Jordan University of Science and Technology

College of Computer Sciences & Information Technology

"EZ Store"

Inventory management system (IMS)

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor in Software Engineering

By

Own Naim Ali Malkawi (121643)

Zaid Ahmad Ali Daoud (127532)

Amjad Ibrahim Hasan Abu Ghazleh (122015)

Supervised by

Dr.Hamza Al-Kofahi

ACKNOWLEDGEMENT

We Would like to express our special thanks of gratitude to **Dr.Hamza AL-Kofahi** for his guidance ,support and for providing us with all the facility that was required throughout this project.

UNDERTAKING

Our project is Inventory management system it shall be called

“EZ STORE”

is an original work done by undersigned, in partial fulfillment of the requirements for the degree “Bachelor in Software Engineering” at the Software Engineering Department, College of Computer and Information Technology, Jordan University of Science and Technology. All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

1. Own Naim Ali Malkawi
2. Zaid Ahmad Ali Daoud
3. Amjad Ibrahim Hasan Abu Ghazleh

ABSTRACT

The issue of inventory management is very paramount in any store or organisation that has to store moving items (stocks) .Therefore we intend to provide our services to help with the efficiency of moving stocks, These services will be defined as an inventory management system, this project involves the use of application programming interface more specifically Android studio using java language , this is the primary development environment and the outcome will be a program that manages inventories with low cost and high efficiency.

Table of Contents

LIST OF TABLES

CHAPTER 2

CHAPTER 6

LIST OF FIGURES

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 1: Introduction (Heading 1)

With the increase in small shops (shops, mini markets, warehouses ... etc) The need to use something affordable and easy to use has increased; to track goods and warehouses and it should be available for everyone who owns a store, especially in Jordan, as if his shop and his source of livelihood are in his hands.

For this type of small businesses, you are already thinking of reducing the cost as much as possible, because you cannot spend at least a thousand dinars on a system that helps you in the inventory process, you need something easy to use and affordable, all you need is your phone, with a monthly fee that does not exceed 10 dinars, because we are targeting small companies and this means the number of customers will be high, so 10 dinars per month may not be much, but when we look at our target customers who make up more than 66% of the market, we are looking for thousands of customers who will be happy using this system , as it will help them grow and manage their business.

1.1 Overview

This project facilitates the annual inventory process and keeps the stakeholder informed of the state of his store and warehouses by daily, monthly and annual report, and all of these functions can be done easily by his mobile phone with low monthly fees , suitable for most small stores.

You don’t need a high-spec phone to use this application, as it won’t use your phone's memory to store data, as all you need is an internet connection.

1.2 Project Motivation

Most stores still use paper and pen in the process of inventorying goods and this method has many problems, these papers can be lost or destroyed by mistake or even theft. Among its disadvantages, this method also consumes considerable time and effort, so it was necessary to find a practical solution to this problem.

This project will find great turnout by the stakeholders because it will help them to monitor the status of their warehouses, and it will also help them to know the goods that have been

carried out previously, so he will not be surprised by the exhaustion of a certain type of goods, and this, in turn, will increase the value of profits.

One of the new features is your phone will guide you to the exact location of the goods in your warehouses (ease of use and availability is our main motivation)

1.3 Problem statement

Small business owners face a challenge with their storage tracing what they have and don't have, every time they need to check the availability for something, they have to spend a minute up to 15 minutes to make sure of the availability but if there is something affordable, they will gladly take it, so small businesses are affected by manually, making it an automated process will surely increase profits.

1.4 Project Aim and Objectives

The goal of this project is a method for tracking inventory to maximize profit margins because a well-managed inventory can save you from stockpiles and lost sales and it helps with excess inventory costs, with this application functionalities the owner can watch over the employees for what's getting out and what's entering the warehouse.

1.5 Project Scope

The scope can include valuing the inventory and warn the user about low stocks of any individual item in the inventory, a daily, monthly, annual report to allow the user to plan for future needs for the inventory

- Users can choose a plan, trial, monthly, biannually, annually.
- Users can decide the size of the inventory upon subscribing.
- Users can print daily, monthly, annual reports with administrative rights
- Users can pay either cash or visa
- Administrator can add or remove any item
- Administrators can restrict employees from specific functionalities.

1.6 Project Software and Hardware Requirements

- Deployment environment:

  - ❏ Software Requirements:
    - ➢ Google Play
    - ➢ Internet
    - ➢ Visa card / mastercard / zain cash / orange money/dinark or any available application for the payment process.
  - ❏ Hardware Requirements:
    - ➢ Mobile Phone
    - ➢ PC
    - ➢ Android 7.0 or higher Operating System

- Development environment:

  - ❏ Software Requirements:
    - ➢ Android Studio
    - ➢ firebase
    - ➢ External libraries
    - ➢ Adobe xd
    - ➢ PowerPoint, word
    - ➢ Tool for gantt chart
    - ➢ Draw.io
  - ❏ Hardware Requirements:
    - ➢ PC
    - ➢ Mobile Phone
    - ➢ Operating System

- Server to save the Database (Firebase).

1.7 Project Limitations

We have tried our best to make this software flexible and easy to use but we can't avoid the limitations.

We made the system offer a wide range of options to the users but some options were not covered because the time restrictions compelled us to ignore some parts.

Considerable efforts have made the program easy to operate but it is recognized that the average person may find some problems during use.

The user is provided help at each step for his convenience in working with the software.

Here is a list of limitations in the system:

- Our application only supports the android platform and only the mobile phones that have a version 5.0 and more.
- The absence of internet connectivity.
- A mobile phone that doesn't support Google services.

1.8 Project Expected Output

- The system should help business owners to allocate whether the items in need are available or not.
- The system should allow owners to choose any administrative rights to the employees.
- The system shall cover all needs of small business owners.

CHAPTER 2: Related Existing System

2.1 Introduction

Inventory management system is a system that combines both software and hardware to supervise capitalized assets and stock items , simplify stock arrangement , what goes in and what goes out.

Here we are going to look at the difference or what will our system bring to the consumers/users differently from other IMSs , considering that there is a vast amount of IMSs out in the market

2.2 Existing Systems

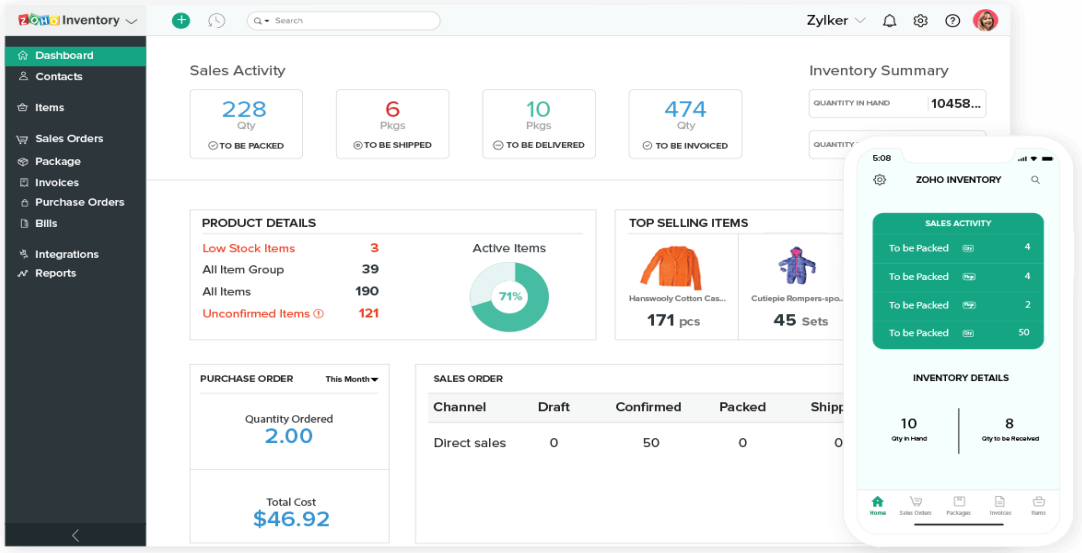
Googling “Inventory Management System” brings out 497 million results ! That's pretty high for something that is not essential in most people's lives, but once someone who is in need of it uses it or hears of it , he will be astonished at how much more work he can do with a fraction of the time of doing it manually.

Hence ,there came to be a high demand on better systems to be even more efficient than ever at managing inventories, a lot of programs/applications came to be.

One of them.

Zoho inventory is the most used IMS worldwide , it became a mainstream for using an IMSs , for its simplicity and security , but it has some downsides that we will be talking about in the next segment.

Figure 2.2.1.



Looking at other apps on google play store.

Inventory Management from Industrial IT , claiming to be easy to use and covering all functionalities that any warehouse/business would need , with its 50k downloads and a rating of 3.8/5 stars on google play store , everything seems to be promising until we dissect its problems.

Figure 2.2.2.

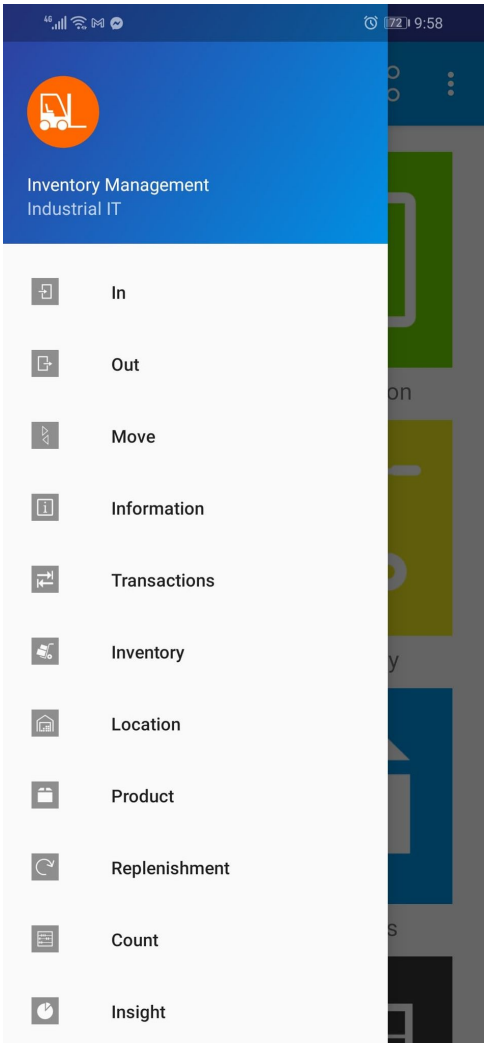
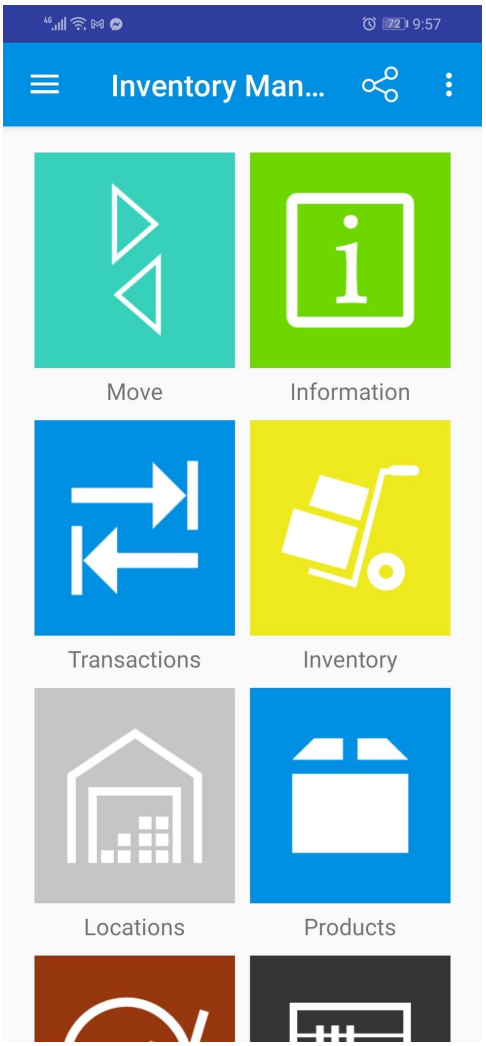


Figure 2.2.3.



2.3 Overall Problems of Existing Systems

An IMS like Zoho is web based ,meaning you have to use it on a dedicated internet browser

it can’t be used without an internet connection , you can’t even check your inventory when you are offline.

Dissecting The problems with Inventory Management from Industrial IT , its trying to be the opposite of bland by adding colors making it appear childish and unpleasant to the eyes

Plus there are no profiles , it's a single person use , unless you want to manually sync every stock movement , this defeats the purpose of an efficient IMS , in view of the fact that you will be in need of multiple users.

