COMPUTER SCIENCE PROGRAM

MOBILE APPLICATION PLATFORM FOR ONLINE PHARMACY CALLED PHARMAC

Capstone Project Report

Prepared By:

Kapil Reddy Kuchekulla

Master of Science in Computer Science

Faculty Advisor:

Visvasuresh Govindaswamy

Associate Professor of Computer Science

May 7, 2021

MOBILE APPLICATION PLATFORM FOR ONLINE PHARMACY CALLED PHARMAC

Capstone Project Report

Approved by the Supervisor

Signature:

Visvasuresh Govindaswamy,
Associate Professor of Computer Science

TABLE OF CONTENTS

INTRODUCTION 6

PharmaC Abstract 6
Market Need 6
Medicines E-Commerce app 6
Easy interface 7
Good Discounts 8
Competitors in the market 9
Future Work Plans 10

USER MANUAL 11

Installation guide 11
Using the application 11
Creating an Account 11
Forgot Password 12
Creating Order 12
Tracking an Order 13
Creating shop 13
Adding order to shop 13
Managing customer Order 14
Adding promotion codes 15

APPLICATION DESIGN 17

Purpose of Design Models 17
Design Diagrams 18
Graphical User Interface 19
Use Case Designs 23
Login System Use Case 23
Place Order Use Case 24
Hierarchical Task Analysis 25
Finite State Diagrams 26
Complete Finite State Diagram 26
Forgot Password Level 2 Diagram 27
Data Flow Diagram 28
Sequence Diagrams 29
Sign Up Sequence Diagram 29
Login Sequence Diagram 30

CLASSES 31

class explanations 31
LoginActivity.java 31
ShopDetailsActivity.java 31
AddProductActivity.java 31
EditProductActivity.java 32
PromotionCodesActivity.java 32
AdapterCartItem.java 32

TESTING 34

Test Used 34

Black-Box *35*
White-Box *36*

CONCLUSION 40

conclusion 41
personal mentions 41

CODE 43

Java Code 43
references 47

ABOUT ME 48

Biography 48
MY Goals 48
My CS Experience 48

END PAGE 53

INTRODUCTION :

PharmaC Abstract:

- PharmaC App is designed and implemented to allow users to purchase medicines online. This will enable users to buy medicines directly from the app where they place their order, pay for it right in the app and get them delivered excluding the need for going physically to the pharmacy.

- The purpose of the app is to provide reliable and easy access to medicines for the medical community including physicians and a normal buyer with doorstep delivery. Many customers consider this to be more convenient than traveling to medical shops on a day-to-day basis. Prescription medications are very expensive and customers may turn to online pharmacies to save money as we offer promotion/coupon codes as well as help the seller to thrive their business. Many pharmacy owners and managers, as well as entrepreneurs will benefit from this. In comparison to other apps, PharmaC app gives the user the ability to compare and select a product for the best lowest price across multiple stores and suggest over the counter medicines to users based on their selection of the illness.

- The app is developed using the Android Studio Integrated Development Environment (IDE), which contains Java and Extensible Markup (XML) programming languages. It is designed using software design tools such as Adobe XD and *diagrams.net* (formerly *draw.io*). To keep the data securely stored and easily accessible, Google's Firebase, a Backend-as-a-Service (BaaS) platform, is used. Testing, consisting of Black-box, White-box, Grey-Box, Unit Testing techniques, are used. Future work includes extending the app to suggest over the counter medicines to users based on their selection of the illness and scale it.

Market Need :

- *Medicines E-Commerce app*

  This medicines E-commerce app is important because there is little information about pharmacies and EC in drugstore This application provides a way to create an E-commerce website on Medical and Pharmacy utilities.

  By the help of this application customer will be able to search and buy the medicine from anywhere and as per his requirements

- *Easy interface*

This project includes to give the list of the medicines to the customer/patient based on the category he selects

Application will provide information about different offers on buying of product/medicines.

It would also provide methods to change the quantity of products purchased and edit the cart.

This application is evolution in the medical Drug shopping, it minimize the effort and save the time to buy the medicine

- Good Discounts

This has been equipped with the coupon mechanism where the buyer can add promotion codes based on several factors and multiple coupon methods for multiple discounting procedures can be applied.

User can use these coupon codes while opting to place an order and will be getting an instant discounting on the products/items in the cart of the buyer.

Sprint Schedule

- **Sprint 1 (Jan 25 - Feb 8)**
- Market analysis for competition
- Developing E-commerce app objective
- Presentation of project objective

- **Sprint 2 (Feb 8 – Feb 22)**
- Updated project idea
- Market analysis for existing competition
- Presentation of New project objective

- **Sprint 3 (Feb 22 – March 8)**
- Create Design prototype in Adobe XD
- Design diagram and models for the Login and Sign Up
- Login and Sign Up pages

- **Sprint 4 (March 8 – March 22)**
- Create Homepage Layout
- Set up API integration with firebase

- **Sprint 5 (Mar 22 – April 5)**
- Add customizations like category, filter and search functionality
- Create pages for the shops and medicines

- **Sprint 6 (April 5 – April 19)**
- Create Profile section and edit User details
- Testing the functionalities and events for any errors and updating

- **Sprint 7 (April19 – May 3)**
- Prepare for Investor Pitch and demo
- Update the diagrams
- Include updates in Documentation prep

