|  |
| --- |
|  |
| Programming Project Documentation |
| My Little Shop – team F |

Nguyễn Quý Vinh - 10001 (Team Leader)  
Nguyễn Tiến Đức - 9560  
Trần Phương Nam - 9980  
 Lê Quốc Hưng –11084

Table of Contents

1. [**Introduction**](#Intro) **3**
2. [Purpose](#purpose) 3
3. [Document](#abb) Abbreviations 3
4. [References](#ref) 3
5. [**Overall Description**](#descript) **3**
6. [Product perspective 3](#perspective)
7. [Product Features](#feature) 3
8. [User Classes and Characteristics 4](#character)
9. [Operating Environment](#operate) 4
10. [Assumptions and Dependencies](#dependency) 4
11. [**System Features**](#sysfeat) **4**
    1. [Access](#access) Control 4
    2. [Transaction](#transaction) Management 5
    3. [Database](#database) Management 5
12. [**Functional Requirements**](#Functional) **5**
13. [**Non-functional Requirements**](#nonfunc) **6**
14. **[Use Case 6](#usecase)**
15. **External Interface 7**
16. **Design Architecture 10**
17. [Activity Diagram](#activity) 10
18. [Class Diagram](#class) 12
19. [Deployment Diagram](#deploy) 13
20. [**References**](#R) **13**

# Introduction

1. **Purpose**

The report was created to deliver our final project in details as the final solution to our assignment given by Prof. Clavel at Vietnamese-German University.

This report will cover all the completed and uncompleted functions of the software, regarding the given requirements. It will also inform Prof. Clavel all the activities of our team members in the process.

The team’s result will be evaluated by this report and our final project.

1. **Document Abbreviations**

The document includes some acronyms. Their full writing forms will be provided after its first use. The following lines are the list of abbreviations:

* POS: Point of Sale.
* API: Application Programming Interface.

# Overall Description

1. **Product Perspective**

The My Little Shop Project aims to replace the old fashion recording system (recording manually with pen and papers) with the new computer system which can provide user a comprehensive sales report from many locations (the number of shops that the user own) to one destination (their current position).

1. **Product Features**

The new system still allows users to execute all previously available tasks that the old system can fulfill:

* Find the details of a product that the shop is selling.
* Keep track with the transactions of the shop.
* Manage the storage of the shop (keep track with how many products left in the storage).
* Add and remove detail of products.
* Provide the sale reports in a period of time.

1. **User Classes and Characteristics**

* Manager: can access to all the data which is stored in the database and has all authorities to modify them. The manager also has the ability to put the shop’s transactions under surveillance.
* Salesperson: only access to the transactions’ history and doesn’t have the ability to modify, delete anything in the history. They cannot access to the database to adjust any information which is stored in it.

1. **Operating Environment**

The software is based on the webpage platform using JavaScript, HTML and CSS languages and Firebase Real-time Database for backend service.

# System Features

1. **Access Control**
2. **Description and Priority Level**

The users must provide their credentials on the login screen, their credentials then sent to the database. After checking users’ identification, the confirmation will be returned and the users will be allowed to access. This function has medium priority.

1. **Responsive Sequences**

* The salesperson provides the system with his/her credential.
* The system checks and grants access to the salesperson.
* A list of available function which can be executed by the salesperson provided by the system.

The manager has more access authorities to more functions of the system, but the process is the same.

1. **Transaction Management**
2. **Description and Priority Level**

The user can use any built-in or external webcam of the computer to scan the barcode. The extracted information will be sent to the database. After checking, the database will return the name and the price of the product. The system will calculate and show the price of the bill on the computer screen.

This function has high priority because it is the basic characteristic of the system which makes the users to use our product.

1. **Responsive Sequences**

* The salesperson put the barcode in front of the camera. The barcode is captured, processed and its value’s extracted.
* Barcode’s value will be sent to the database. The system will compare the value with the stored data. Then the data will be sent back to the user’s computer.
* The computer calculates the bill based on the price returned from the database and the quantity which is inserted by the salesperson.
* After calculating, the bill will be shown on the computer’s screen.
* The transaction’s details will be sent to the database to be stored.