2.4 Overall Solution Approach

Our IMS shall provide most of the urgent functionalities without internet connection, even though it still needs internet connection , it is 100% operable without internet , Database is stored locally and on the cloud , it is store in the cloud for security reasons , cloud storage is safer than local storage , that's why it is possible to opt out of local storage and stay with only cloud storage.

Every user of this app must login to use the full functionalities and with those profiles/users there will be a peer to peer connection between employees and the owner, the database will be held by nothing but its users , this is also optional.

The conclusion is that there will be three types of database in our system

- Cloud based database
- Locally held database
- Peer to peer database

you can use the three of them or atleast one of them accordingly.

CHAPTER 3: Requirement Engineering and Analysis

Requirement engineering and analysis is the discipline that involves establishing , documenting requirements and defining user expectations for a new software being built or modified ,it could be called requirements gathering.

So in this chapter we will be listing the entities that will be affected by this project and how they are affected by it.

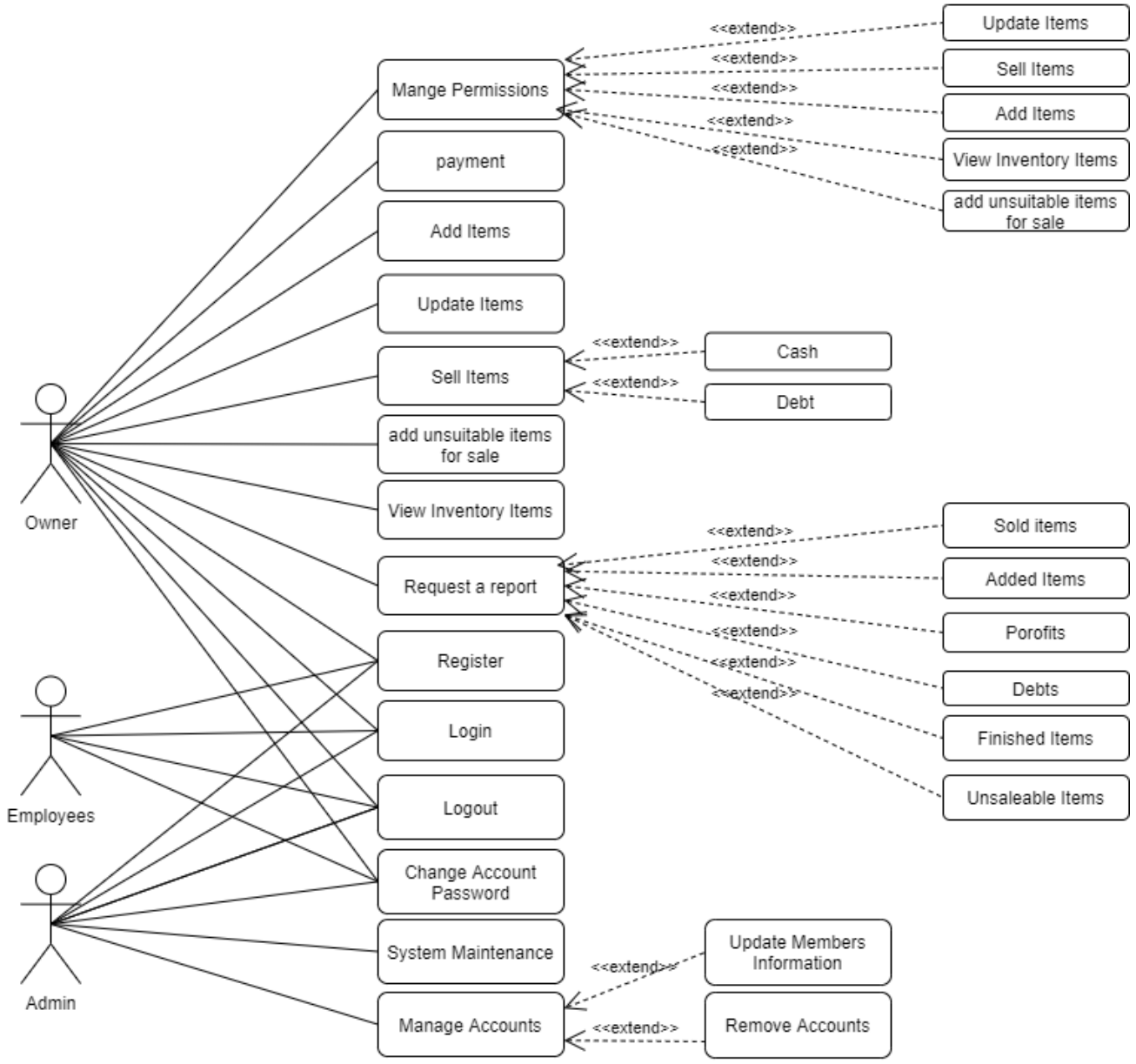
3.1 Stakeholders

- Primary stakeholders :
    1. Owner
    2. Employee
    3. Admin
- Secondary stakeholders :
    1. Developers
    2. System QA

3.2 Use case diagram

- The figure below shows a use case diagram contains three types of users and the activities available for each of them.

Figure 3.2.1



-

3.2.1 Use Case Description
3.2.1.1 Alternative Flow

Use Case	**Manage Permissions**
ID	**1**
Brief description	**Give/revoke specific permissions to/form specific Employees.**
Primary Actor	**Owner**
Secondary Actor	**Employee**
Pre-condition	**1. Owner must have an account.** **2. Employee must have an account** **3. System must be connected to the network**
Main Flow	**1. The Owner logs in** **2. The Owner selects one employee or more.** **3. The Owner gives or revokes certain permissions to the employee** **Extension point : TypeOfPermissions , Extensions: Update Items, AddItems, Sell Items, View Inventory Items, View Items Location, Add Unsellable Items.**
Post-condition	**Permissions were either given or taken from the employee**
Alternative Flow	● **If you are employee you can't give or revoke permissions** ● **If the employee doesn’t have an account, the owner can't give him permissions.** ● **If the owner does not connect to the network, he can't give or revoke permissions.**

(Table 0-1) Manage permissions , it allows the owner to control the employee's authority.

Extension Use Case	**Add Items**
ID	**2**
Brief description	**Segment 1 : Permission allows employees to enter the purchased goods into the database .**
Primary Actor	**Owner**
Secondary Actor	**Employee**
Segment 1 Pre-condition	1. **The users must have an account.** 2. **System must be connected to the network.**
Segment 1 Main Flow	1. **The Employee takes or misses (Add items) permission.**
Segment 1 Post-condition	1. **The employee can add items like the owner if he takes the permission, and if he loses it he can't.**
Segment 1 Alternative Flow	• **None**

(Table 0-2) Add Items is extension of Manage permissions , it allows the owner to control the employee's authority to add items.

Extension Use Case	**Update Items**
ID	**3**
Brief description	**Segment 2 : s employees to update items info.**
Primary Actor	**Owner**
Secondary Actor	**Employee**

Segment 2 Pre-condition	1. **The users must have an account.** 2. **System must be connected to the network.**
Segment 2 Main Flow	1. **The Employee takes or misses (Update items) permission.**
Segment 2 Post-condition	1. **The employee can update items like the owner if he takes the permission, and if he loses it he can't.**
Segment 2 Alternative Flow	● **None**

(Table 0-3) Update Items is an extension of Manage permissions , it allows the owner to control the employee's authority to update items.

Extension Use Case	**Sell Items**
ID	**4**
Brief description	**Segment 3 : Permission allows employees to Sell Items**
Primary Actor	**Owner**
Secondary Actor	**Employee**
Segment 3 Pre-condition	1. **The users must have an account.** 2. **System must be connected to the network.**
Segment 3 Main Flow	1. **The Employee takes or misses (Sell items) permission.**
Segment 3 Post-condition	1. **The employee can sell items like the owner if he takes the permission, and if he loses it he can't.**
Segment 3 Alternative Flow	● **None**

(Table 0-4) Sell Items is an extension of Manage permissions , it allows the owner to control the employee's authority to sell items.

Extension Use Case	**Add unsuitable items for sale**
ID	**5**
Brief description	**Segment 4 : Permission allows employees to Add unsuitable items for sale**
Primary Actor	**Owner**
Secondary Actor	**Employee**
Segment 4 Pre-condition	1. **The users must have an account.** 2. **System must be connected to the network.**
Segment 4 Main Flow	1. **The Employee takes or misses (Add unsuitable items for sale) permission.**
Segment 4 Post-condition	1. **The employee can Add unsuitable items for sale like the owner if he takes the permission, and if he loses it he can't.**
Segment 4 Alternative Flow	● **None**

(Table 0-5)Adding items unsuitable for sale is an extension of management permissions, as it allows the owner to control the employee's authority to add damaged goods, expired goods, and other goods that cannot be sold

Extension Use Case	**View Inventory Items**
ID	**6**
Brief description	**Segment 5 : Permission allows employees to see the items inside the inventory**
Primary Actor	**Owner**

Secondary Actor	**Employee**
Segment 5 Pre-condition	**1. The users must have an account.** **2. System must be connected to the network.**
Segment 5 Main Flow	**1. The Employee takes or misses (View Inventory Items) permission.**
Segment 5 Post-condition	**1. The employee can view inventory items like the owner if he takes the permission, and if he loses it he can't.**
Segment 5 Alternative Flow	● **None**

(Table 0-6) View inventory items is an extension of Manage permissions , it allows the owner to control the employee's authority to view inventory items.

Use Case	**Payment**
ID	**7**
Brief description	**The owner can stop or activate the payment process whenever he want**
Primary Actor	**Owner**
Secondary Actor	**None**
Pre-condition	**1. Owner must have an account.** **2. System must be connected to the network.**
Main Flow	**1. The Owner Logs in .** **2. The Owner chooses to Disable payment if it's Enabled before ,or Enable payment if it's disabled before.**
Post-condition	**Payment Process has been Enabled or Disabled**
Alternative Flow	● **If the owner's financial account does not contain sufficient amounts, he will not be able to Enable the payment service and therefore will not benefit from the application's services.** ● **If Owner does not connect to the network, he can’t Enable or Disable his payment process.**

(Table 0-7)Payment , it allows the owner to enable or disable the payment process.

Use Case	**Add Items**
ID	**8**
Brief description	**Enter the purchased goods into the database .**
Primary Actor	**Owner**
Secondary Actor	**None**
Pre-condition	**1. The user must have an account.** **2. System must be connected to the network.**
Main Flow	**1. The User Logs in .** **2. The User Press on Add Items.** **3. The User inserts the item's barcode.** **4. The User inserts the item's Name.** **5. The User inserts the item's Quantity.** **6. The User inserts the item's Category.** **7. The User inserts the item's Price.** **8. The User inserts the item's Expiration date.**
Post-condition	**Purchased item has been added to the database**
Alternative Flow	● **If the user does not connect to the network, he can't add items.**

(Table 0-8)Add items, it allows the user to add new items.

Use Case	**Sell Items**
ID	**9**
Brief description	**Sale of available goods either in cash or in debt**

Primary Actor	**Owner**
Secondary Actor	**None**
Pre-condition	**1. The user must have an account.** **2. System must be connected to the network.**
Main Flow	**1. The user Logs in .** **2. The user presses on sell Item.** **3. The user chooses the payment method.** **Extension point : TypeOfPayment , Extensions: Cash,Debt.**
Post-condition	**1. Bought items have been added to the database.**
Alternative flow	● **If the user does not connect to the network, he can't sell items.**

(Table 0-9)Sell items, it allows the user to sell existing items.

Extension Use Case	**Cash**
ID	**10**
Brief description	**Segment 1 :Sale of available goods in cash**
Primary Actor	**Owner**
Secondary Actor	**None**
Segment 1 Pre-condition	**1. The User must have an account.** **2. System must be connected to the network.** **3. The User chooses cash .**
Segment 1 Main Flow	**1. Camera will open to read the item's barcode.**
Segment 1 Post-condition	**Bought items have been added to the database.**
Segment 1 Alternative Flow	● **If the user does not connect to the network, he can't sell items.**

(Table 0-10) Cash is an extension of Sell items, it allows the user to sell items in cash.

Extension Use Case	**Debt**
ID	**11**
Brief description	**Segment 2 :Sale of available goods in debt**
Primary Actor	**Owner**
Secondary Actor	**None**
Segment 2 Pre-condition	1. **The User must have an account.** 2. **System must be connected to the network.** 3. **The User chooses debt**
Segment 2 Main Flow	1. **Camera will open to read the item's barcode.**
Segment 2 Post-condition	**Bought items have been added to the database.**
Segment 2 Alternative Flow	● **If the user does not connect to the network, he can't sell items.**

(Table 0-11) Debt is an extension of Sell items, it allows the user to sell items in debt.