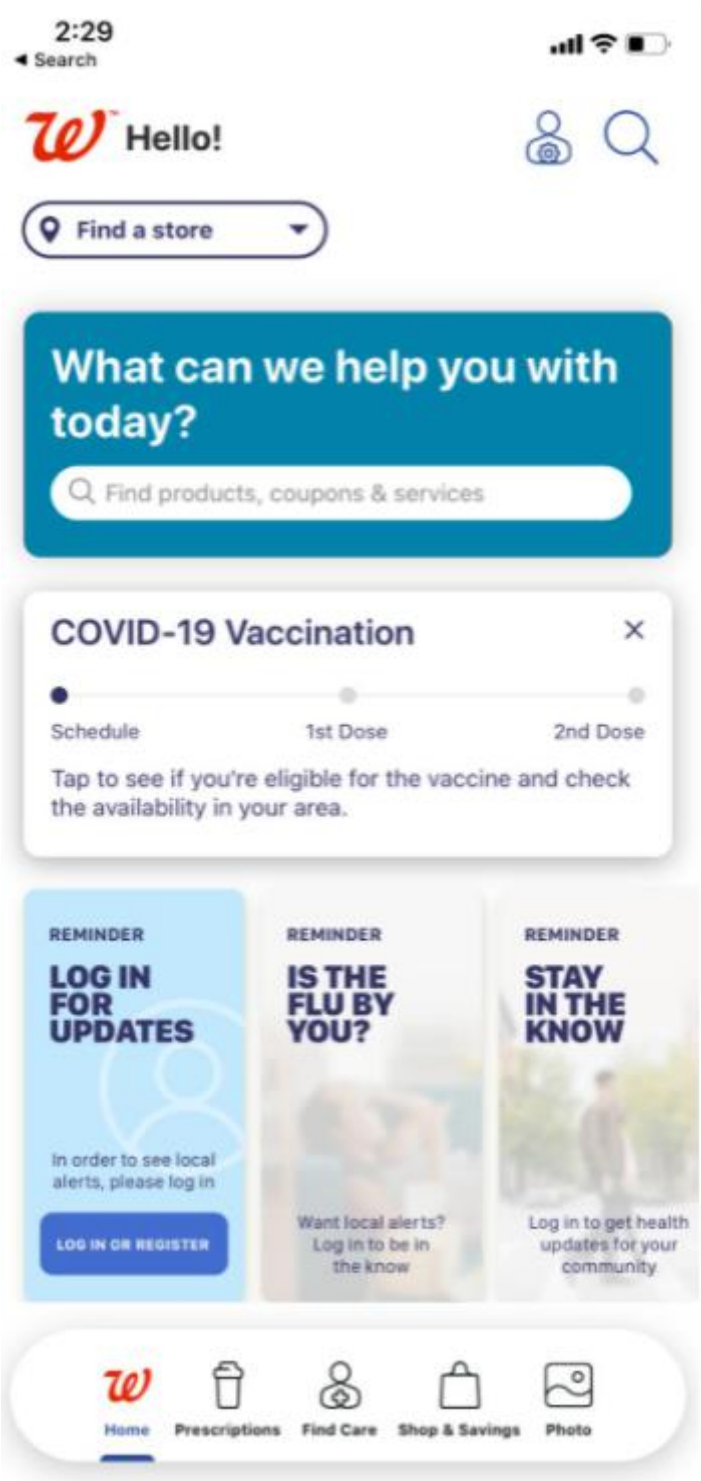
Competitors in the market :

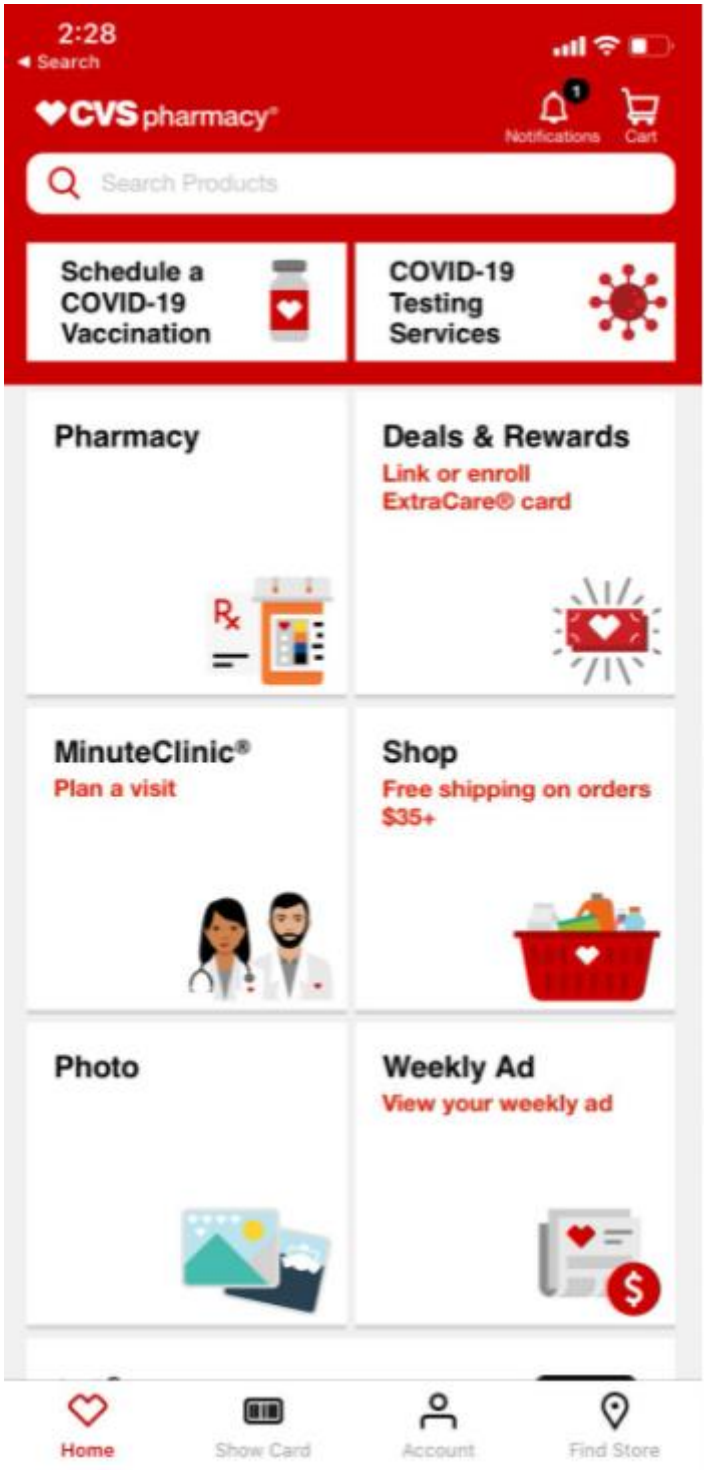
Lack of potential competition is a real plus for us to excel in this path of the business. Many companies may provide you with the capabilities to buy or sell your products or others online buy are they safe for you, Do they fit all your needs is what you need to decide.

