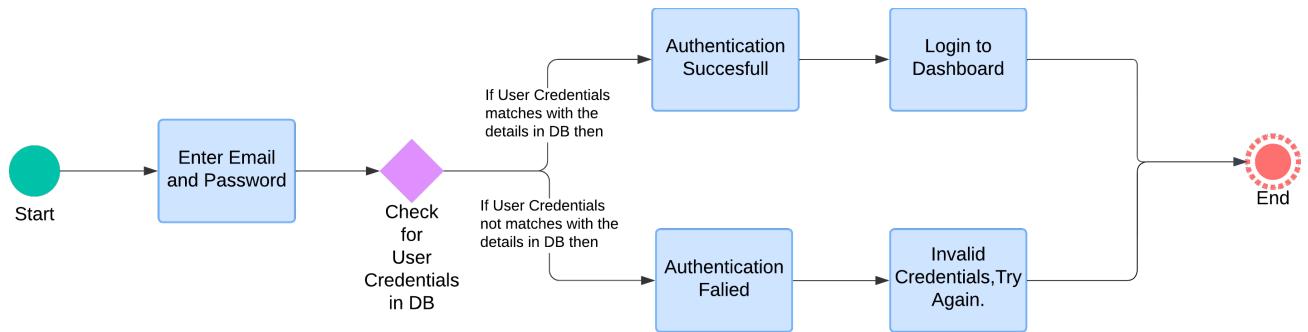


Figure 2: State chart diagram for user authentication

Algorithm 1: User Authentication Algorithm

Input : Email and Password
Output: Authentication status (Success/Failure message with token if successful)

- 1 Input the Email and Password from the user.
- 2 Search for the Email in the user database.
- 3 **if** *the user with the Email is found* **then**
- 4 Validate the Password.
- 5 **if** *the Password matches* **then**
- 6 Generate an Authentication Token.
- 7 Grant access to the system.
- 8 Display "Authentication Successful"
- 9 **end**
- 10 **else**
- 11 | Display "Invalid Password,Try Again!"
- 12 **end**
- 13 **end**
- 14 **else**
- 15 | Display "User Not Found."
- 16 **end**

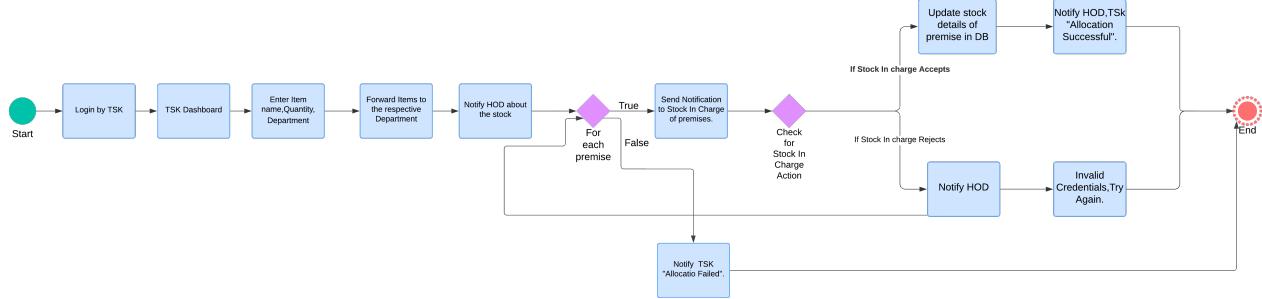


Figure 3: State chart diagram for stock allocation

Algorithm 2: Stock Allocation

Input : Item Name, Quantity, Department (from TSK)
Output: Allocation Successful: Notifications sent to Stock-In-Charge, HOD, and TSK
Allocation Failed: Notification sent to TSK

- 1 Input Item Name, Quantity, and Department (from TSK).
- 2 Forward the items to the selected department by TSK.
- 3 Notify the HOD about the stock allocation request.
- 4 if *HOD forward items to the specific Premise.* then
 - 5 Send Notification to the Stock-In-Charge of the premise with the allocation request.
 - 6 Wait for Response from the Stock-In-Charge.
 - 7 if *Stock-In-Charge Accepts* then
 - 8 Update the stock for the premise and update database data.
 - 9 Notify HOD, Custodian, and TSK: "Allocation Successful".
 - 10 end
 - 11 if *Stock-In-Charge Rejects* then
 - 12 Notify the HOD about the rejection.
 - 13 Notify TSK: "Allocation Failed".
 - 14 end
- 15 end

