Title : Market Sales Management System

Name	ID	
Yonas Ayele	4528	
Hayat Jemal	5292	
Tashel Mekoro	4541	
Segni Terakegn	4737	adviser : Segni B.

List of Acronyms

POS ---------- point of sale

Table Content

Figure

Table

Chapter one: Introduction

1. 1 Introduction

A market is a platform where buyers and sellers come together to exchange goods and services. Markets can take many forms, such as physical markets and Online markets. In a physical market, buyers and sellers meet in person to negotiate prices and exchange goods. Online markets, such as e-commerce websites, allow buyers and sellers to connect and trade goods electronically.in this topic we will be dealing with the physical market.

A physical market is a physical location where goods and services are bought and sold. Physical markets can range from large international exchanges to small local bazaars. Physical markets create a face-to-face opportunity for buyers and sellers to interact with each other while trading goods and services.in most developed country's the market system have been developing and the use technology to make sealing and cash management easy by using POS (pint of sale) system. POS is device that have scanner ,cash register and screen to display info. The bar-code and QR-code on product is scanned at the POS (point of sale), which allows the retailer to quickly and accurately track the sale of a particular item. A bar-code and QR-code is a method of representing data in a visual format on a product. It is typically used in retail settings to track inventory and pricing information.

When we came to our proposal we will be dealing with system that help improve the price control method over our country. its simpler to POS system its device used mostly in developed country this device used in super market and scan every product as user buys it and the system will register every activity in the super market and at the end of the day, week or month the owner can view every activity of his shop from price of every day to amount of product sealed and unsold also system view most sealing product and other infos.and our our system will be application form of this device it can perform every activity the POS machine perform and many more.

The system will have to pricing methods first there is manual pricing method in this system the user will decide the price of every product in the shop in the second method the government decide the pricing of product.

Also by using our system government will be able to control price of product as u know there is problem of uncontrolled pricing in our country this made buying and sealing of product very hard so in this admin side the government or the admin will post price of every product and the system will calculate the price for every sealer in the country by identifying the length from product source point to its destination.

1.2 Background

Super markets and other shop like electronics are modern way of exchanging product they enable easy access of product for there customers and also they mainly include products that pass the manufacturing step means that item like unpacked fruits and vegetable don't included. passing the manufacturing steps is very important. And when we came to the reality of us using those product identifies we are very fare behind.

So the main concern of our system is on shops and market place like super market ,electronics and pharmacy etc… as we know shopping is one of the basic human activity's seen long time ago unlike us in developed country's it have developed very rapidly there is POS system that manage every activity of there store. But when we see our country shopping haven't grown that match from the traditional ways of shopping sealer use paper to register infos. And as same of the sealers tolled they were wanting to use POS system for there mini market and they couldn't afford it.

1.3 Existing System

Current system called POS system its used in large super markets like sehwa super market and others. It look like computer but its modify include same other hardware like scanner and cash register to do shopping task easy. mainly control cash registration and product existing amount.

1.4 Statement of the Problem

The current system is good system but same of its native impact listed below.

- Very expensive user with small shops couldn't afford it
- Complexity of the system
- Because its hardware it require maintenance if it fail
- Fixed at one place user cant move it to home to review profit after work

1.5 Objective

General Objective

Our general objective is to develop android based market sales management system for mattu town

Specific objective

Our system concern about mattu town saler's and its specific Objective is listed below.

Review problem of the current system: focusing on the problem and limitation of the current system while so that we can develop system that is useful then existing system

gather requirements: gathering information my be by observation or by interview and quetionary so that we can have good view of the existing problem

Analyse gathered data: manage and try to review the information gathered so that we can develop system that compatible with system

Review similar work done by other people: searching for similar work so that we can easily identify how our system while in use and to easily pass the problem they encounter.

Develop the system using java and commit table library: this the main part of the project and the most time consuming in this step we will be developing the backed using java and the front-end using XML

Taste the system for bug and error : in this step we will be tasting and handling existing exception so that the system will not crash will in use

Finally publish System for users: after accomplishing all that work and taste finally the system will be published for use.

1.6 scope of project

Scope of the project cover the following listed areas.

- Enable sealer's to register shop activity including costs in database instead of using notebook.
- Enable reliability of info user can't lose any data of the activity
- Availability at any time the app is available at any time so the user can review data at any time and place
- Enable government to collect vat
- Also help avoid importing vat machine this will decrease unwanted expenditure
- View report like list of item on the market and top sealing item also profit using XY graph it my be daily ,weekly,monthly or end of the year.

1.7 Project limitation

As fare as we try there is always a limitation for any system so our system my have the following limitations.

- Our system don't include other shop like mini shop,construction material shop etc… because most of the product in those shops don't have bar-code or qr-code.
- Our system sore data only if user use it all the time
- If user don't have smart phone or if the smart phone sdk version is low they can't use this system

1.8 Significance of the project

our project will help the organization modernize info registration method and cost management ways.currently organization do this using notebook. Our system will enable the organization increase income by viewing every days profit and show alert if profit decrease.The main significance of the project is to help the people in the following ways.

- To perform their work,faster,efficiently and accurately.
- To help people to adapt to new technology.
- Help government reduce cost.

1.9 Methodology of the Project

1.9.1 Requirement Gathering Method

We have been collecting data by observation ,interview and same shops , pharmacy and electronics. We used three method of data gathering.

Observation: we have been gathering requirements this method as we observed most sealer's they sale product and they don't use any type of way to register sealed product info and they cut vat using 40,000 bir vat machine

Interview: and also we have been interviewing same shops for info and as they tolled us same of them use notebook to register sealed product info

Questionnaire: we have been asking series of YES/NO as we observed from this sealer don't have basic info about bar-code on product and use of bar-code and qr code

1.9.2 Methodology for Analysis and Modelling

In our project to analyse and model the input scenarios that we will gather, we will use object oriented system analysis and design (OOSAD) methods. During this phase the team will use to model the function of the system (use case modelling), find and identify the business objects, organize the objects, identify the relationship between them and finally model the behaviour of the objects. This is because an OOSAD provides the following advantages

- **Provide better understanding**
- **Maintainability** OOP systems are convenient and easy to operate compared to structured programs.
- **Allows to breakdown** complicated systems into smaller, clearly defined and more manageable parts.
- **Re-usable** enables reuse of old data in new applications

We choose the **Iterative model** to develop system because In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.

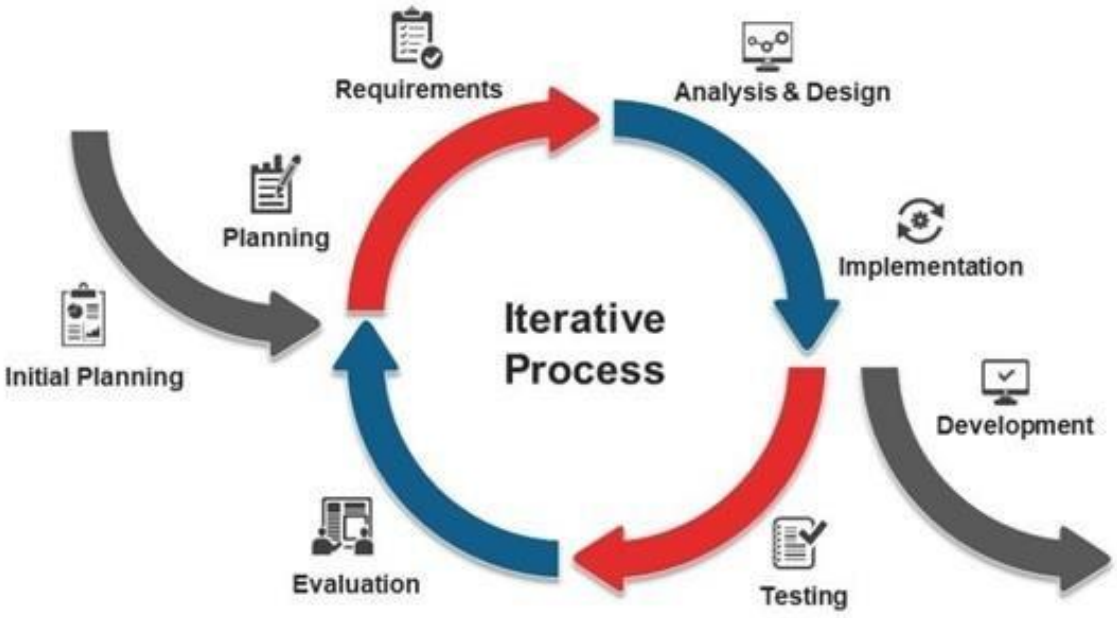


Figure:1.1 ***Iterative model***

1.9.3 System Implementation Method

We will be using same hardware ,programming and software tool to develop the system they are:

Hardware tool

name	Description
Laptop(Pc)	For coding designing,tasting
Android phone	For tasting and dubbing
Usb cable	For running the app
Usb flash	For transferring files
Internet cable	For getting internet to search in web

Table 1.1 **Programming tool**

Name	Description
Java	For writing general functional part of the app
XML	For writing the front end part of the app
fire-base	For writing the database script

Table:1.2 Software tools

name	description
Android studio	For making the android app
Adobe XD	For designing the UI/UX parts
Adobe Photo-shop	For making app icons and backgrounds
Microsoft Word	For writing proposal and documentation
Edarw-max	For designing the models

1.9.4 Testing Methodology

tasting will be done after finishing the code to identify error or check if system work properly by using **unit tasting** which enable as to taste single unit or part from the system and also we will be using **system tasting** to taste the all system.

Using the Unit tasting will benefit in :

- help identify error easily
- help us to fix bugs early in the development cycle
- Helps know which part of our code is wrong

Also System tasting will benefit in:

- To debug app before publish
- Identify if app crash while in use

1.10 Feasibility

1.10.1 Economical Feasibility

The economical feasibility have been done in relative to the app cost my be required to accomplish this project it my include tangible and intangible costs tangible include cost for travel to town for information gathering and other human power the can't be determined in money.

Hardware and software cost

No	Name	Amount	Price per unit	price
1	Cost for travel	--	10 birr	100 birr
2	Cost for paper	30	1 birr	30 birr
3	Cost for pen	3	20 birr	60 birr
4	android studio	--	--	Open source
5	Adobe XD	--	--	Open source
6	Edraw-max	--	--	Open source
	total			190 birr

Table:2.3 software and hardware

1.10.2 Technical Feasibility

Technical view user can easily able to know how to use the system because the user app will be developed in very interactive way so user can communicate with app. Also the app will note include very broad and complicated contents so the it will note confuse the users. So any user with the general knowledge android phone is able to use this app.

1.11.3 Schedule Feasibility

As anything that not done without plan can fail we must have basic schedule for developing our system its views as follow using Gant chart

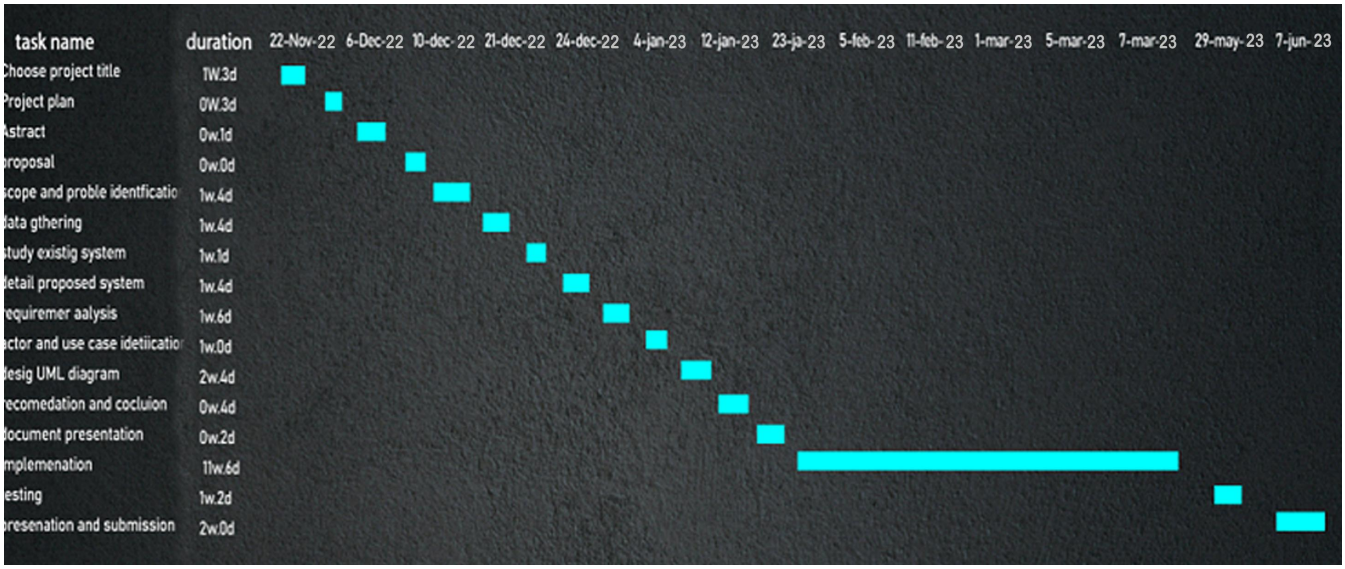


Fig 1.2 schedule feasibility

1.11 Risk

During the project development time same risk my arise same of them are.