1. **Database Management**
2. **Description and Priority Level**

The database must record name, quantity of the currently available products of each separated shop. It also keeps track with the business by keep the records of each transaction with date, amount of items and the total money the customer has paid. After each transaction, the database must automatically adjust the quantity in the storage of the products which are bought.

This function has the high priority.

1. **Responsive Sequences**

* After the customers pay the bill, the bill’s record will be transferred to the database. The information will be dated and stored.
* The salesperson can only see the transactions’ history but cannot modify anything in the database.
* The manager can see all the transactions, can modify the price and the quantity of the products, can search the transactions by date.

# Functional Requirements

* The client must provide a quick and easy form for the users to login.
* The server must have a persistent and secure method to store the credentials.
* The barcode-scanner can capture and identify the barcode in at most 10 seconds. If not, the salesperson must enter the value of the barcode into the textbox.
* The returned value of the product must be consistent with the value stored on the database.
* The stored transactions must be consistent with the actual transactions.
* Ensure that only the manager has full authorities to modify the database, and the salesperson only has limited authorities on the database.
* The sale reports must be integrated.

# Non-functional Requirements

1. **Performance Requirements**

The Graphic User Interface (GUI) should be easy to use for both customers and staff. At the Home page, all users could see all kinds of goods, and also their total price. Only managers could access User and Product Management pages. At the left-hand side of the website, managers could add products or users, while all products and users’ data are displayed at the right-hand side.

1. **Security Requirements**

Each user must have a specific username and password to login the My Little Shop application. Only managers are allowed to access User and Product Management to add or delete users, insert the quantity, the initial and selling price of products, while employees compute the total chosen goods, and customers could see the transaction.

1. **Accessibility Requirements**

Managers are able to access every data and backup data stored on the Firebase Real-time system, including justifying products, shops, and users.

# Use Case

1. **Login**

* Primary Actors: Employees, Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: The application requires users to login by texting the user names as well as passwords. After login, customers are only allowed to see goods, employees could calculate the total price of chosen products and regard the sales, meanwhile managers are able to previous tasks and also both add products and users and shops and delete users.

1. **View products**

* Primary Actors: Employees, Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: All users could watch all kinds of goods and their quantity.

1. **Sell products**

* Primary Actors: Employees, Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: Employees and Managers scan or enter the barcodes. After checking in the database, the items will be added to the table on the computer screen and the values will be calculated.

1. **Commit transaction**

* Primary Actors: Employees, Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: The transaction is displayed after the products’ price calculation.

1. **View Sales Report**

* Primary Actors: Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: Managers can see the sale report by choosing the start and end date, then the sale report of that period will be returned from the database.

1. **Add products**

* Primary Actors: Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: New products’ barcode, the initial and selling price are added to the list, which is displayed on “Items” page.

1. **Update inventory**

* Primary Actors: Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: Managers modify the details (quantity, initial and selling price) of the available products in the stock.

