Fashion Store Android App

Yukti Khurana

Malaviya National Institute of Technology Jaipur

20th November 2019

—

Software Engineering

—

Under the Guidance of

Dr. Girdhari Singh

ABSTRACT

Internet is the most used source for businesses and shopping. Online shopping is an extension for business owners to sell their products and increase profits. This increases the sales and profits of the business. The advent of online shopping has remarkably increased the profits of such businesses majorly due its ease of access. Software applications are used in every part of our daily life but still there is need to use management software for handling billing details, customers information, stock details. In order to full fill this gap I designed boutique management software application for cloth stores. This application will reduce manual work and maintain all details in database.

This is an **online fashion store application named – "Fashionista"** for selling clothing and accessories online. There are a variety of products available and customizable.

The robustness and scalability of any application are the most important factors. Hence, the framework and coding are done on android studio using Java, the most widely used object-oriented language for applications and XML for front-end. The need of such an application is proven by the tremendous popularity of Smartphone applications and ease of online shopping for both the vendor, supplier and the customer. This application provides a one-click use for the customers making it highly attractive in today's fast-paced world.

INDEX

INTRODUCTION

1.1 Overview

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, its requirements with respect to consumers. The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. Developing an online platform is a job that requires equal share of technological expertise and sound decision making. Principally built on android, this app offers human experience which makes the shopping experience as satisfying as in a real store. Electronic commerce has expanded rapidly over the past five years and is predicted to continue at this rate, or even accelerate.

1.2 Outline of SRS

This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client and admin see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

This system provides an easy solution for customers to buy the product without going to the shop and also to shop owner to sale the product. This proposed system can be used by any users and it does not require any educational level, experience or technical expertise in computer field but it will be of good use if user has the knowledge of how to operate a smartphone.

1.3 Purpose of this project

Today, Internet is the most used source for businesses and shopping. Online shopping is an extension for business owners to sell their products and increase profits. The complete collaboration of business, internet and software comes under a common term E-commerce Technically, Electronic commerce defines as a term used for buying and selling products using electronic systems such as computers, networks and Internet. This completely involves the developing of software applications, marketing, selling, shares and business.

This project mobile application for clothing store which contains several options to buy, sell and display products. The project also includes showing an admin panel, to control sales and inventory. The mobile applications will be executed by the emulators. The Application also contains Databases. The development process will be object oriented using java for backend. All the project schedules and time taken for different steps in developing the application are given in other sections below.

1.4 Feasibility of this project

Presently handheld devices such as smartphones are in heavy use around the word. In short, we are using them to ease the work of both vendor and customer.

This android application will help vendors in managing the various types of Records such as client details, supplier details, product list and provide ease in this otherwise hectic process. The vendor recording the details of the customers, major suppliers, profit/loss on paper is a time-consuming and error-prone process.

Moreover, it is difficult to retrieve the previous records of customers, orders, preferences suppliers, product details. This app aims to provide one-click functionality for all these tasks without any hassle.

Android products are cost-effective and are in wide use in today’s world. The overall cost of the system is much less than the revenue generated after deployment. The basic technological requirements of the system are minimal in the view of current availability of common software and hardware platforms.

The project “Fashionista” is developed with the objective of making the system reliable, user-friendly, easier, robust, fast, and more informative and therefore its development is feasible.

1.5 Why Android?

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android has become one of the most popular platforms due to its robust offerings with many new android devices.

It’s most important advantages that makes it a suitable platform for this project are:

1. Open Source
2. Easy to integrate
3. Multiple sales channels and easy deployment
4. Easy Adoption
5. Low Investment & High ROI (Return on Investment)
6. Use of java with a rich set of libraries
7. Universal chargers for mobile applications
8. Several brands of Android to choose from with affordable prices
9. Removable Storage and battery High flexibility
10. Better hardware such as faster processor, more RAM, increased battery capacity and better screen resolution.
11. Google Play Is More User-Friendly
12. Easy to maintain data.
13. Less time-consuming.

Android comparatively has a low barrier to entry. Android provides freely its Software Development Kit (SDK) to the developer community which minimizes the development and licensing costs. The development costs can be divided into three stages: Application Development, Testing and deploying the android mobile application. This application developed in android technology can install in any smartphone device. All these factors contribute to the choice of this platform for this project.

1.6 References

https://developer.android.com/reference/android/app/package-summary

https://stackoverflow.com/users/878126/android-developer

https://www.tutorialspoint.com/android/index.htm

https://www.studytonight.com/android/

SOFTWARE REQUIREMENTS SPECIFICATION REPORT

2.1 Introduction

This document is meant to delineate the features of OSS (Online Shopping System), so as to server as a guide to the developers on one hand and a software validation document for the prospective client on the other. The online Shopping app "Fashionista" is intended to provide complete solutions for vendors as well as customers through a single get way using the internet. It will enable the vendors to setup online shops, customer to browse through the shop and purchase them online without having to visit the shop physically. The administration module will enable a system administrator to approve and reject requests for new shops and maintain various lists of shop category.

2.2 Purpose

The purpose of this mobile application system is to implement the computerization of the clothes inventory and sales.

2.3 The perspective of the product

This app is a solution that will allow online shopping through mobile devices like the Tablet/smartphones and manage the operation various businesses.

2.4 Assumptions and Dependencies

- It should be developed for Android devices.
- Administrator is created in the system already.
- Roles and tasks are predefined.

2.5 Predictable System Evolution

The system must be portable and scalable so that its user—friendly and can be adapted to any type of establishment e.g. neighbourhood or shop etc.

2.6 Existing system

Earlier in general vendors or boutique owners who used to record their shop details like customer, product selling price, product purchase price, discount on that product on paper and searching in the book. He/she used to search in books whether the client has to pay any amount in past and how much quantity of material still remaining in the shop. He uses a calculator to calculate the total amount of the bill.

To maintain different databases, the garment manufacturing companies use the following systems:

- Paper-based
- MS-Excel
- VB Software

2.7 Disadvantages of the existing system

- Time-consuming.
- Waste of paper.
- High cost
- Not reliable
- Difficult to search and remember the details.

2.8 Proposed system

The proposed system aim is to reduce the difficulty of remembering the details and manage the shop details easily. At the same time, digitalising the commerce of clothes and women's accessories for increased sales and ease of use for vendors and customers.

2.9 Advantages of the proposed system

- Less time-consuming.
- No paperwork.
- No use of Calculator.
- Fast and Robust
- Reliable and easy
- Interactive and user-friendly
- Less human effort.

2.10 Interface Requirements

2.10.1 User Interface

- Login and Signup page
- Home Page containing store's products
- If user select any product, it should open another screen for displaying all the products' information
- Cart Activity where user can add products of choice while browsing.
- After user places an order, a separate shipment details form should open up to confirm the order
- A form for editing user profile info and profile pic which will be displaced in home page to signify the user account.

2.10.2 Hardware Requirements

- System: Intel i3 3rd generation.
- Storage: Minimum GB
- Ram: Minimum of 2GB, 4 GB is recommended for optimal performance

- OS: 64-bit environment preferably for Android 2.3.x and higher versions.
- Video: 1280 x 800 pixels or higher on a 10-inch device
- Processor: Intel Atom® Processor Z2520 1.2 GHz, or faster processor
- Android Smartphone/device
- Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable. The system requires Database also for the store the any transaction of the system like MYSQL etc. system also require DNS (domain name space) for the naming on the internet.

2.10.3 Software Environment

The following are the environments for smart phone application.

- Platform: Windows
- The Operating System: Windows 7/8/10
- Front-End Tool: XML
- Back-End Tool: Java
- Tool Kit: Android SDK
- IDE: Android Studio (version 3.0 or higher)
- Minimum API Level: 23 - Android 6.0 Marshmallow App will run on 62.6% devices approx.)
- Test Deployment: Nexus 5X API 25
- Final Deployment: Mi Redmi Note 4 (installable over all Android devices)
- Database: Firebase and SQLite Database
- Storage: Firebase Storage (Online)

2.11 Functional Requirements

2.11.1 Project Description

The project develops a clothing and accessory store mobile phone application running on windows and Android. Using this mobile application people can start shopping online, buy goods and pay the amount.

This application has various modules or pages, All the functions possible in this module are described separately. Each Module or function have a set of tasks to accomplish. The operation of this project is divided into two different interfaces – Admin and Customer.

2.11.2 Customer Panel

The customer interface is the page for customers to browse and search products. It is the page which is visible to the customers and users. It cannot make any changes to the website or the database. It is accessed by the end user to shop products and make payments. The Customer Interface contains of several functions and modules. Each module is defined separately with explanations.

2.11.3 Vendor/ Admin Panel

To manage all the customer interface modules, we need to have a master admin. This module provides the entry to the backend of the clothing store where admin can add product of various categories listed. On clicking the Admin Login button, the admin can enter the private activity page only if he/she has login credentials and registration for the same is not allowed.

The admin activity opens up showing a variety of options to the admin including adding, viewing, maintaining and editing new products in various categories like tops, skirts, shoes, accessories, makeup, dressed, pants, goggles, hats, winterwear and much more, in the database. The admin can keep a track of all the products ordered including the date and time of orders placed. Only after the approval of the admin, will the user's order be successfully placed and sent out for delivery.

2.11.4 Login

Each and every user have separate profile in the application, which stores their profile information, recent orders, and shipping addresses. Before checking out from the cart, Customers should login into their accounts. If they do not have an account or profile in the application, they should create one.

The application will ask for a username and password, previously created in the Database by both the user and the admin window. The system must ensure that the username and password are valid, which otherwise will be left to the user in the login screen. It provides a remember me option to prevent automatic logout for easy use by users on their smartphones.

3.1.5 Register

For first time users, the app will provide a functionality of easy signup by filling up just three required fields: name, phone and password. These profiles are all saved in databases. The databases are designed in form of these profiles

3.1.6 Home Page

After becoming a member, the user can login and enjoy the variety of products available. After login the user is send to a home activity with products and their information and various options to buy, place order, add to cart or update profile.

3.1.7 Forgot Password

The user may also set security questions in case they forget their password and add a profile picture too. All the user activity shall remain stored in each login. No user data is leaked or used by any admins and remains completely discreet.