- Pc which we are working on my get broken or crash
- Internet connection my be not working
- Same library my not work
- User my not be interested in using the app
- For same device app my be incompatible
- Lack of time due to exit exam

1.12 Risk Assessment and management

We can avoid same of the risk from ceasing problem same of them include.

- We can avoid pc problem by getting backup pc in case of problem occur.
- In case of internet connection problem we can use package data.
- In case of library not installing we can use you-tube instruction.
- If user not interested in using the app try to give guide about it functionality and it benefit
- In case of app in comparability we can expand the Min sdk version to hold more android phone as possible.
- We can solve the time problem by making proper schedule plan.

Chapter Two: Analysis

2.1 Introduction

Analysis is the process of breaking down a complex concept or phenomenon into smaller parts in order to gain a better understanding of it. It is often used in data analysis, where data is collected and then analysed in order to identify patterns or make conclusions.There are many advantages to analysis, including understanding the root causes of problems and finding solutions. Analysis can help you identify the best ways to solve problems, and it can also help you identify potential solutions.When it comes to analysing a mobile point of sale system, there are several key areas to consider. These include the ease of set-up and use, user interface design, security and reliability, customer service, and long-term scalability. Consider the metrics associated with each of these aspects to develop a comprehensive analysis of the system. Additionally, you may also want to look at the features offered and any customer feedback or reviews as part of your assessment.

2.2 Existing System

2.2.1 Existing System Description

A POS (point of sale) system is a type of software and hardware that is used to process sales transactions in a retail or hospitality setting. POS systems are typically made up of a combination of a computer, a cash drawer, a card reader, a receipt printer, and a bar-code scanner. The software component of a POS system is used to manage inventory, track customer information, and process payments. Some POS systems also include additional features such as customer-facing displays, kitchen displays for restaurants, and integrated reporting and analysis's.

One of the main advantages of using a POS system is that it allows businesses to streamline their sales and operations by automating many of the tasks that are typically done manually. This can help to improve accuracy and efficiency, while also providing valuable insights into sales and customer data. POS systems also typically offer advanced security features to protect sensitive customer information and transactions. Additionally, many modern POS systems are now cloud-based, allowing businesses to access their sales data and manage their operations remotely.

2.2.2 Supplementary Requirements

1. BUSINESS RULES

Business rules for a POS (Point of Sale) system are the guidelines and procedures that govern how the system is used and how data is entered and processed. These rules ensure that the system is used consistently and accurately, and that the data it produces is reliable and meaningful.

- **Authorization:** Access to the POS system and related data must be restricted to authorized personnel only.
- **Item entry:** specifying how products and services are entered into the system, including the use of unique identifiers such as bar-codes or QR-code, and the required fields for each item.
- **Pricing:** determining how prices are set, including the use of discounts, promotions, and special pricing for different customer groups.
- **Taxation:** specifying how taxes are calculated and applied, including the use of different tax rates for different items or regions.
- P**ayment processing:** if off-line it can only accept cash
- **Reporting:** specifying how data is reported, including the types of reports that are produced, and the frequency and format of those reports.
- **Security:** specifying the procedures for maintaining the security of the system, including the use of passwords and other forms of authentication, and the procedures for protecting sensitive data.

Overall, Business rules for POS systems are important for businesses to ensure the process in a consistent and accurate way and to make sure that all the data produced by the system is reliable and meaningful.

2. CONSTRAINTS

There are several constraints of existing point of sale (POS) systems, including:

Limited scalability: Many POS systems are designed for small businesses and may not be able to handle the needs of larger businesses.

High cost: Some POS systems can be quite expensive, especially if they include additional features like inventory management or customer relationship management.

Limited flexibility: Some POS systems are not very customizable and may not be able to adapt to the specific needs of a particular business.

Complex set-up: Some POS systems can be difficult to set up and configure, which can be a problem for businesses that don't have IT personnel on staff.

Limited integration: Some POS systems do not integrate well with other software or systems that a business may be using, such as accounting or inventory management systems.

Outdated hardware: Traditional POS system are often dependent on specific hardware, and upgrading these hardware can be costly and time-consuming.

Security risks: POS systems typically handle sensitive financial data, and if they are not properly secured, they can be vulnerable to data breaches and other security risks.

2.3 New System

2.3.1 Software requirement specification (SRS)

FUNCTIONAL REQUIREMENTS:

Here are some functional requirements for a mobile-based point of sale (market sales management) system:

Off-line functionality: The system should be able to work off-line in case of internet connectivity issues, and should be able to sync transactions once the device is back on-line.

Product management: The system should allow for the creation and management of products, including the ability to add, edit, and delete items.

Inventory management: The system should track inventory levels and generate alerts when stock is running low.

Reporting and analysis's: The system should generate sales reports, inventory reports, and other analysis's that can be used to track performance and make business decisions.

Customer management: The system should allow for the creation and management of customer records, including the ability to view customer history and purchase data.

User-friendly interface: The system should be easy to navigate and have an intuitive interface for users with different levels of technical expertise.

Mobile compatibility: The system should be optimized for use on mobile devices, including smart phones and tablets.

Integration with other systems: The system should be able to integrate with other systems such as accounting software or e-commerce platforms for smooth and automated transactions

Send Recite Via Sms: our system will let you send recite Sms to reduce use of paper the cost match also will keep customer from losing the receipt

Scan Both Bar-Code And Qr-Code: the app must be able to scan code that mostly used on products (bar-code and Qr-code)

Calculate Total Price: to help sealer from using calculator the app must be able to calculate total price.

NON-FUNCTIONAL REQUIREMENTS:

Compatibility: The market sales management system must be compatible with both Android and iOS mobile devices.

User interface: The market sales management system must have a user-friendly interface that is easy to navigate and understand.

Security: The market sales management system must provide secure storage and transmission of sensitive customer and financial information.

Scalability: The market sales management system must be able to handle a large number of transactions and customers.

Data Export: The market sales management system must provide options for exporting data, such as sales reports and customer information, in a variety of formats.

USER OF THE SYSTEM

For every system there is user or actor that is part of the system with it the system can't function properly those actors are

Sealer: the main actor of the system is the sealer which mostly gain from this system he interact with the system through the app and manage his store.

Customer(Buyer): is the second main actor of this system which can do only one thing using the app pay via mobile

GOV agent: is the government employee hows task is to view the pricing of item and tax information of users.

Admin: is also the manager and all privilege holder of the system that also view every activity on the system.

2.3.2 Essential Use case Diagram

Essential use case diagrams for a Market sales management system can be used to map out the steps that customers and merchants need to take in order to complete a sealing and buying actions also view the class of GOV agent and admin. Generally, these diagrams will include the following components:

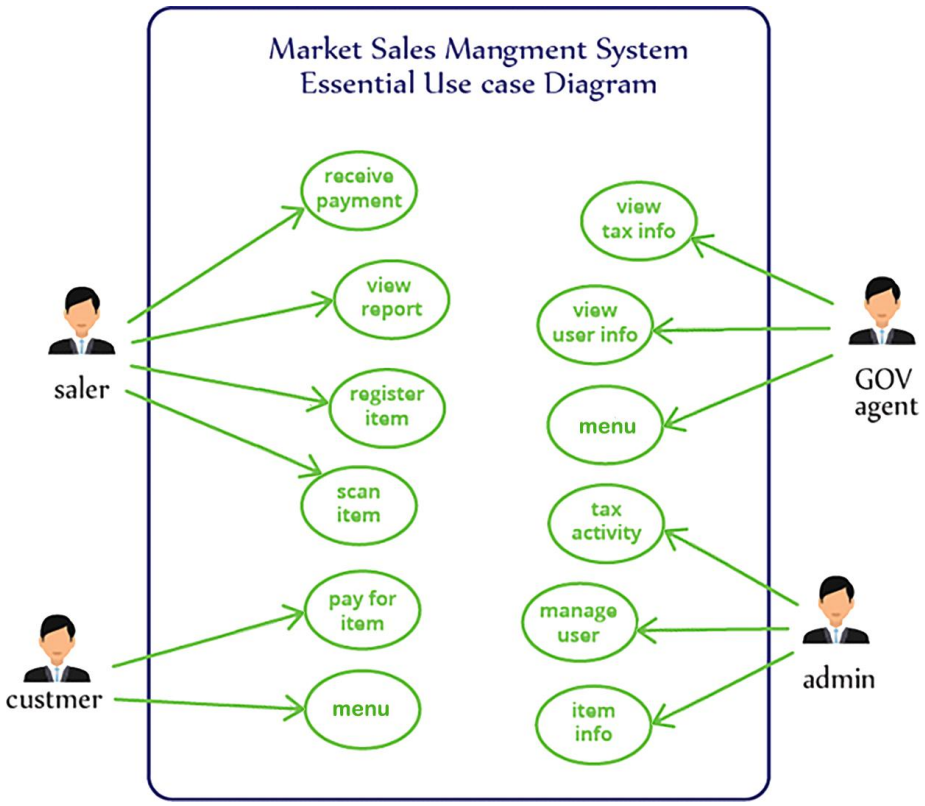


Figure :2.1 Essential Use case Diagram

2.3.3 System Use case diagram

A system use case diagram is a visual representation of the relationships and interactions between different elements of a system. It is a type of UML diagram that focuses on the system's use cases and the different actors that interact with the system. A system use case diagram for a market sales management system will include the following elements:

Actors: Customers, Sale , admin and Gov agent

Use Cases: Scanning items, Payment processing, and Reporting

Connectors: Connecting customer with sale, customer with vendor, and sale with vendor

Dependencies: Database, Payment gateway, and Inventory Management

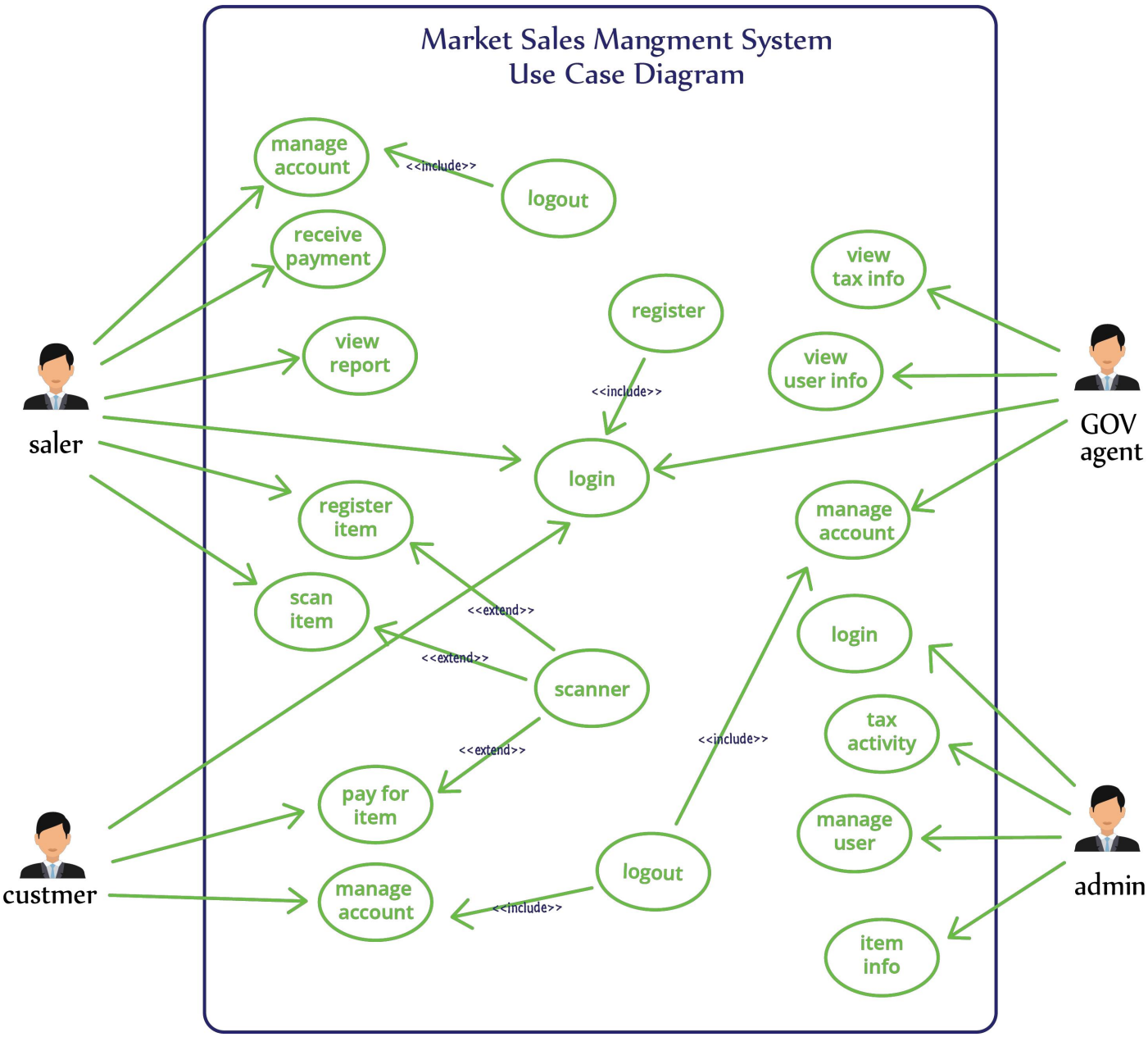


Figure:2.2 System Use case diagram

2.3.1 Use Case documentation

use case documentation should include a description of the use case, a description of the features, a description of any special considerations, and examples of how the system can be used. When we cam e to market sales management system the in order to use the system user must have android smart phone with medium camera quality so that user can scan accurately.