- Are too strong or too weak.
- Have dangerous ingredients.
- Have expired (are out-of-date).
- Aren't FDA-approved (haven't been checked for safety and effectiveness).
- Aren't made using safe standards. Aren't safe to use with other medicine or products you use.

We tackle this by having a transparent system between the user and the buyer and by giving additional benefits of door step delivery easy tracking and flexible discounting.


2:29
Search
PharmEasy
Deliver to Hyderabad 500034
Search medicines/healthcare products
PharmEasy
Go Corona Go Initiative
SUPERHEROES WANTED!
Get your COVID-19 vaccine shot and get rewarded!
FLAT ₹150 COUPON
for next medicine order
Order Medicine
FLAT 15% OFF
Healthcare Products
UPTO 60% OFF
Lab Tests
UPTO 70% OFF
Plus
Save flat 5% extra on medicines & enjoy FREE delivery with PLUS membership
Know more >
5 Step safety Assured
Temperature Check
Secure Packaging
Over 30,00,000 orders safely delivered
Home
Healthcare
Diagnostics
Notifications
Account


2:29
Search
Hello!
Find a store
What can we help you with today?
Find products, coupons & services
COVID-19 Vaccination
Schedule
1st Dose
2nd Dose
Tap to see if you're eligible for the vaccine and check the availability in your area.
REMINDER
LOG IN FOR UPDATES
In order to see local alerts, please log in
LOG IN OR REGISTER
REMINDER
IS THE FLU BY YOU?
Want local alerts? Log in to be in the know
REMINDER
STAY IN THE KNOW
Log in to get health updates for your community
Home
Prescriptions
Find Care
Shop & Savings
Photo


2:28
Search
CVS pharmacy
Notifications
Cart
Search Products
Schedule a COVID-19 Vaccination
COVID-19 Testing Services
Pharmacy
Deals & Rewards
Link or enroll ExtraCare® card
MinuteClinic®
Plan a visit
Shop
Free shipping on orders $35+
Photo
Weekly Ad
View your weekly ad
Home
Show Card
Account
Find Store

Future Work Plans :

- To make this application platform independent and provide a system compatible with any device.
- Adding support for the doctors in the system to aid the patients online so that they can suggest medicines as well as stores and medicine links directly to the user.
- Making online prescriptions a trustable thing to every individual.
- Scaling the app.
- To suggest over the counter medicines to user based on their selection of the illness.
- Future work includes extending the app to suggest over the counter medicines to users based on their selection of the illness and scale it.

USER MANUAL

Installation guide :

You can download the application from the play store and install on your system. For you to find the application you need to type in "PharmaC Seller" for downloading and installing the seller application and "PharmaC Buyer" for downloading and installing the buyer application.

- Software requirements

Zip file of the project with all java files and libraries

- Hardware requirements

At least 25 MB Hard Disk space, RAM that satisfies Android studio 4.1.2

Step by Step guide:

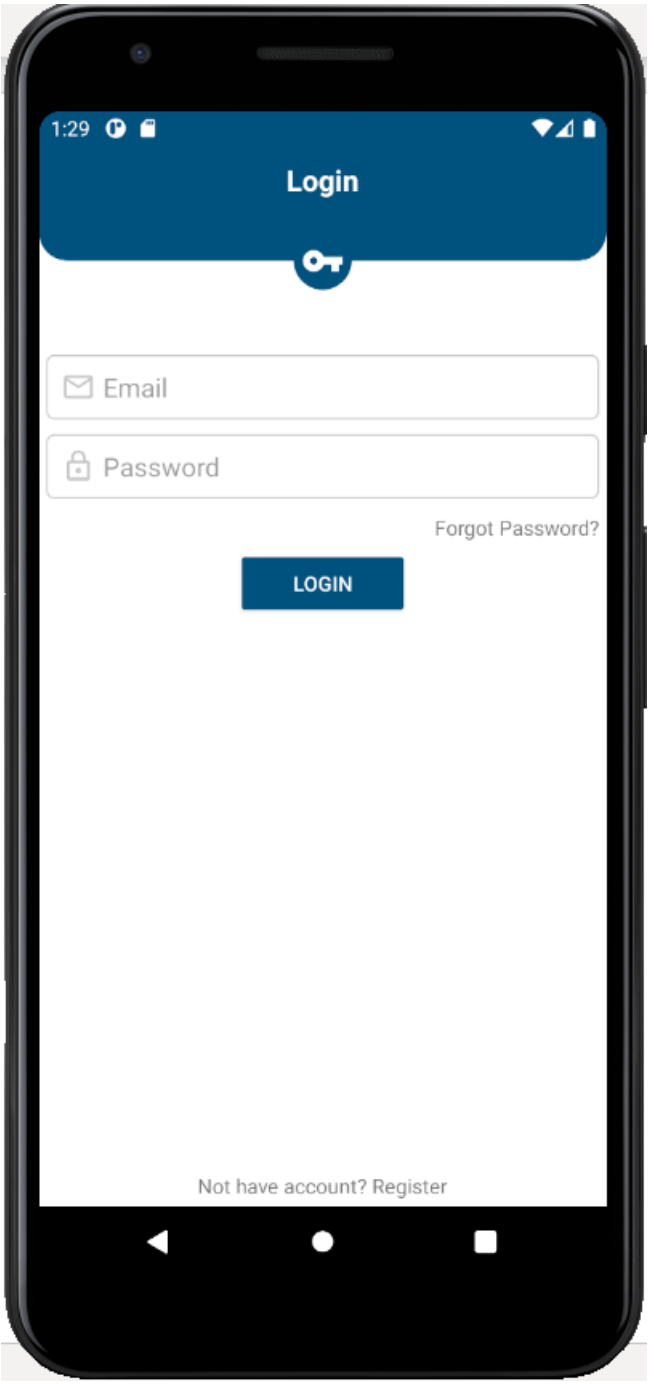
- Download the .zip file and unzip both the application of it to any desired location
- Copy the unzipped file path
- Open Android Studio
- From the left corner menu choose >>file>>open for both the apps
- Paste the address of the unzipped file and select the project
- Click select/open
- All necessary libraries have to be up to date
- On "Gradle.Module" file on the left open all the libraries and dependencies
- Green color indicated libraries have to be updated by right clicking on it

Using the application :

- *Creating an Account* :

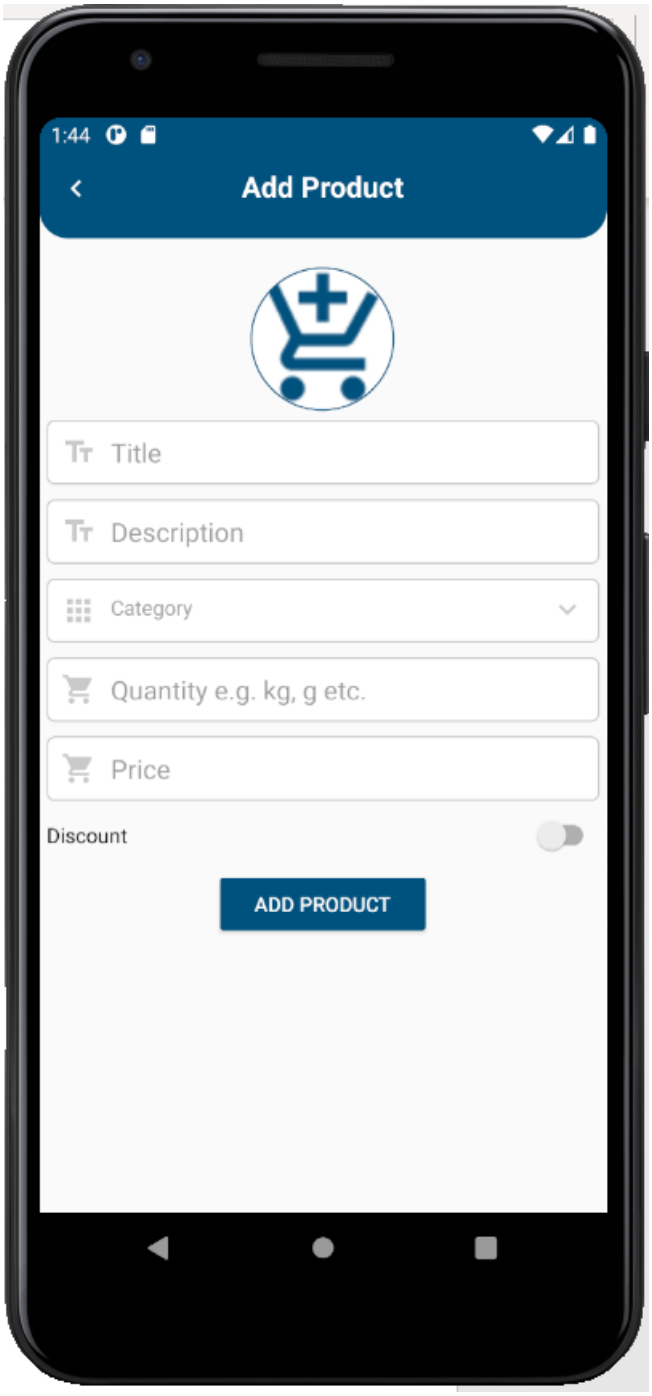
    - *After you install any one of the seller or buyer applications you will be prompted to LOGIN.*
    - *You can enter your credentials here to login into the system.*
    - *If you are a new user you can click on the REGISTER.*
    - *Enter mandatory details and register.*
- *Forgot Password:*
    - When you are in the login page and want to change your credentials or forgot your password
    - *You can choose the forget password option to get your password changed.*
    - You will be receiving the link to reset the password to your registered email.
- Creating Order:
    - Login into the PharmaC Buyer application
    - Select one shop from the list of shops provided in the home page of the application.
    - Now you can be able to see the list of medicines.
    - Click on the ADD TO CART option in the card of the medicine.
    - You will be able to see a pop-up dialog.
    - Add the required number in the quantity by clicking '+' or '-'.
    - Now click on the ADD TO CART in the pop-up dialog.
    - Go to the cart by clicking on the cart icon at the top of the page.
    - Now check your order details and add promotions code if any.
    - Then click on the CONFIRM ORDER button to place the order.
- Tracking an order:
    - In the ORDER DETAILS page you can check the 'order status' to track the order.
    - Order status will be in 'In progress' by default when you place the order.
    - Order status will be in 'Completed' state when the order gets delivered.
    - Order status will be in 'Cancelled' state when the seller cancels the order.
- *Creating shop:*
    - *In the 'PharmaC Seller' application when you REGISTER into the system you will be prompted to add a shop to the app to continue.*
    - *You can enter your shop details and create your shop.*
- *Logging in into the application:*
    - *Open the application.*