2.11.8 Orders

The products added in cart can be placed for order. On placing any order, the user will be given a final order amount and will have to fill a small form about delivery address and payment amount. The app will allow one-click shopping experience for users and the order is sent for delivery only after being approved by the admin. The date and time of order shall be recorded and sent to the admin for approval.

2.11.9 Update Profile

Member clients must have an option to edit customer data after logging into their accounts, in where it will display a window with the data of the client, allowing you to modify all data except for password that can only be edited after answering the security questions set by the user. The editable fields should include: Name, Address and Phone. Security questions can also be updated once logged in successfully.

2.11.10 Inventory/ Product Module

The admin can view all the products with their names, descriptions and prices and edit them based on present stock and availability. Every product will be identified by a product id. The product details must include its price, brief description, image for user's ease of selection and price.

2.11.11 Unlimited Products

This module or functions states that there can be unlimited products listed for sale. There is no limitation for browsing the product. The products displayed should be in stock and should be available for sale for getting added in to the cart.

2.11.12 Unlimited Product Categories

As there are Unlimited Products with various categories like dresses, shoes, hats, pants, sportswear, winter wear, heels, goggles and much more. There is no limitation in the categories of products also.

2.11.13 Shopping Cart

After shopping and browsing the products, the customers can add the products to the cart (Shopping Cart). The cart application is coded in such a way, that it sums to give the total amount of all products added.

2.11.14 Check-out

The shipping address and billing address of the customer should be completed before check out. This information of the customer should be secure and this page should not have any loopholes in the code, that makes it easy for eavesdropping.

2.11.15 Shopping Cart Application

After shopping and browsing the products, the customers add the products to the cart (Shopping Cart). The cart application is coded in such a way, that it adds tax according to the given state. Also gives the option to pay the outstanding balances. It has an option for entering any codes for reducing prices such as gift cards or promotional cash, etc.

2.11.16 Search Products

This option is the most important of all the modules, because every customer type in the product he want in the search bar to make his/her shopping experience easy.

2.11.17 Order Status

Customers should be able to track their products once the shipping from the ware house is done.

2.11.18 Databases

All the customers and the users having profiles or accounts in application are saved in the user accounts database. All the products bought by the customers are stored separately in orders table while the unlimited products in the app are added through the product database.

2.11.19 Update Prices and Product Details

Prices of the products change eventually, so admin panel should be able to update prices and banners. Banners are posters that are displayed in the home page for bringing up the deals and offers.

2.11.20 More Informative

The application must be more extensively informative storing all the necessary information in reliable database without any scope of redundancy and glitch.

2.11. 21 Smartphone Application Activities

The Activities involving for smart phone application development are given as follows:

- Application Software Architecture
- Designing Database Tables
- Coding for the Application in Java for backend, XML for front-end and SQL for database.
- Designing framework for Smartphone Application
- Implementation using mobile emulators like Nexus 5X API 25.

- Testing and debugging the Smartphone application on Android emulator as well as deployment on the Android Smart phone.

2.11.22 Project Schedule

The time taken for completing the whole project is one semester approximately i.e. 8 weeks. The division of work for 8 weeks is divided. The Schedule is done taking the project activities into consideration.

2.12 Non-Functional Requirements

2.12.1 Security

The login into any customer and especially admin account must not happen without proper authentication. The change of password for users is allowed only after meeting specific security requirements like security questions. The system should not leave any cookies on the customer's computer containing user's password. The system's backend servers shall only be accessible to back-end administrators. Sensitive data will be encrypted before being sent over insecure connections like the internet. Even the Admins must not have access to any personal customer information specially passwords.

2.12.2 Reliability

It must have good component design to get better performance. The system must provide storage of all database on redundant computers with automatic switchover. The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backbone of the database which is continuously maintained and updated without any anomalies to reflect the most recent changes especially when users place orders. The app should not crash except due to errors on the underlying device on which it is being run. Whenever user perform some important action it should be acknowledged with confirmation.

2.12.3 Availability

The system should be available at all times, meaning the user can access it using the mobile app, only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24X7 availability.

3.12.4 Maintainability

The software design will be done with modularity in mind so that maintenance can be done efficiently. The addition of functionalities must not require the change in the structure of complete application.

2.12.5 Portability

The application is android-based so it must work with at least 60% of all the android devices available in the market and must be compatible with minimum system requirements, so that it can be easily installed by the users across various devices.

2.12.6 User-Friendly

It must be easy to use for even the users not efficient on electronic devices. The application must provide interactive and attractive GUI for better understanding of the users. User should be able to understand the flow of App easily i.e. users should able to use App without any guideline or help from experts/manuals.

2.12.7 Reusability

The software should be cost-effective and different modules should be coded in a generic pattern to aid in functional reusability. Moreover, modules must be constructed with utmost care to prevent duplication of code/modules.

2.12.8 Correctness

All the calculations specially the price during placement of order has to be accurate according to the latest changes in the database and any updates from the end-user side must immediately be reflected at the user side to avoid anomalies or mismatch of records. All transactions must be accompanied by time stamps and properly linked to concerned user’s account activity.

2.12.9 Efficiency

The mobile application must use minimum resource to provide optimal performance without any delays in the shopping experience of users. Order placement and browsing must be delay-free with the exception only being a slow internet connection. Fast profile updates and search of products is necessary.

2.12.10 Responsiveness

Application should be responsive to the user input or to any external interrupt which is of highest priority and return back to same state. When app gets interrupted by call , then app should able to save state and return to same state/ page which was there before it got interrupted.

2.12.11 Performance

The performance of the Application can be determined by it responsive time, time to complete the given task. When app is made to start up it shouldn’t take more than 3 second to load initial screen. Likewise, it should be made sure that app will not cause any hindrance to the user Input.

2.12.12 Scalability

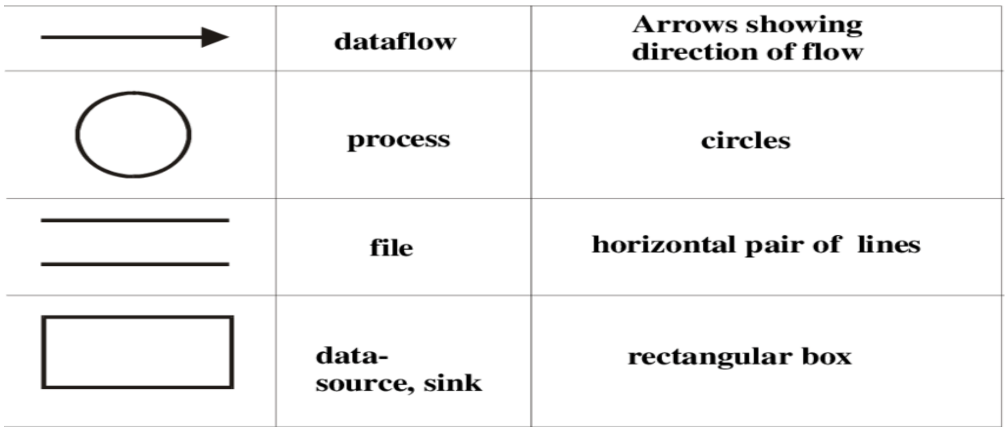
App should able to adopt itself to increased usage or able to handle more data as time progress. When the user data (caches, stored data etc) increases, app should be capable of handling them without delay by optimising the way storage is done and accessed. The increase in the load i.e. large number of user logins at the same time should not cause delay or crashes or any kind of disturbance in online shopping experience.

SYSTEM DESIGNS SPECIFICATIONS

3.0 Data Flow Diagram

A Data Flow Diagrams is a structured analysis and design tool that can be used for flowcharting in place of, or in association with, information-oriented and process-oriented systems flowcharts. A DFD is a network that describes the flow of data and the processes that change, or transform, data throughout a system. This network is constructed by using a set of symbols that do not imply a physical implementation. It has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design phase that functionality decomposes the requirement specifications down to the lowest level of detail.

