**Ali ARSLAN**

**CS-360 3-3 Project One**

**Selected Application:** Option 1-Inventory App

**Inventory Application Project Plan**

**Introduction and Goals of the Application**

The chosen application for this project is the Inventory App. The primary purpose of this app is to manage warehouse inventory efficiently, ensuring accurate tracking of items, updating their quantities, and maintaining stock levels. This application will be beneficial for warehouse managers, inventory personnel, and small business owners, providing them with a streamlined tool to oversee their inventory operations.

**Mobile App Objectives**

The main objective of the Inventory App is to facilitate the tracking and management of items in a warehouse environment. The app aims to streamline inventory management processes with real-time updates, simplify item recording, and promptly notify users when stock levels are low or depleted. Additionally, this app will help users reduce costs by providing efficient inventory management. The app will feature a grid system to display inventory, a search function for detailed searches, and screens for adding, deleting, and editing inventory items. User data will be displayed on the screen, and a notification system will alert users based on specified conditions.

**Major Components**

1. **Two-Table Database**
   * **Inventory Items Table**: Stores information about each item, including item ID, name, description, quantity, and location.
   * **User Logins Table**: Manages user credentials securely, providing access and authentication.
   * **Additional table** : Additional tables may need to be added to add features such as notification and alert.
2. **Login Screen**
   * Allows users to log into the application or create a new account, ensuring secure access.
3. **Inventory Grid Screen**
   * Provides a comprehensive view of all items in inventory in a grid format, enabling users to view current stock levels.
   * Includes a search function above the columns for quick searches and buttons for transitioning to the editing screen, alert screen, or processing actions.
4. **Item Management Screen**
   * Users can add, remove, update items, and adjust quantities to ensure the database remains current and accurate.
   * Automatically fills relevant fields when updating items.
5. **Notification System and Screen**
   * Notifies users when the quantity of any item reaches zero, preventing stockouts and enabling timely replenishment.

**Potential Users and Assumptions**

1. **Individual Sellers Online**

For individuals selling online, managing inventory can be challenging as products may be located in different places such as their homes, warehouses, or Amazon depots. This application can help them organize these locations and quantities and can also reduce their costs by providing low-cost the way.

* + **Goals**: Optimize stock management, track inventory, improve sales operations, and increase customer satisfaction.
  + **Needs**: Real-time stock updates, fast and accurate inventory tracking, critical stock notifications.
  + **Interaction**: Daily stock management to improve performance.

1. **Warehouse Staff**

These people may need an application where they can follow everything more easily and mobile, instead of complex applications. For this reason, this application can comply with their living standards.

* + **Goals**: Quickly add, update, and remove inventory items, ensure accurate stock counts.
  + **Needs**: A user-friendly interface, fast item browsing and updating, notifications for low stock.
  + **Interaction**: Frequent interaction throughout shifts to manage inventory tasks efficiently.

1. **Small Business Owners**

Since small business owners do not have very complex operations and cannot pay very high costs for such simple operations, this type of application will meet their needs and be compatible with their lifestyles

* + **Goals**: Oversee overall inventory management, ensure business continuity, make informed decisions based on inventory data.
  + **Needs**: Access to inventory reports, insights on stock levels, notifications about critical inventory changes.
  + **Interaction**: Periodic use to review inventory status and make strategic decisions.

1. **Small Manufacturer:**

Small manufacturer needs may have a more complex structure than other users, and these users may want to separately track the stocks of the end products they produce and the items they use as raw materials, which may reveal the need for an advanced categorization system.

* + **Goals:** Tracking the raw materials they use in production and keeping and tracking the stock quantity of the products they produce. Planning an uninterrupted production process in this way.
  + **Needs:** Ability to separate stocks into different categories (production, final product, etc.) and to do this in detail and track them.
  + **Interaction:** Periodic use to review inventory status and make strategic decisions.

**Necessary Screens and Features**

1. **Login Screen**
   * Allows user authentication and account creation.
   * Transitions to the Inventory Grid Screen upon successful login.
2. **Inventory Grid Screen**
   * Displays inventory items in a grid format.
   * Search function above columns for quick searches.
   * Buttons for transitioning to the Item Management Screen and Notification Screen.
3. **Item Management Screen**
   * Allows adding, updating, and removing inventory items.
   * Displays current item details for updates.
4. **Notification Screen**
   * Lists alerts for low stock items.
   * Allows users to configure alert settings.

**High-Level Design and Navigation**

**A diagram of a diagram

Description automatically generated with medium confidence**

1. **User Flow**:
   * Users log in or create an account on the Login Screen.
   * Upon successful login, users are directed to the Inventory Grid Screen.
   * From the Inventory Grid Screen, users can navigate to the Item Management Screen to add, update, or remove items.
   * Users can access the Notification Screen to view alerts and configure settings.
   * To achieve these, a more intuitive structure should be designed instead of the standard navigation system. For example, alert adding, editing or slime buttons should be placed next to the relevant item in the DataGrid and the add button should be visible, and profile settings can be accessed by clicking on the person's profile picture. (Sample Provided Above)
2. **Justification**:
   * The design adheres to Android Design and Quality Guidelines, ensuring a user-centered UI with intuitive navigation and accessibility.

**Code Design and UI Integration**

1. **Functional App Requirements**:
   * Secure user authentication and account management.
   * Real-time inventory updates and accurate item tracking.
   * Push notifications or in-app alerts for low stock items.
2. **Data Flow**:
   * All Data will be stored in online database.
   * The app will use RESTful API calls to interact with the database, ensuring real-time data updates.
   * UI components will display data retrieved from the database and accept user inputs for item management.
3. **Data Transfer Between Screens.**

Data Transfer between Pages is quite limited. We can adjust these in two ways. Data transfers between the Login Screen and Application screen and items on the Grid Screen. Ancestor transfers between the deletion or editing screen.

* + **Data transfers between the Login Screen and Application screens:** After the user successfully logs in to the application, the user ID will be stored in a constant and only the user's information will be retrieved from the online database on the relevant pages. Therefore, the user ID must be kept in a global constant in order to be transferred between pages.
  + **items on the Grid Screen:** When the update or delete item button is clicked, the id of the relevant item should be sent to the piece of code that will perform the relevant action. In this way, when editing the product, the necessary information can be obtained from the database or the deletion process can be performed successfully.

1. **Major UI Components**:
   * **Login Screen**: Input fields for username and password, login and sign-up buttons.
   * **Inventory Grid Screen**: Grid view of items, search textbox, buttons for item management and alerts.
   * **Item Management Screen**: Input fields for item details, buttons for adding, updating, and removing items.
   * **Notification Screen**: List of alerts, settings for configuring notifications.
2. **Data Management**:
   * Inventory data will be fetched from the Inventory Items Table and displayed on the Inventory Grid Screen.
   * User inputs on the Item Management Screen will update the database accordingly.
   * Notification settings and alerts will be managed through the Notification Screen, with data stored in the database.

This plan outlines the goals, major components, user types, necessary screens, and features for the Inventory App, ensuring a comprehensive and user-centered design.