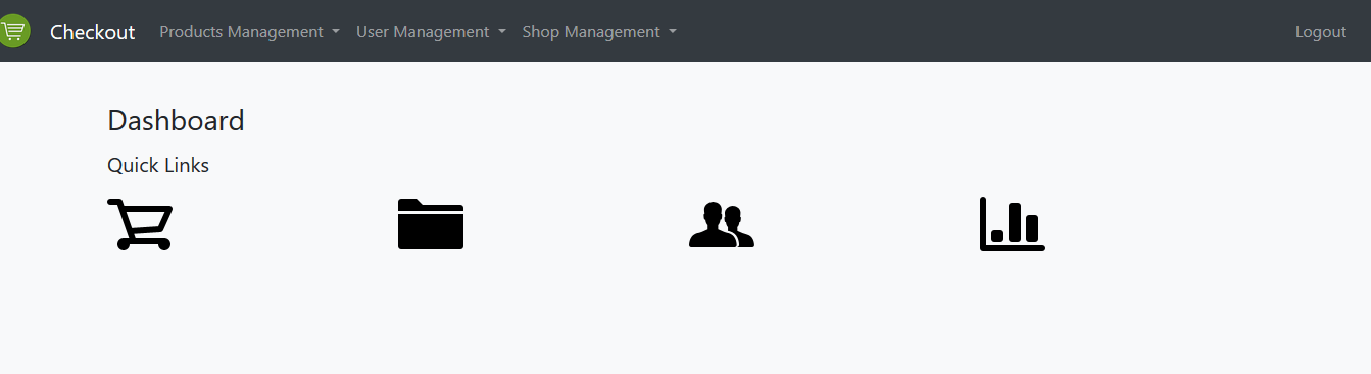
1. **Add users**

* Primary Actors: Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: New users are added to the list which is displayed on “Add User” page.

1. **Add Shops**

* Primary Actors: Managers
* Scope: Online retailers
* Precondition: None
* Main Success Scenario: They add new shops to the application

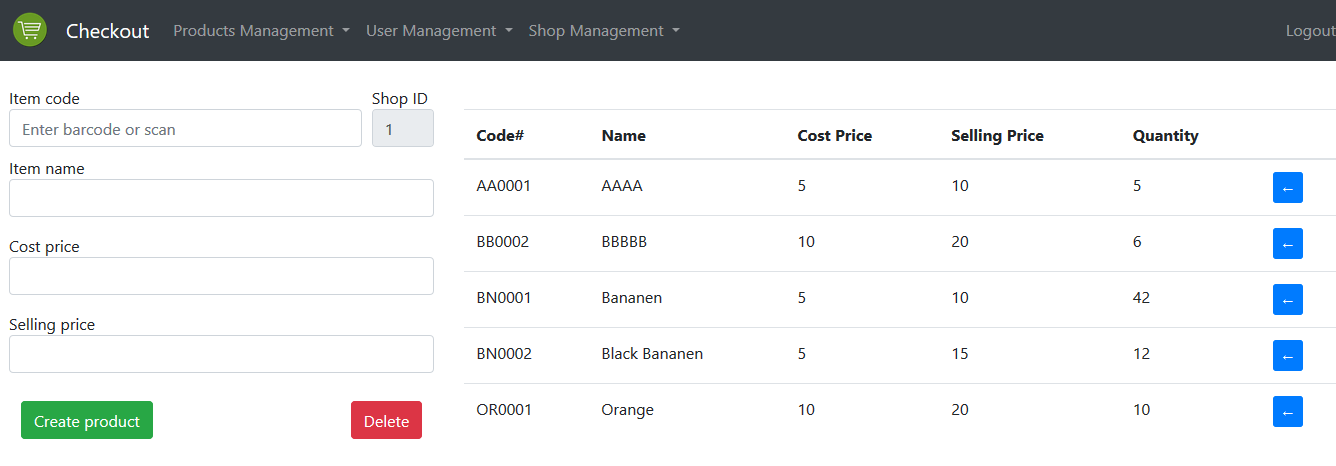
# External Interface Requirements

My Little Shop application requires users to login, then only managers are allowed to access every pages, meanwhile employees could just access the Checkout page.

Quick links for easy access: Checkout, Products Management, User Management, and Shop Management. After access the Checkout