The system have many features like mobile payment system and ability to integrate with other payment systems like CBE birr and Amole in order to use those features user must provide basic information like type of payment method and there payment PIN also user must be user of either CBE birr or Amole to pay via app other wise they can pay in cash

For security propose the user must always use there PIN to use the app in-case person with out authorization wants to access the app also the app will ask for your CBE birr or Amole Pin this information will not be shared with any other third party the info will be stored locally on the phone.

Finally both sealer and buyer must use the same payment system to my via app for example if the sealer is using CBE birr the buyer also must use CBE birr.

attribute	Purpose
Use Case ID	01
Use Case Name	Register
Description	In this Use case user can register
Actor	Custmaer and sealer
preconditon	In-order to use this function the app must be installed on user phone and internet must be enabled
Basic flow	-user click on register -enters name and PIN and other info -the app store on db
Post conditon	User must be registered inoreder to use the app
Basic course of action	Enter or accessing the app
alternative course of action	User input invalid or no connection

Table 2.1

attribute	Purpose
Use Case ID	02
Use Case Name	Login
Description	In this use case user can login using user-name and password
Actor	Custmar ,sealer and agent
preconditon	In-order to login in to the app user must enter correct user-name and password
Basic flow	-open the app -enter user name and password -login
Post conditon	The user
Basic course of action	The home page should be displayed after success registration
alternative course of action	Incorrect user-name or password

Table 2.2

attribute	Purpose
Use Case ID	03
Use Case Name	Paying for item
Description	User(buyer) can pay for item via his mobile
Actor	Customer
preconditon	In order for buyer to use payment via mobile he must have CBE birr account
Basic flow	-enter CBE birr PIN -scan sale QR code or info

	-approve payment
Post conditon	User will receive recite from saler
Basic course of action	Paying via mobile app
alternative course of action	Insufficient balance or no connection

attribute	Purpose
Use Case ID	04
Use Case Name	Receive payment
Description	Saler can receive payment via mobile app
Actor	sealer
preconditon	In-order for user to receive payment via mobile he also must have CBE birr account
Basic flow	-after scanning item for sale clicks on gon qr code button -receive CBE birr SMS
Post conditon	Network must be available
Basic course of action	Receive payment via mobile app
alternative course of action	SMS my net be received

Table:2.3

attribute	Purpose

Use Case ID	05
Use Case Name	View Report
Description	User can view report about the store
Actor	sealer
preconditon	First user must provide this info for the app
Basic flow	-open the app -open analytics
Post conditon	If user didn't use the app before there will be no report
Basic course of action	Manage store
alternative course of action	No report

Table:2.4

attribute	Purpose
Use Case ID	06
Use Case Name	Scan item
Description	User can register item by scanning
Actor	sealer
preconditon	User must enable acccess for app
Basic flow	-open scanner -scan item and enetr info
Post conditon	Qr code must be visible
Basic course of action	Store item
alternative course of action	Scanner can't scan

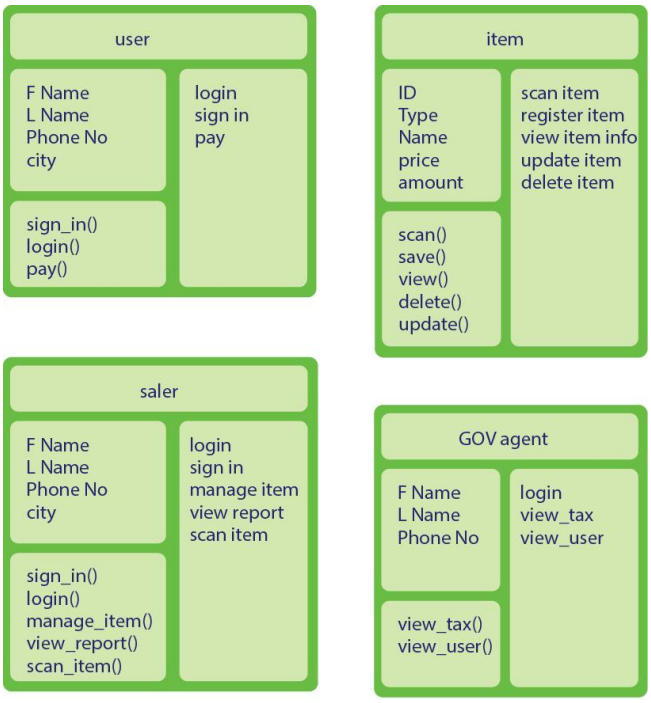
Table:2.5

attribute	Purpose
Use Case ID	07
Use Case Name	View tax info
Description	Gov agent can view tax info and other user infos
Actor	Gov Agent
preconditon	User must have privilage
Basic flow	-open the app -open view info
Post conditon	No info to view
Basic course of action	Manage taxs
alternative course of action	No data

Table:2.6

2.3.4 Key abstraction with CRC analysis

A CRC card is a standard index card that has been divided into three sections, one indicating the name of the class the card represents, listing the responsibilities of the class, and the third listing the names of the other classes that this one collaborates with to fulfil its responsibilities.



2.3.5 Sequence diagram

A sequence diagram for a market sales management system would show the interactions between the different components of the system, including the mobile device, the database. The diagram would show the sequence of messages exchanged between these components, and the order in which they occur. making it easier to understand and troubleshoot any issues that may arise.

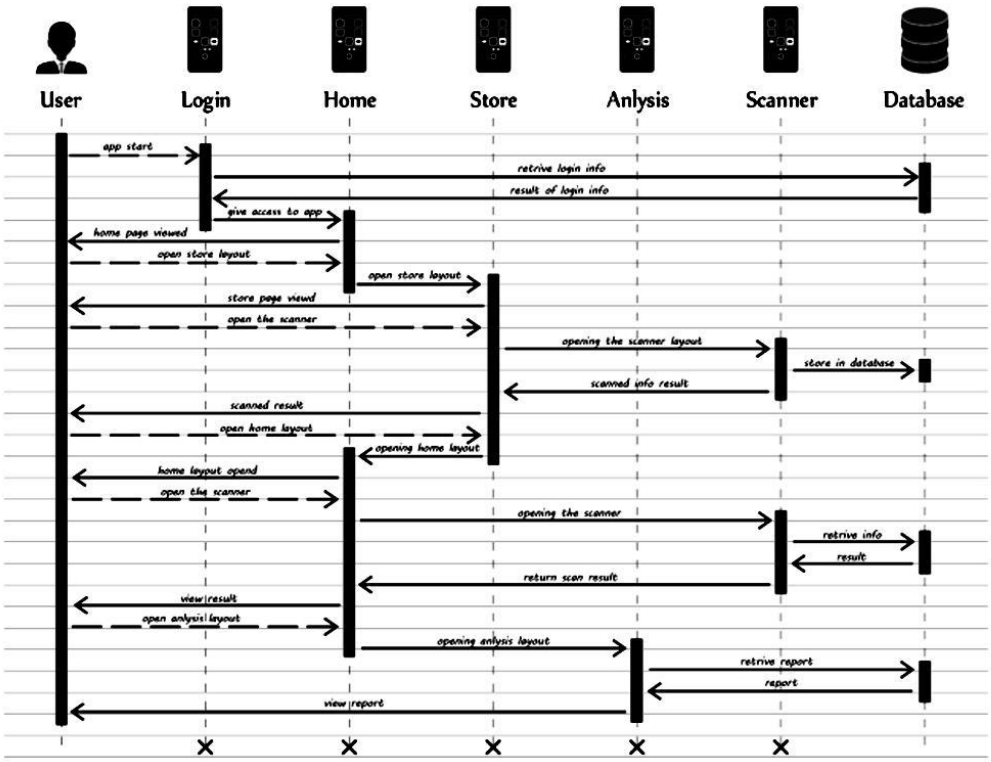


Figure:2.4 Sequence diagram

On the above sequence diagram shows how sealer(the user) scan and register on the data base for later use and again the sealer try to scan and retrieve stored info.

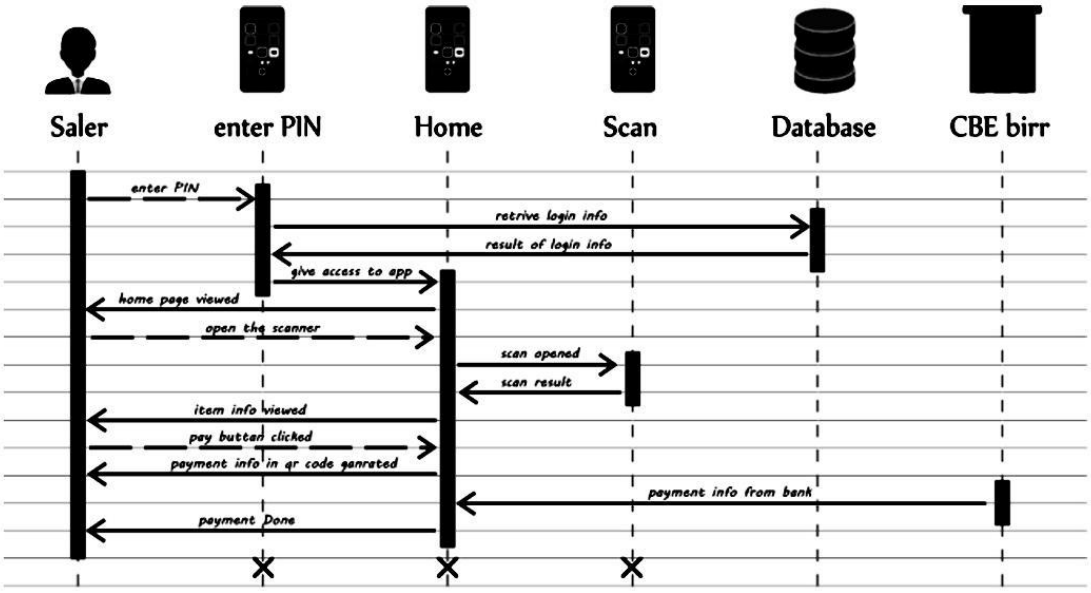


Figure:2.5 Sequence diagram

on the above diagram shows basic step taken for sealer to receive payment via mobile it includes scanning an item and getting the stored info about that item then the sealer click button the app proper qr-code that include the payment information so that the buyer can scan it.

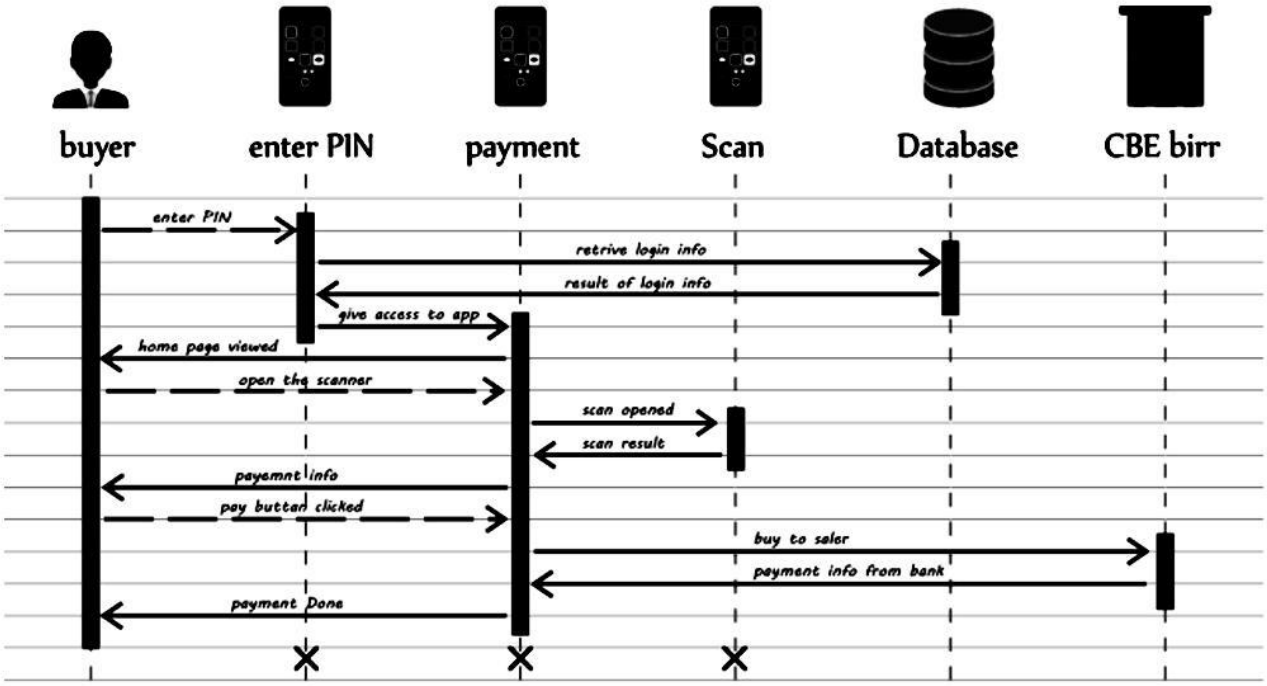


Figure:2.6 Sequence diagram

on the above diagram shows the sequence activity to pay to an item using app and CBE birr account and the CBE Birr only included to show the prose's how transaction.