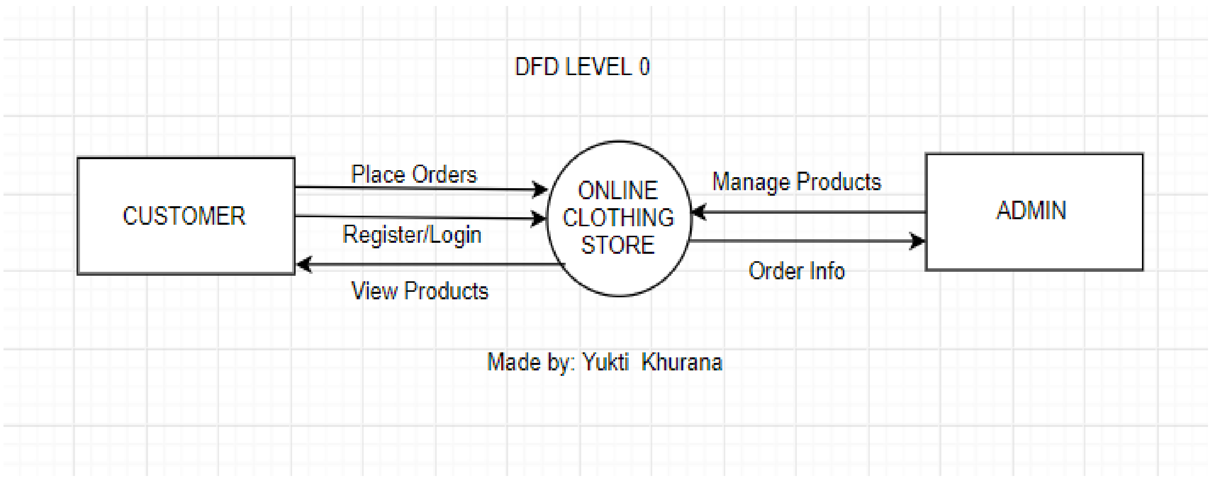
3.1 Data flow Diagrams Symbols



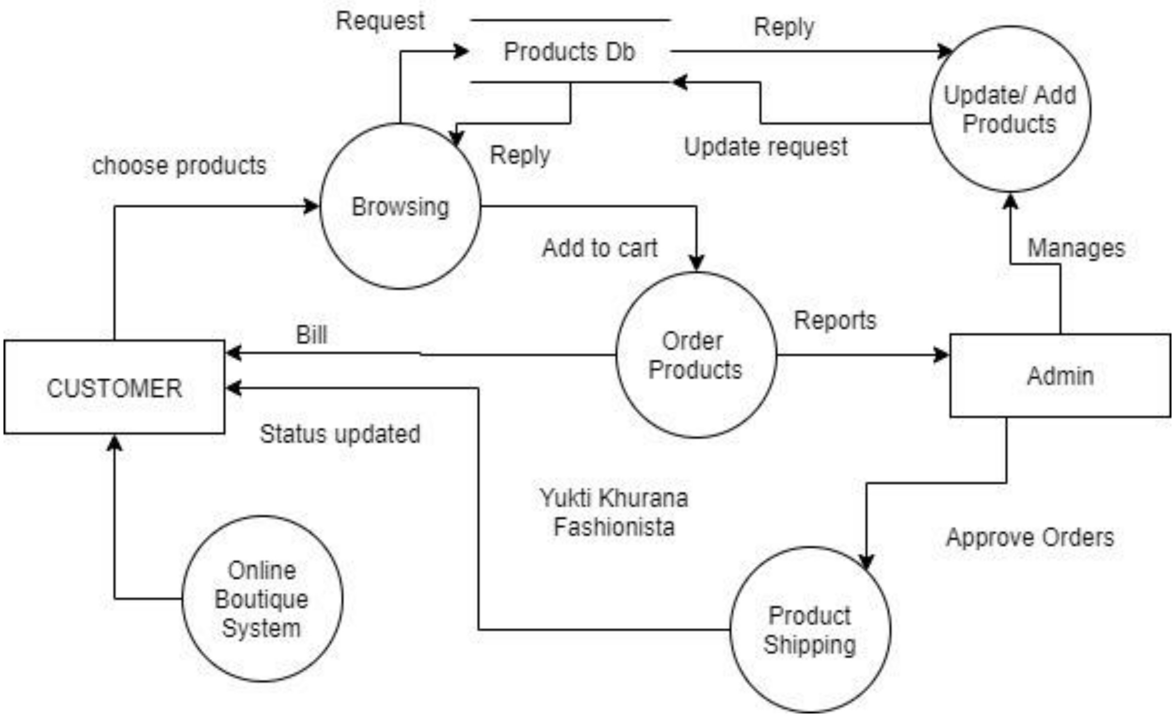
→	**dataflow**	**Arrows showing direction of flow**
○	**process**	**circles**
=	**file**	**horizontal pair of lines**
▭	**data-source, sink**	**rectangular box**

3.2 DFD Level 0

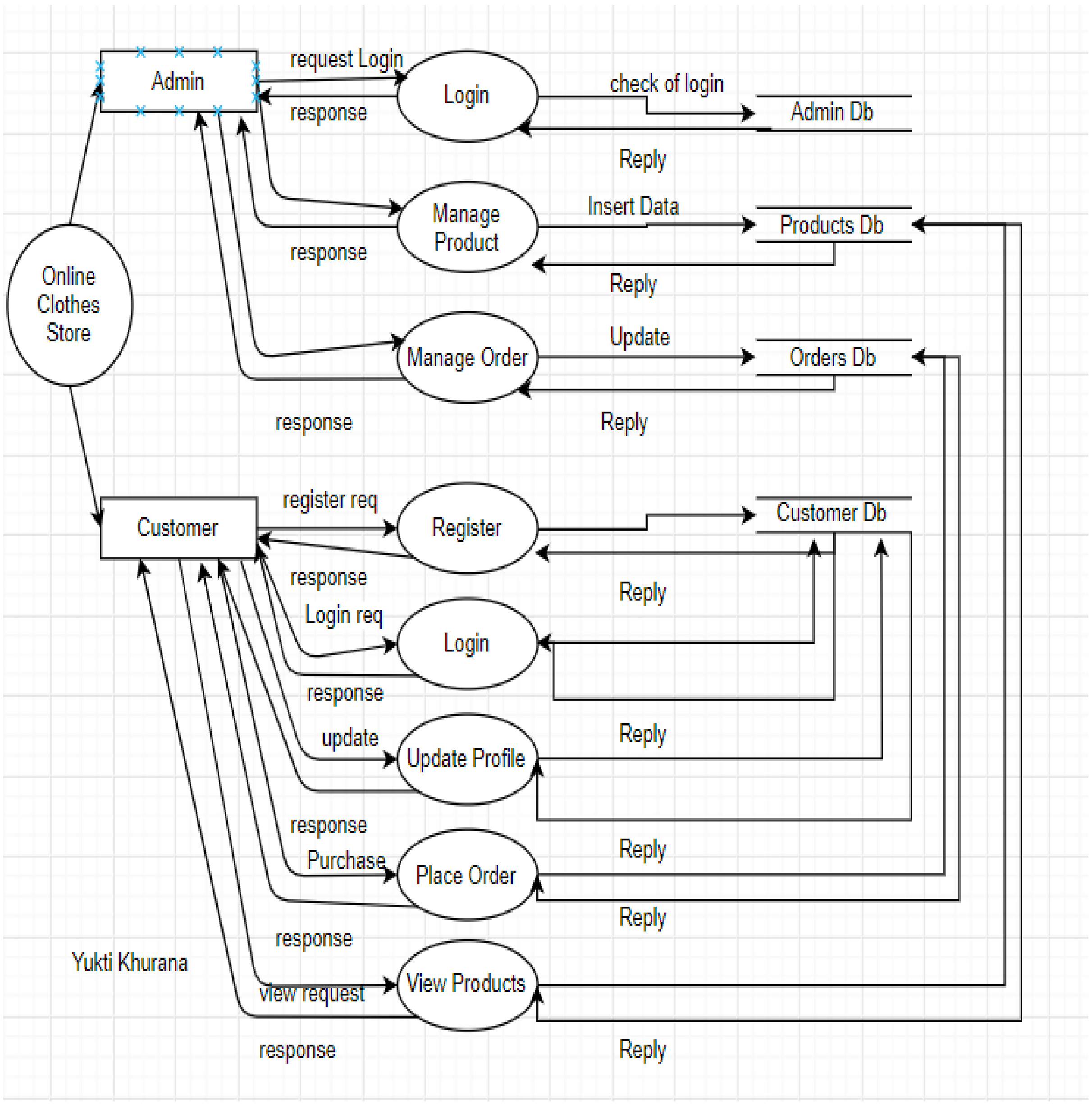
The 0-level DFD describe the all user module who operate the system. Below data flow diagram of fashionista shows the two users can operate the system: Admin and Member user.
The admin will have the access to create Products on the app. Only after the product is added by the admin, the user can browse that product. So, the app is acting as a medium between the admin and customer (buyer). Besides that, the app will serve to provide the details of all product listed there, if a user intends to buy.



3.3 DFD Level 1



3.4 DFD Level 2

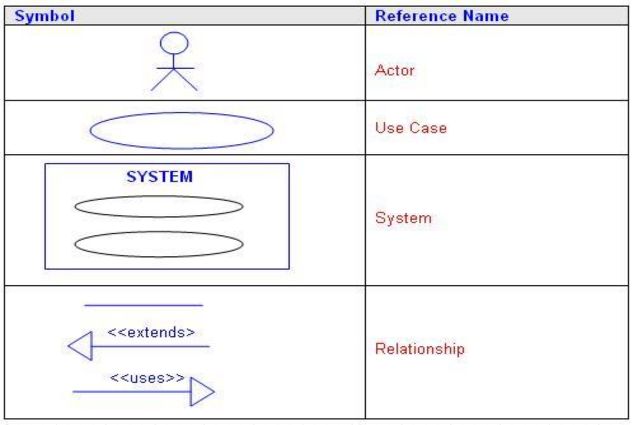


3.5 UML Diagrams

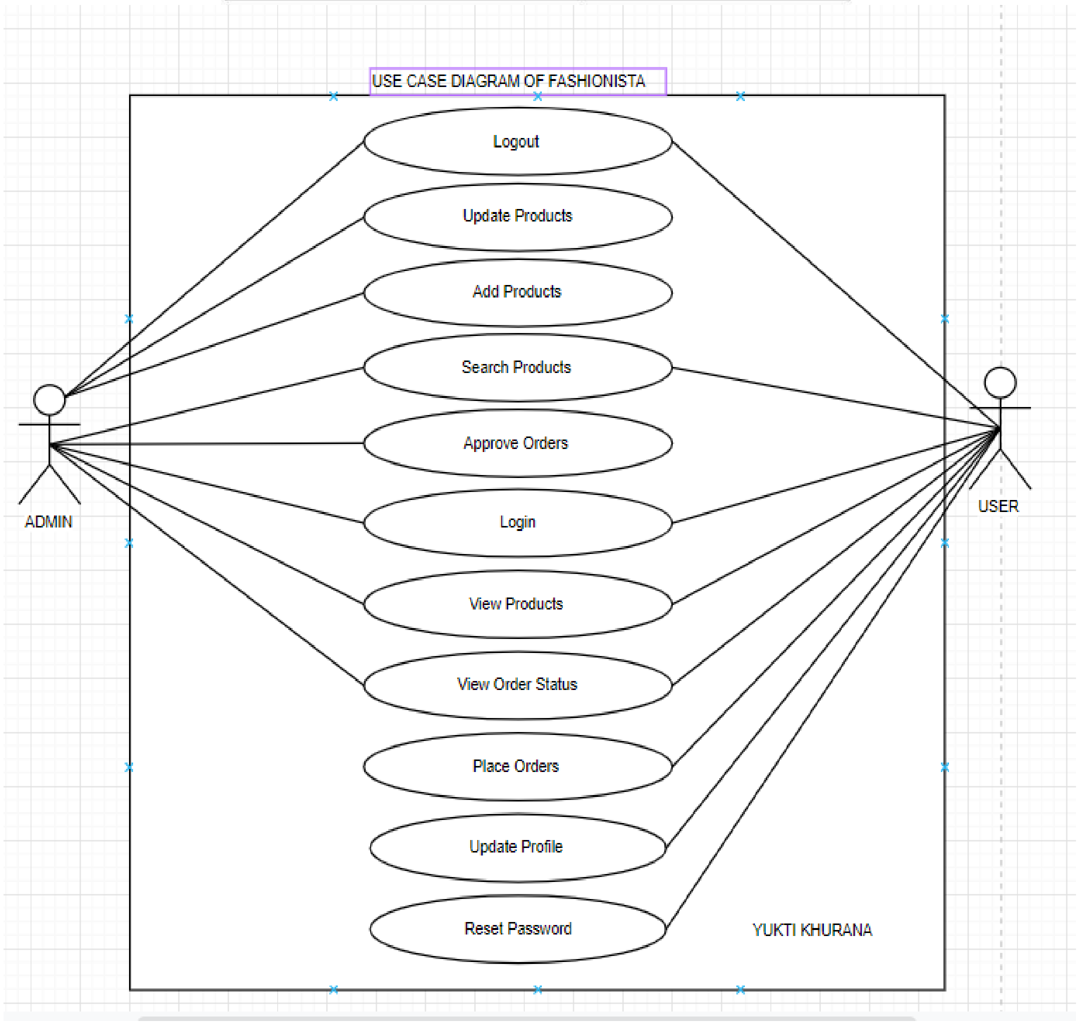
UML is a standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems. UML diagrams are not only made for developers but also for business users, common people, and anybody interested to understand the system. The system can be a software or non-software system. Thus, it must be clear that UML is not a development method rather it companies with processes to make it a successful system. UML is a modelling language used to model software and non-software systems. Although UML is used for non-software systems, the emphasis is on modelling OO software applications. Some of the UML diagrams include: class diagram, object diagram, collaboration diagram, activity diagrams all would basically be designed based on the objects.