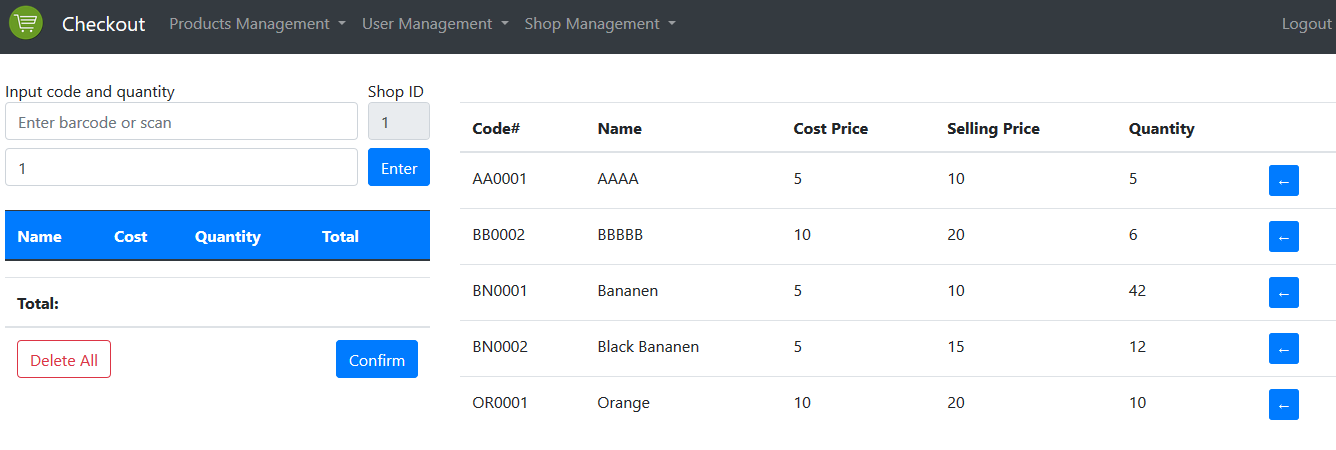
# At the left hand-side:

* Enter: enter the barcode incase cannot scan it.
* Confirm: confirm all the chosen products.
* Delete All: delete all the chosen products.

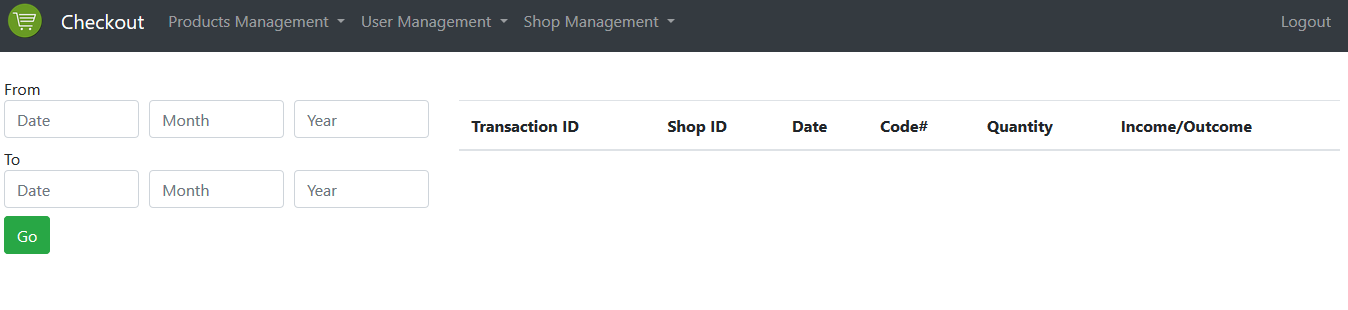
In the Products Management page, there are Create Items, Import Items, Sales Report button. Click Create Items:

Enter Item code, Item name, Cost price, and selling price, then click “Create product” to add product to the list at the right hand-side of the page. Click “delete” to eliminate all the information texted.