- *After you see the splash screen you will be able to see the login page where you can enter your credentials.*



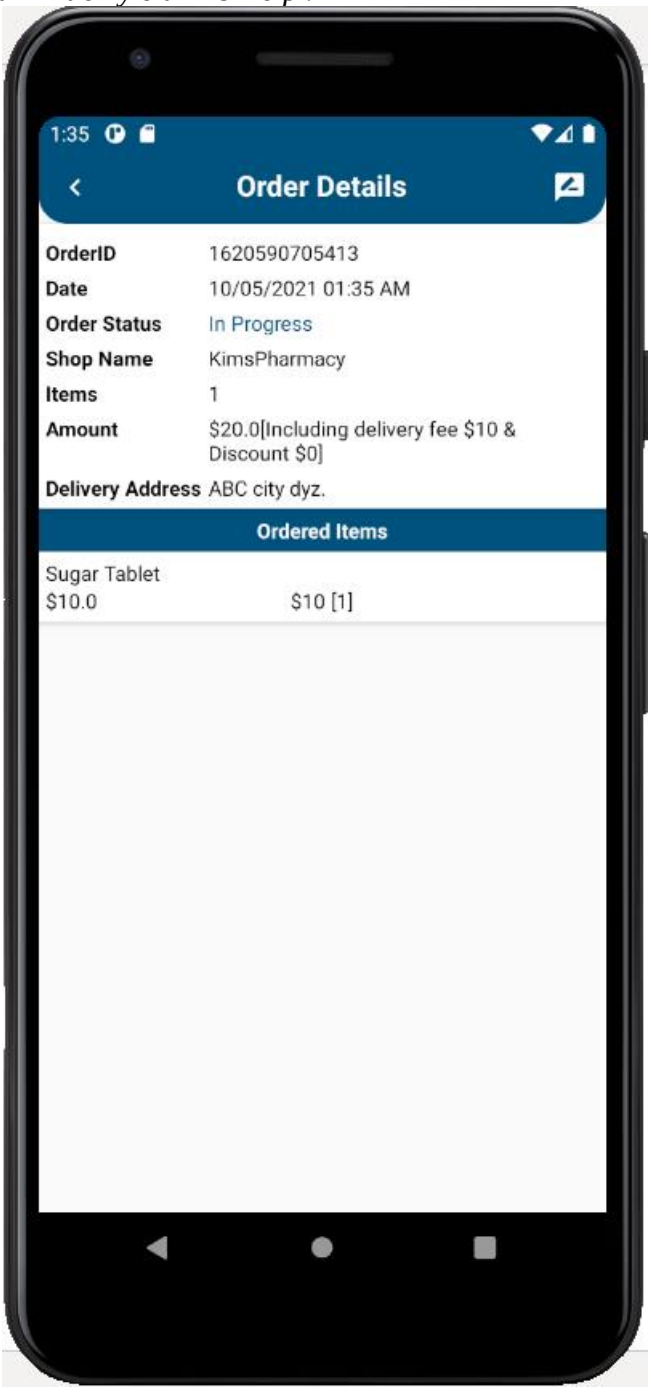
- *Adding order to shop:*
  - *In the HOME page of the 'PharmaC Seller' application click on the 'cart with plus' icon.*
  - *You will be redirected to ADD PRODUCT page.*
  - *Enter the Title that is displayed to the user.*

- *Enter the Description of the product include all the product related details into this section of the page.*
- *Select a desired category from the page that suits the product.*
- *Add the quantity of the product in the quantity section.*
- *Add the price of the product in the price section.*
- *You can toggle the discount section to add an extra discount to the product you are adding into the shop.*
- *Discount price defines the amount that is reduced from the actual price of the product.*
- *You can add a note to attract more users into Discount Note.*
- *This is customised message so that you can be more flexible with your discounts.*
- *Click on ADD PRODUCT button to add this product into your shop.*