3.5.1 Use-Case Diagram

The UML provides the use case diagram notation to illustrate the name of the use case actors and relationship between them. User case diagrams are used to model the functional interaction between users and system.



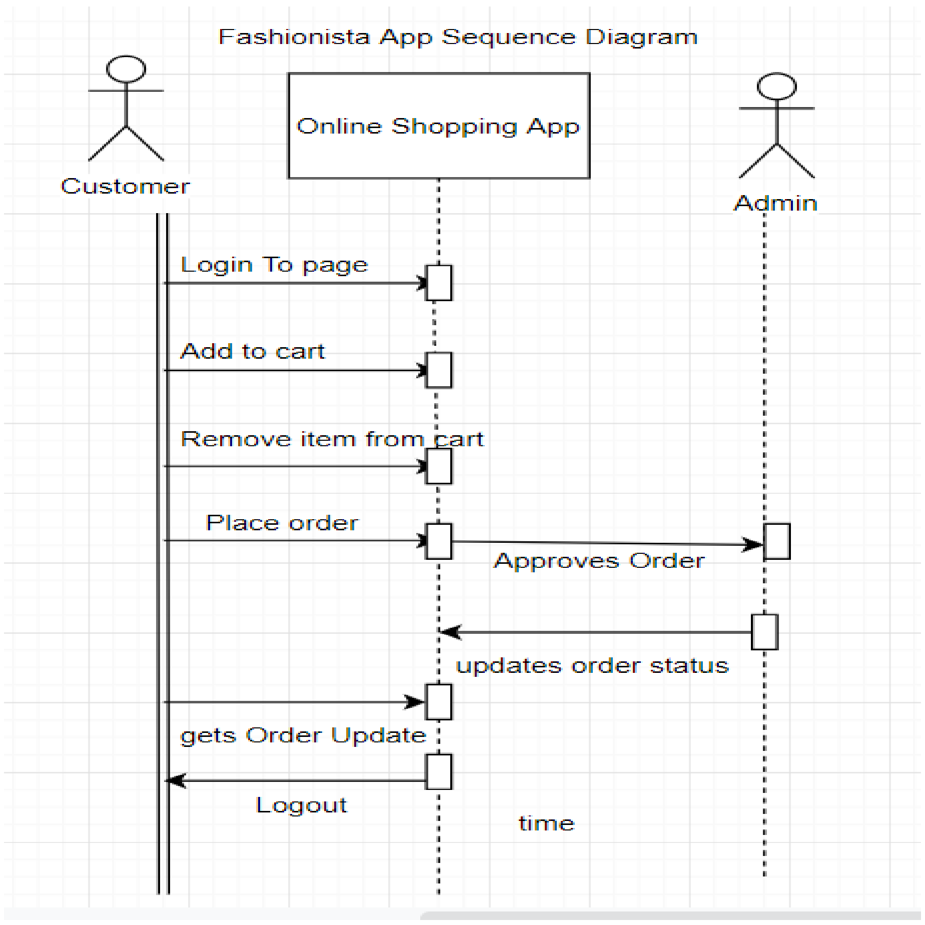
Symbol	Reference Name
	Actor
	Use Case
SYSTEM	System
<<extends> <<uses>>	Relationship



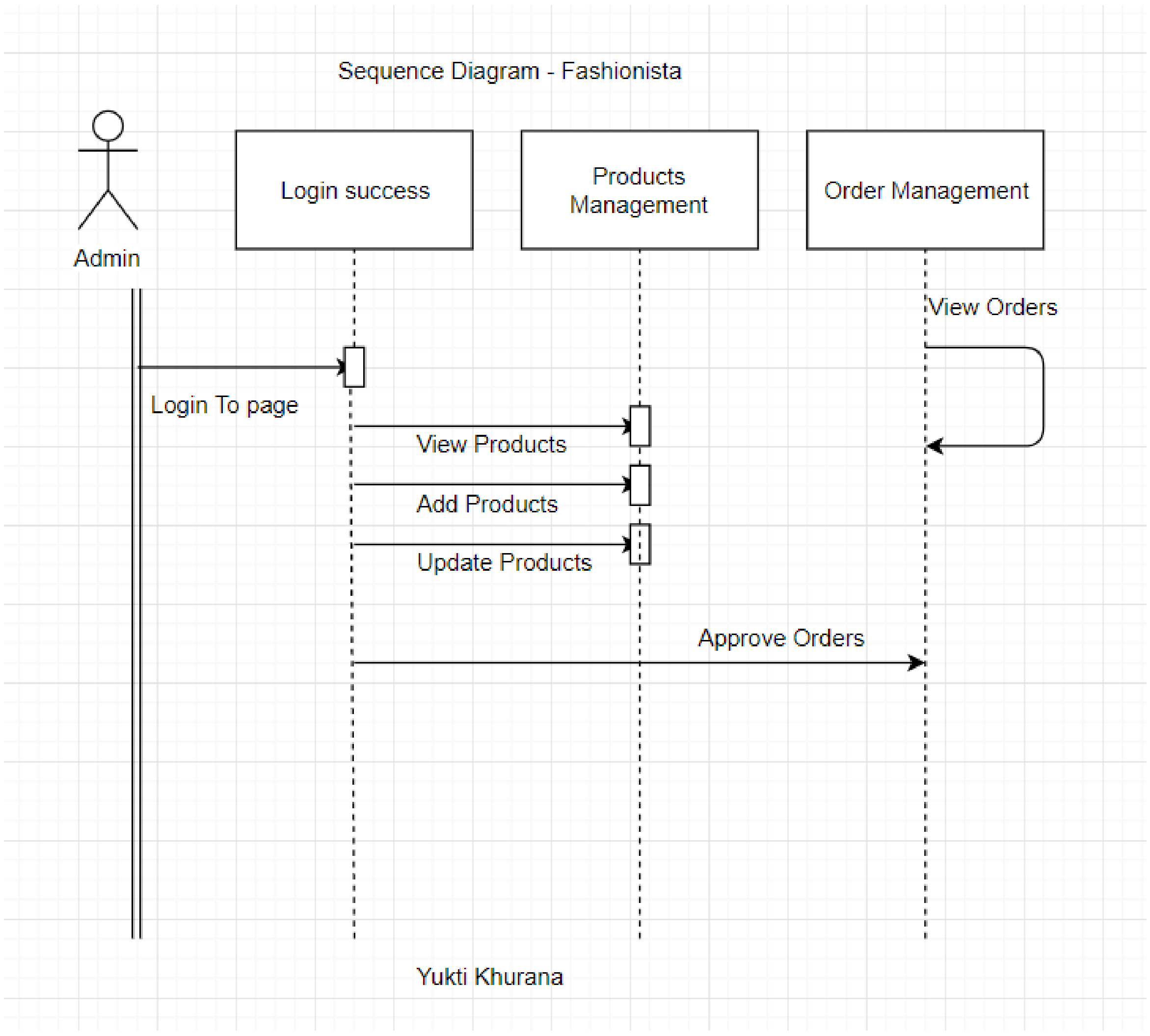
3.5.2 Sequence Diagram

A sequence diagram illustrates in a kind of format in which each object interacts via messages. It is generalization between two or more specification diagram. Sequence diagram is an interaction over view diagram. It provides a big picture over view of now a set of interaction is related in terms of logic and process flow.