Use Case	**Update Items**
ID	**12**
Brief description	**Make modifications to the name, quantity or location of the existing elements**
Primary Actor	**Owner**
Secondary Actor	**None**

Pre-condition	1. **The User must have an account.** 2. **System must be connected to the network.**
Main Flow	1. **The User Logs in .** 2. **The User presses on update Item.** 3. **Camera will open to read the item's barcode.** 4. **The User set new information.**
Post-condition	**The information of items have been Updated**
Alternative Flow	● **If the user does not connect to the network, he can’t update items.**

(Table 0-12) Update items , it allows the user to update items information.

Use Case	**Add unsuitable items for sale**
ID	**13**
Brief description	**Add damaged or expired items to the list of unsuitable items.**
Primary Actor	**Owner**
Secondary Actor	**None**
Pre-condition	1. **The User must have an account.** 2. **System must be connected to the network.**
Main Flow	1. **The User Logs in .** 2. **The User presses on Add unsuitable item for sale.** 3. **Camera will open to read the item's barcode.**
Post-condition	**1- The unsuitable items for sale have been added.** **2- The unsuitable items for sale have been removed from the list of existing items.**
Alternative flow	● **If the user does not connect to the network, he can’t add unsuitable items.**

(Table 0-13) Add unsuitable items for sale , it allows the user to add damaged goods, expired goods, and other goods that cannot be sold.

.

Use Case	**Request a report**
ID	**14**
Brief description	**A report showing the amount of sales, purchases, profits, damaged goods, goods sold on debt, finished goods, and adjustments made within a day, week, month or year**
Primary Actor	**Owner**
Secondary Actor	**None**
Pre-condition	**1. The User must have an account.** **2. System must be connected to the network.**
Main Flow	**1. The User Logs in .** **2. The User presses on Repot.** **3. The User Choose the type of wanted report.** **Extension point : TypeOfReport , Extensions: Sold Items,Added Items, Unsellable Items, Debt, Profits, Finished Items.**
Post-condition	**The required report is displayed**
Alternative Flow	• **If the user does not connect to the network, he can't see reports.**

(Table 0-14) Request a report , it allows the owner to request any type of reports about(sold items , added items ..etc).

Extension Use Case	**Sold Items**
ID	**15**
Brief description	**Segment 1 : Show report of sold items.**
Primary Actor	**Owner**

Secondary Actor	**None**
Segment 1 Pre-condition	**1. System must be connected to the network.** **2. The User Choose sold items report**
Segment 1 Main Flow	**1. The user chooses the duration of the report**
Segment 1 Post-condition	**The sold items report are displayed**
Segment 1 Alternative Flow	**None**

(Table 0-15)Sold items is an extension of Request a report, it allows the owner to see the sold items during a specific period.

Extension Use Case	**Added Items**
ID	**16**
Brief description	**Segment 2 : Show report of added items.**
Primary Actor	**Owner**
Secondary Actor	**None**
Segment 2 Pre-condition	**1. System must be connected to the network.** **2. The User Choose added items report**
Segment 2 Main Flow	**1. The user chooses the duration of the report**
Segment2 Post-condition	**The added items report are displayed**
Segment 2 Alternative Flow	**None**

(Table 0-16)Added items is an extension of Request a report, it allows the owner to see the added items during a specific period.

Extension Use Case	**Add unsuitable items for sale**
ID	**17**
Brief description	**Segment 3 : Show report of unsuitable items for sale.**
Primary Actor	**Owner**
Secondary Actor	**None**
Segment 3 Pre-condition	**1. System must be connected to the network.** **2. The User Choose unsuitable items for sale report**
Segment 3 Main Flow	**1. The user chooses the duration of the report**
Segment 3 Post-condition	**The unsuitable items report are displayed**
Segment 3 Alternative Flow	**None**

(Table 0-17)Add unsuitable items for sale is an extension of Request a report, it allows the owner to see the damaged goods, expired goods, and other goods that cannot be sold during a specific period.

Extension Use Case	**Debt**
ID	**18**
Brief description	**Segment 4 : Show report of items sold in debt.**
Primary Actor	**Owner**

Secondary Actor	**None**
Segment 4 Pre-condition	**1. System must be connected to the network.** **2. The User Choose debt report**
Segment 4 Main Flow	**1. The user chooses the duration of the report**
Segment4 Post-condition	**The debt report are displayed**
Segment 4 Alternative Flow	**None**

(Table 0-18) Debt is an extension of Request a report, it allows the owner to see the items sold in debt during a specific period.

Extension Use Case	**Profits**
ID	**19**
Brief description	**Segment 5 : Show report of Profits.**
Primary Actor	**Owner**
Secondary Actor	**None**
Segment 5 Pre-condition	**1. System must be connected to the network.** **2. The User Choose Profits report**
Segment 5 Main Flow	**1. The user chooses the duration of the report**
Segment 5 Post-condition	**The Profits report are displayed**
Segment 5 Alternative Flow	**None**

(Table 0-19) Profits is an extension of Request a report, it allows the owner to see his profits during a specific period.

Extension Use Case	**Finished Items**
ID	**20**
Brief description	**Segment 6 : Show report of finished items.**
Primary Actor	**Owner**
Secondary Actor	**None**
Segment 6 Pre-condition	1. **System must be connected to the network.** 2. **The User Choose finished items report**
Segment 6 Main Flow	1. **The user chooses the duration of the report**
Segment 6 Post-condition	**The finished items report are displayed**
Segment 6 Alternative Flow	**None**

(Table 0-20) Finished Items is an extension of Request a report, it allows the owner to see the finished items in his store.

Use Case	**View inventory Items**
ID	**21**
Brief description	**Show all existing items with their information**
Primary Actor	**Owner**
Secondary Actor	**None**
Pre-condition	1. **The User must have an account.**

	2. **System must be connected to the network.f**
Main Flow	1. **The User Logs in .** 2. **The User presses on View inventory items.**
Post-condition	● **list of all existing items appears**
Alternative Flow	**None**

(Table 0-21)View inventory items , it allows the owner to see the existing items.

Use Case	**Register**
ID	**22**
Brief description	**Register to the system**
Primary Actor	**Owner , Employee , Admin**
Secondary Actor	**None**
Pre-condition	1. **System must be connected to the network.**
Main Flow	1. **The User inserts his name.** 2. **The User inserts his email.** 3. **The User inserts his phone number.** 4. **The User inserts a password.** 5. **The User confirms the password.** 6. **The User Choose his role (Admin,Employee).** 7. **The User presses on the register button.**
Post-condition	**User has been registered and his information is saved**
Alternative Flow	● **If the inserted phone exists before the user can't create this account.** ● **If the inserted email exists before the user can't create this account.** ● **If Password doesn't match, the register button won’t work.**

(Table 0-22)Register , it allows the user to create an account.

Use Case	**Login**
ID	**23**
Brief description	**A user login to system to access the functionality of the system**
Primary Actor	**Owner , Admin , Employee**
Secondary Actor	**None**
Pre-condition	**1. System must be connected to the network.** **2. The user must have an account.**
Main flow	**1. The user inserts his email.** **2. The user inserts the password.** **3. The user presses the login button.**
Post-condition	● **After successful login, the user's home page will be opened**
Alternative flow	● **If the email or password doesn’t exist the user can’t login .** ● **If the email or password is wrong the user can’t login .**

(Table 0-23)Login , it allows the user to enter his account.

Use Case	**Logout**
ID	**24**
Brief description	**Exit from the system**
Primary Actor	**Owner , Admin , Employee**
Secondary Actor	**None**
Pre-condition	**1. System must be connected to the network.** **2. The user must have an account.**
Main Flow	**1. The user should press the logout button.**

Post-condition	**You have been logged out of the system**
Alternative Flow	**None**

(Table 0-24)Logout , it allows the user to exist from his account.

Use Case	**Change Account password**
ID	**25**
Brief description	**You can change your account password whenever you want**
Primary Actor	**Owner , Employee , Admin**
Secondary Actor	**None**
Pre-condition	**1. System must be connected to the network.** **2. The user must have an account.**
Main Flow	**1. The user should press on change account password .** **2. The user will receive code in his phone number.** **3. The user should insert this code .** **4. The user should insert the new password.** **5. The user should confirm this password.**
Post-condition	● **Password has changed**
Alternative Flow	● **If Password doesn't match, the user can’t change his password.** ● **If the inserted code was wrong the user can’t change his account password.**

(Table 0-25)Change Account Password , it allows the user to create a new password .

Use Case	**System Maintenance**
ID	**26**
Brief description	**The admin can change the way the system is shown or enable more features or disable them .**
Primary Actor	**Admin**
Secondary Actor	**None**
Pre-condition	**1. System must be connected to the network.** **2. The user must have an account.**
Main Flow	**1. The Admin Logs in.** **2. The Admin fixes specific problems or creates new features or disables some features.**
Post-condition	**The modifications made by the admin to the system have been completed**
Alternative Flow	**None**

(Table 0-26)System maintenance , it allows the admin to change the way the system is shown or enable more features or disable them .

Use Case	**Manage Accounts**
ID	**27**
Brief description	**Admin can access the database and update it or delete accounts.**
Primary Actor	**Admin**
Secondary Actor	**Owner, Employee**
Pre-condition	**1. System must be connected to the network.** **2. The user must have an account.**

Main Flow	1. **The Admin Logs in .** 2. **The Admin chooses what type of management he wants.** **Extension point : TypeOfManagement , Extensions: Update Members Information,Remove Accounts.**
Post-condition	**The modifications made by the admin to the database have been completed**
Alternative Flow	**None**

(Table 0-27) Manage Accounts , it allows the admin to manage users accounts.

Extension Use Case	**Update Members Information**
ID	**28**
Brief description	**Segment 1 :Change members' information.**
Primary Actor	**Admin**
Secondary Actor	**Owner, Employee**
Segment 1 Pre-condition	1. **System must be connected to the network.** 2. **The admin Choosed (Update Members Information).**
Segment 1 Main Flow	1. **The admin chooses the member.** 2. **The admin changes this member information.**
Segment 1 Post-condition	**The modifications made by the admin to the database have been saved.**
Segment 1 Alternative Flow	**None**

(Table 0-28)Update members information , it allows the admin to change users information .

Extension Use Case	**Remove Accounts**
ID	**29**
Brief description	**Segment 2 :Delete unwanted accounts.**
Primary Actor	**Admin**
Secondary Actor	**Owner, Employee**
Pre-condition	**1. System must be connected to the network.** **2. The admin Choosed (Remove Accounts).**
Segment 2 Main Flow	**1. The admin chooses the member .** **2. The admin presses delete account .**
Segment 2 Post-condition	**The modifications made by the admin to the database have been saved.**
Segment 2 Alternative Flow	**None**

(Table 0-29)Remove Accounts , it allows the admin to delete users accounts.

3.3 Functional Requirements

R1: The owner should be able to give permissions to specific employees.
R2: The owner should be able to revoke the permissions.
R3: The owner will manage the payment process.
R4: The owner can add new items.
R5: The owner can look up items manually.
R6: The owner can look up items by reading the barcode using his phone camera.
R7: The owner can update the item's information.
R8: The owner can sell items in cash.
R9: The owner can sell items in debt.
R10: The owner can add items not suitable for sale to reduce the existing quantity.
R10: The owner can request a report of sold items with any period as selected by him.
R11: The owner can request a report of added items with any period as selected by him.
R12: The owner can request a report of the debt list with any period as selected by him.
R13: The owner can request a report of the profits with any period as selected by him.
R14: The owner can request a report of unsellable items with any period as selected by him.
R15: The owner can request a report of finished items with any period as selected by him.
R16: The employee can sell items if he takes permission.
R17: The employee can add items if he takes permission.
R18: The employee can update items information if he takes permission.
R19: The employee can add items not suitable for sale if he takes permission.
R20: The employee can view inventory items if he takes permission.
R21: Owner, admin, and employees must register before using the application
R22: Owner, admin, and employees can log in to the application.
R23: Owner, admin, and employees can log out of the application.
R24: Owner, admin, employees shall be able to change the account password
R25: The admin will be responsible for system maintenance
R26: The admin will be responsible for accounts removal and updating members' information.

3.4 Non-functional Requirements

Non Functional Requirements: Performance, availability, security, usability, recoverability
These non-functional requirements will be discussed below.

- Performance: The proposed system will be highly efficient, effective.
- Availability: In terms of availability, the system will work for almost 24*7. The target customers can gain the required customer service at any time in a day.
- Security: The system should provide confidentiality for user data using database encryption and local encryption to protect data in the event of device theft (phone)
- Recovery: In terms of recovery, the system is developed with high quality and effective recovery systems.