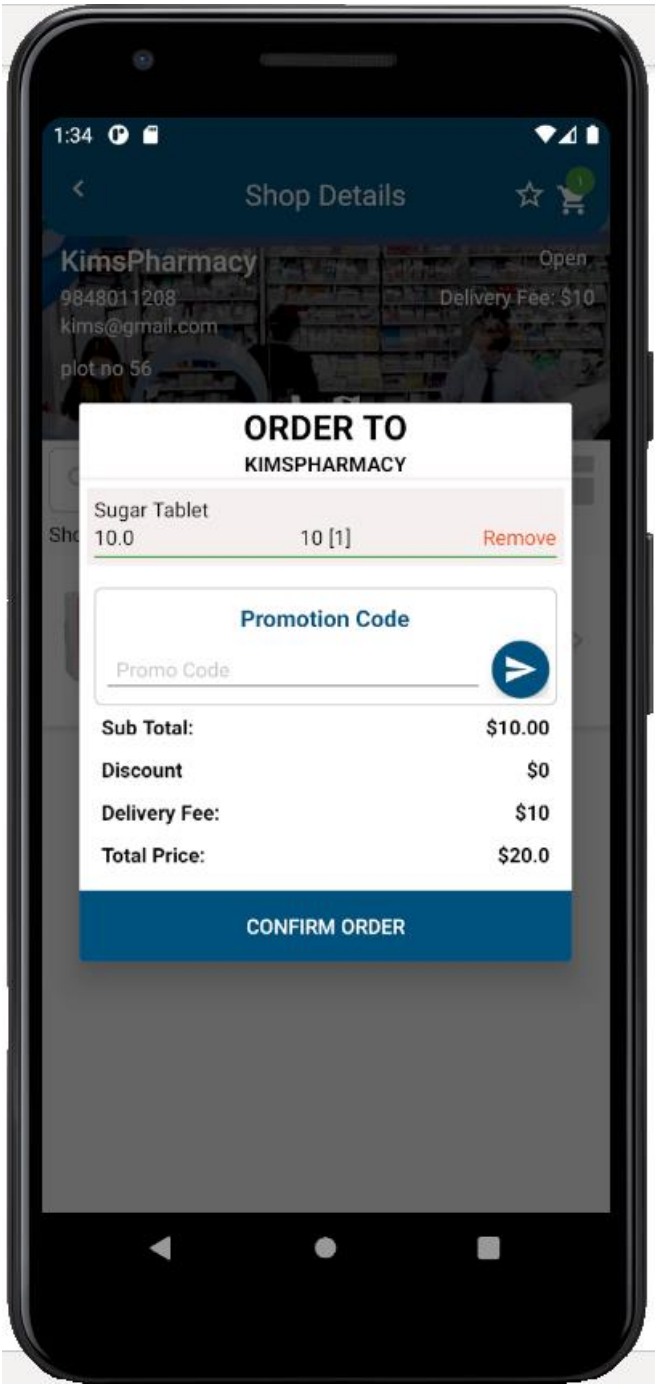


- *Managing Customer Order:*
  - *Click on any product, then you will be shown an edit option.*
  - *Click on the pen icon to edit the product.*
  - *You will be redirected to the EDIT PRODUCT page where you can change the order details.*
  - *Enter the Title that is displayed to the user.*
  - *Enter the Description of the product include all the product related details into this section of the page.*
  - *Select a desired category from the page that suits the product.*

- *Add the quantity of the product in the quantity section.*
- *Add the price of the product in the price section.*
- *You can toggle the discount section to add an extra discount to the product you are adding into the shop.*
- *Discount price defines the amount that is reduced from the actual price of the product*
- *You can add a note to attract more users into Discount Note.*
- *Click on UPDATE PRODUCT button to add this product into your shop.*



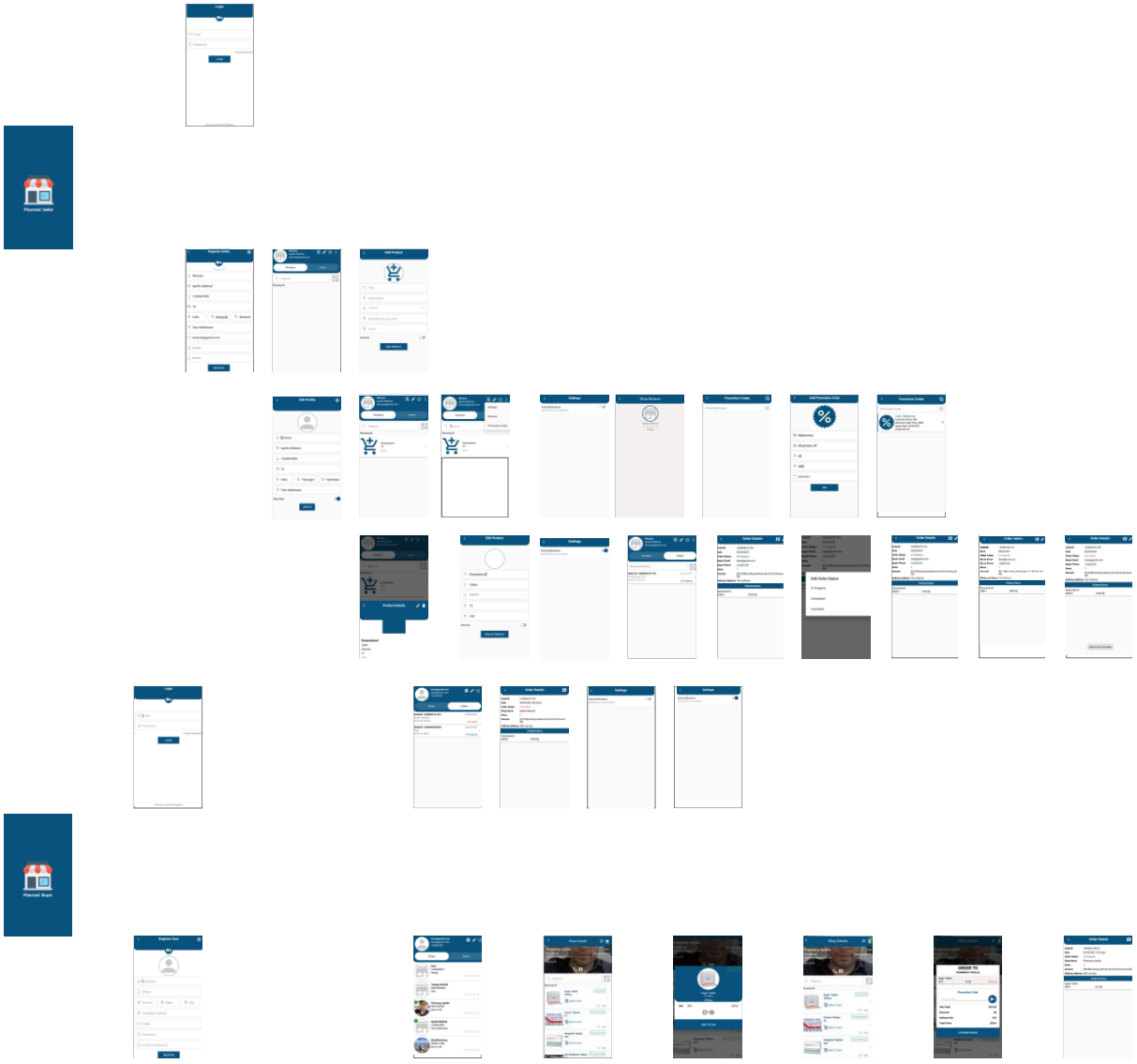
- *Adding Promotion Codes:*
  - *Click on the three dots icon at the top right corner of the screen.*
  - *Now select the promotion codes from the dialog.*
  - *You will be redirected to the PROMOTION CODES page.*
  - *Now click on the plus icon on the top right corner of the page.*
  - *Add the title of the promotion code in the Promotion Code section.*
  - *Add the promotion code description in the description section.*
  - *Add the price that you wanted to reduce on the actual price.*
  - *Add minimum order price in the minimum order price section.*
  - *Enter expiry date of the promotion code.*
  - *Now click on ADD button to add the promotion code.*



