SOFTWARE REQUIREMENTS SPECIFICATION

FOR

Stock Inventory Application

Prepared By

Meenakshi Sundaram C

Vaishnavi O

Arivazhagan G

Academic Year: 2023-2024

1.Introduction

1.1 Purpose

The primary purpose of the Stock Inventory Application is to efficiently and accurately manage and control the organization's inventory. This application aims to provide real-time visibility into the organization's inventory, including item quantities, locations, and statuses. By doing so, it facilitates effective cost control by tracking purchase costs, storage expenses, and sales prices, allowing the organization to optimize stocking levels and minimize the risk of overstocking or stockouts .Through alerts for low stock levels, the application helps prevent stockouts, ensuring operational continuity and customer satisfaction, while also minimizing overstock situations to reduce holding costs.

1.2 Document Convention

- ➢ Entire document should be justified.
- ➢ Convention for Main title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 14
- ➢ Convention for Sub title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 12
- ➢ Convention for body
  - Font face: Times New Roman
  - Font Size: 12

1.3 Scope of development project

The Stock Inventory Application is designed to efficiently manage and optimize inventory operations within the organization. This application will serve as a central hub for tracking, monitoring, and controlling all aspects of the inventory management process. Its primary functions will include real-time inventory tracking, user access control, detailed item tracking, automated stock alerts, and comprehensive reporting and analytics. The system will be designed to cater to the needs of various stakeholders, including inventory managers, procurement teams, warehouse staff, and administrators. By automating and streamlining inventory-related tasks and processes, the application aims to enhance operational efficiency, reduce costs, prevent stockouts, and provide valuable insights into inventory performance for informed decision-making

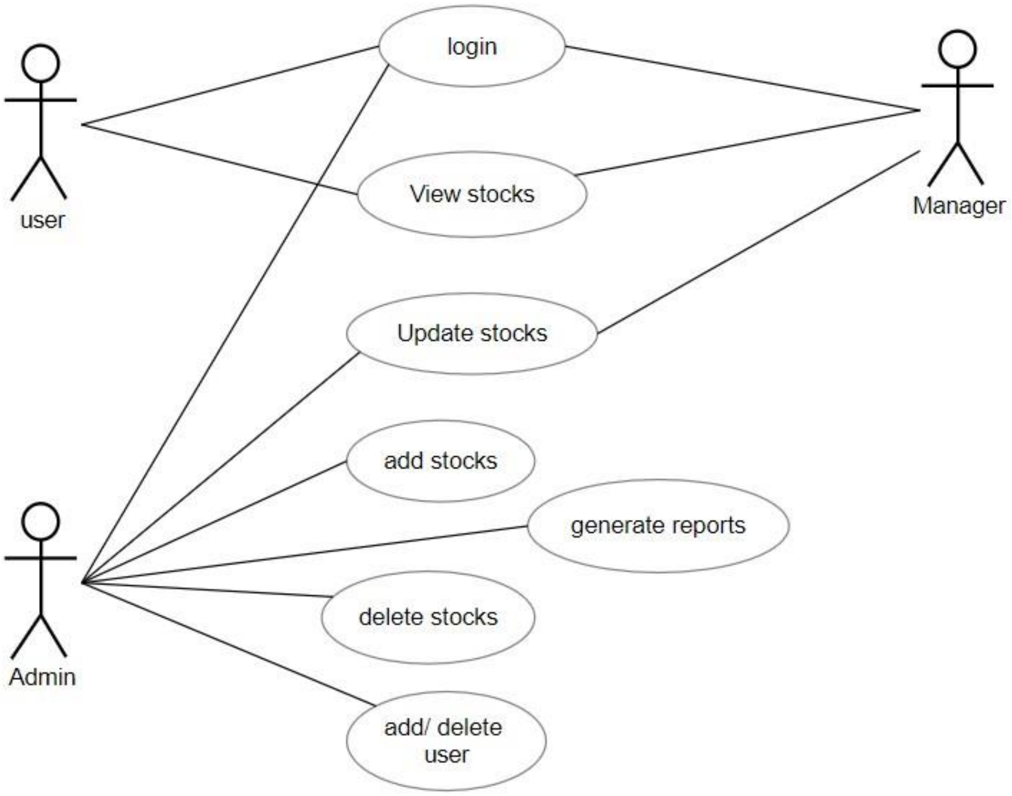
1.4 Definition, Acronyms and Abbreviation

UML - Unified Modelling Language
ER - Entity Relationship
SRS - Software Requirement Specification
SQL - Structured Query Language
IDE - Integrated Development Environment