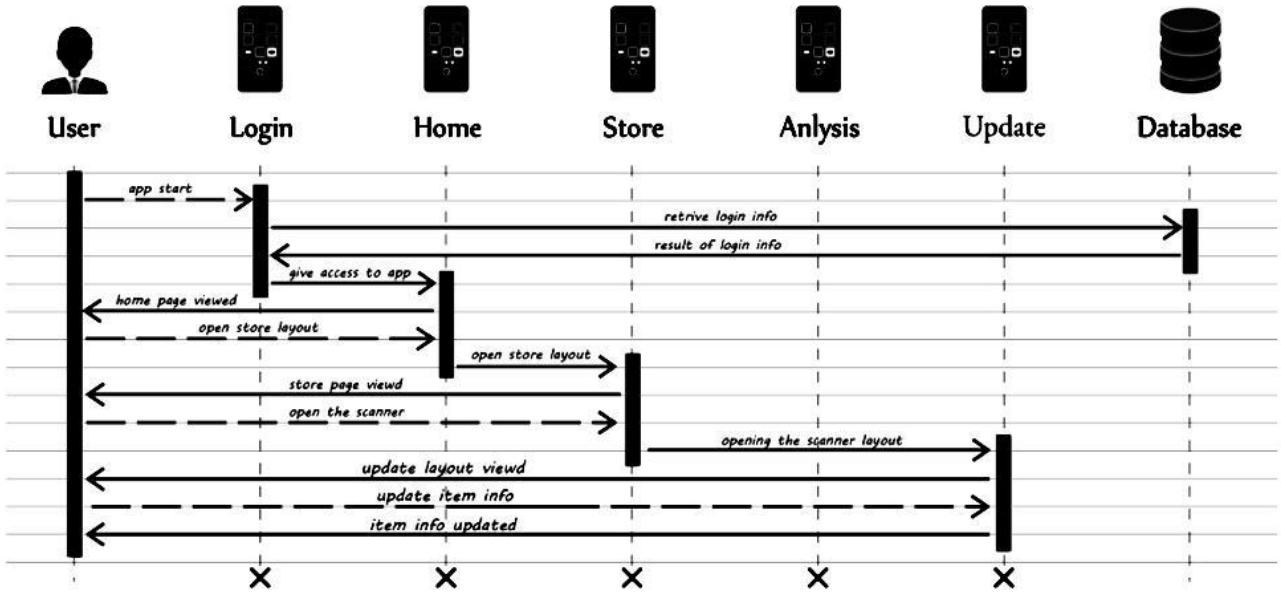


Figure:2.7 Sequence diagram

Above diagram show how user update item info.it includes login layout database also home layout, store layout , update layout and also the user

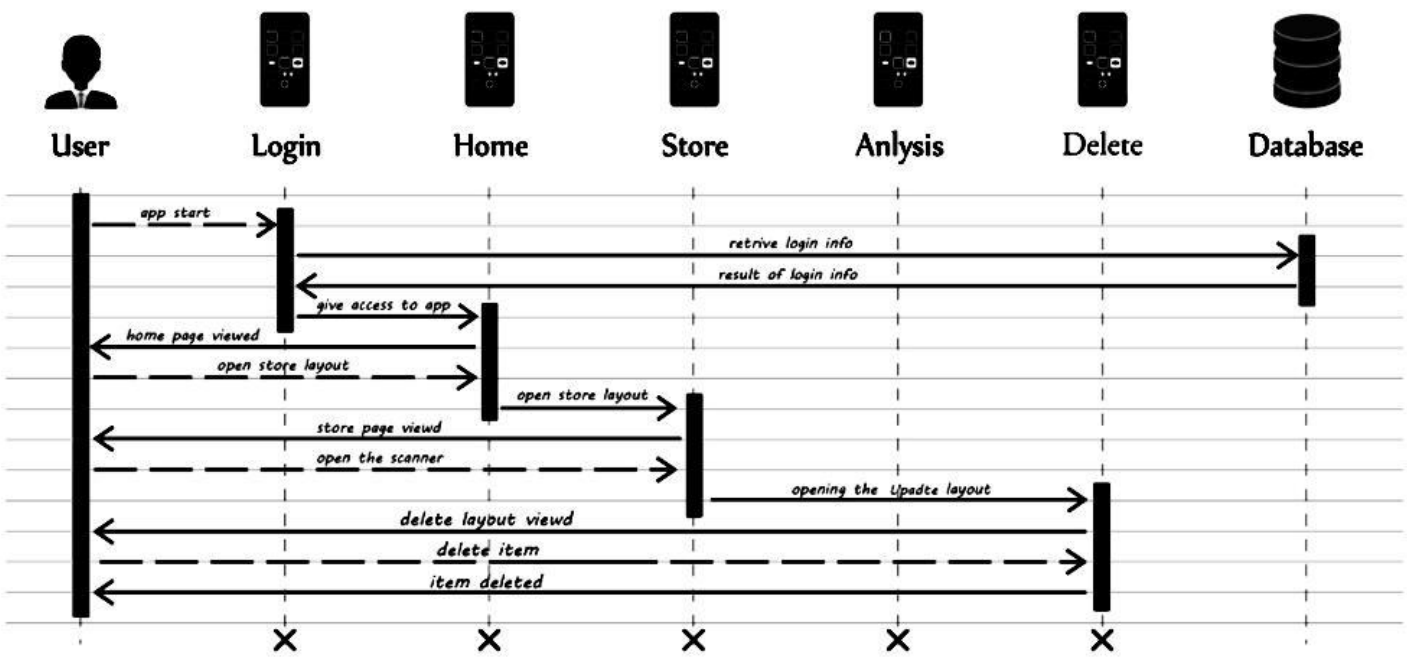


Figure:2.8 Sequence diagram

Above diagram shows how user can delete item from the app data base this my happen in same case like if user want to remove item form the store.

2.3.6 Activity diagram

An activity diagram for a market sales management system system would show the different steps involved in a transaction, from the customer initiating the purchase to the final receipt being issued. Additionally, it may include activities such as customer login, account creation, and order history management. The following diagram shows how product info registered and scanning product and retrieve data is done.

The below activity diagram show how sealer start app and scan item and store it and scan it again and the system search for that item and display result if found else it show not found.

In this path app starts and try to validate user from local database if user exists the next activity start else the it return to login activity

In this part user views the home page and click on store navigate button the store activity viewed then user click on add button Then the scanner activity is views

After scanner starts the result will be displayed then the user saves and goes to next activity

And again the scanner starts and search for the scanned id

Then if id found result will be displayed else the not found message will be displayed

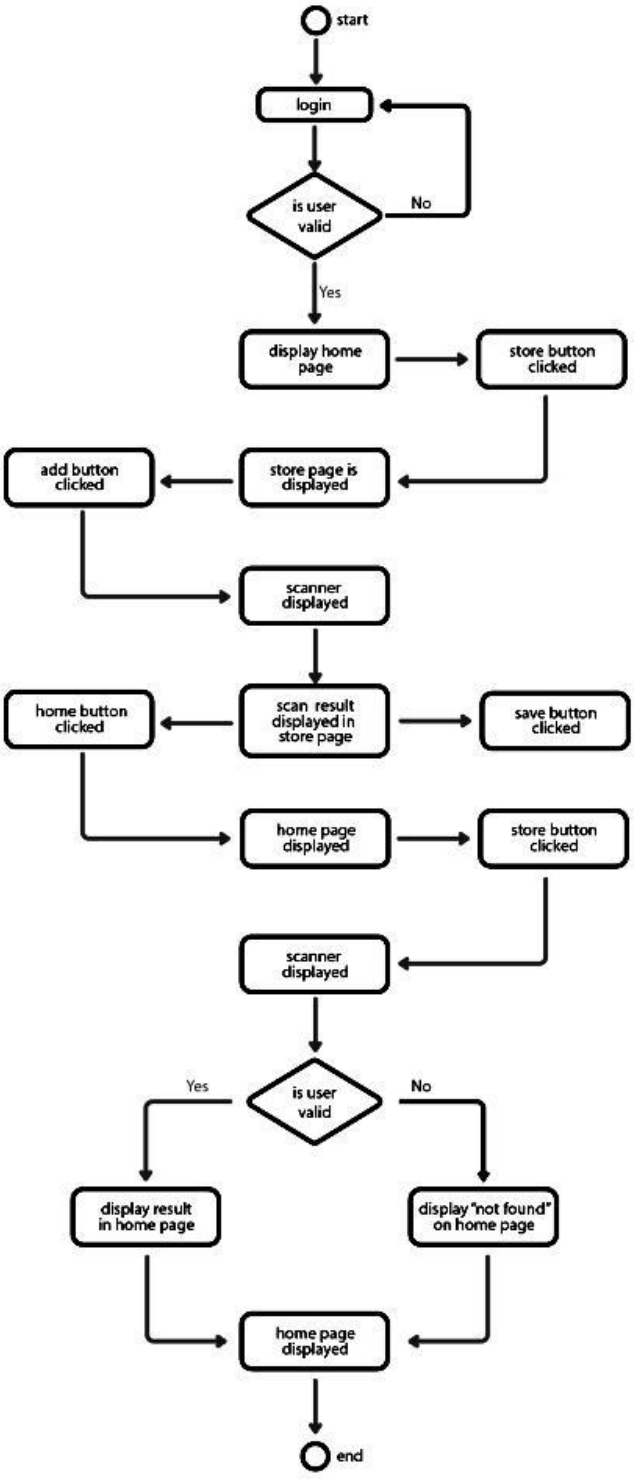


Figure 2.9 activity diagram

In this activity the sealer can receive payment either via cash or via mobile and it start form scanning item code then info displayed if found then payment starts.