APPLICATION DESIGN

Purpose of Design Models:

Design Models are representations that can aid in defining, analysing, and communicating a set of concepts.
Design **models** are specifically developed to support analysis, specification, **design**, verification, and validation of a system, as well as to communicate certain information.



Design Diagrams :

- ***Graphical User Interface***
  - *This is a glace of how the user interface of the application looks*

    like.

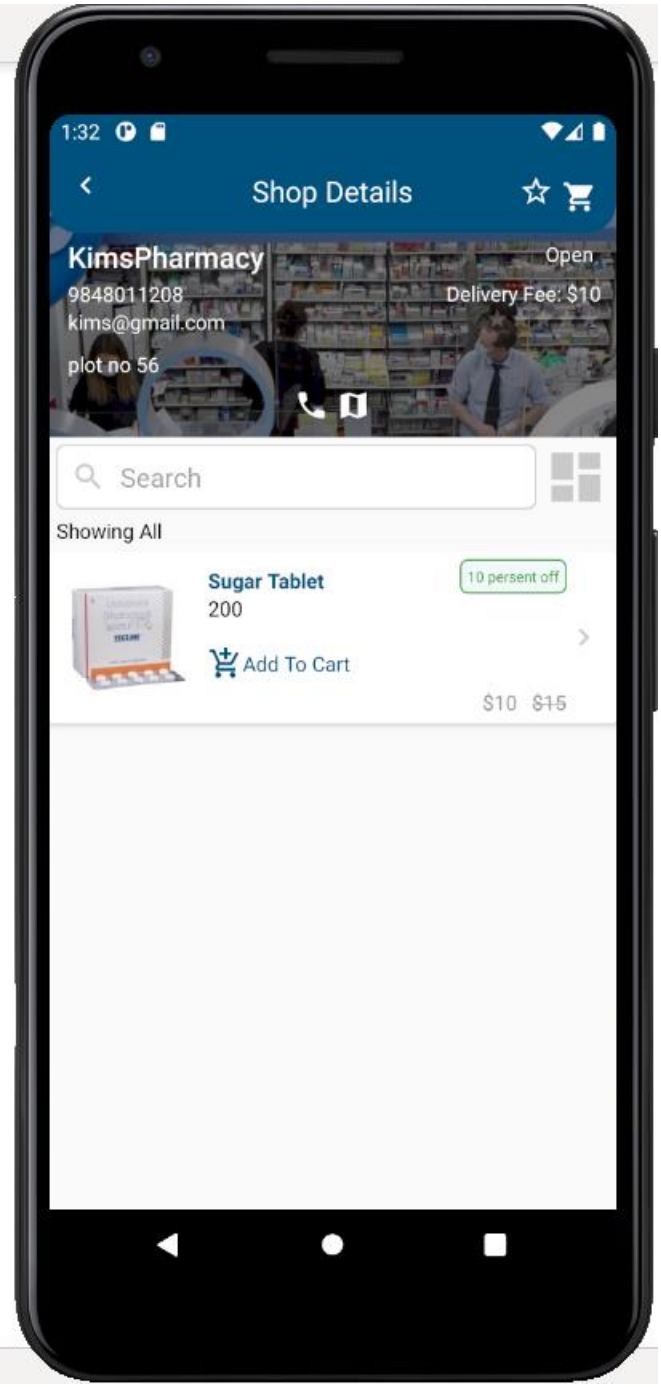
  - The following are the diagrams related to the LOGIN, REGISTER and the HOME page of the application.
  - A good user interface is the heart of the application.
  - We have simplified the user interface as much as we can to improve the understandability, reliability and to save some time of the user in their shopping experience.