3.5.2.1 Customer side Sequence Diagram

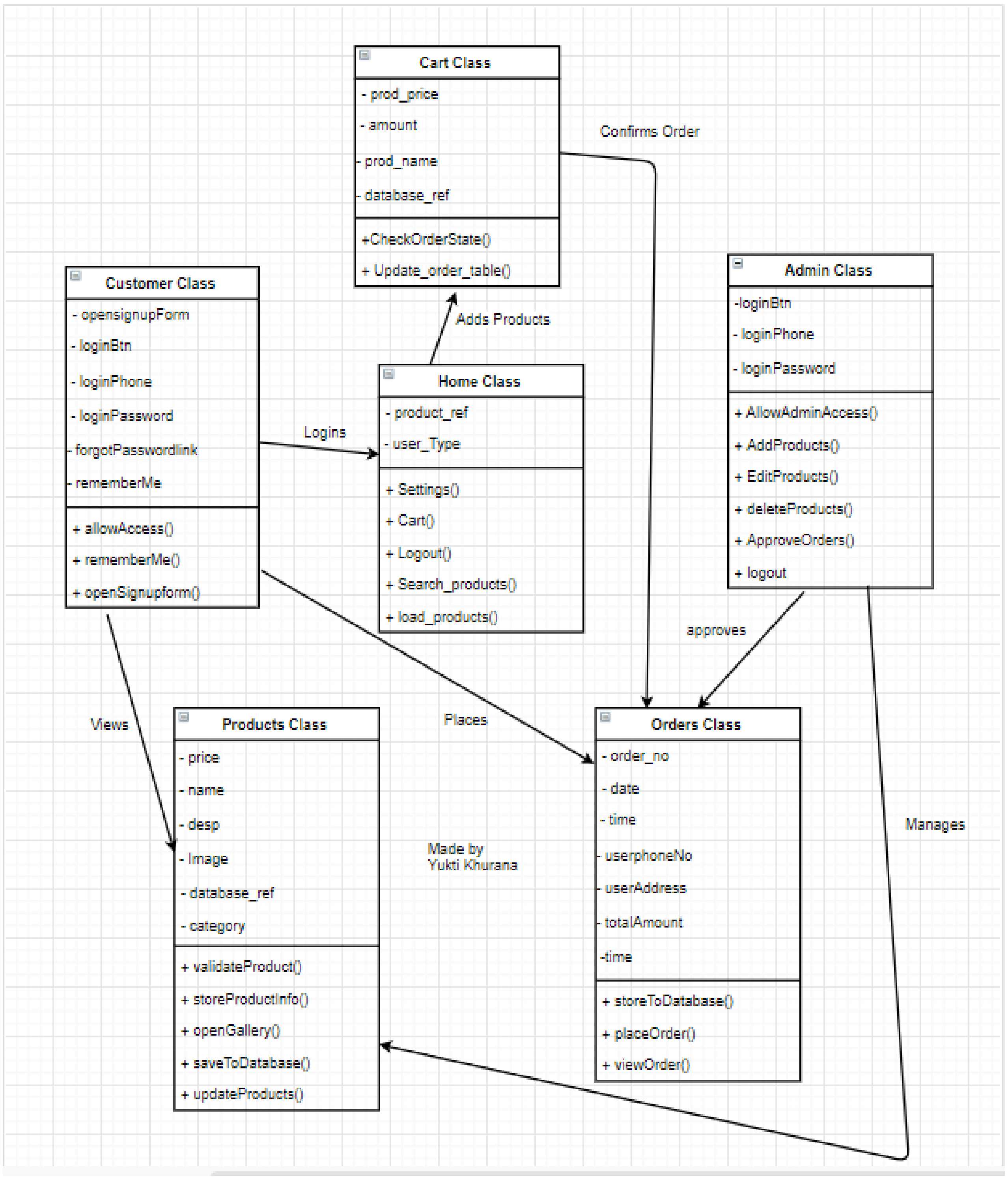


3.5.2.2 Admin side Sequence Diagram



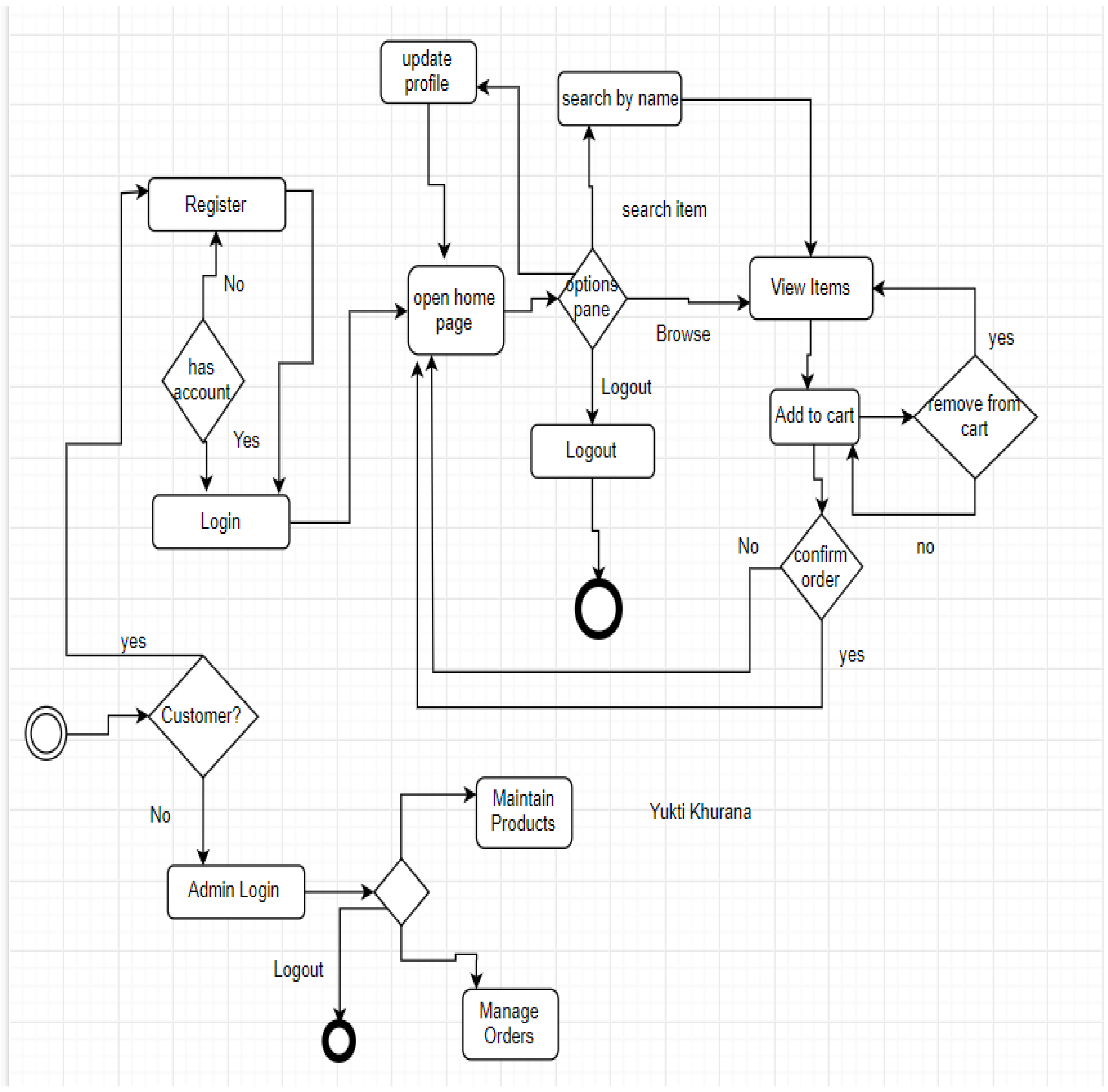
3.5.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.



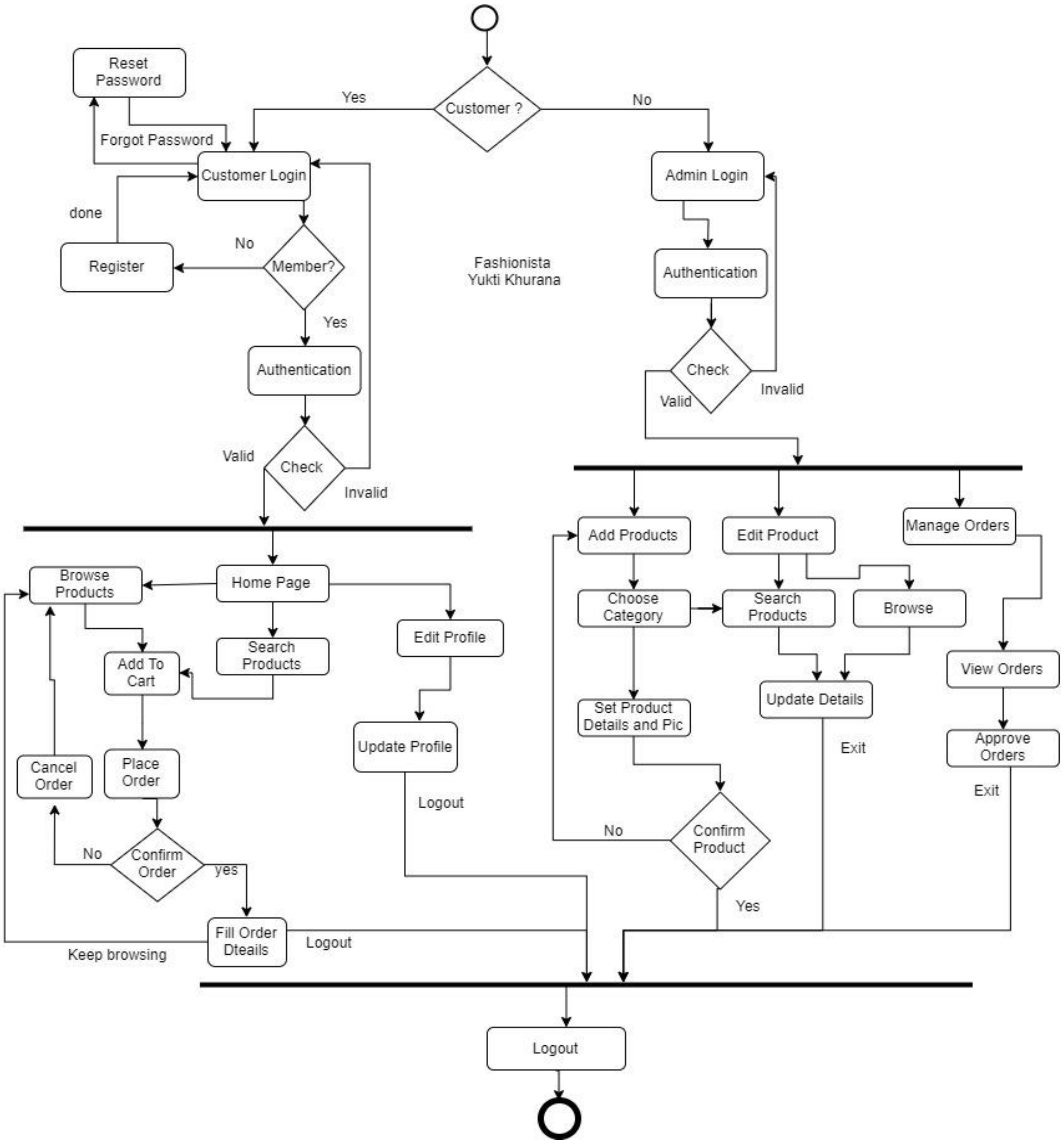
3.5.4 State Chart Diagram

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioural diagram and it represents the behaviour using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. It is used to model the dynamic behaviour of a class in response to time and changing external stimuli.