  This is acceptable that the integrated disaster recovery technique will assist the company as well as the system to recover the destroyed magnetic tapes, which are used to collect and store valuable data and information.
- Usability: The application provides high availability and high

3.5 Constraints

1. An always-on internet connection connected to Google's Firebase Realtime database storage.
2. A device running the Android operating system.
3. The admin should have a visa card or MasterCard or zain cash or any available app for the payment process.

CHAPTER 4: Architecture and Design

4.1 Overview

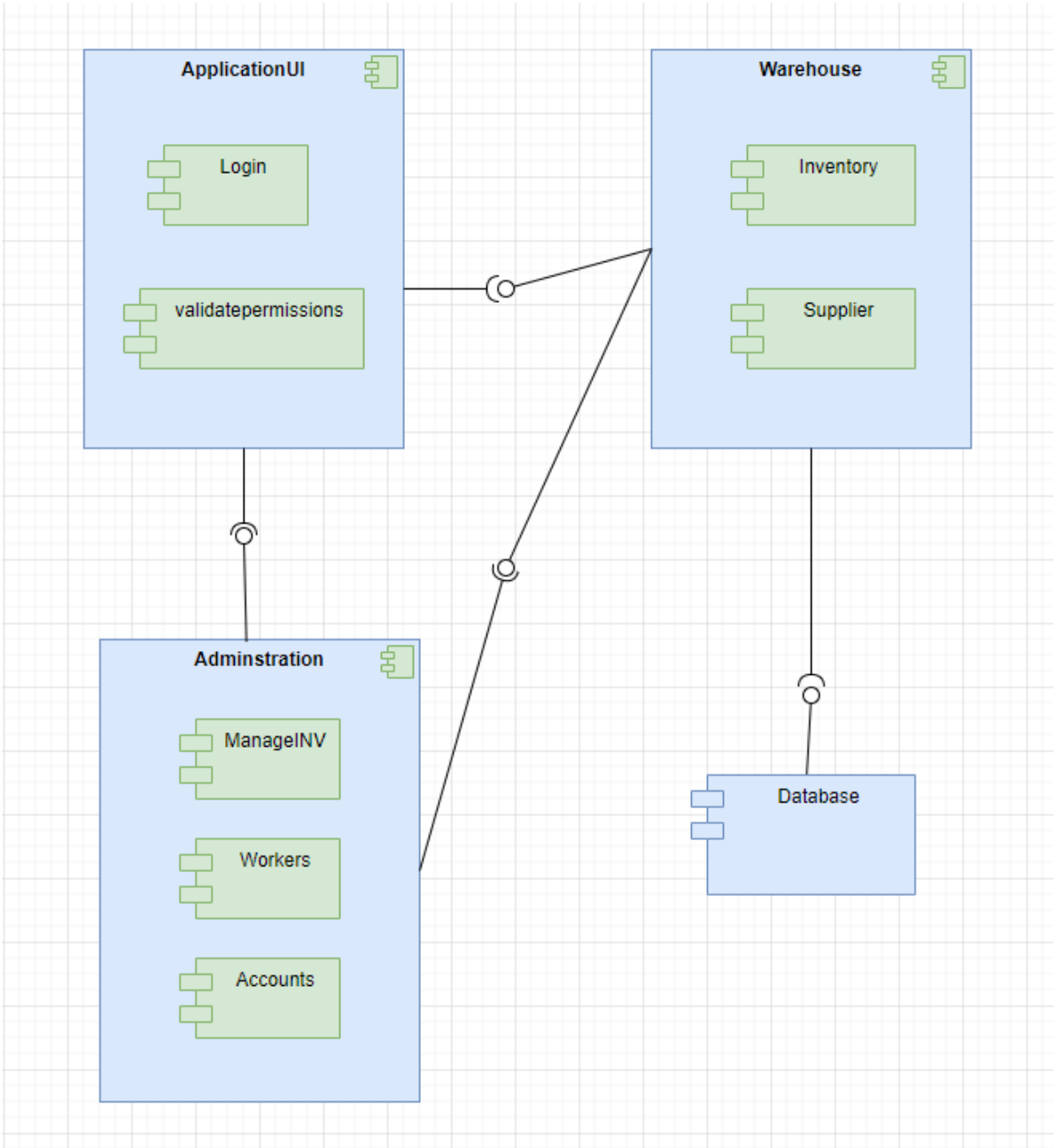
Visually representing a system with its main actors , actions , roles and classes. In order to better understand information about the system.

4.2 Software architecture

4.2.1 Logical view

The figure below explains the wiring of the physical components in a system.

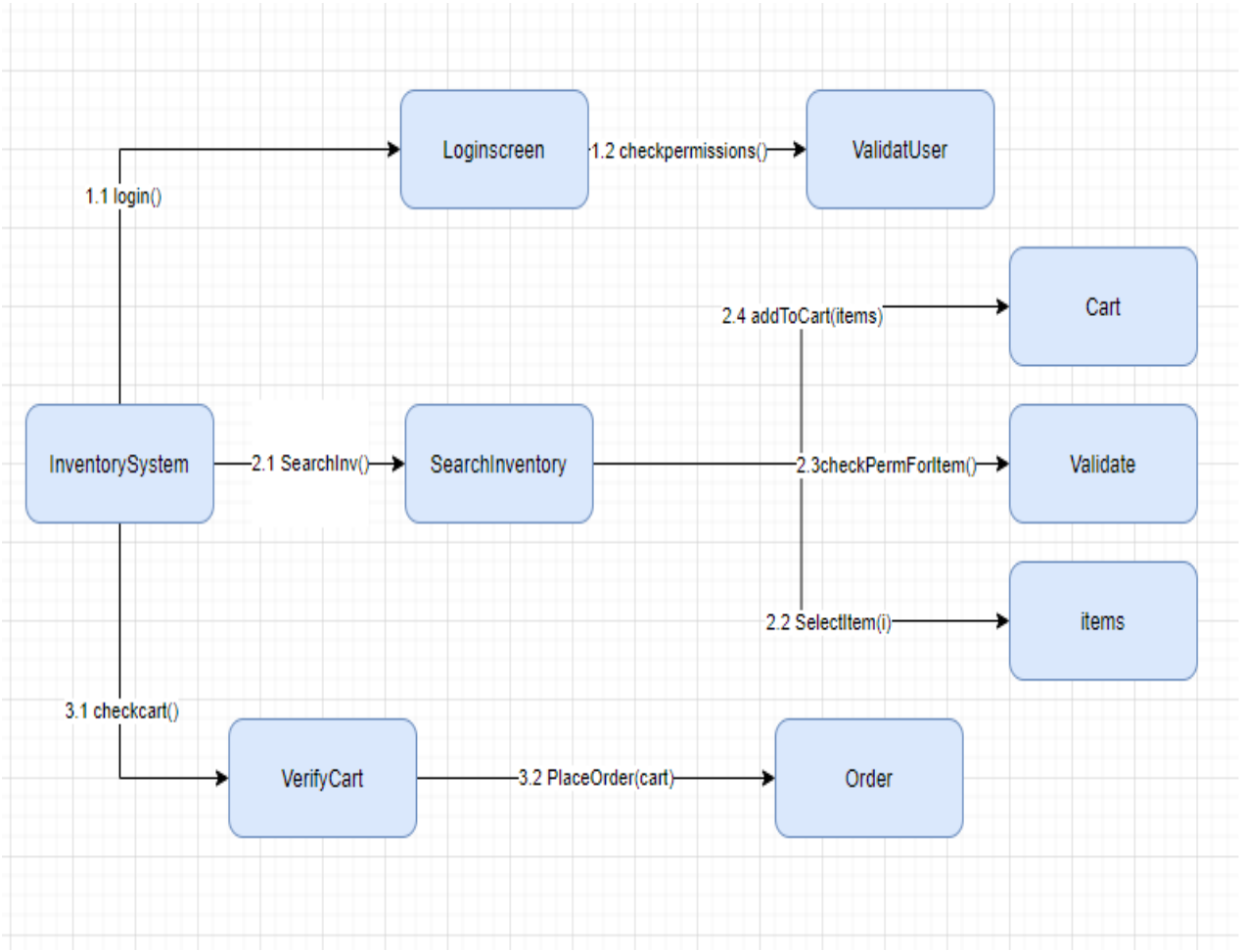
Figure 4.2.1.1.



4.2.2 Process view

As shown in the figure 4.2.2. below , a communication diagram is an interaction diagram that shows object relationships

Figure 4.2.2.1.



4.2.3 Physical view

In the figure below this deploying diagram shows the configuration of run time processing nodes and the components that live on them.