2.Overall description

2.1 Product Perspective

Use case diagram



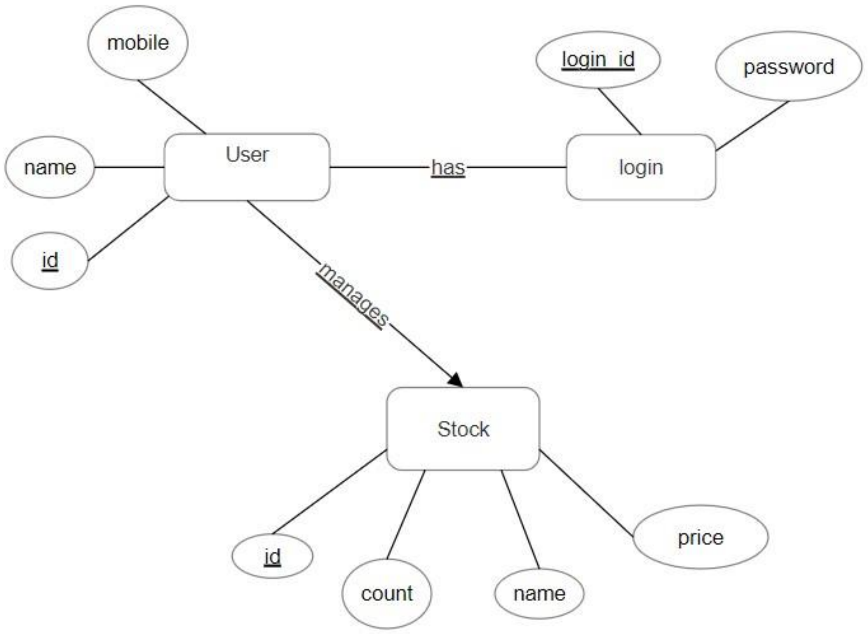
A use case diagram is a type of diagram that represents a system's functionality and the ways users or external systems interact with that functionality. It's a high-level visual representation that helps to define the system's behaviour and requirements from a user's perspective.

The main actors of the system are the users who can be the employees or the suppliers, managers and Admin.

All the actors can login to the system. The user can only view the stock availability. The manager can view as well as update the stocks based on the supply. The admin has the maximum functionalities. He can view, update , add and delete the stocks . He can also generate reports on the stock availability. He can add or delete a user .

2.2 Product Function

ER diagram



An Entity-Relationship (ER) diagram is a type of visual representation that illustrates how different entities in a system are related to each other. It is commonly used in database design to model the relationships between entities and the attributes associated with those entities.

The main entities of the system are the users, stocks and login. Every user must require a login in order to access the system .The attributes of login includes login id and password . The attributes of users are user id, username and mobile number. Here the user is represented generally and can be the admin , employee or the manager. The user manages the stocks available. It includes the features like adding , deleting , viewing and updating .

2.3 User classes and characteristics

The system allows different users with different types of functionalities. The Three type of users are Admin ,Inventory Managers and employees.

Admin :

- Full control over all the operations of the system.
- Manage the user accounts and permissions.
- Adding new stocks to the inventory.
- Deleting and updating existing stocks.
- Generate reports

Inventory Mangers:

- View all stocks available.
- Update stocks.

Employees:

- View stock availability.
- No operations on stocks.

2.4 Operating environment

The stock inventory application is designed for Android. Android version must be greater than 8.0. The device must have a RAM greater than or equal to 4 GB. It must have a minimum of 2 GB available space. The application may require various permissions on the user's mobile device, such as location access for GPS-based features, and storage access for saving data and images locally. It also requires a stable internet connection with a minimum bandwidth of 1.5 gbps.

2.5 Assumption and dependencies

The assumptions are,

- The code must be error free
- The application must be user friendly so that it is easy to use for the users
- The stock details and availability must be stored in the database
- Accessing and updating the stock details in database must be faster
- The application must provide search facility.
- The user must have a proper login to access the application.
- The application is accessible any time.

The dependencies are,

- The specific hardware and software due to which the product will be run
- On the basis of listing requirements and specification the project will be developed and run
- The end users should have proper understanding of the application
- The system should have the general report stored
- The information of all the stocks must be stored in a database that is accessible by the Application
- Any update regarding in the stock is to be recorded to the database and the data entered should be correct.

2.6 Requirements

Software Configurations

- Programming Language – Java
- IDE – Android Studio
- Database – SQLite

Hardware Configurations

- RAM – 4 GB or more
- Storage – Minimum 2 GB free space
- Connectivity – Wifi , 3G or more

2.7 Data requirement

The input consists of queries to database and output is the results of the queries. All the operations like adding a new stock , deleting and updating the existing stocks is given as a query and the corresponding operation is performed on the database. The Stock details are displayed to the users when they request the server. The stock details, user details and the supplier details is required to be stored in separate tables in database.The stock details include stock price, availability , supplier . The supplier details include supplier name, supplier id, stock supplied.

3.External Interface requirement

3.1 GUI

The System provides a simple graphical interface for the users and the admin can perform any operations on the system link adding , deleting and updating the stocks.

- The design should be simple and aesthetic
- All the modules available should fit within the graphical user interface.
- It must be easily understandable by any new user.
- It must provide search facility based on different criteria.
- The details must be displayed with proper spacing and standards.