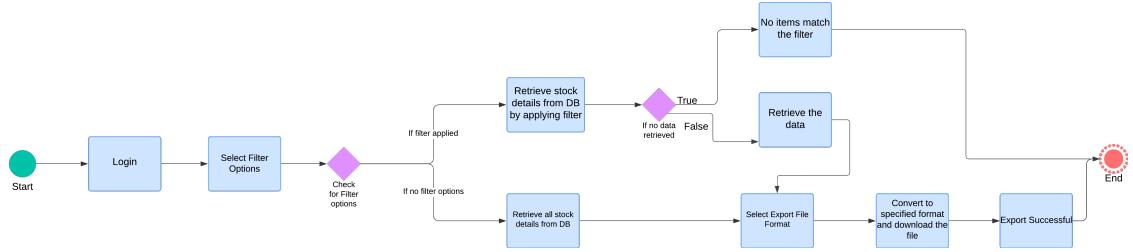


Figure 4: State chart diagram for filter and report generation

Algorithm 3: Filter And Report Generation

Input : Filter options: Item Type, condition, other specifications
 Export options: PDF, Excel

Output: Filtered Inventory Report or Full Stock Report in the chosen format (PDF or Excel).

- 1 Input filter options (Item Type, Condition, etc.) and export format (PDF or Excel). check filters.
- 2 **if** no Filters are applied **then**
- 3 | Query the database for all stock details
- 4 **else**
- 5 | Query the database for records matching the selected filter options.
- 6 **end**
- 7 Check the Query result **if** no records are found **then**
- 8 | Display "No items match the selected filters." Exit
- 9 **else**
- 10 | Retrieve the matching records or full stock details.
- 11 **end**
- 12 Choose the Export format by user
- 13 **if** Export format chosen by user = PDF **then**
- 14 | Format the data (filtered or full stock) into a PDF report using a PDF generation library.
- 15 | Save the PDF file.
- 16 | Display "PDF Report Generated Successfully."
- 17 **else**
- 18 | **if** Export format chosen by user=Excel **then**
- 19 | Format the data (filtered or full stock) into a Excel report using a Excel generation library.
- 20 | Save the Excel file.
- 21 | Display "Excel Report Generated Successfully."
- 22 **end**
- 23 **end**
- 24 **if** User clicks download option **then**
- 25 | Download the generated report.
- 26 **end**

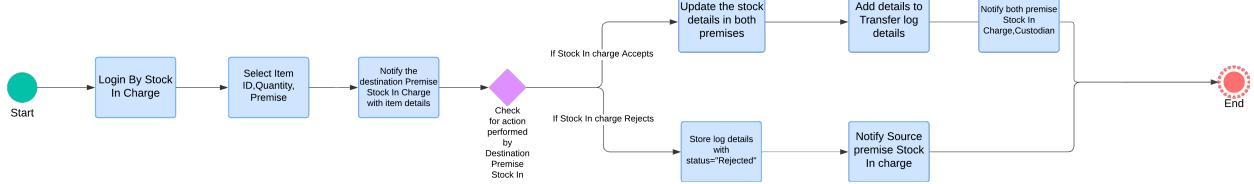


Figure 5: State chart diagram for stock transfer

Algorithm 4: Stock Transfer

Input : Item name, Quantity, Destination premise name by Stock in charge
Output: Transfer Successful: Notification sent to Destination,Source Premise's Stock-In-Charge,Custodian.
Transfer Failed: Notification sent to Destination,Source Premise's Stock-In-Charge.

- 1 Select Item ID, Quantity, and Destination Premise by the Stock-In-Charge of the source premise.
- 2 Send Notification to the Stock-In-Charge of the selected Destination Premise with transfer details:Item name,Quantity,Source Premise name.
- 3 **if** the Stock-In-Charge of the Destination Premise accepts the transfer request **then**
- 4 Update the stock database.
- 5 Deduct the specified Quantity from the Source Premise's stock.
- 6 Add the specified Quantity to the Destination Premise's stock.
- 7 Store the transfer Log details : Source Premise Name, Destination Premise Name, Item Name, Quantity, Date of Transfer, Status: Accepted.
- 8 Notify the sender,receiver Stock-In-Charge,Custodian with a successful transfer message.
- 9 **else**
- 10 **if** the Stock-In-Charge of the Destination Premise rejects the transfer request **then**
- 11 Store the transfer Log details : Source Premise Name, Destination Premise Name, Item Name, Quantity, Date of Transfer, Status: Rejected.
- 12 Notify the sender Stock-In-Charge with an unsuccessful transfer message.
- 13 **end**
- 14 **end**

Algorithm 5: Stock Clearance

Input : Item name, Quantity, Department by Stock in charge
Output: Clearance Successful: Updates the inventory,Notification sent to HOD, TSK, Custodian.
Transfer Failed: Notification sent to Stock-In-Charge.