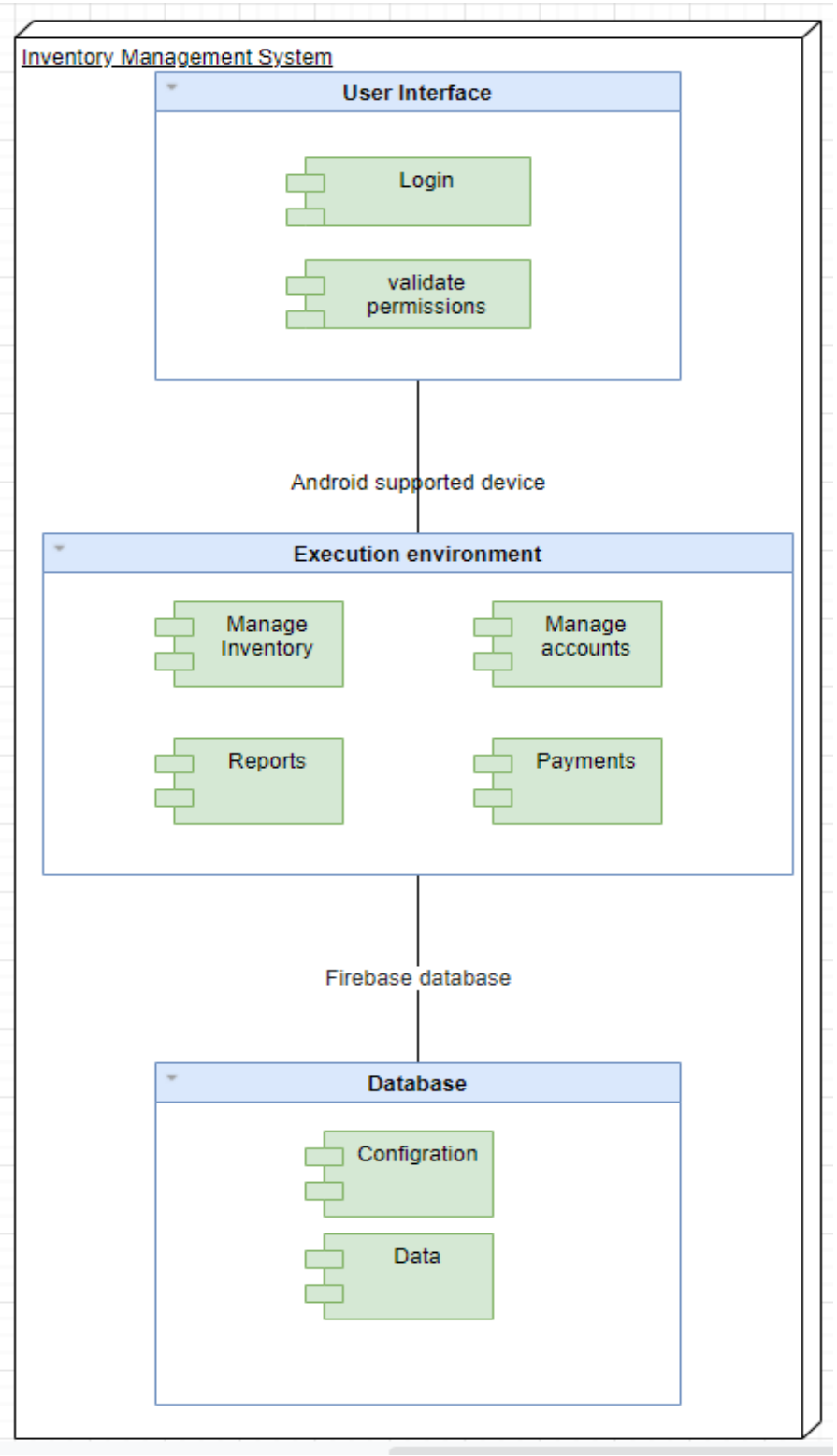


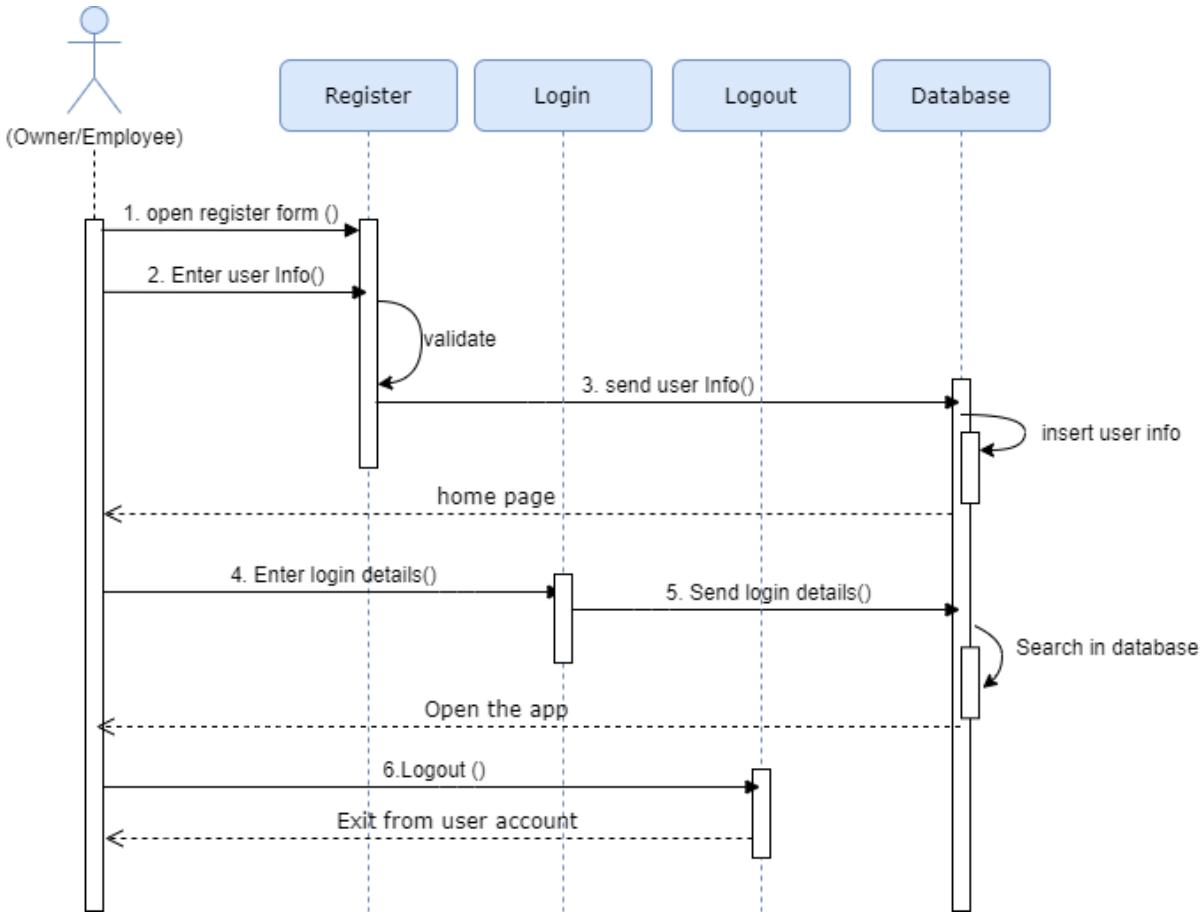
Figure 4.2.3.1.

4.3 Software design

4.3.1 UML sequence

(Register and Login Scenario)

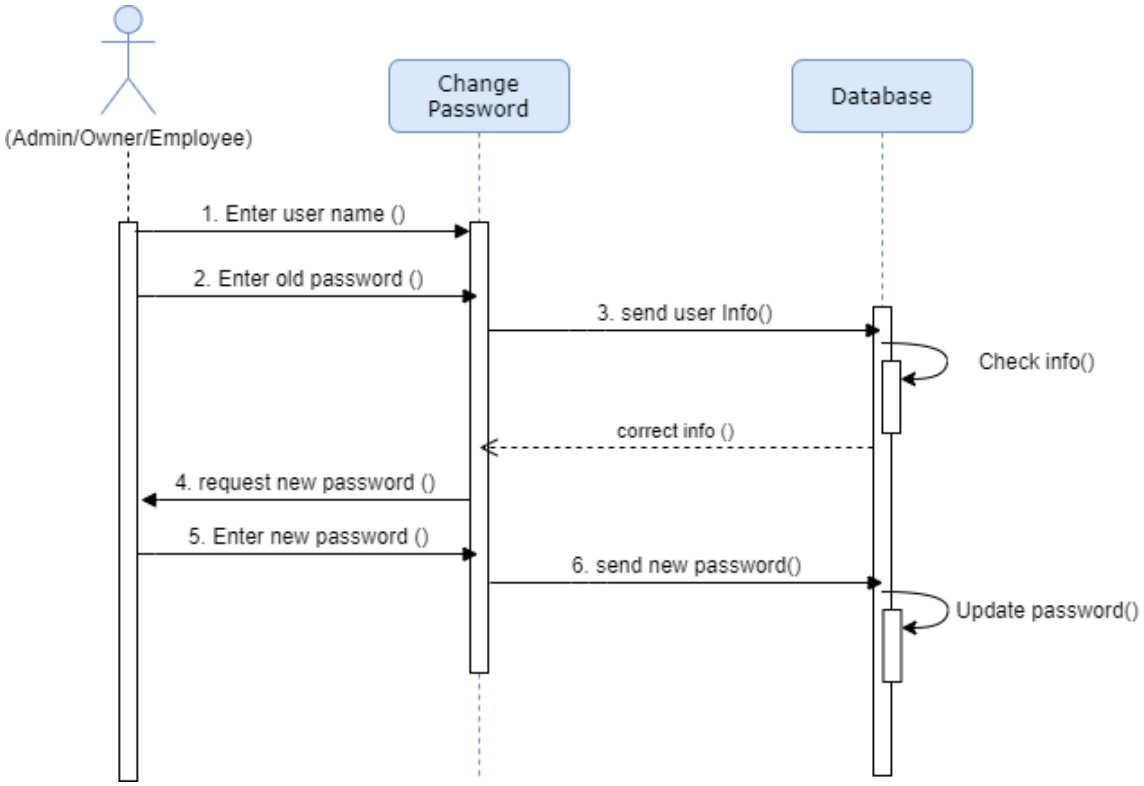
Figure 4.3.1.1.



- Owner and Employee should register to the system before they can login and then they can logout from the system.

(Change password scenario)

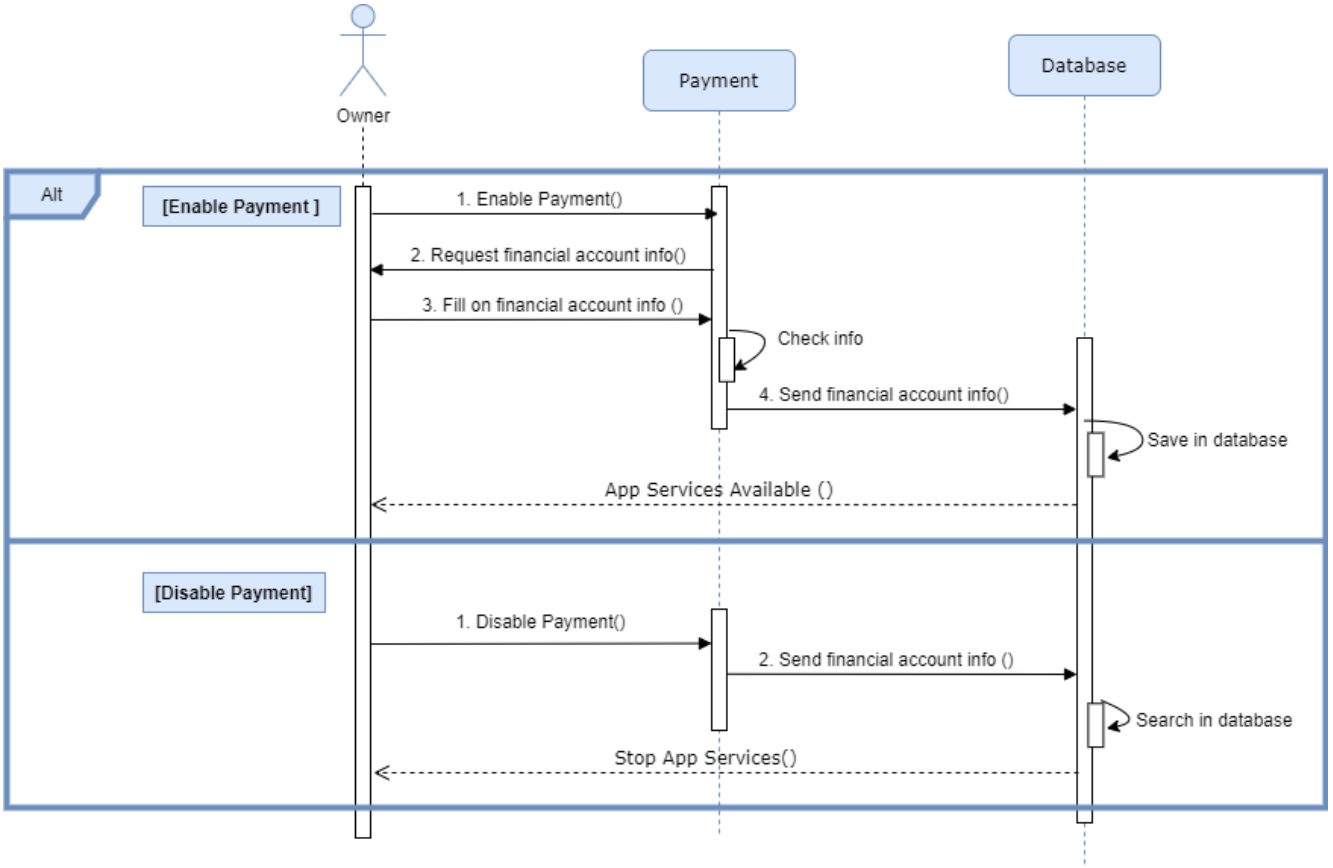
Figure 4.3.1.2.



- Admin or owner or employee enters the user name and password and he will be able to enter a new password.

(Payment scenario)

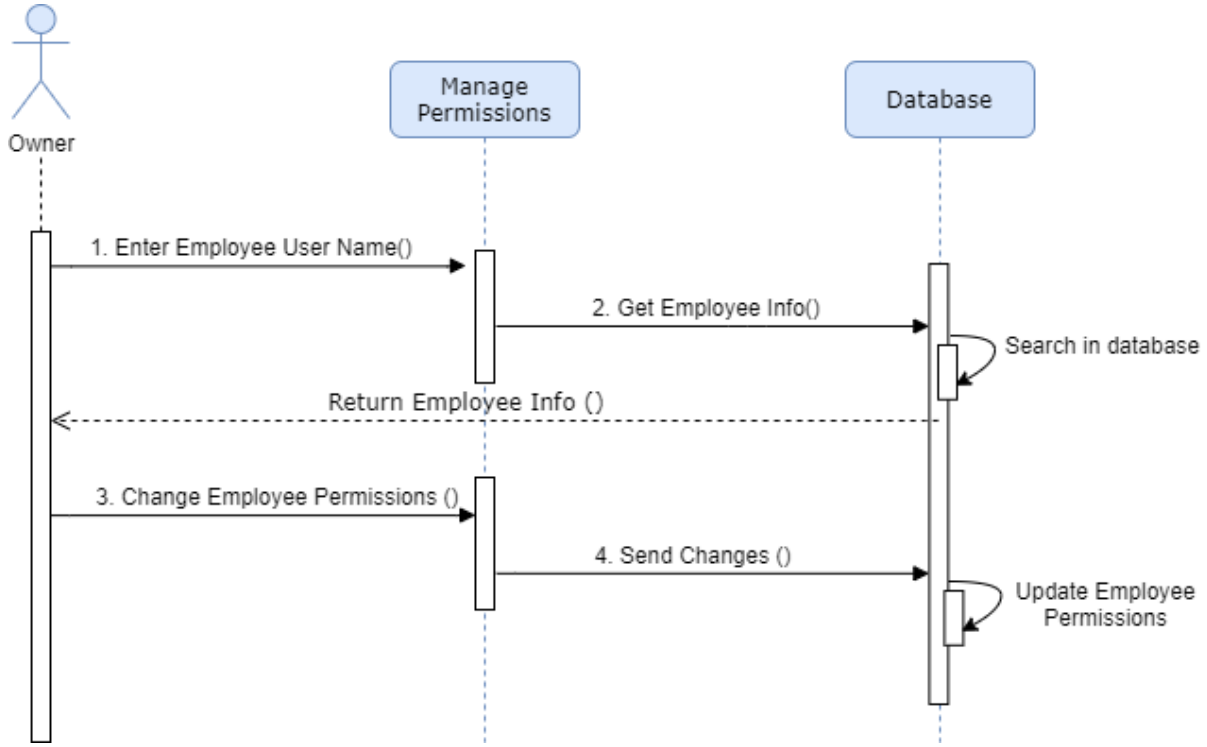
Figure 4.3.1.3.