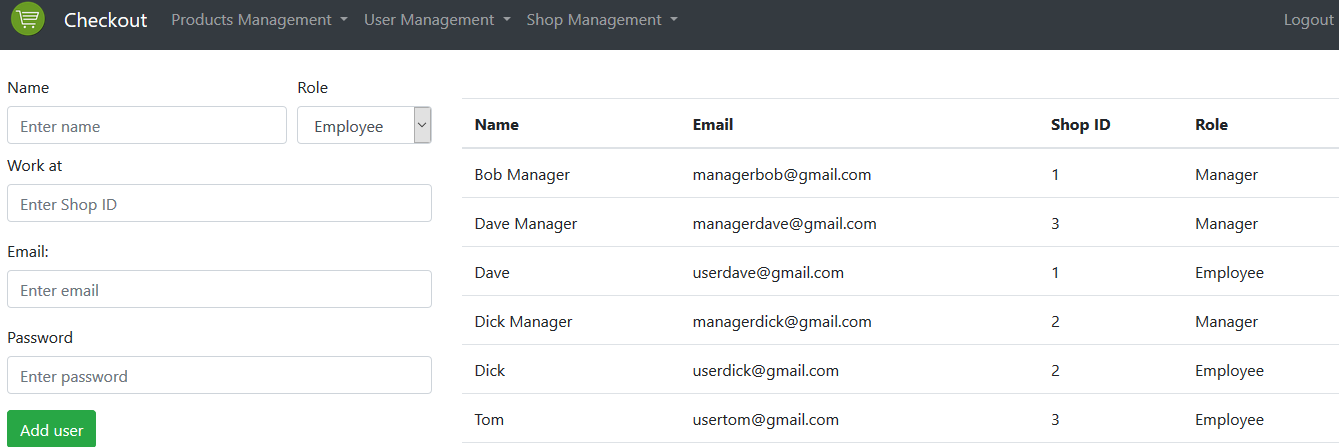
Click Import Items:



Enter barcode or scan and add the number of products required. Click “Delete All” to delete all data texted, or click “Confirm” to add products’ quantity to the list at the right hand-side of the page.

Click Sale Report:

Enter the start date and end date, then click “Go” to display the sale report.

Click User Management, choose “Add User”

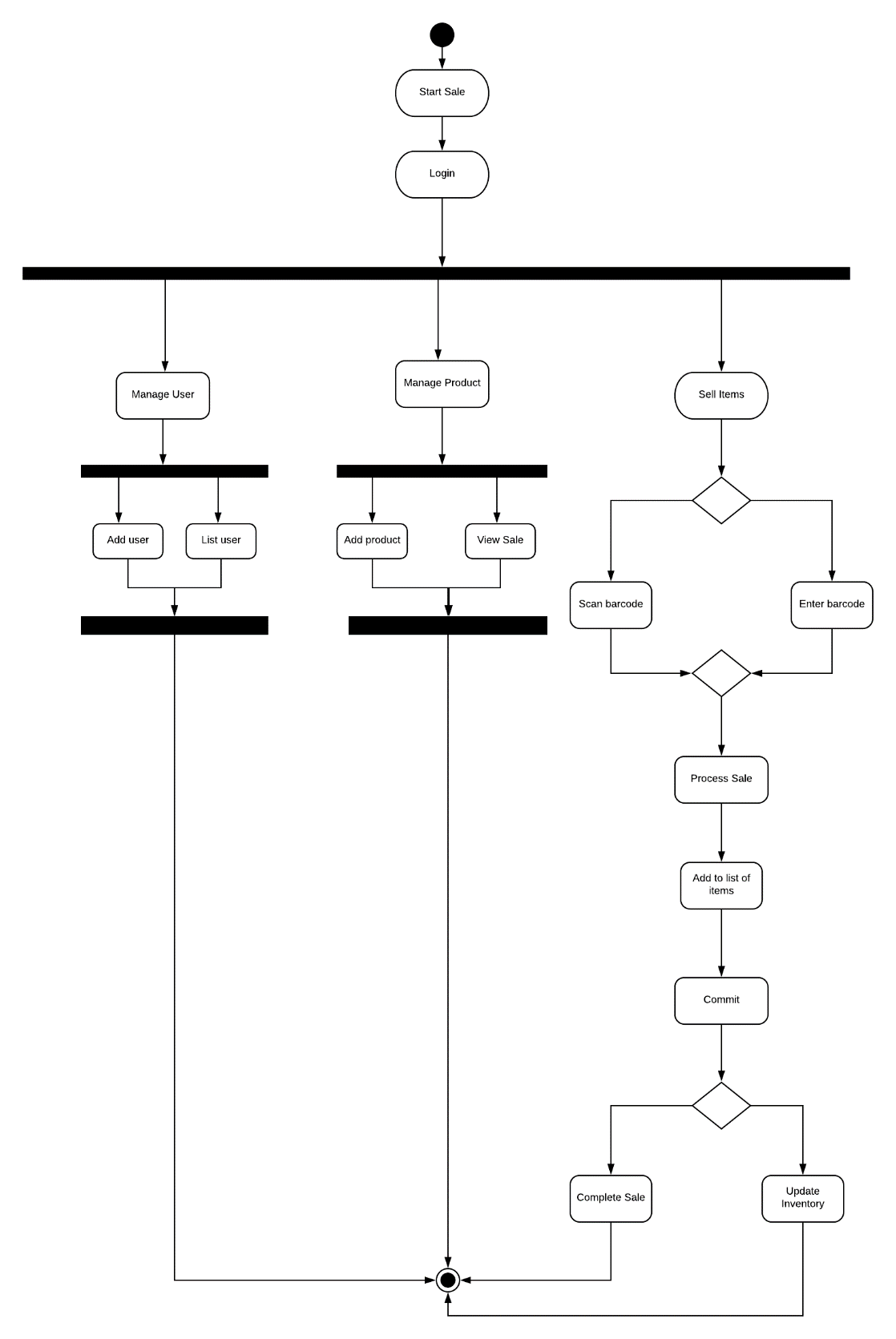
Enter name, shop ID, email, Password, then click “Add user” to add new user to the list at the right hand-side.