- 1 Select Item Name, Quantity by the Stock-In-Charge.
- 2 Send Notification to HOD, TSK, Custodian, the Stock-In-Charge with clearance details:Item name,Quantity.
- 3 **if** the Stock-In-Charge marks the damaged items for clearance **then**
- 4 Update the stock database.
- 5 Deduct the specified Quantity from the Premise's stock.
- 6 Store the Clearance Log details : Premise Name, Item Name, Quantity, Date of Clearance.
- 7 Notify the Stock-In-Charge,Custodian with a successful transfer message.
- 8 **else**
- 9 | Notify the Stock-In-Charge with an unsuccessful clearance message.
- 10 **end**

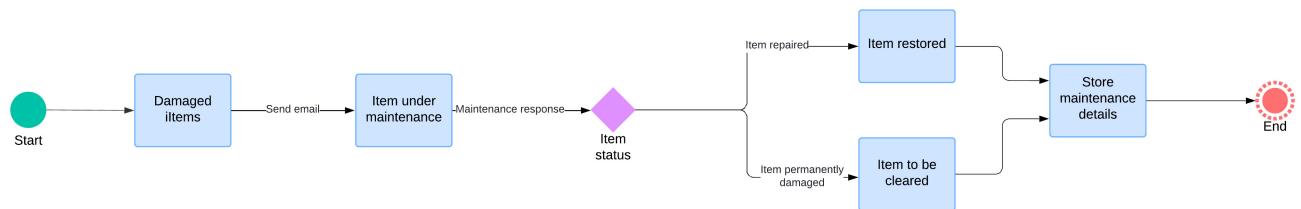


Figure 6: State chart diagram for stock maintenance

Algorithm 6: Stock Maintenance

Input : Sends a notification to service provider by Stock in charge containing details of item requiring repair
Output: After repair: Update item status, Notification sent to Stock-In-Charge, Custodian.

- 1 Send an email to the service provider by the Stock-In-Charge with details:Item name, Quantity, and remark regarding the issue.
- 2 **if** the email was successfully send **then**
- 3 | Update the item status to under maintenance.
- 4 | Notify the Stock-In-Charge, Custodian with a successful email message.
- 5 **else**
- 6 | Notify the Stock-In-Charge with an unsuccessful email message.
- 7 **end**
- 8 Wait for confirmation of task completion from the service provider.
- 9 **if** the item is repaired **then**
- 10 | Update the item status as available.
- 11 | Notify the Stock-In-Charge, Custodian with a successful repair message.
- 12 **else**
- 13 | **if** the item is unrepairable **then**
- 14 | | Update the item status as unserviceable.
- 15 | | Notify the Stock-In-Charge, Custodian with the unrepairable message.
- 16 | **end**
- 17 **end**

Algorithm 7: Stock Handover

Input : Handover request details (from current stock-in-charge), new stock-in-charge information (appointed by HOD).

Output: Stock responsibility transferred, notifications sent, and login credentials updated.

```
1 if the Stock-In-Charge requests stock handover then
2   | Notify the HOD about request.
3 end
4 if the HOD accepts the handover request then
5   | Input details of new stock in charge.
6   | Creates temporary login credentials.
7   | Terminates the login credential of former stock in charge.
8   | Notify both parties (current and new stock-in-charge) about the stock handover.
9 else
10  | if the HOD rejects the handover request then
11    |   | Notify the requesting stock in charge that the request has been rejected.
12 end
13 end
```