- The owner can enable app services when he enables the payment process and he can stop it if he disables the payment method.

(Manage Permissions scenario)

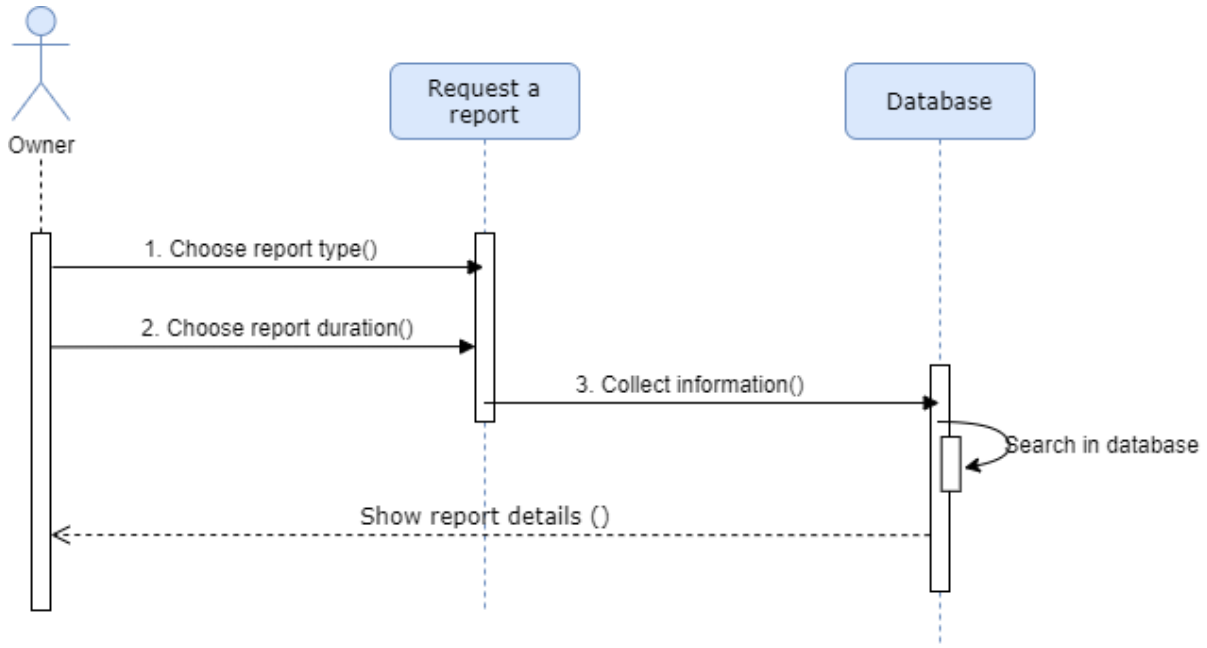
Figure 4.3.1.4.



- The owner can choose his employees and choose the type of permissions appropriate for each employee.

(Request a report scenario)

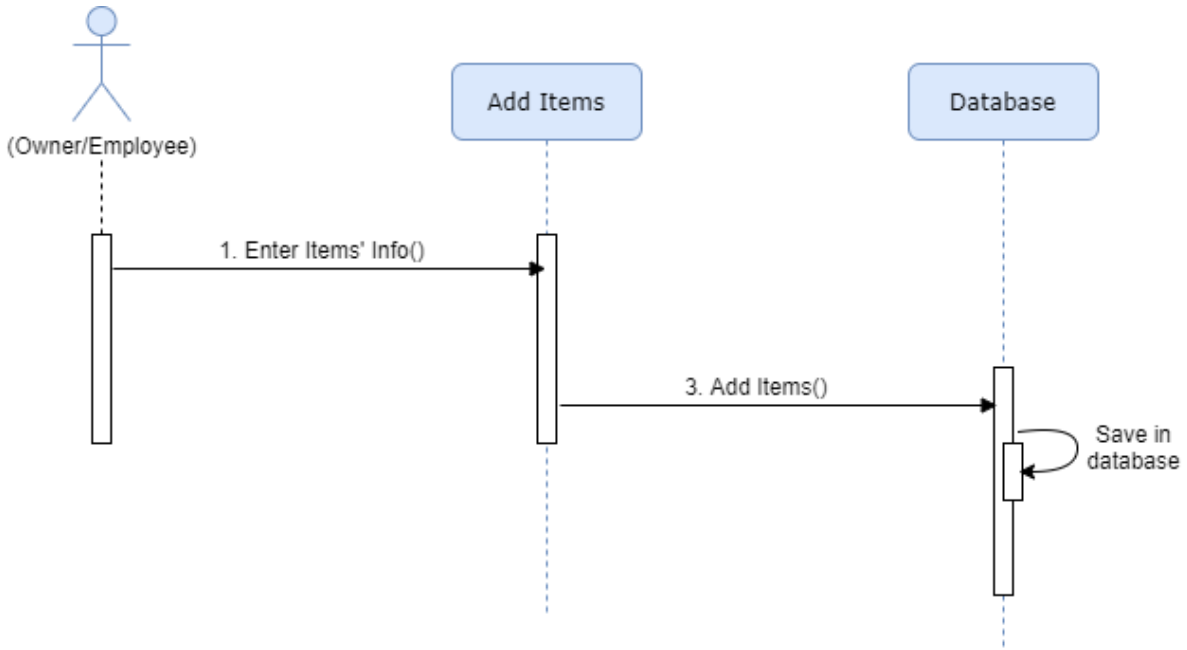
Figure 4.3.1.5.



- The owner chooses the report type and duration and the details will be shown on his screen.

(Add items scenario)

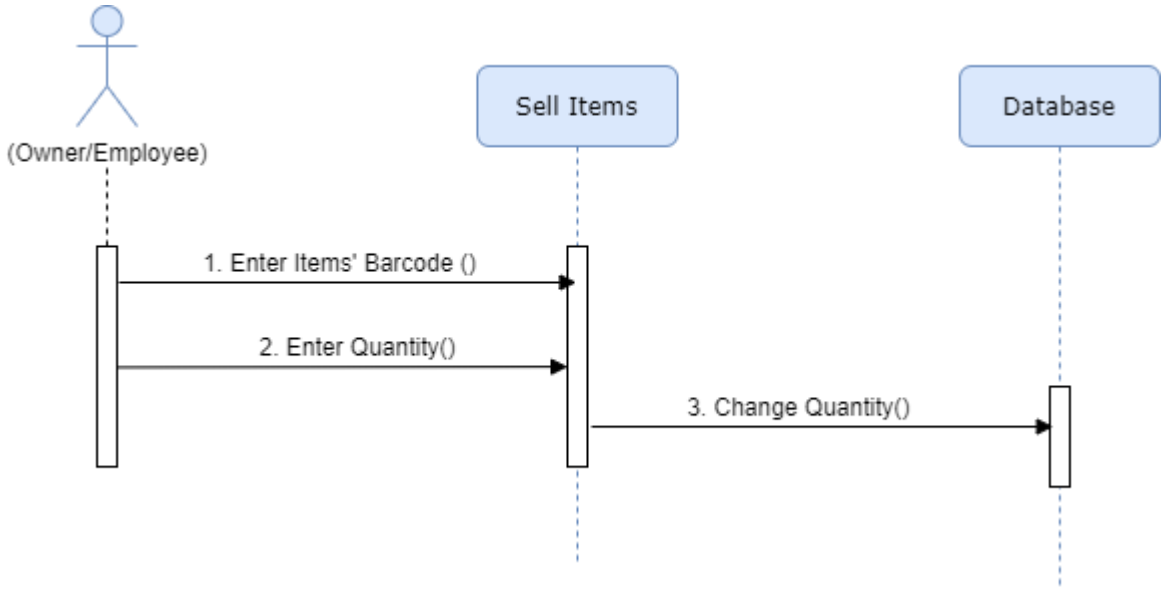
Figure 4.3.1.6.



- The owner and employee add items .

(Sell items scenario)

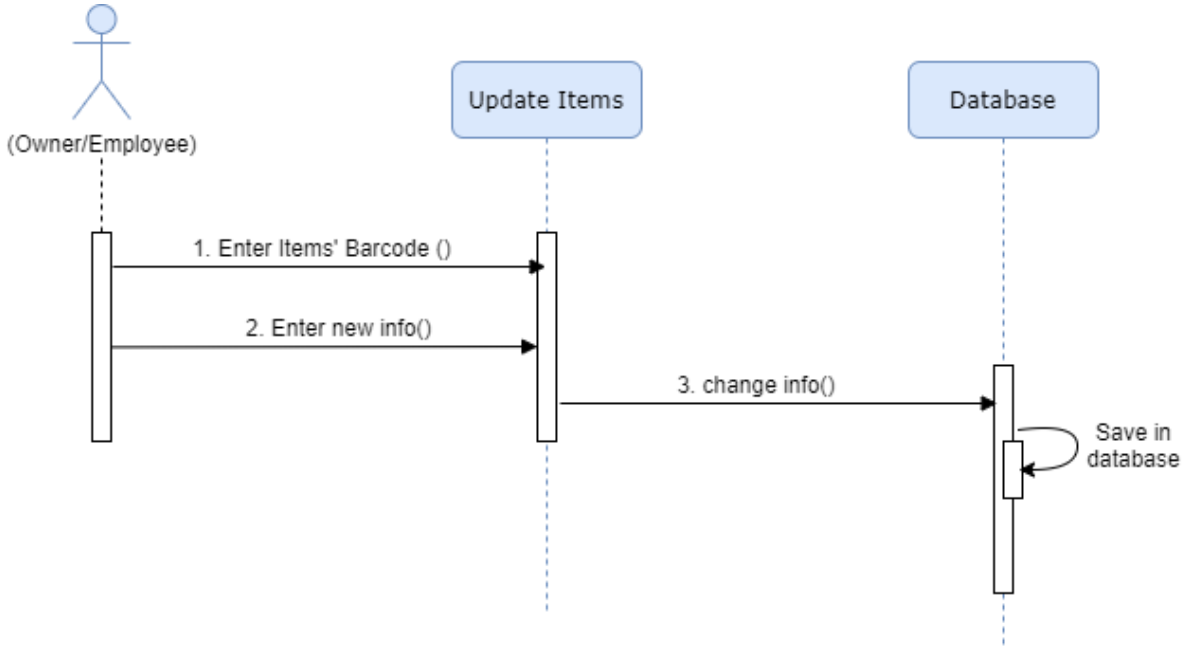
Figure 4.3.1.7.



- The owner and employee Sell items by entering the barcode manually or using a phone camera.

(Update items scenario)

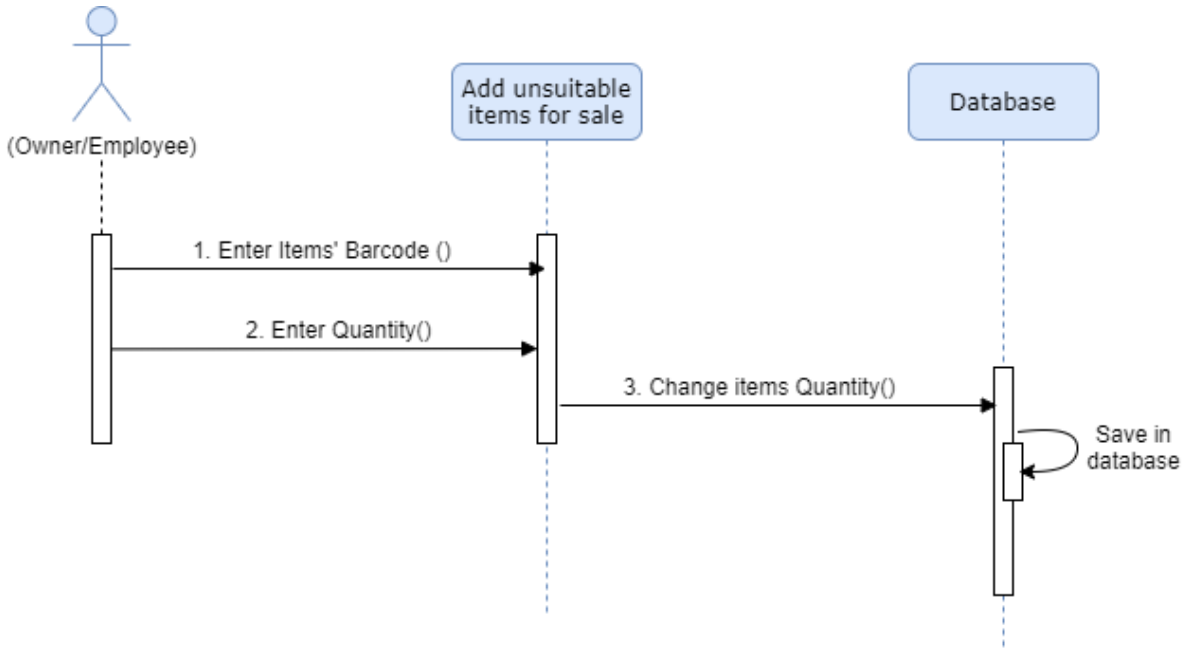
Figure 4.3.1.8.