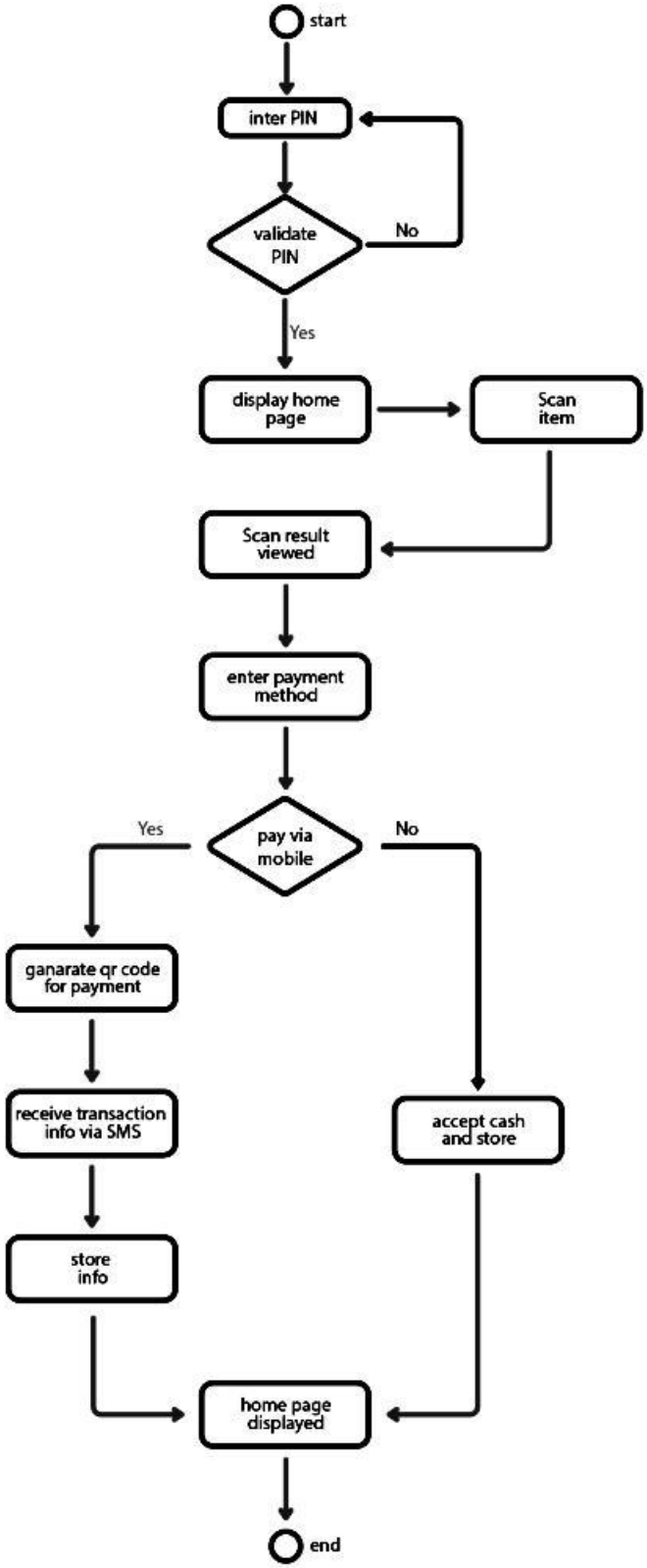
The activity start user enter pin if valid goes to next activity else return back to enter pin

Home page displayed and user click add button scanner start and scan item code

The user asked for payment method if user enter via mobile the next activity on the right will start else on the left

I user goes to activity on the left user accept payment via cash and save info if on right activity App create qr code for payment so the buyer can scan it then payment notification will be received

Then the info will be stored and payment is done



Figur 2.10 activity diagram

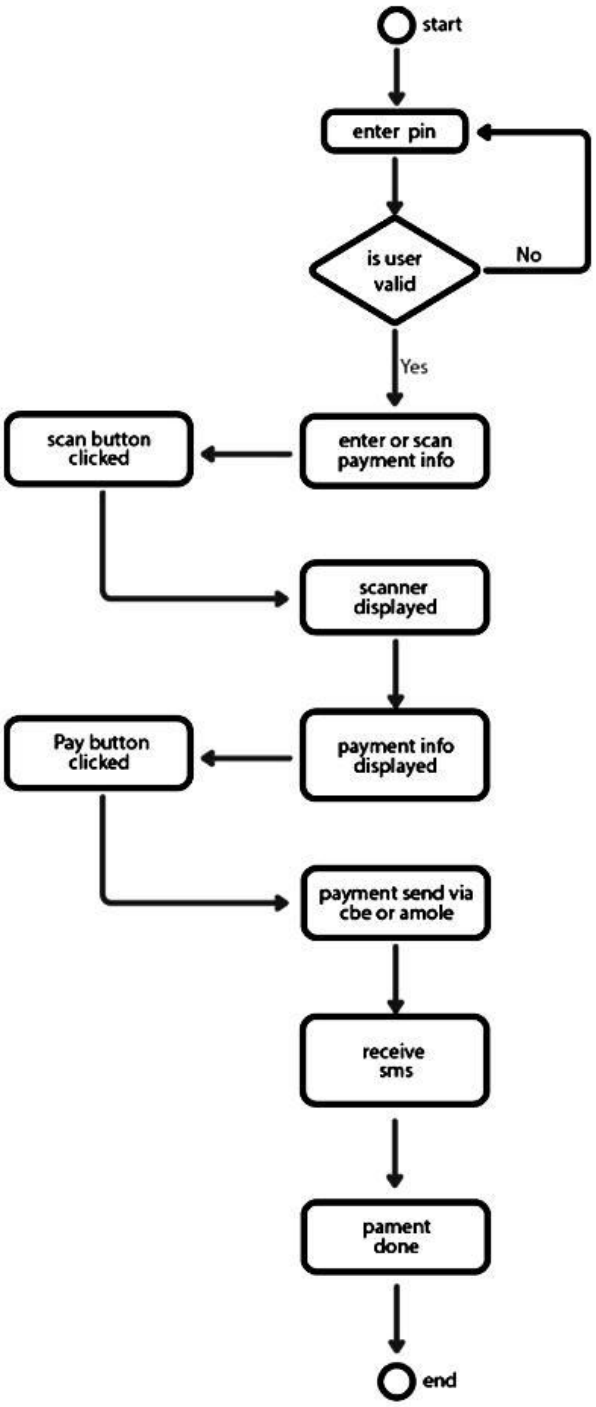
In in this activity diagram user can use the app to pay for item via CBE birr of amole on either of them it must be the same to what the buyer use. Then the app must first register payment method and PIN then the app replace this information to the dialler and dials and receive SMS back.

The activity start user enter pin if valid goes to next activity else return back to enter pin

Then the scan info written into USSD code Like this:
Dial="*847#2*1*1*"+price+"*"+pin+"*1#"
And dials

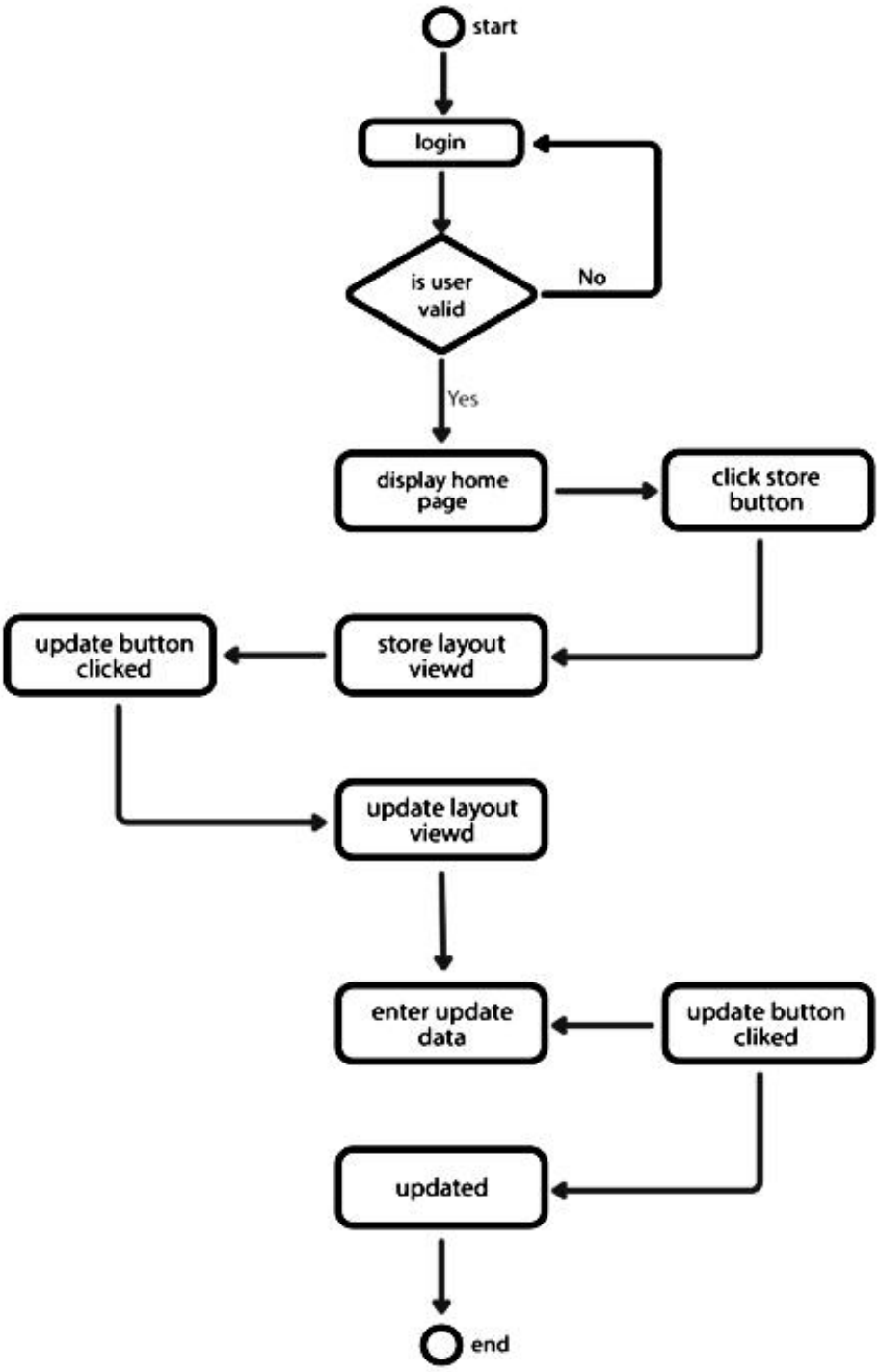
Then when pay button clicked the app dial the USSD code

Then the app read sms from the bank and register info



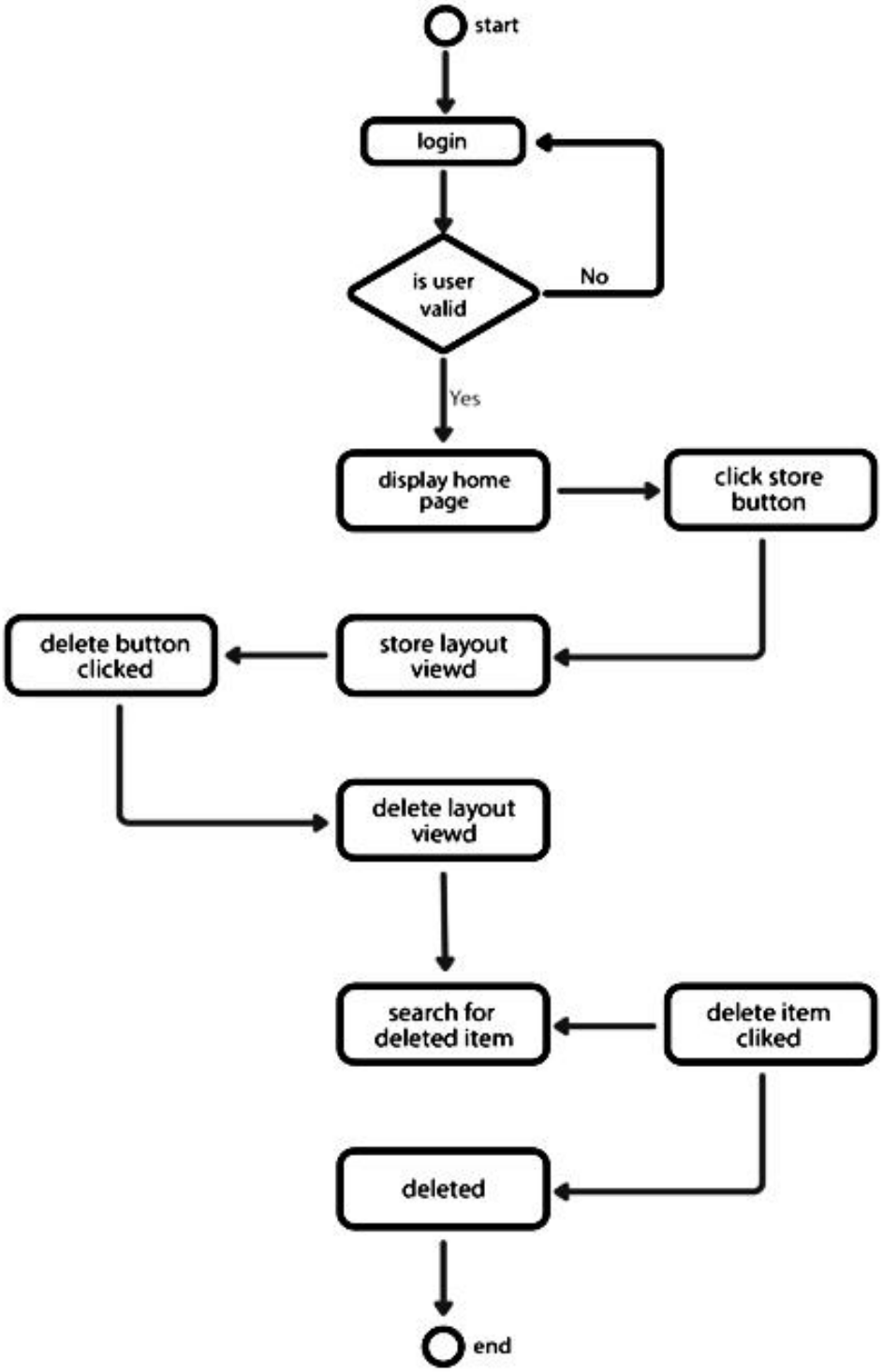
Figur diagram 2.11 activity diagrm

In this layout user can update item info like price, name and product type in-case same product price changed or name changed or other info changed. First user validated in login layout then home layout displayed then user click store button then update button clicked then update layout viewed then user inter item info then click enter then the item info is updated.