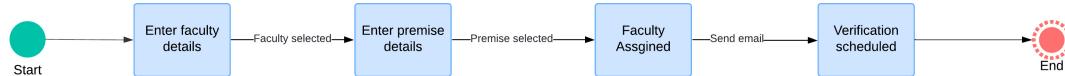


Figure 7: State chart diagram for assign faculty for verification

Algorithm 8: Assign faculty for verification

Input : Name and email-ID of the faculty , department and premise to be verified

Output: Notification send to HOD of the specified department

- 1 Enter the name of the faculty
- 2 Enter email id of the faculty
- 3 Select the department where verification is to be performed
- 4 Select the premise within the department
- 5 Enter the deadline date
- 6 Assign faculty

Algorithm 9: Create temporary login for verifying faculty

Input : Notification from principle regarding assigned faculty

Output: Temporary login created for faculty and credentials is sent via mail

- 1 Input email ID of the faculty
- 2 Enter a password
- 3 Give access to specified inventory
- 4 Share login credentials via email

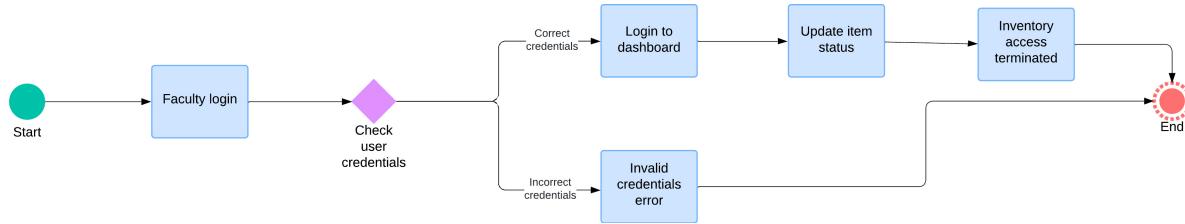


Figure 8: State chart diagram for stock verification

Algorithm 10: Stock Verification

Input : Login credentials received in mail
Output: Verification report sent to principal

- 1 Enter email ID and password
- 2 Fetch email ID and password from database
- 3 if *email ID and password are correct* then
 - 4 for each item in inventory do
 - 5 Mark item if it is present
 - 6 Mark status of item as working or not working
 - 7 Enter remarks if any
 - 8 end
 - 9 Submit verification
 - 10 Send report to principal
 - 11 Send notification to principal
- 12 else
 - 13 Display error message: "Invalid Credentials"
- 14 end

Algorithm 11: Approve stock verification report

Input : Verification report received by principal
Output: Approved report

- 1 Select the report submitted by the faculty
- 2 Verify the report
- 3 Approve report
- 4 Terminate access to inventory for the faculty

Algorithm 12: Add a new stock system

Input : Name, room no and type of new stock system
Output: A new database is created

- 1 Enter name of the premise
- 2 Enter the room no
- 3 Enter the type of the stock system
- 4 Create the stock system

Algorithm 13: Remove a faculty

Input : email-ID of the faculty to be removed
Output: The details of the faculty is removed from the database

- 1 Enter the email-ID of the faculty
- 2 **if** *User with email-ID is found then*
- 3 | Remove the details from the database
- 4 **else**
- 5 | Display error message: "User not found!"
- 6 **end**

Algorithm 14: Add an item to the inventory

Input : Name,ID,date of purchase,date of intend,description,warranty period and room no of the item
Output: The item gets allocated to the premise and notifications are send to stock-in-charge,HOD and TSK

- 1 Enter the item name
- 2 Enter the item ID
- 3 Enter the date of purchase
- 4 Enter the date of intend
- 5 Enter the item description
- 6 Enter the warranty period
- 7 Enter the room no of the premise
- 8 Add the item to the premise
- 9 Send the notifications to stock-in-charge,HOD and TSK

Algorithm 15: Assign faculty in-charge for an inventory

Input : Name,role,email-ID and password of the faculty and name of the inventory
Output: Faculty in-charge is assigned for the inventory

- 1 Enter the name of the faculty
- 2 Enter the role of the faculty
- 3 Enter the email-ID
- 4 Enter the password
- 5 Enter the name of the inventory
- 6 Assign faculty in-charge for the inventory

2.2.2 Module 2.2 : Database

The detailed design of the database module involves representing the system using an Entity-Relationship (ER) Diagram and developing the schema for each entity. The ER diagram captures the relationships between entities such as User,Stocks,Items,Premises etc emphasizing their interactions. Each entity corresponds to a table in the database, and the attributes are carefully chosen to reflect the project's requirements.

ER Diagram