4.System Features

- The users must be authorised and validated to access the system. Only admin can add new users .
- Alert admin in case of low stock or overstock.
- The admin can add new stocks and delete stocks .
- Allows user to generate reports on stock usage.

5. Other Non-Functional Requirements

5.1 Performance Requirements

- The system should respond to user actions within under normal load conditions.
- Data retrieval and display operations, such as loading inventory lists, should not exceed more than 5 seconds.
- The application should be scalable to accommodate a minimum of concurrent users without significant performance degradation.
- The application should have limited offline functionality, allowing users to access and view data even when not connected to the internet.

5.2 Safety Requirements

- The application must maintain the integrity of inventory data to prevent data corruption or loss.
- Regular data backups should be scheduled to ensure data recovery in case of system failures.
- In the event of critical issues or data breaches, the application should provide a mechanism for emergency shutdown to protect sensitive information.

5.3 Security Requirements

- System should use secured database
- Proper user authentication should be provided
- No one should be able to hack users' password
- There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

5.4 Business Rules

This defines specific rules and algorithms for determining when to generate purchase orders based on inventory levels, sales trends, and lead times. Establishes rules for pricing calculations, including profit margins and pricing adjustment. Also define how items are categorized and classified within the inventory system.

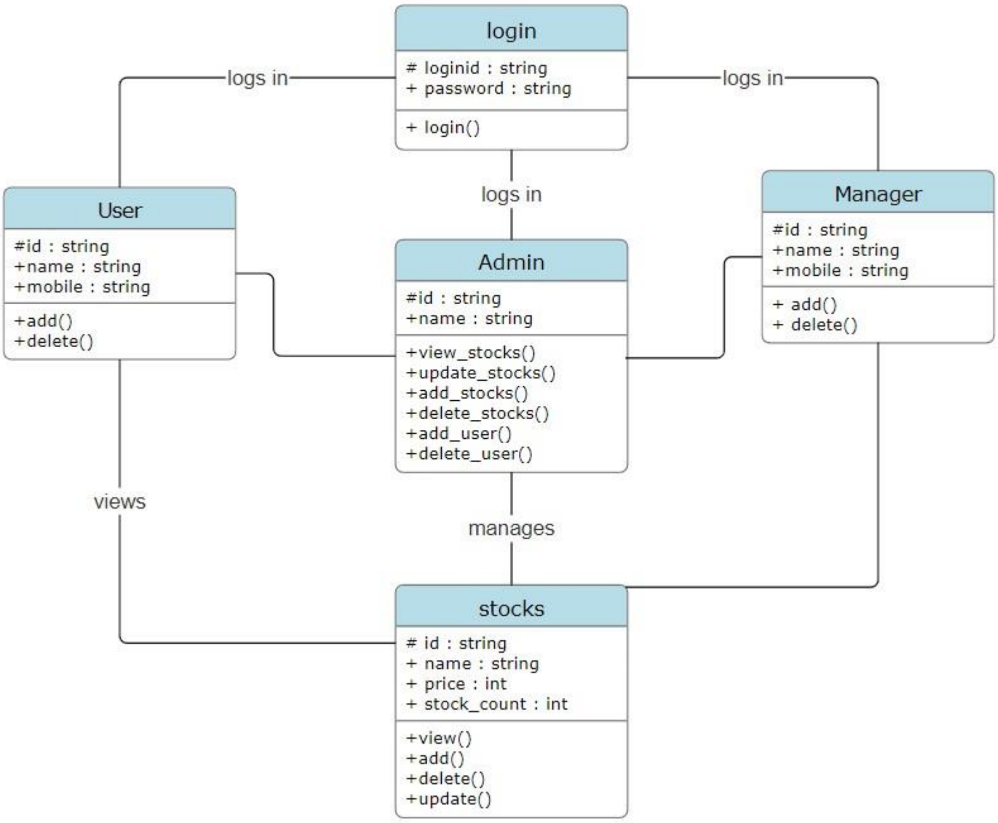
5.5 User Requirements

- Provide training materials and resources to educate users on how to effectively use the mobile application.
- Implement a mechanism for users to provide feedback and report issues, with a commitment to addressing user concerns in a timely manner.
- Ensure that the application is accessible to users with disabilities, adhering to relevant accessibility standards and guidelines (e.g., WCAG).

6.Class diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes' structure and their relationships to each other frozen in time represent the static model.

Here the main classes are login , admin , user , manager and stocks. Each of these class has its own set of attributes and methods. There exists relationships between these classes like association and aggregation .


login
# loginid : string
+ password : string
+ login()
logs in
logs in
logs in
User
#id : string
+name : string
+mobile : string
+add()
+delete()
Admin
#id : string
+name : string
+view_stocks()
+update_stocks()
+add_stocks()
+delete_stocks()
+add_user()
+delete_user()
Manager
#id : string
+name : string
+mobile : string
+ add()
+ delete()
views
manages
stocks
# id : string
+ name : string
+ price : int
+ stock_count : int
+view()
+add()
+delete()
+update()