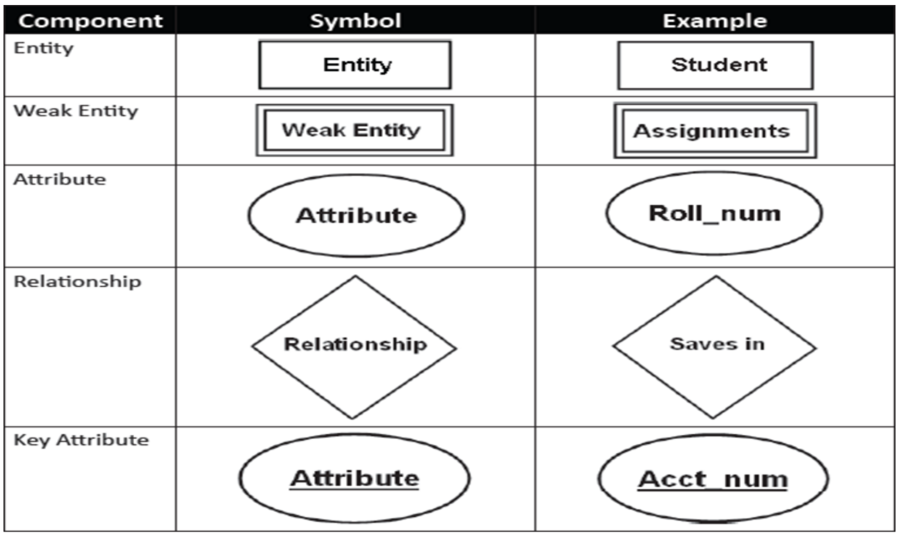
3.5.5 Activity diagram

The activity diagram used to describe flow of activity through a series of actions. Activity diagram is an important diagram to describe the system. The activity described as an action or operation of the system. Activity diagram shows sequential and parallel activities in a process. They are useful for modelling business, workflows, the data flows and complex algorithm. A UML activity diagram offers rich notation to flows a sequential of activities. It may be including parallel activities. It may be applied to any purpose, but it is popular for visualization of business workflows and use case.



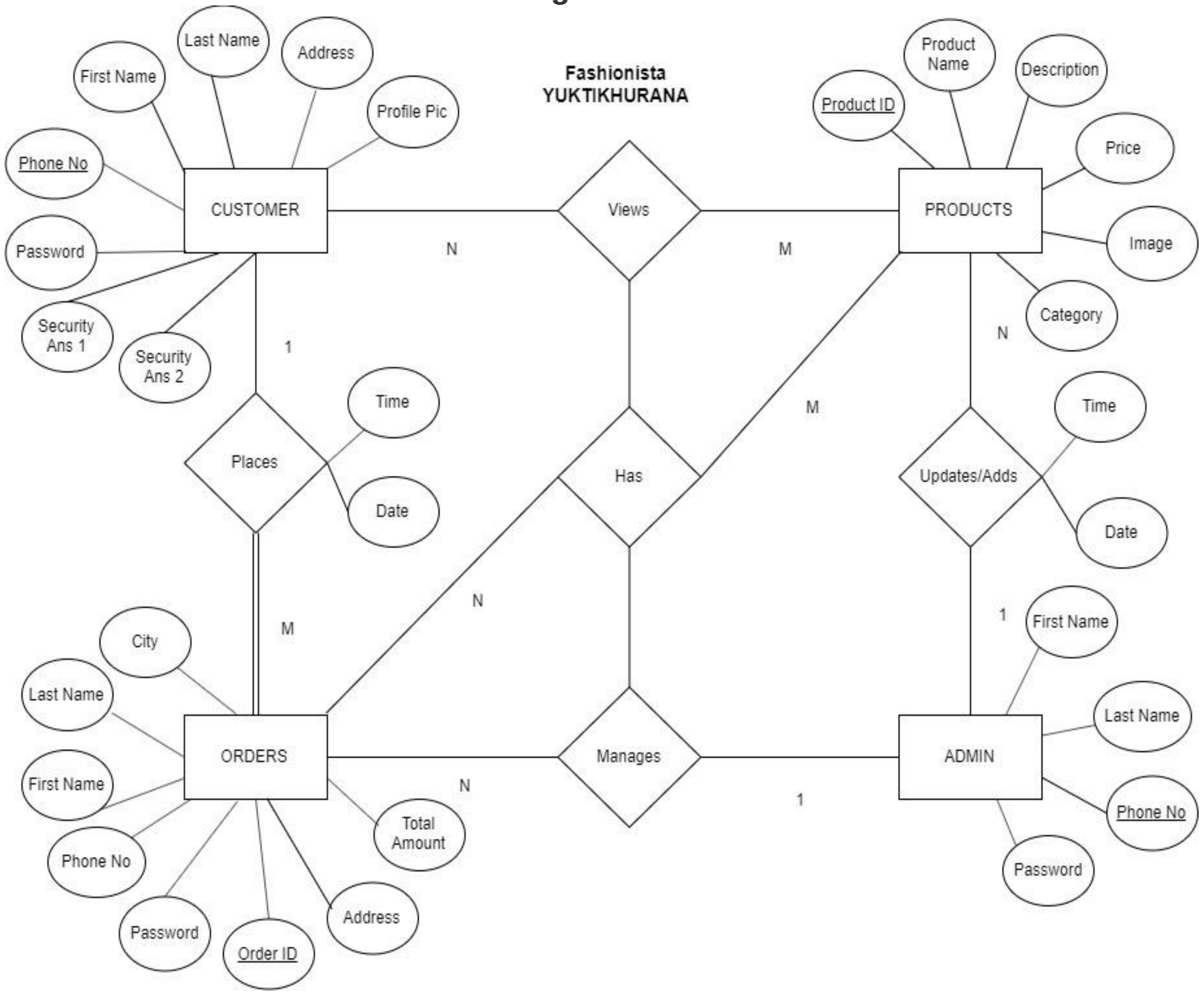
3.5.6 ER Diagram

The purpose of draw this diagram to show the relationship among the objects and personal attributes that belong to them.



Component	Symbol	Example
Entity	Entity	Student
Weak Entity	Weak Entity	Assignments
Attribute	Attribute	Roll_num
Relationship	Relationship	Saves in
Key Attribute	Attribute	Acct_num

ER Diagram for APP



3.5.6.1 ER Diagram to Relations

Admins

Admins (fname, lname, phone, password)

Customers

Users (fname, lname, phone, password, address, profilePhoto, securityAns1, securityAns2)

Products

Products (pid, date, time, description, image, category, price, pname)

Orders

Orders (totalAmount, fname, lname, phone, address, city, date, time, state)

3.5.6.1 Entities and their attributes

- **Customer**
  - First Name
  - Last Name
  - Phone No
  - Password
  - Address
  - Profile Pic
- **Products**
  - Product Id
  - Date
  - Time
  - Description
  - Image
  - Category
  - Price
  - Product Name
- **Admin**
  - First Name
  - Last Name
  - Phone
  - Password
- **Orders**
  - Total Amount
  - First Name
  - Last Name
  - Phone

- Address
- City
- Date
- Time
- State

SOFTWARE IMPLEMENTATION

Algorithms of different modules

4.0 Main Activity

Shows two buttons:

- Customer Login
- Admin Log

```
openCustomerLogin.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        openCustomerLoginForm();
    }
});
```

On clicking the customer login button, the user can view the customer login activity to open his/her account to browse fashionista’s products

4.1 Customer Login

1. This module gives the user an option to login to an existing account or create a new one.

```
private void CustomerLogin(){
      if userPhone and userPassword are correct
             send to HomeActivity
      else
             inform user “Invalid credentials”
}
```

2. If user clicks on the register option, then he/she is directed to a new activity page giving a form to fill in necessary details and sign up on the app as a member.

```
public void openSignupForm(){
    Intent intent = new Intent(this, SignupForm.class);
    startActivity(intent);
}
```

3. Otherwise the user can fill his account details: phone number and password to proceed to shopping.

```
private void AllowAccessToUserAccount (final String phone, final
String password)
```

The details entered by the user are verified by retrieving the information from the database named: “UsersInfo.Db” and using table: “Users”. These tables are made using the “SQLiteOpenHelper” in android.

4. The user has the option to go to forgot password window in case she forgets her password and intends to reset it using previously entered security keys.

5. There is a “Remember Me” button in the user-Interface that keeps the user logged in even after closing the app. The user can directly enter the app and start shopping the next time without having to enter his account credentials.

```
       private void AllowDirectAccess (final String phone, final
String password) {
       if “Remember me” option ticked {
                retain user state using Paper database and allow
                user direct access
       }
}
```

4.2 Signup Form

This module creates a user account after taking in user credentials. The functions of this module are:

```
CreateAccount(){

        If user entered valid Phone and Valid Password and Valid
        Name then{
              Enter user data in database tables and create an
              account.
              openHomePage()
        }else{
              Prompt user to enter valid credentials.
        }


}
```

```
openHomePage(){

      opens user home activity specific to each user

}
```

4.3 Home Activity

If the user is Customer

Shows a wide range of products to shop from with their prices and description and an image also. Clicking on the products lets the user see all the details of the product as well as select the products and add a specific quantity to user cart.

```
private DatabaseReference ProductsRef;
ProductsRef =
FirebaseDatabase.getInstance().getReference().child("Products");
Paper.init(this);
private RecyclerView recyclerView;
```

Navigator Pane Action in Home activity:

```
public boolean onCreateOptionsMenu(Menu menu)
```

- It displays user photo and name at the top using database information.

  ```
  DatabaseReference UsersRef =
  FirebaseDatabase.getInstance().getReference().child("Users").chil
  d(currentOnlineUser.getPhone());
  ```

- Settings: Provides user the option to edit his/her account details

  ```
  Intent intent = new Intent(HomeActivity.this,
  SearchProductsActivity.class);
  startActivity(intent);
  ```

- My Cart: button to open up the user shopping cart to view products added and proceed for placing order

  ```
  Intent intent = new Intent(HomeActivity.this,CartActivity.class);
  ```

- Search: lets the user search any product of choice using name or any keyword that describes it.

  ```
  Intent intent = new Intent(HomeActivity.this,
  SettingsActivity.class);
  ```

- Logout: button to logout from the application. Ends the application.

```
   Paper.book().destroy();
   Intent intent = new Intent(HomeActivity.this,
MainActivity.class);
```

If the user is Admin

Shows all the products with their images and details. Clicking on the products lets the admin to edit the product’s price, name, description or image according to the stocks in the inventory.

```
Intent intent = new Intent(HomeActivity.this,
AdminCategoryActivity.class);
```

Admin also has the search option for faster access of specific products to make management of products easy.