Click Shop Management:

# Enter Shop ID, Address, then click “Create Shop” to add shops to the right hand-side of the page.

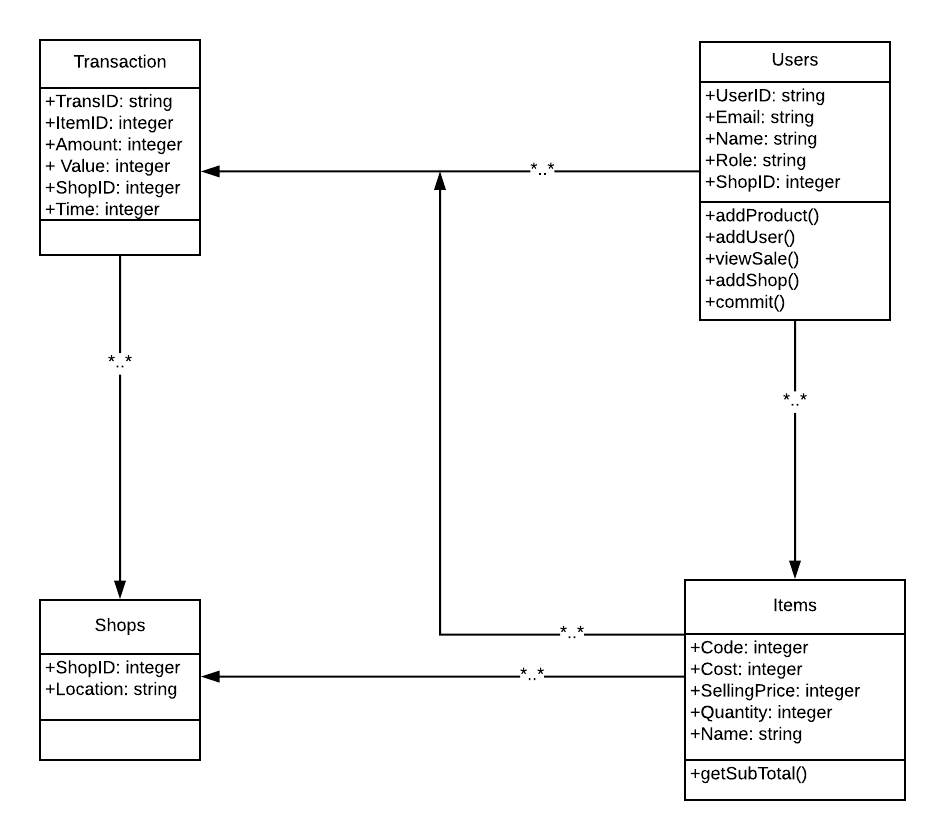
# Design Architecture

1. **Activity Diagram**

*Fig 1. Activity Diagram*

This diagram illustrates the operations of My Little Shop application. After login, both managers and employees can view sales, scan or enter barcode to process sale, and commit transaction, but only managers are allowed to add both users and products, and update inventory.