Figur 2.12 activity diagram

In this layout user can delete item that previously register by interning the store layout and then delete layout this enable user to remove item that not found on the store. First as other activity user is validated if valid processed to home layout the the user click store button the app display store layout then the user click delete button from the buttons then the delete layout is displayed



Figur diagram 2.13 activity diagram

2.3.7 Conceptual modelling:Class diagram

A conceptual model for Market sales management system would outline the key components and their relationships in the system. This can include the different types of devices that can be used (such as smart phones or tablets), the software that runs on these devices, the payment methods that are accepted, and the various data flows that occur within the system.

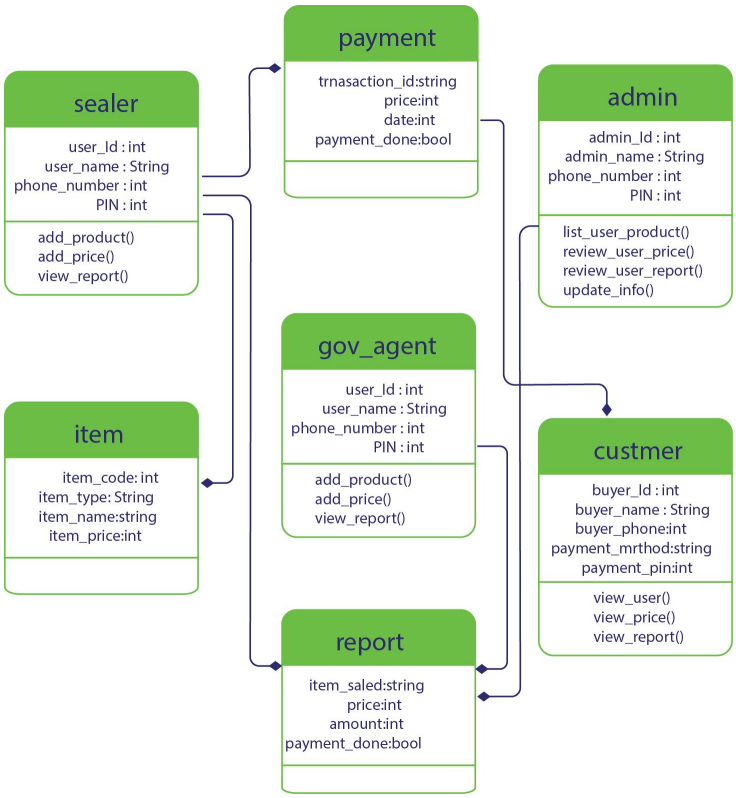


Figure:2.14 Conceptual modelling

2.3.8 User Interface Prototyping

User interface (UI) prototyping is an important step in the development of a market sales management system. A UI prototype is a visual representation of the user interface that simulates the functionality of the final product. It allows designers and developers to test and iterate on the design before the actual development of the system begins.

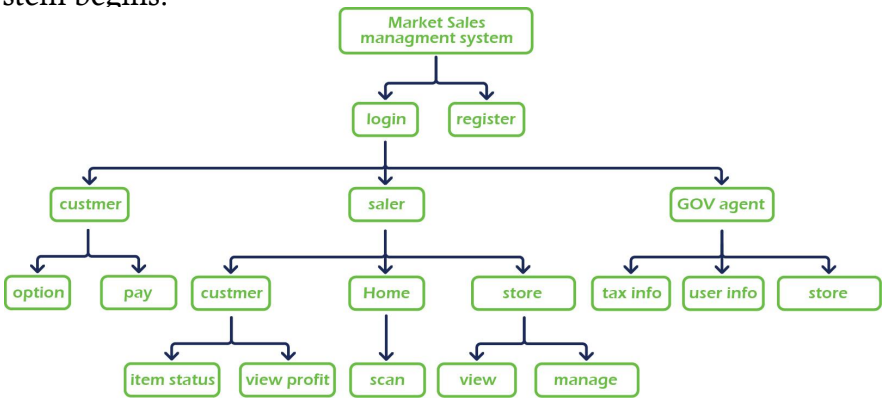


Figure:2.15 User Interface Prototyping

2.3.9 Identifying change cases

A change case for a market sales management system could include:

Improved mobility and flexibility for sales associates: With a mobile market sales management system, associates can make transactions and assist customers from anywhere in the store, rather than being tethered to a fixed register.

Increased efficiency and productivity: By eliminating the need to wait in line at a register, customers can move through the store more quickly, which can lead to more sales and a higher overall throughput.

Improved inventory management: A mobile market sales management system can integrate with inventory management software, allowing associates to quickly check stock levels, place orders, and transfer items between locations.

Cost savings: A mobile market sales management system can be less expensive to implement and maintain than a traditional market sales management system, as it eliminates the need for bulky, expensive hardware.

Increased data collection and reporting: A mobile market sales management system can provide valuable data on customer behaviour, inventory levels, and sales, which can be used to make informed business decisions.

Chapter 3: System Design

3.1 Purpose and goals of design

The goal of system design is to allocate the requirements of a large system to hardware and software components. The system design activity starts after the system requirements analysis has been completed. The starting point is a system requirements specification and the documentation created during the system requirements analysis. The transitioning from the system requirements analysis to system design is usually a non-trivial exercise as there is seldom a single, obvious solution that presents itself to the designers. Sometimes, the transition is made easier by customer documentation that includes a partitioning. The first step in the system design phase is the partitioning activity where a large system is decomposed into smaller entities to reduce the overall complexity, and to allow subsequent development of architectural modules by several teams of developers. The primary purpose of partitioning is to decompose the requirements of a large real-time system into smaller, manageable units that can be designed as architectural entities by teams of hardware and software engineers. The system design phase consists of the decomposition of a large system into partitions that can execute as independent entities in a distributed configuration. The configuring step is performed in close concert with the partitioning.

3.2 Class Modelling Diagram

A class modelling diagram is a type of diagram used in object-oriented programming to represent the classes and their relationships within a system.In a class modelling diagram, classes are represented as boxes that contain the class name and its properties (attributes) and methods (operations). The properties and methods are typically listed inside the box, with the properties above the methods. Overall, class modelling diagrams are useful for visualizing the design of a system and understanding how the different classes and their relationships work together to perform the system's functionality.

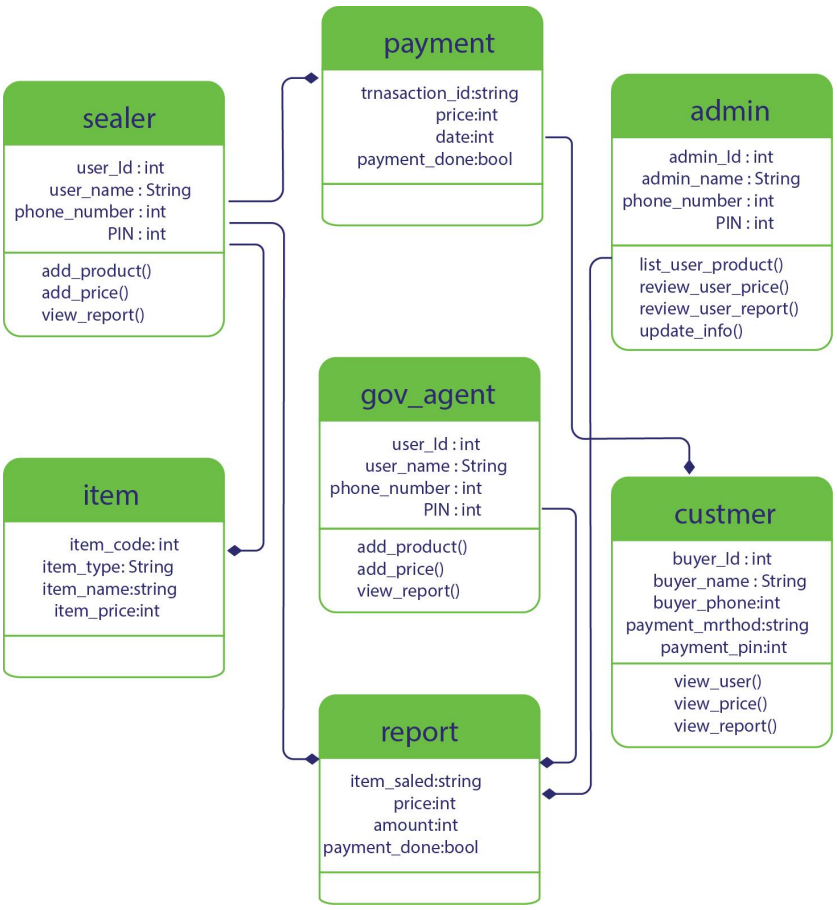


Figure:3.1 Class Modelling Diagram

3.3 Current software architecture

The Current software architecture of a POS system typically includes the following components:

User Interface: This is the front-end of the system where the user interacts with the system to perform transactions and manage inventory.

Database: This component stores all the information related to transactions, inventory, customers, and employees.

Business Logic: This component contains the rules and algorithms that govern the system's behaviour. It processes the transactions and updates the inventory accordingly.

Payment Processing: This component is responsible for handling all the payment-related tasks. It communicates with payment gateways to process credit and debit card transactions.

Reporting: This component generates reports such as sales reports, inventory reports, and customer reports.

Communication: This component is responsible for communicating with external systems such as a CRM or ERP system, or with other POS systems in a multi-location set-up.

These components interact with each other to perform the various functions of a POS system, such as processing transactions, managing inventory, and generating reports. Some market sales management systems are cloud-based, which means that the software and data are hosted on a remote server and can be accessed from any device with an internet connection.

3.4 Proposed software architecture

A Market sales management system typically includes the following software architecture components:

User Interface: A user-friendly interface for customers to make purchases, view their transaction history, and access other features of the system.

Back-end Server: A server that handles all the business logic, such as inventory management, reporting, and customer management. This server communicates with the mobile application through APIs.

Database: A centralized system that stores all the data related to the system, including customer information, transaction history, and inventory.

Analytic: A tool to analyse the data and generate reports for business insights.

Cloud-based infrastructure: A scalable and reliable infrastructure that can handle a large number of transactions and support the system's growth.

Integration with other systems: The ability to integrate with other systems, such as accounting software, CRM, and ERP systems, to ensure data consistency and accuracy.

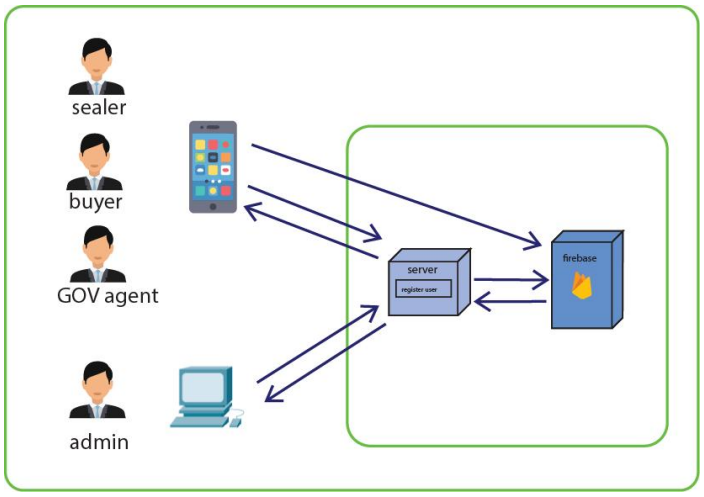


Figure:3.2 Proposed software architecture

3.4.1 Subsystem Decomposition

A market sales management system can be decomposed into several subsystems, including:

Hardware subsystem: This includes the physical components of the Market sales management system such as the mobile device (e.g. tablet or smart-phone).

Software subsystem: This includes the software that runs on the mobile device and manages the POS functionality, including the user interface, payment processing, inventory management, and reporting.

Payment processing subsystem: This handles the communication with payment processors to authorize and complete transactions.

Inventory management subsystem: This tracks and updates the inventory levels and pricing information.

Reporting subsystem: This generates and provides sales and financial reports.

Communication subsystem: This manages the communication between the mobile device and the back-end systems, such as the inventory management system and the reporting system.

Network subsystem: This manages the communication between the mobile device and the internet.

3.4.2 Component diagram

Component diagrams are typically used during the design phase of software development and can be helpful in identifying potential issues before the code is written. They can also be used to document and communicate the design of a system to stakeholders.