- The owner and employee update items by entering the barcode manually or using a phone camera and entering new information.

(Add unsuitable items for sale scenario)

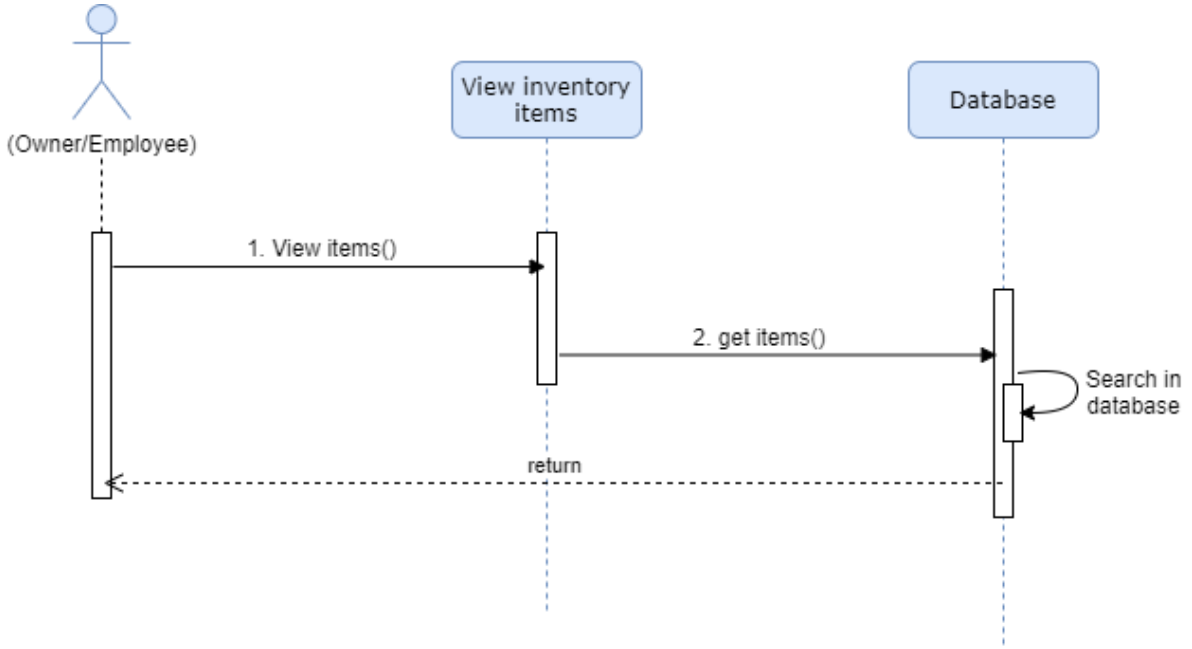
Figure 4.3.1.9.



- The owner and employee add unsuitable items for sale by entering items' barcode.

(View inventory items)

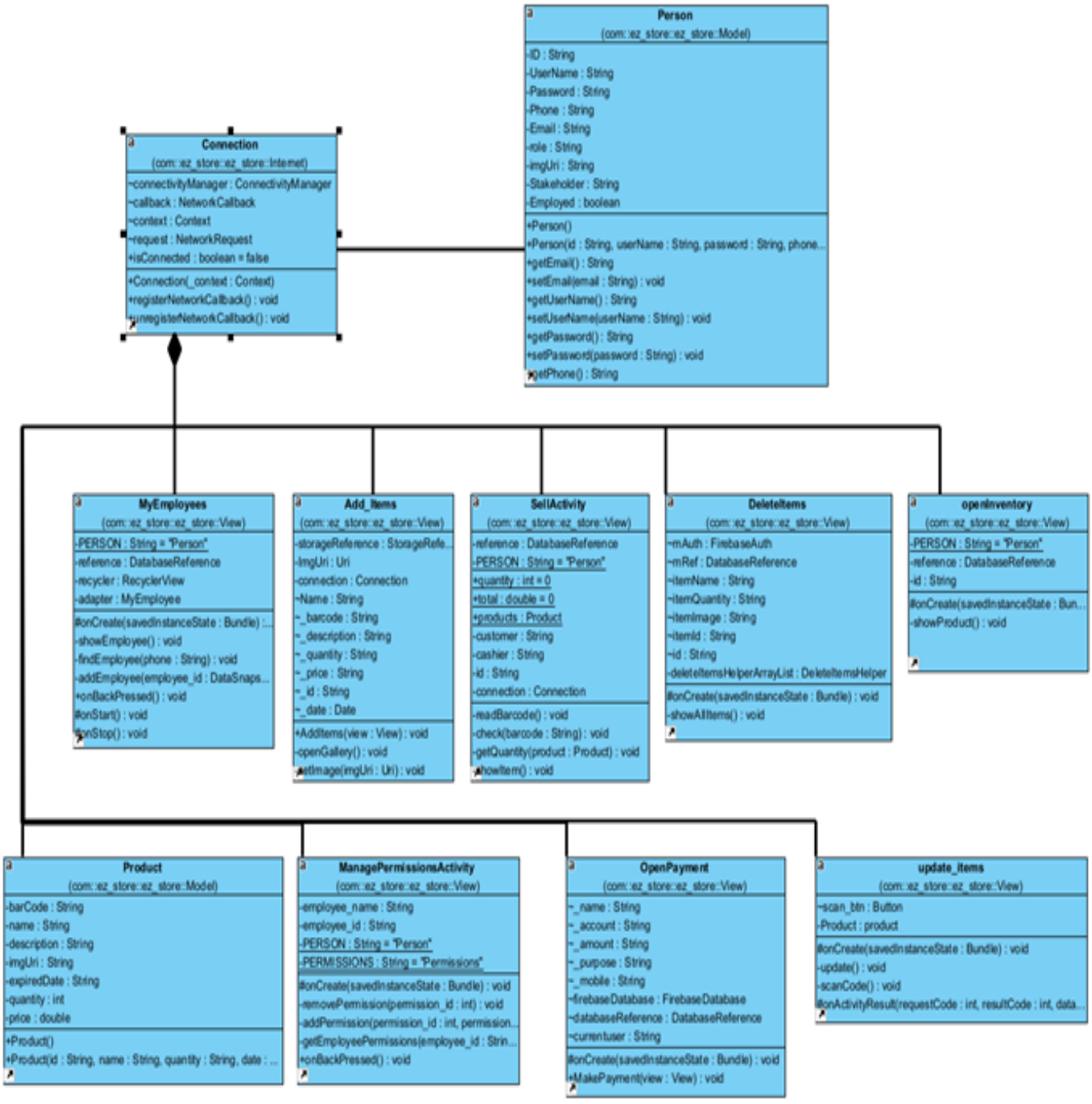
Figure 4.3.1.10.



- The owner and employee can view inventory items.

4.3.2 Class Diagram

Figure 4.3.2.1.

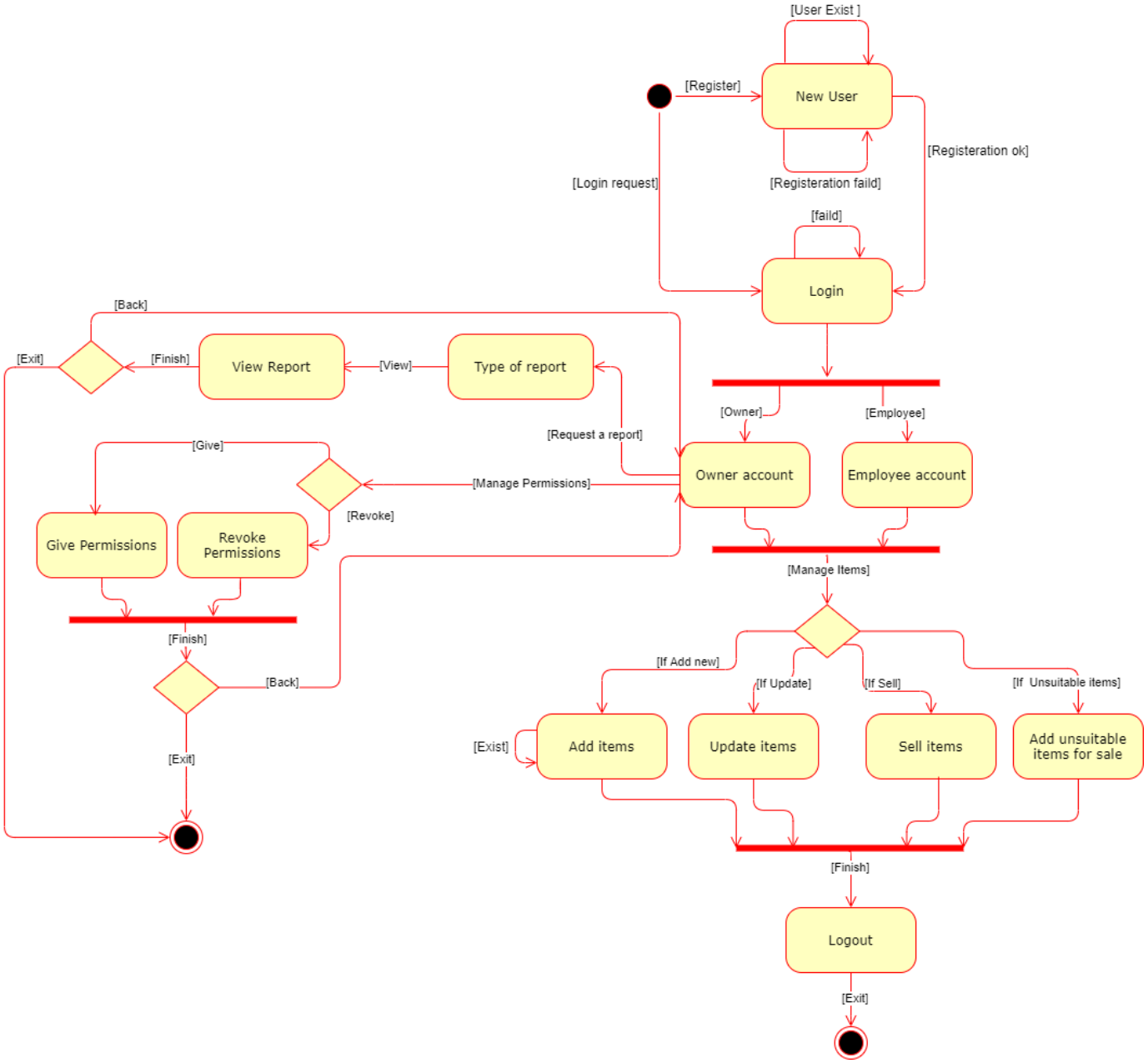


4.3.3 ER diagram

Figure 4.3.3.1.

4.3.4 State transition diagram

Figure 4.3.4.1.



4.4 User interface design

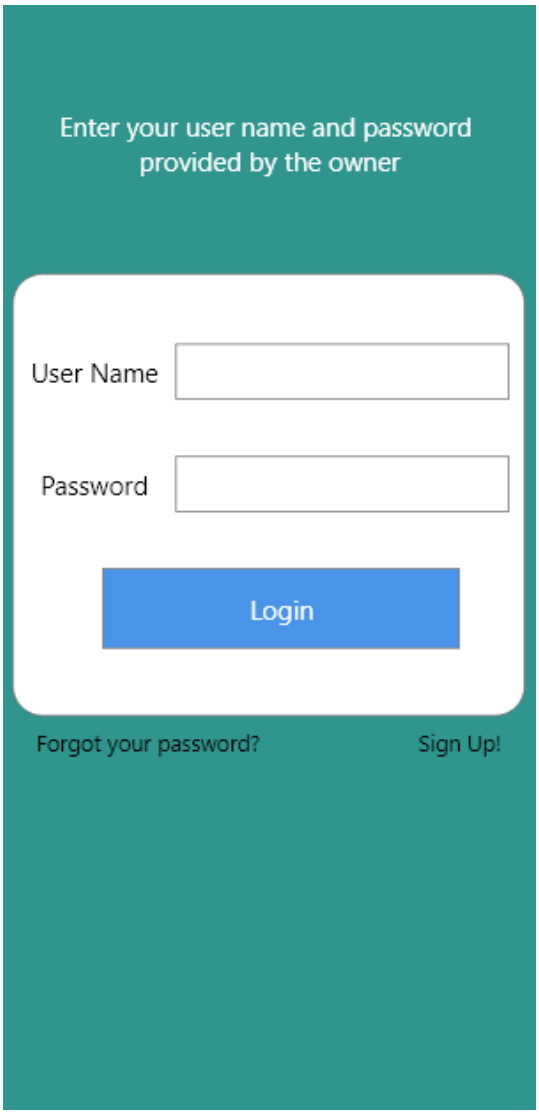
The following figures are how the system is expected to look , can be a subject to change in Graduation project 2

Splash Screen.