4.4 Settings Products Activity

This module gives user an option to update his profile details like Name, profile photo, address and set security answers for future use in resetting the password. It also fetches user information from the database and shows the user all the details previously entered by them.
On clicking the “Update” button,

```
private void updateUserInfo (){
       // to load previous saved info of user from database
       userInfoDisplay()
       get name
       get address
       get user profile photo
       if (fields are valid){
              update table “users” in database
              uploadImage()

       }else{
              Prompt user to re-enter the data
       }

}
private void uploadImage (){
       if (select image button is clicked) {
              open user Gallery after taking permission
              give crop option for selected image
              update image view with user selected image
              insert/update firebase storage linked with user phone
              if (update successful) {
                     prompt user about success status
              }

       }
}
private void userInfoDisplay(final EditText userPhoneEditText,final
EditText firstNameEditText,final EditText lastNameEditText, final
EditText addressEditText){
      Picasso.get().load(model.getImage()).into(holder.imageView);
      DatabaseReference UsersRef =
      FirebaseDatabase.getInstance().getReference().child("Users")
}
```

It also lets user update his/her profile information like:

- Name
- Profile photo
- Address
- Security Questions

```
private void userInfoSaved()
private void updateUserInfo()
private void uploadImage()
```

4.5 Cart Activity

This module allows the user to view all his/her products of choice with the toal amount of all the products.

```
final DatabaseReference cartListRef =
FirebaseDatabase.getInstance().getReference().child("Cart List");
```

It allows the user to edit the amount of each product that user wishes to buy. It gives user two options: Edit or remove from cart.

```
oneTypeProductTPrice = ((Integer.parseInt(model.getPrice()))) *
Integer.parseInt(model.getQuantity());
AlertDialog.Builder builder = new
AlertDialog.Builder(CartActivity.this);
builder.setTitle("Cart Options");
```

If the user has placed any order, he/she can view the order status also here

```
CheckOrderState(){
       If the order has been shipped{
              Inform the user, that order has been shipped and allow
              purchase of more products
       }else{
              Inform the user, that the order will be shipped as
              soon as the vendor approves it.


       }
}
```

4.6 Reset Password Activity

This module gives the user the option to reset the password in case he/she has forgotten it. For security reasons the user can only reset the password if he/she enters his/her account’s security questions correctly. The user is first prompted to enter his phone number linked to his/her “fashionista” account and then answer the following to questions stored in user database.

There are two security questions asked

1. When is your favourite person’s birthday?
2. What is your favourite TV show/ movie?

```
private void verifyUser(){
       get user security password1
       get user security password1
       if (security1 == database.value1 and security ==
       database.value2 ){
              get new password
              if (confirm){
                     private void changeUserPassword(final String
                     phone){
                            change users database info
                     }
              }
       }else{
              Reject user request to change password
              Return to main acitity
       }
}
```

4.7 Search Products Activity

This module gives user the option to search various products using name or any description keyword for faster and easy access amongst thousands of available products.

SearchProducts(){

```
DatabaseReference reference =
FirebaseDatabase.getInstance().getReference().child("Products");
Get name of product from user
//Load products from database with similar or same name/ description
FirebaseRecyclerOptions<Products> options =
        new FirebaseRecyclerOptions.Builder<Products>()
.setQuery(reference.orderByChild("keyword").startAt(SearchInput),
Products.class).build();
```

}

4.8 Product Details Activity

This module displays the product selected by the user to closely view it and can see its price, description, name and select required quantity which the user can then add to cart.

```
private void getProductDetails(final String productID){
       FirebaseDatabase ref = “Products.db”
       Get product price using ref
       Get product name
       Get product description
       Get quantity from user
       DisplayAllProductInfo()
       If (AddTocart button is clicked){
              addingToCartList()


       }

}
private void addingToCartList() {
       get current system time
       get current system date
       generate orderid equals date + time
       create a hash map
       store product price, name, description, quantitity,
       productid,image, date and time to hash map
       store hashmap in the database in firebase storage

}
```

4.9 Confirm Final Order Activity

This module is for placing final order of the products in the cart. It shows the user his/her order amount and prompts them to fill shipment details like name, phone number, home address/ delivery address and city name.

```
Check(){
       Check if the user has entered all the necessary details
              Return true
       Else
              Prompts the user to do so
}
private void ConfirmOrder(){

       if (Check() is true){

              get current system time
              get current system date
              get total amount
              get user details for shipment
              set order status to "not shipped yet"
              final DatabaseReference ordersRef =
              FirebaseDatabase.getInstance().getReference()
                      .child("Orders")
              .child(Prevalent.currentOnlineUser.getPhone())
              store the data in a hash map
              use the hash map to store order details in the
              firebase database
              // sends the order to Admin for approval
              updateAdminOrders();


       }
}
```

Admin side

4.10 Admin Login

This module opens up a special login window for administrators who have special rights to manage the “Fashionista” App’s backend. Admin must enter correct Phone number linked to the account and corresponding password to login into the system.

```
private void AdminLogin(){
       get admin phone
       get admin password
       fetch data from admin table in database
       if (phone and pass are correct){
           private void AllowAdminAccess(phone,password)
       }else{
              Reject admin
              send control back to main
       }
}
private void AllowAdminAccess(String phone,String password){
       open AdminHomeActivity()
}
```

4.11 Admin Category Activity

This module lets the admin add new products to the database according to inventory stock. It shows the admin, various categories in the form of icons like dress, winter wear, shoes, goggles, hats, pants, makeup, hair products, jumpsuits, accessories like jewellery, skirts, traditional wear and much more. On clicking these icons Admin is sent to Admin add product activity.

```
“Category”.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view){
        Intent intent = new Intent(AdminCategoryActivity.this,
AdminAddProductActivity.class);
        intent.putExtra("category", "Category");
        startActivity(intent);
    }
});
```

This module also gives the admin option to Manage Customer orders or edit existing orders in case of any changes in quantity, price or description according to the stock.It also has a “log out” button when Admin wants to leave.

4.12 Admin Add Product Activity

This module helps the admin to add products to the products database and makes these products available for the users to browse.

```
private void ValidateProductData(){
       // It validates whether the product info added by the admin
       Get product price
       Get product description
       Get product product name
       UploadProductImage()
       StoreProductInformation();

}
Private void UploadProductImage(){
       // opens gallery and lets the admin choose and crop product
       image
       OpenGallery()
}

private void StoreProductInformation(){
       get current system time
       get current system date
       generate productid equals date + time
       upload product image to firebase storage using image url
       SaveProductInfoToDatabase();

}
```

4.13 Admin Maintain Product Activity

This module lets admin manage products. The user is sent to Admin Products Activity where he/she can edit product details like name, price, description and product image.

```
private DatabaseReference productsRef
private void displaySpecificProductInfo(){
       fetch product info from the database
       Picasso.get().load(pImage).into(imageView);
       if (deleteProduct option is clicked){
              deleteThisProduct(){
       }
       If (updateProductInfo button is clicked){
              applyChanges()
       }
}

private void deleteThisProduct(){
       the admin can delete any product not available currently
       deletes corresponding data in the firebase database
}
private void applyChanges(){
       gets product name, price and description
       use product id to update data in the products database
       changes applied immediately

}
```

4.14 Admin New Orders Activity

This module lets admin view current orders placed by various customers and gives the admin option to approve those orders. The admin can approve order or may reject it according to some supplier issue or product unavailability. As soon as an order is approved or rejected by the admin, the customer cart is updated of the same and the “shipped status” is updated in database immediately.

```
private void RemoverOrder(String uID){

       remove order from order database
}
public static class AdminOrdersViewHolder {
       loads currently unshipped orders from the database
       display order details like products and total amount
       display Customer Name, order id, date and time of placement
       if (order is selected) {
              prompt the admin for approval of order
              if (yes) {
                     update orders database
                     order status is updated to shipped
                     customer can then make new orders

              } else {
                     Customer is informed of the unsuccessful order
              }
       }

}
```

4.15 Database Design

A database design is a collection of stored data organized in such a way that the data requirements are satisfied by the database. The general objective is to make information access easy, quick, inexpensive and flexible for the user. There are also some specific objectives like controlled redundancy from failure, privacy, security and performance. A collection of relative records makes up a table. To design and store data to the needed forms database tables are prepared.

4.15.1 Database Helper Customer

This module helps all the modules to access data stored in customer table of the database.

```
public static final String DATABASE_USERS = "UserInfo.db”
public static final String TABLE_USERS = "Users";
public void onCreate(SQLiteDatabase db)
public boolean insertDataUsers(String fname,String lname,String
phone,String password)
public boolean enterSecurityQues(String phone,String ques1,String
ques2)
```

```
public byte[] getImage(String phone)
public String getSecurity1(String phone)
public String getSecurity2(String phone)
public boolean changePassword
public String getUserFName(String phone)
public String getUserLName(String phone)
public String getUserAddress(String phone)
public Boolean UserPhoneExists(String phone)
public Boolean UserPasswordCorrect(String Phone,String Password)
public Cursor getAllDataUsers()
public boolean updateDataUsers(String oldPhone,String phone,String
fname,String lname,String address)
public Integer deleteDataUsers(String phone)
public void insertProfilePic(String phone,byte[] img)
```

4.15.2 Database Helper Admin

This module helps all the modules to access data stored in Admin table of the database.

```
public static final String DATABASE_ADMINS = "Owners.db"
public static final String TABLE_ADMINS = "Admins";
public void onCreate(SQLiteDatabase db)
public boolean insertDataAdmins(String fname,String lname,String
phone,String password)
public String getAdminPassord(String phone)
public String getAdminFName(String phone)
public Boolean AdminPhoneExists(String phone)
public Boolean AdminPasswordCorrect(String Phone,String Password)
public Cursor getAllDataAdmins()
public boolean updateDataAdmins(String phone,String fname,String
lname,String password)
public Integer deleteDataAdmins(String phone)
```

SOFTWARE TESTING

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics.

Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

5.1 Unit Testing

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit tests are typically written and run by software developers to ensure that code meets it design and behaves as intended. Its implementation can vary from being very manual (pencil and paper) to being formalized as part of build automation.

5.2 Integration Testing

Integration testing, also known as integration and testing (I&T), is a software development process which program unit are combined and tested as groups in multiple ways. Integration testing can expose problems with the interfaces among program components before trouble occurs in real- world program execution. There are two major ways of carrying out an integration test, called the bottom-up method and the top-down method. Bottom-up integration testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. In top-down integration testing, the highest-level modules are tested first and progressively lower- level modules are tested after that. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

5.3 Validation testing

At the validation level, testing focuses on user visible actions and user recognizable output from the system. Validations testing is said to be successful when software functions in a manner that can be reasonably expected by the customer.

Two types of validation testing are:

- **Alpha testing** is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.
- **Beta testing** comes after alpha testing. Versions of the software, known as beta version, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users

5.4 Test cases

This software explains Test cases for Flipkart Online Shopping Web Application testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing presents an interesting of a system using various test data. Preparation of the test data plays a vital role in the system testing. After preparation, the test data, the system under study is tested those test data. Errors were found and corrected by using the following testing steps and corrections are recorded for future references. Thus, series of testing is performed on the system before it is already for implementation. The development of software systems involves a series of production activities where opportunities for injection of human errors are enormous. Errors may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified as well as in later design and development stages. Because of human in ability to perform and communicate with perfection, software development is followed by assurance activities ER Diagram for Customer.

5.5 Objectives

The test plan for the system should support following objectives:

- Identify which features of the system will be tested.
- Define the pass/fail criteria for each feature to be tested.
- Specify the testing approaches that will be used during testing.
- Identify the deliverables of the testing process.

5.6 Description

- Test Steps - List all the test execution steps in detail. Write test steps in the order in which they should be executed. Make sure to provide as many details as you can.
- Test Data - Use of test data as an input for this test case. You can provide different data sets with exact values to be used as an input.
- Expected Result - What should be the system output after test execution? Describe the expected result in detail including message/error that should be displayed on the screen.
- Actual result - Actual test result should be filled after test execution. Describe system behaviour after test execution.
- Status (Pass/Fail) - If actual result is not as per the expected result mark this test as failed. Otherwise, update it as passed.
- Notes/Comments/Questions - If there are some special conditions to support the above fields, which can't be described above or if there are any questions related to expected or actual results then mention them here.

Features to be tested

This section outlines all the features that will be tested: Type of User Feature Identifier Description User Case-1 System Register Case-2 Edit Shopping Cart Case-3 Add to Cart Case-4 Place Order Seller Case-5 Create and Delete product from Category Case-6 Manage Orders

CASE 1: System Register Purpose: Test that users can register with the proper username and password 1. Visit Customer's Login web page 2. Enter First name, Last name, username 3. Enter password, confirm password, e-mail, address 4. Security question and Security answer 5. Click Signup button S/N Input data Expected Results Actual Results Pass Fail Remarks 1 Enter empty value for First Name Display error message to enter some valid text. Error: "Name is required, Please Enter Correct details". ✓ 2 Enter empty value for Last Name No error No error ✓ 3 Enter empty value for User ID Display error message to enter some valid text. Error: "This Field is required". ✓ 4. 6. Enter a username already in use with an existing user Display error message to the user and he should not be allowed to register for an account with that username. Error Message: "The input username is already taken please enter some other username". ✓

40. 5. Enter a Username not in use with other existing users. User should be able to register with the website and directed to the secure Web page requested. No Error ✓ 6. Enter empty value for Contact Number Display error message "Enter valid contact number" Error message: "Enter valid contact number" ✓ 7. Enter non numeric value in Contact Number field like abcde Display error message to enter a valid contact number" Error message: "Invalid contact number, Please enter

correct Phone no.” ✓ 8. Enter empty value for Email Address Display error message “Enter a valid address” Error message: “Enter a valid address” ✓ 9. Enter empty value for Confirm Email Address Display error message “Enter your email address” Error message: “Enter your email address” ✓ 10 Enter different email format likeYahoo.com.sg Display error message “Enter a valid email Address” Error message “Enter a valid email Address” ✓ 11. Enter empty value for either Password or Confirm Password Display error message “Password does not match.” Error message: “Password does not match.” ✓ 12. Enter password less than 8 characters long 123456 Error message “Password length must be at least 8 characters” ✓ 13. Enter empty value for Address No error No error ✓

41. CASE 2: Edit Shopping Cart Purpose: Test that clicking Update Quantities will update the cart summary accordingly. 1. After selecting a product, go to Shopping cart Web page. 2. Check after entering incorrect input, an appropriate message should be displayed. 3. If entered a valid number, check if the total quantity and relative price is updated after clicking update or delete button. S/N Input data Expected output Actual Result Pass Fail Remarks 1. Negative input number or input other than integer number in “Quantity” field Display error message to notify the given input is invalid Error message: “Please input valid no. of items” ✓ 2. Enter a Positive integer number in the “Quantity” field The product quantity should be updated or deleted according to the specified input number No Error. “The item is updated successfully” ✓ CASE 3: Add to Cart Purpose: Test that clicking add to cart button, product is getting added in the cart 1. Click Add to Cart button 2. Check whether the cart shows the product 3. Check if the quantity count is one, if product is not already in the cart. 4. Check if the quantity count is increased by one, if product is already in the cart S/N Input data Expected Results Actual Results Pass Fail Remarks 1. Click “Add to Cart” Button for product not already in the cart. The product should be moved to wish list. No Error. “The product is successfully added to Cart ” ✓ 2. Click “Add to Cart” Button for product already in the cart. If product already exists then its quantity is updated by 1. No Error. “The product quantity is successfully increased by one in the Cart ” ✓

42. CASE 4: Place Order Purpose: Test that user can place an order with valid profile information 1. Select a product to order. 2. Click Place Order button with one of the fields empty. 3. Check if the message is to enter all required fields. S/N Input data Expected Results Actual Results Pass Fail Remarks 1. An empty required field. (First name and last name, Street address, city, state, zip code, email, credit card information, shipping address) An appropriate message should be displayed and the user should not be allowed to place the order. Error message: “Please enter all the required details to place the order” ✓ 2. The user should be able to place an order and redirected to the order confirmation page. The product quantity should be updated or deleted according to the specified input number. No Error. The product is ready to be shipped within the week. ✓ CASE 5: Edit a Product from Category for Seller Test that after clicking Delete Product, Update

Product, selected product is being removed from the catalog. The carts before and up to that date are removed 1. Click Delete Product button 2. Check for the message- product has been removed from the category. 3. Check that after clicking Show products in category, the deleted product is not shown. S/N Input data Expected Results Actual Results Pass Fail Remarks 1. The seller selects a product to be deleted, the category from which it is to be deleted and clicks delete button The product should be deleted from the category only for that category. Error message: “Product is successfully removed from that category but not from others, if it belongs to more than one category” ✓ 2. Negative input number or input other than integer Display error message to Error message: “Please input ✓

43. number in “Quantity” field notify the given input is invalid valid no. of items” 3. Enter a Positive integer number in the “Quantity” field The product quantity should be updated or deleted according to the specified input number No Error. “The item is updated successfully” ✓ 4. Enter a Positive integer number in the “Price” field The product price should be updated according to the specified input number No Error. “The item is updated successfully” ✓ CASE 6: Manage Orders Purpose: Test that order status is correctly updated 1. Select the order from order list 2. Click Edit button 3. Check for the message- Order status successfully updated 4. Check that status of other orders in database is unchanged S/N Input data Expected Results Actual Results Pass Fail Remarks 1. Enter a Positive integer number in the “Price” field The product price should be updated according to the specified input number No Error. “The item is updated successfully” ✓ 2. Seller will enter the date range, number of orders and type of order, to view the existing orders in database. He/she will then select the order for which the status is to be updated where its details could be edited. The status of that order is updated, but the status of all other orders remains same No Error. “The Details of the Product is successfully Updated” ✓

44. 6.4.4 Approach: Only functional black box testing will be performed to test the functionality of the system. The features mentioned above describe how the user will interact with the system, so the testing will require the tester to interact with the system in the same way a typical user would. The user actions will be simulated through a set of test scenarios. Each scenario will trace back to a requirement listed in the Vision Document. 6.5 Performance Testing: This will be performed to test the entire system to see whether all driving requirements are satisfied. Allowing multiple users log into the system and perform the operations at the same time using the JMeter testing tool will do this. This test verifies that the components of the systems meet the stated requirements for speed. The following components of the system would be analyzed for performance: • Buying: Includes browsing the catalog, selecting a product, add to shopping cart, checkout, enter personal details and place order. • Searching for a Product. • Getting Product recommendation Using JMETER tool, approximately 100 concurrent virtual users with a minimum of 50 requests per user will be inputted to calculate the response times for each of the above components. The above tests would be done in different environments like: • Local connection of 54Mbps • LAN connection of 100Mbps • Wired

connection 6.6 Suspension Criteria and Resumption Requirements: 6.6.1. Suspension Criteria: If a test case fails, testing will be suspended for all dependent features. The failed test case will be logged into a test log along with a description of the failure. 6.6.2. Resumption Requirement: Test cases, not dependent on the case in which a bug is reported, will continue to be executed in parallel to bug fixing. Testing for the failed test case will resume after the bug has been identified and resolved.

CONCLUSION

After the completion of the project, one can clearly say that our developed application will help shopkeepers in many ways to record the details. This is the easy way to record the details manage them. It can use by any persons who have a shop and who does not want to record the details of the shop not on paper to reduce his effort and time-consuming.