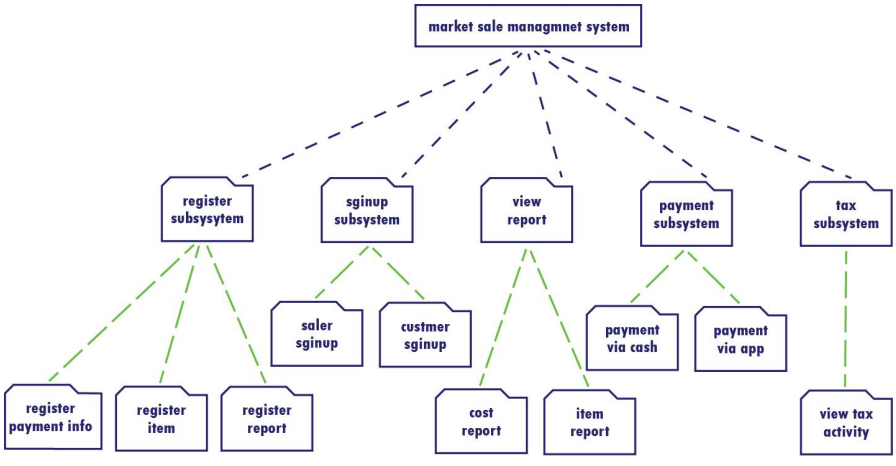


Figure:3.3 Component diagram

3.4.3 Deployment diagram

Deployment digram for Market sales management system is a simple and efficient way to manage and deploy your market sales management system. This tool can help you plan and schedule your market sales management system deployments, track the status of your systems, and manage your finances.

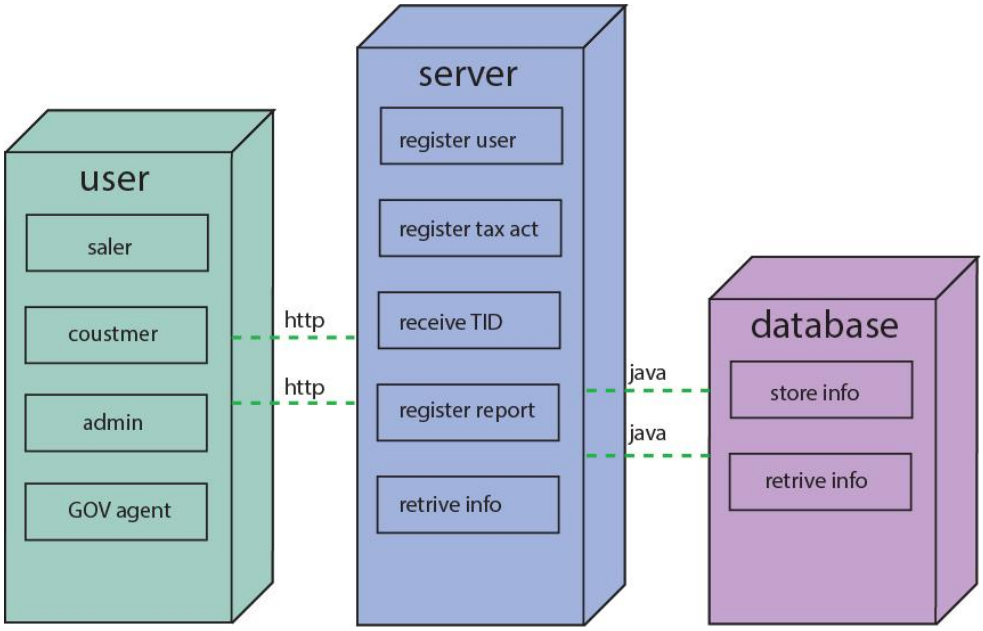


Figure:3.4 Deployment diagram

3.4.4 Persistence modelling for object-oriented database

Persistence modelling for an object-oriented database for a Market sales management system involves defining the data elements that need to be stored in order to support the system¡¯s functionality. This includes the objects, their attributes, and the relationships between them. Generally, the modelling process includes the following steps:

Define the objects: this involves determining the data elements that need to be stored in the database in order to support the system¡¯s functionality.

Define the attributes: this involves specifying the attributes of each object that will be stored in the database.

Define the relationships: this involves specifying the relationships between the objects that will be stored in the database.

Define the database structure: this involves creating the database structure that will store the objects, their attributes, and their relationships.

Define the database queries: this involves creating the database queries that will be used to retrieve the data from the database.

3.4.5 Access control and security

Access control and security for Market sales management systems can be implemented in several different ways. One way is to use a secure access control system that utilizes biometric authentication such as fingerprint scanning, iris scanning, or facial recognition. This can help to ensure that only authorized personnel can access the system. Other methods of access control and security can include using encryption, anti-virus software, and secure networks.

actor	class	operation
sealer	Manage item	Store(),delete(),update() and search()
	View report	View_item_info(),view_cost_info()
customer	Pay for item	Pay()
	Scan payment code	Scan()
admin	View user info	List_user()
	View price info	List_price()
GOV agent	View tax info	View_tax()
	View sealer pricing	View_Price()

3.4.6 Boundary condition and exception handling

. In order to ensure the system is robust and secure, boundary conditions and exception handling should be implemented. Boundary conditions are used to ensure that the system can handle unexpected input or conditions. This can include validating user input, checking the data type of variables, and ensuring that the system can handle requests outside of the expected range. Exception handling is used to handle errors and unexpected conditions. This can include catching errors, logging them, providing informative error messages, and gracefully recovering from the error

3.5 User Interface Design

User interface design for mobile point of sale systems is a complex task that must take into account the needs of both customers and merchants. To ensure a smooth user experience, design features should be streamlined, intuitive, and lead users through the checkout process intuitively. Market sales management systems should consider items such as intuitive navigation, one-click checkout, and integration with payment processing and loyalty programs. Additionally, a well-designed user interface should have an attractive visual style that encourages customers to shop.

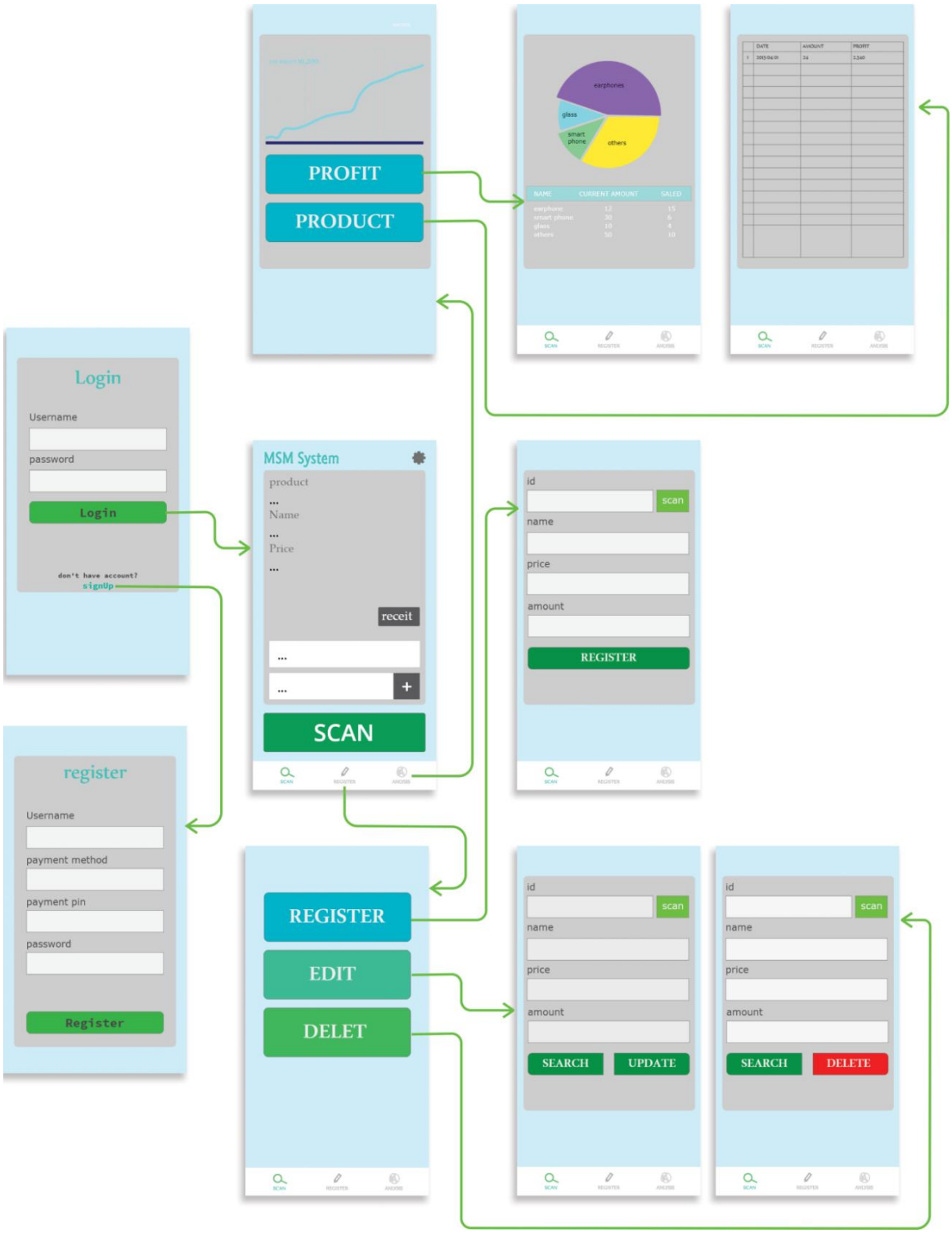


Figure:3.5 User Interface Design

Chapter 4: Implementation and Testing

4.1 Implementation

Code for scanning the Qr code or barcode using the zxing library

```
private void scanCode(){
    IntentIntegrator integrator = new IntentIntegrator(this);
    integrator.setCaptureActivity(CaptureAct.class);
    integrator.setOrientationLocked(true);
    integrator.setPrompt("Scanning...");
    integrator.initiateScan();
}
```

code for showing notification when items sold out

```
public void show_notification(String itemString item_name){
    new Handler().postDelayed(new Runnable() {
        @Override
        public void run() {
            CookieBar.build(getActivity())
                    .setTitle("Manage Sales")
                    .setMessage("The Item "+ item + " (" + item_name + ") " + "is sold out!")
                    .setBackgroundColor(Integer.parseInt("000000"))
                    .setCookiePosition(CookieBar.TOP)
                    .setDuration(5000)
                    .show();

        }
    }500);
}
```

code for generateing qr code

```
MultiFormatWriter multiFormatWriter = new MultiFormatWriter();
try {
    float tax = 0;
    BitMatrix bitMatrix =
multiFormatWriter.encode(r_phone+"><"+Tid+"><"+hold_item+"><"+hold_price+"><"+tax
BarcodeFormat.QR_CODE11151127);
    BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
    Bitmap firstImage = barcodeEncoder.createBitmap(bitMatrix);
    Bitmap secondImage = null;
    try {
        InputStream bitmap1=getActivity().getAssets().open("qr_frame.png");
        secondImage = BitmapFactory.decodeStream(bitmap1);
    } catch (IOException e1) {
        e1.printStackTrace();
    }
    Bitmap mergedImages = createSingleImageFromMultipleImages(firstImage secondImage);
    qr_image.setImageBitmap(mergedImages);
} catch (WriterException e) {
    e.printStackTrace();
}
```

Code for detecting the if internet and if internet is not detected it shows red also if internet is detected it show green and hide.

```
if (activeNetworkInfo != null && activeNetworkInfo.isConnected()) {
    // There is an active network connection
    layout.setBackgroundColor(ContextCompat.getColor(((Activity)context)R.color.green));
((Activity)context).getWindow().setStatusBarColor(ContextCompat.getColor(((Activity)conte
xt)R.color.green));
    i_status.setText("Internet Connected");

    new Handler().postDelayed(new Runnable() {
        @Override
        public void run() {
            layout.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT 0));

((Activity)context).getWindow().setStatusBarColor(ContextCompat.getColor(((Activity)conte
xt)R.color.white));
            decor.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
        }
    }500);
} else
{layout.setBackgroundColor(ContextCompat.getColor(((Activity)context)R.color.dark_red));

((Activity)context).getWindow().setStatusBarColor(ContextCompat.getColor(((Activity)conte
xt)R.color.dark_red));
    i_status.setText("No Internet connection");
    layout.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT 30));
    decor.setSystemUiVisibility(0);
}
```

Code for prosessing payment by using dialer

```
private void dialNumber(String number) {

    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.setView(view);
    alert.setCancelable(false);
    final AlertDialog dialog = alert.show();

    close_pay.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            dialog.dismiss();
        }
    });

    TelephonyManager manager = (TelephonyManager) getSystemService(TELEPHONY_SERVICE);
    if (ActivityCompat.checkSelfPermission(this Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED) {
        return;
    }
    manager.sendUssdRequest(number new TelephonyManager.UssdResponseCallback() {
        @Override
        public void onReceiveUssdResponse(TelephonyManager telephonyManager String request
CharSequence response) {
            super.onReceiveUssdResponse(telephonyManager request response);

            if("Please try again".equals(response.toString())){
                progressBar.setVisibility(View.GONE);
                alert_text.setText("PLEASE TRY AGAIN");
            }else if("Your request is being processed. Please wait for CBE Birr confirmation
message.".equals(response.toString())) {
                alert_text.setText("PAYMENT SENT");
                progressBar.setVisibility(View.GONE);
                done_img.setVisibility(View.VISIBLE);
            }
        }
        @Override
        public void onReceiveUssdResponseFailed(TelephonyManager telephonyManager String
request int failureCode) {
            super.onReceiveUssdResponseFailed(telephonyManager request failureCode);
            alert_text.setText("FAILED");
            progressBar.setVisibility(View.GONE);
        }
    } new Handler());
}
```

4.2 Unit tasting

Unit testing is an essential part of software development, including Android-based Point of Sale (POS) systems. It helps ensure the reliability, functionality, and quality of the codebase. In this context, unit testing involves testing individual units or components of the system in isolation to verify that they function correctly.