Figure 4.4.1.

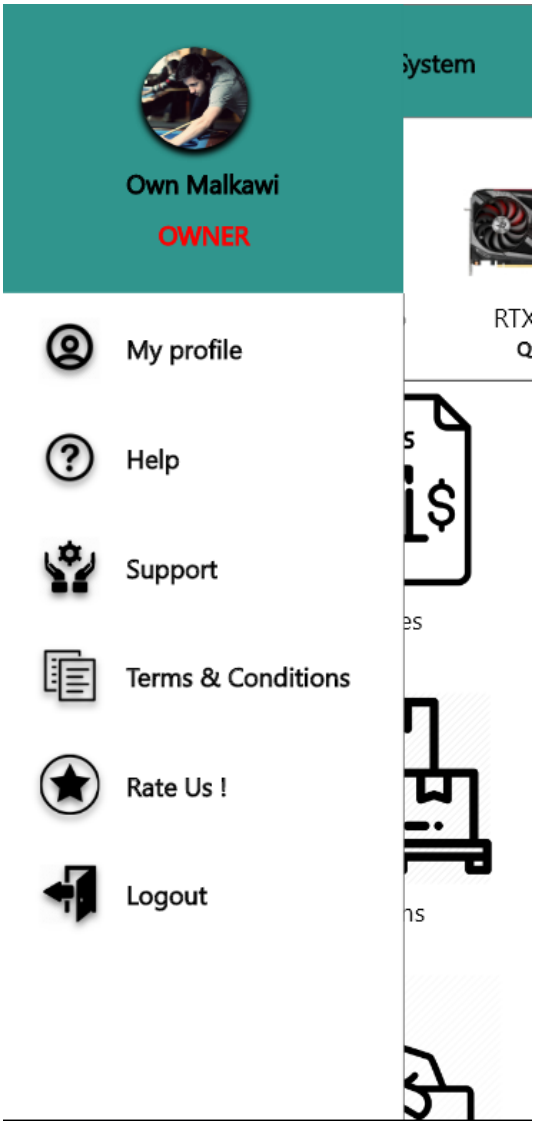
Sign in screen.

Figure 4.4.2.



Side menu.

Figure 4.4.3.



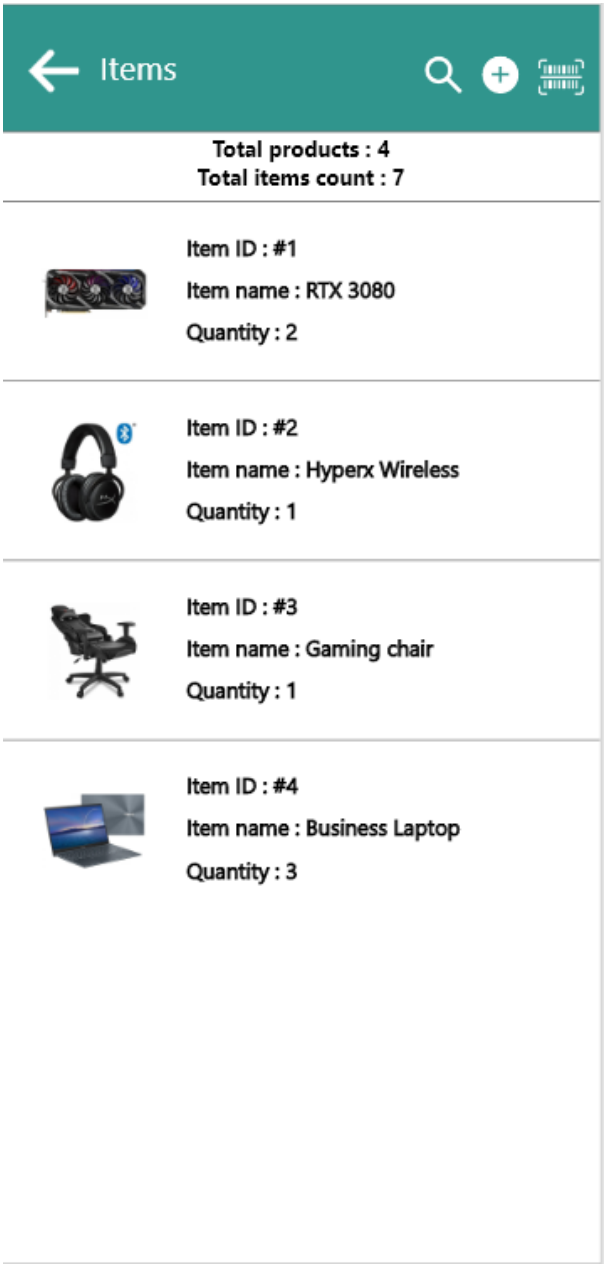
Home Page.

Figure 4.4.4.

Scrolling through items in stock and available features (notes , sales ….).

One of the main functionalities.

Figure 4.4.5.



Here you can search up items , add manually or by a barcode scan

CHAPTER 5: Implementation Plan

5.1 Description of Implementation

Using what's available on the internet or in a correct word in the market we used (API) as the main source of developing the and the reason why we choose it because APIs can access the app components, the delivery of services and information is more flexible, we will use the PHP backend to connect our server that been created with MySQL the view for our application will have two types:

1. A web page development using PHP just for the database and admin
2. Android mobile application
3. Using google servers, we can easily deploy our application and it's very easy to install just head to the play store and type our application name and after registration, it's ready to use.

Or use our local server to use the web page or use our localhost:4000/home.
The main components of our system are :

- Payments
- Profiles and Permissions management
- Inventory analysis (financial reports)
- Suppliers management
- Inventory control (stock, the order of items, placing of items)

for the development, we need supported hardware such as:
A local server : To test database support for users , e.g. retrieving lost data of users.
A PC : main development interface to use the specified API
A lan network
An Android mobile phone.

For the requirements, we need an API for a PHP server license and an API for an online payment(credit card) license to be able to use the testing tools.

5.2 Programming language and technology

Using the latest technologies available at the market, we preferred and discussed it with **Doctor Hamza Al-kofahi** to use API as the main source of development on android studio using JAVA language

And Adobe xd for the prototypes which we showcased earlier.

Technologies used :

- Java se 15 language and (“SQL Server 2019 v15“or “Google’s Firebase”) as a back-end
- Android Studio v4.1.2 xml as a front-end
- Adobe xd for modeling mock-ups of interfaces

5.3 part of implementation if possible

A project file will be disclosed with this document.

CHAPTER 6: Testing Plan

6.1 Black-box

Black box testing is based on the requirements and we will check the system to validate against predefined requirements , And To ensure that the requirement match the system is functional and the input matches the outcome.

In our project we will use it To Test the Functional and None - Functional Requirement And the techniques we'll be using here are:

- Equivalence partitioning Technique
- Boundary Value Analysis Technique
- Decision Table Testing Technique

Test Register scenario

Condition	**validPartitions**	**InvalidPartition**	**ValidBoundary**	**InValidBoundary**
User Name	3-32 chars	< 3chars > 32 chars	2 chars 32 chars	1 chars 33 chars
User ID	4 digits	< 4 digits >4 digits	4 digits	3 digits 5 digits
User Email	15 - 30 chars	> 30 chars < 15 chars	15 chars 30 chars	14 chars 31 chars
User premID	2 digits	< 2 digits > 2 digits	2 digits	1 digits 3 digits

(Table 1-1)Test Register scenario.

Test Login Scenario

condition	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
Valid ID	T	T	T	T	F	F	F	F
ValidPassword	T	T	F	F	-	-	-	-
ID Employee/Owner	E	O	-	-	-	-	-	-
Actions								
Login accepted	T	T	F	F	F	F	F	F
Login as Employee	T	F	F	F	F	F	F	F
Login as Owner	F	T	F	F	F	F	F	F

(Table 1-2)Test Login scenario.

Test Payment Scenario

TC	inputs	Expected outcome	Actually outcome
1	Valid card number Sufficient balance Security code correct	Successfully Payment	
2	Enter card number or security code over 3 times	Decline the payment and ask for card confirmation	
3	Invalid card	Error message “Card is Invalid”	
4	Insufficient balance	Error message “Your balance not enough”	

(Table 1-3)Test payment scenario.

Test Add Item Scenario

TC	Inputs	Expected outcome	Actually outcome
1	Enter Correct itemName and item Id	Successfully Addition	
2	Enter Id as a String	Error message “Characters not allowed”	
3	Item you want to add is already exists	Error message “Item is Already Exists”	
4	Enter the quantity in negative	Error message “Negative quantity not allowed”	

(Table 1-4)Test add item scenario.

Test sell Item Scenario

TC	Inputs	Expected outcome	Actually outcome
1	Enter barcode correctly Enter positive quantity	Successfully sale	
2	Barcode does not work	Error message “Try Manual method ”	
3	Id containing string	Error message “Characters not allowed”	
4	Negative Quantity	Error message “Negative Quantity not allowed”	

(Table 1-5)Test sell items scenario.

6.2 White-box

White box testing Is testing software internal structure, design, coding and the tester must verify the flow of inputs and outputs through the application, to improve design, usability, security.

We have three basic types of test coverage:

1- Statement coverage - the percentage of statements tested.

2- Branch coverage - the percentage of branches tested.

3- Basic path - the percentage of basic paths tested.

We will add it to the graduation project 2*

6.3 Testing automation:

"JUnit"

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development and is one of a family of unit testing frameworks that are collectively known as xUnit that originated with SUnit.

All test cases are mentioned in the black-box

Looking at the market , there are some programs that run flawlessly and we can't have room for error so we decided to use another automation testing plan by using a fully automated testing tool, we will be using **"TestComplete"** , as their promise is **"Ensure the quality of your application without sacrificing speed or agility with an easy-to-use, GUI test automation tool. Our AI-powered object recognition engine and script or scriptless flexibility is unmatched, letting you test every desktop, web, and mobile application with ease."**

CHAPTER 7: Conclusion and Results

This application will result in less waste in every aspect of inventory management, which means you have very little room for error, you can't keep products that are no use to you or your customers, we made sure that this application will alert you and cover anything that will upset the owners of businesses wallets! Of Course with a higher inventory turnover rate and lower costs and faster operations because you are not wasting time organizing items in aisles alphabetically or by the description on paper.

The semester concludes that the project was completely and slowly developed by us. We also conclude that this project has helped us gain more knowledge about the topic that we are indulging ourselves in " android studio, UML diagrams and design " and **The great help and guidance that was provided to us by doctor Hamza Al-kofahi**.

We would be glad to enhance and promote this project if given the chance and help ourselves and especially small businesses, shortly The developed application is tested with sample inputs and outputs obtained according to the requirement we put. Even though we have tried our level best to make it a dream project. Due to time constraints, we could not add more facilities to it. The efficiency of the developed system can be enhanced with some minor modifications. Future development can be made in the proposed system by integrating more services like:

- Implementing it through web pages.
- Enhance the application by adding more functionalities like providing our suppliers to any kind of business associated or using our application
- New effective modules can be added from time to time

In a conclusion we think that our application will solve the problem of expensive inventory management systems, all other applications offer a one-time payment and receive your application, which will be available with our app but the renting part is a crucial part of our system as we guarantee better inventory management, if not ? a refund will be issued, so anyone, especially small businesses will not have any fears with opting in with our services

CHAPTER 8: References

[1]: Doctor Hamza Al-Kofahi

[2]: UML diagrams help
https://www.visual-paradigm.com/

[3]: Android studio user guide
https://developer.android.com/studio/intro/?gclid=CjwKCAiAuqHwBRAQEiwAD-zr3dWfblv64rGya11i9szm-fmekJP4N1so4G6tmLVGl9P9xucWMXFC4xoCNMYQAvD_BwE

[4]: Zoho Inventory management system
https://www.zoho.com/inventory/kb/general-overview/

[5]: SE342 Course: Software architecture & design slides.

[6]: SE430 course: Software testing slides

[7]: Testing information https://medium.com