



COLLEGE OF ENGINEERING

Department of Software Engineering

Software Component Design

Group Project

Project Title: Inventory Management System(mPOS) using Fimga and Agile Model

SECTION: A

Group Members	ID
1. Abdi Getachew	ETS 0003/13
2. Ahmed Abdulhakim	ETS 0091/13
3. Akrem Ahmed	ETS 0097/13
4. Ali Salahadin	ETS 0105/13
5. Mahfouz Teyib	ETS 0786/13

Submitted to: Lecturer Gizatie D.

Submission date: December 18, 2024

Table of Contents

Introduction	1
Functional Requirements	1
2.1 User Authentication (for Both Users and Admins)	1
2.2 Inventory Management	1
2.3 Order Processing	2
2.4 Admin Dashboard and Reports	2
2.5 Purchase Process	3
2.6 Additional Features	3
Technical Requirements	4
Pages of the App	5
I. Admin Side Page	5
II. User side page	6



Introduction

- **Purpose:** The mPOS application is designed for small businesses to manage inventory, process sales, and handle payments using mobile devices, with support for payments via Telebirr. This version also includes an **admin interface** for business owners or system administrators to manage inventory, users, and view analytics. The app is built using **Flutter** for cross-platform support and **Firebase** for the backend.
- **Scope:** The MVP will provide basic features for both users (merchants) and admins. Admins will have additional privileges like managing stock, monitoring transactions, and viewing business analytics.

Functional Requirements

2.1 User Authentication (for Both Users and Admins)

1. Login and Registration:

- Users and admins can log in using email and password.
- Firebase will handle authentication for secure login.

2. Role-Based Access:

- Admins have elevated privileges compared to regular users (e.g., merchants).

2.2 Inventory Management

1. View Stock:

- Users and admins can view a list of items in stock, including product details such as name, price, and quantity.

2. Add/Edit Items (Admin Only):

- Admins can add new products, update existing ones, and adjust pricing or stock levels.

2.3 Order Processing

1. Add Items to Orders:

- Users can add items from stock to customer orders.

2. Review and Finalize Orders:

- Users can review order details before completing the transaction.

3. Order Completion:

- Once orders are finalized:
 - Stock levels are automatically updated.
 - Admins can access and review the order history.

2.4 Admin Dashboard and Reports

1. User Management:

- Admins can add, remove, or update user accounts (e.g., merchant accounts).

2. Transaction Reports:

- Admins can generate daily, weekly, and monthly transaction reports.

3. Inventory Reports:

- Reports include stock usage, out-of-stock alerts, and inventory trends.

4. Sales Analytics:

- Data visualizations show total sales, top-selling items, and payment trends.

2.5 Purchase Process

1. User Side:

- Users can log in to their accounts.
- Users can select items to purchase, and the system will automatically decrease inventory after the transaction.

2. Admin Side:

- Secure login for system management.
- Dashboard with an overview of:
 - Total sales.
 - Stock levels.
 - Recent transactions.
 - Revenue and profit.

2.6 Additional Features

1. Low Stock Alerts:

- Automated notifications inform admins when inventory levels are low.

2. Loan Management:

- A system for managing loans as part of the operations (further details to be defined).



Technical Requirements

3.1. Platforms

- The app will be built using **Flutter** to support Android and iOS.

3.2. Backend

- **Firebase Authentication:** For managing user and admin login.
- **Firebase Firestore:** For storing inventory, orders, and transaction data.
- **Firebase Functions:** To handle server-side logic .
- **Firebase Storage:** For storing images (e.g., product images).

3.3. Connectivity

- The app will have **offline mode** capabilities for users, storing data locally and syncing with Firebase once the device reconnects. Will be done if time allows.

3.4. Security

- **Firebase Auth:** To handle secure user authentication.
- **Firestore Rules:** Role-based access for users and admins to ensure data security.

3.5. Admin-Specific Features

- **Role-Based Access Control:** Admins will have access to all data, including stock management, sales reports, and user management.

- **Data Analytics:** Firebase analytics or external tools (e.g., Google Analytics) to track app usage and sales.

Pages of the App

I. Admin Side Page

1. Signup Page

- Admin can register to the system.

2. Login Page

- Secure login for admins.

3. Dashboard

- Overview of key metrics:
 - Revenue
 - Total Sales
 - Profit
 - Recent Transactions

4. Inventory Management

- **Inventory List:** View all available items with details like name, price, and quantity.
- **Add New Item:** Add new stock items (product name, price, stock levels, etc.).
- **Edit Item:** Modify details of existing inventory items.

5. Analytics and Reports

- **Revenue Analytics:** Visualizations of total revenue.
- **Profit Analytics:** Track profits.
- **Lent Analytics:** View loaned amounts or items.

- **Sales Analytics:** Display sales trends and top-selling items.
- **History Page:** View past transactions and logs.

6. User Management

- **Add New User:** Add merchant/user accounts.
- **View User List:** Manage users, update, or remove them.

II. User side page

1. Login Page

- Users can log in to access their accounts.

2. Catalog Pages

- **Catalogue View 1:** Display available products by categories.
- **Catalogue View 2:** Alternative layout for viewing products.

3. Cart Page

- **View Cart:** See selected items, quantity, and total cost.
- **Edit Cart:** Add, update, or remove items in the cart.

4. Checkout Page

- Finalize the purchase, showing total cost.

5. Profile Management

- **Edit Password:** Update the user account password.