When it comes to unit testing an Android-based market sales management system, there are several key aspects to consider:

Testing Frameworks: Android provides built-in testing frameworks like JUnit and AndroidJUnitRunner for running unit tests. These frameworks allow you to write test cases and execute them on Android devices or emulators.

Testable Components: In a market sales management system, there are various components that can be tested individually, such as payment processing, inventory management, receipt generation, and user authentication. Each component should have its own set of unit tests.

Isolation: Unit tests should be designed to run independently of other components, which requires isolating the unit under test from external dependencies. This can be achieved by using mocking frameworks like Mockito or creating test doubles manually.

Test Coverage: It's important to aim for high test coverage to ensure that all critical parts of the market sales management system are adequately tested. This includes covering different scenarios, edge cases, and error conditions that the system may encounter during real-world usage.

Test Data: Unit tests should use specific and controlled test data to ensure consistent and predictable results. This may involve creating test data sets for different scenarios, including valid and invalid inputs, to thoroughly exercise the unit being tested.

Test Automation: Automating unit tests allows for frequent and efficient testing as the codebase evolves. Continuous Integration (CI) tools like Jenkins or CircleCI can be integrated to automatically trigger unit tests on code changes or regular intervals.

Test Reporting: Generating test reports helps track the status and progress of unit tests. Various reporting tools are available, such as Android Test Orchestrator, which provides detailed test results and logs for analysis.

Regression Testing: As the market sales management system evolves, it's crucial to maintain a comprehensive suite of unit tests and perform regular regression testing. This ensures that existing functionalities are not inadvertently broken by new code changes.

By following these best practices for unit testing, you can enhance the stability, maintainability, and overall quality of your Android-based market sales management system. Remember that unit testing is just one aspect of a broader testing strategy, and other testing techniques like integration testing and UI testing should also be considered to provide a comprehensive quality assurance process.

4.3 Functional Test Specifications

Action	Test Procedure	Input	Expected Test Result	Pass/Fail
User login	Enter valid username and password	Enter valid username and password	Successful login	Pass
	Enter invalid username/password	Enter invalid username/password	Login failure	Pass
Add new product	Provide product details	Product details	Product added	Pass
	Provide missing required details	Missing required details	Error message displayed	Pass
Update product details	Select a product and update its details	Valid product ID	Product details updated	Pass
	Select an invalid product and attempt update	Invalid product ID	Error message displayed	Pass
Delete product	Select a product and delete it	Valid product ID	Product deleted	Pass

	Select an invalid product and attempt deletion	Invalid product ID	Error message displayed	Pass
sales report	Specify date range and desired filters	Date range and filters	Report generated	Pass
	Specify an invalid date range	Invalid date range	Error message displayed	Pass
Search product	Enter a valid search keyword	Valid search keyword	Relevant results shown	Pass
	Enter an invalid search keyword	Invalid search keyword	No results displayed	Pass
View sales statistics	Select relevant filters for statistics display	Relevant filters selected	Statistics displayed	Pass
	Select invalid filters for statistics display	Invalid filters selected	No statistics displayed	Pass

Table 4.1

Test 1:

- Test Name: User Login

- Test Description: Verify the user login functionality.

- Purpose of Test: To ensure that users can log in successfully and handle invalid login attempts.

- Testing Objects: User login functionality.

- Test Focus: Validating the login process and error handling.

- Test Process:

  - Starting Condition: The user is on the login screen.

- Input:
    - Valid username and password for successful login.
    - Invalid username or password for login failure.
- Expected Result:
    - For valid credentials: Successful login.
    - For invalid credentials: Login failure or error message displayed.
- Failure Condition: Unexpected login result or error message not displayed.

Test 2:

- Test Name: Add New Product
- Test Description: Verify the functionality to add new products.
- Purpose of Test: To ensure that products can be added to the system accurately.
- Testing Objects: Add new product functionality.
- Test Focus: Validating the addition of new products.
- Test Process:
    - Starting Condition: The user is on the product creation screen.
    - Input:
        - Valid product details for successful addition.
        - Missing required details for error handling.
    - Expected Result:
        - For valid details: Product added successfully.
        - For missing details: Error message displayed.
    - Failure Condition: Unexpected addition result or missing details not handled properly.

Test 3:

- Test Name: Update Product Details

- Test Description: Verify the functionality to update product details.

- Purpose of Test: To ensure that product details can be successfully updated and handle invalid product IDs.

- Testing Objects: Update product details functionality.

- Test Focus: Validating the update process and error handling.

- Test Process:

  - Starting Condition: The user is on the product details screen.

  - Input:

    - Valid product ID and updated details for successful update.

    - Invalid product ID for error handling.

  - Expected Result:

    - For valid product ID: Product details updated successfully.

    - For invalid product ID: Error message displayed.

  - Failure Condition: Unexpected update result or invalid product ID not handled properly.

Test 4:

- Test Name: Delete Product

- Test Description: Verify the functionality to delete a product.

- Purpose of Test: To ensure that products can be successfully deleted from the system and handle invalid product IDs.

- Testing Objects: Delete product functionality.

- Test Focus: Validating the deletion process and error handling.

- Test Process:

  - Starting Condition: The user is on the product details screen.

  - Input:

    - Valid product ID for successful deletion.

    - Invalid product ID for error handling.

  - Expected Result:

    - For valid product ID: Product deleted successfully.

    - For invalid product ID: Error message displayed.

  - Failure Condition: Unexpected deletion result or invalid product ID not handled properly.

Test 5:

- Test Name: Generate Sales Report

- Test Description: Verify the functionality to generate sales reports.

- Purpose of Test: To ensure that sales reports can be generated based on specified criteria and handle invalid date ranges.

- Testing Objects: Generate sales report functionality.

- Test Focus: Validating the report generation process and error handling.

- Test Process:

  - Starting Condition: The user is on the sales report generation screen.

  - Input:

    - Valid date range and filters for successful report generation.

    - Invalid date range for error handling.

  - Expected Result:

    - For valid date range: Report generated successfully.

- For invalid date range: Error message displayed.

- Failure Condition: Unexpected report generation result or invalid date range not handled properly.

Certainly! Here are a few more tests combining similar actions into a single test:

Test 6:

- Test Name: Search Product

- Test Description: Verify the functionality to search for products.

- Purpose of Test: To ensure that products can be searched using keywords and handle invalid search inputs.

- Testing Objects: Product search functionality.

- Test Focus: Validating the search process and error handling.

- Test Process:

  - Starting Condition: The user is on the product search screen.

  - Input:

    - Valid search keyword for successful search.

    - Invalid search keyword for error handling.

  - Expected Result:

    - For valid search keyword: Relevant results shown.

    - For invalid search keyword: No results displayed or error message shown.

  - Failure Condition: Unexpected search result or invalid search input not handled properly.

Test 7:

- Test Name: View Sales Statistics

- Test Description: Verify the functionality to view sales statistics.

- Purpose of Test: To ensure that sales statistics can be displayed based on selected filters and handle invalid filter selections.

- Testing Objects: Sales statistics display functionality.

- Test Focus: Validating the statistics display and error handling.

- Test Process:

  - Starting Condition: The user is on the sales statistics screen.

  - Input:

    - Relevant filters selected for successful statistics display.

    - Invalid filters selected for error handling.

  - Expected Result:

    - For valid filter selections: Statistics displayed accurately.

    - For invalid filter selections: No statistics displayed or error message shown.

  - Failure Condition: Unexpected statistics display result or invalid filter selections not handled properly.

Chapter Five: Conclusion and Recommendation

conclusion

Through this project we have successfully developed a robust and efficient point-of-sale solution that leverages the power of Android technology.

First and foremost the Android market sales management system has greatly enhanced our operational efficiency. With its user-friendly interface and intuitive design our staff can quickly process transactions manage inventory and generate reports with ease. The system's real-time data synchronization capabilities have streamlined our operations enabling us to make informed decisions promptly.

Additionally the Android market sales management system has improved the overall customer experience. The system's fast and accurate transaction processing coupled with its ability to handle payment methods has reduced waiting times and enhanced customer satisfaction. Furthermore the system's integration with our loyalty program and CRM has allowed us to personalize interactions reward loyal customers and build lasting relationships.

Furthermore the Android market sales management system's scalability and adaptability have proven to be invaluable. As our business grows the system can easily accommodate additional registers locations and features. Its modular architecture allows for seamless integration with third-party applications and services ensuring that we can meet evolving customer needs and industry trends.

Throughout the project we have prioritized data security and compliance. The Android market sales management system incorporates robust encryption mechanisms access controls and regular data backups to protect sensitive customer information and ensure regulatory compliance. This commitment to security has fostered trust among our customers and strengthened our reputation in the market.

In conclusion the Android market sales management system project has revolutionized our organization's point-of-sale operations. By leveraging the power of Android technology we have achieved greater efficiency improved customer experiences and ensured scalability for future growth. This project stands as a testament to our commitment to innovation and excellence and we are confident that it will continue to drive our success in the ever-evolving retail landscape.

Recommendation

As we wrap up the current Android-based POS system project it is essential to outline recommendations for future projects in this domain. Here are several key recommendations to consider:

Enhanced Integration Capabilities: In future projects focus on expanding the system's integration capabilities with other business systems and services. This can include integrating with popular e-commerce platforms accounting software customer relationship management (CRM) systems and inventory management tools. Seamless integration will allow for more efficient and accurate data synchronization across different platforms enabling a holistic view of business operations.

Mobile Payments and Digital Wallets: With the increasing adoption of mobile payments and digital wallets it is crucial to incorporate support for popular payment platforms such as telbirr and other local payment solutions. This will provide customers with convenient and secure payment options enhancing their overall experience and keeping pace with the evolving payment landscape.

Advanced Analytic and Reporting: Implement advanced analytic and reporting features in future projects. This will empower businesses to gain valuable insights into sales patterns inventory management customer behaviour and other key performance indicators. Advanced reporting capabilities will enable data-driven decision-making helping businesses identify opportunities optimize operations and drive growth.

Customization and Flexibility: Recognize the diverse needs of businesses operating in various industries and aim to provide a flexible and customize market sales management system. Consider offering a modular architecture that allows businesses to tailor the system to their specific requirements. This could involve customize user interfaces configurable work flows and the ability to add or remove features based on individual business needs.

Cloud-Based Infrastructure: Consider developing a cloud-based infrastructure for future projects. Cloud-based solutions offer scalability easy access to data from multiple locations and automatic backups ensuring business continuity and reducing reliance on local hardware infrastructure. Cloud-based systems can also facilitate seamless updates and maintenance minimizing disruptions to business operations.

Enhanced Security Measures: Continuously invest in security measures to protect sensitive customer data and ensure compliance with industry regulations. Stay updated with the latest security standards implement robust encryption mechanisms conduct regular security audits and stay proactive in addressing emerging threats. This will help build trust with customers and maintain the integrity of the system.

Continuous Improvement and User Feedback: Prioritize user feedback and continuously seek input from customers staff and other stakeholders. Regularly gather feedback on system usability performance and features to identify areas for improvement and address pain points. Actively involve end-users in the development process through beta testing and self-centred design methodologies to ensure the system meets their needs effectively.

By considering these recommendations in future projects we can continue to develop innovative and cutting-edge Android-based market sales management systems that meet the evolving needs of businesses and deliver exceptional value to both merchants and customers.

Reference

https://www.investopedia.com/terms/p/point-of-sale.asp

https://www.scribd.com/document/407789810/documentation-docx

2014 and previous project in our department

Appendx 1

List of question and answer asked for the sealer in mattw town.

Qa what is main problem found in store related to management?

Answer: recording report

Qa what your knowledge about pos system?

Answer: I have seen it in shewa super market and its very costy

Qa how your reviewing and recording store activity?

Answer: using big notebook

Qa how Mach item do you sale per day?

Answer: unknown

Qa do you know bar-code? if yes what is its function?

Answer: yes if have seen it on items but I don't know its function

Qa what is price of vat machine your using?

Answer: about 35000 to 40000 birr

Qa are you intersted in using mobile based market sales management system?

Answer: yes if you develop it