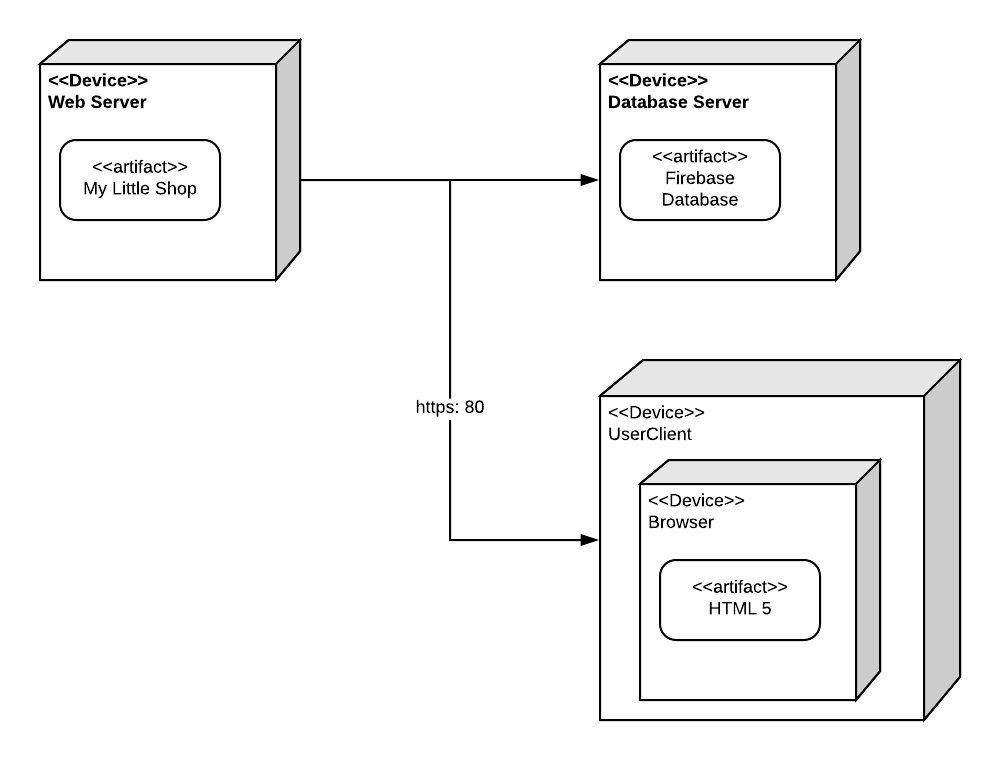
1. **Class Diagram**

****

*Fig 2. Class Diagram*

There are 4 components. Employees and Managers could see the sales reports, and also all specifications of a items. Furthermore, managers ought to both add products or users or shops.

1. **Deployment Diagram**

*Fig 3. Deployment Diagram*

My Little Shop application connects to Google Firebase Database to get Employees, Managers, Sales Reports, Transaction, and Items data, then users are able to view the website programmed based on HTML 5.

# References.

[1] S. Johnson, "Software Architecture Document generated using Rational SoDA template and Rational Rose model.”, 1999, version 1.0.

[2] Dr Dave Sloggett, “Technical Design Document”, 2001.

[3] Prof. B. Meyer, Prof. P. Kolb, “Successful Software Outsourcing and Offshoring”.

[4] QuaggaJS [Online] Available at: https://serratus.github.io/quaggaJS/