- [illegible]

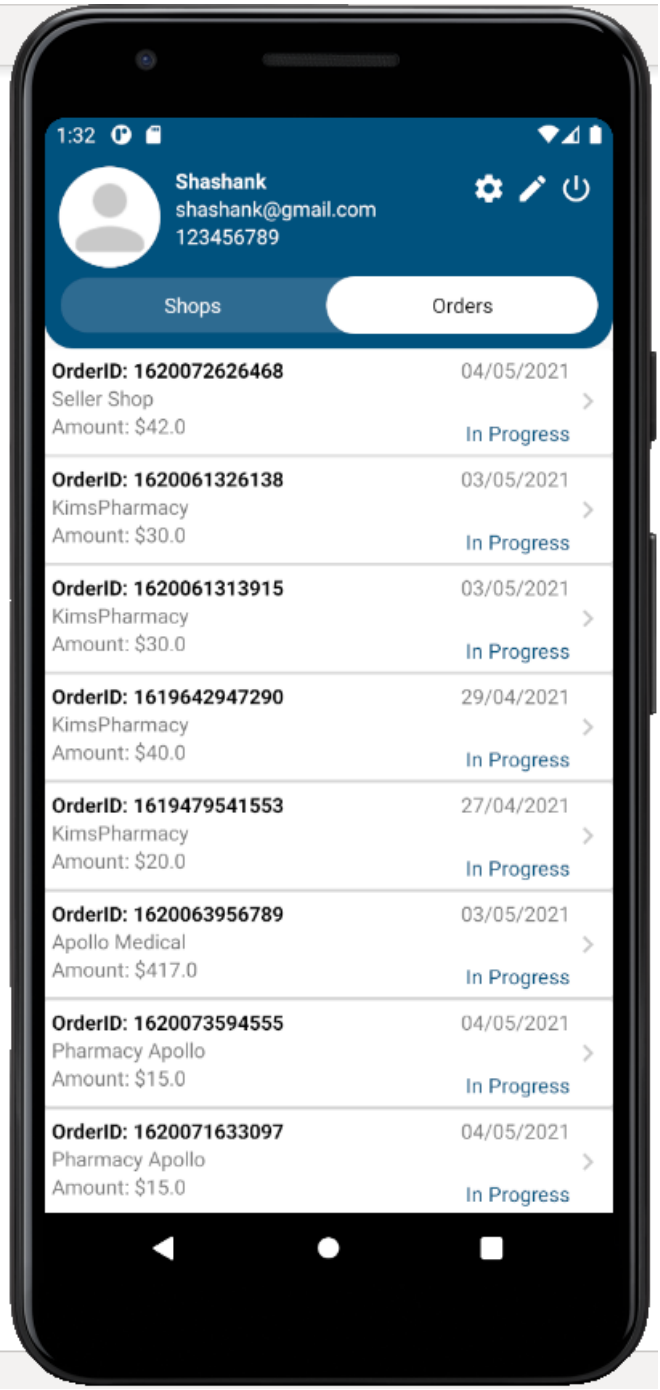
a)Login page

b)Register page of seller

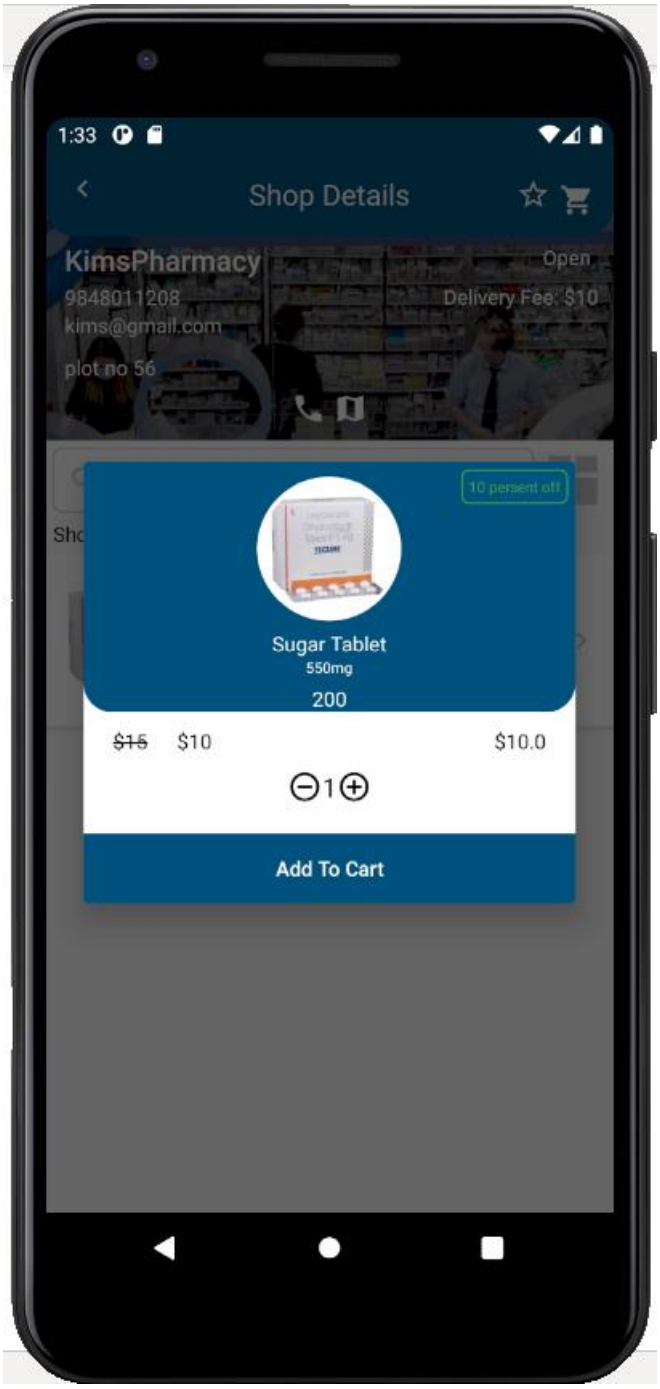
c)Products page of user



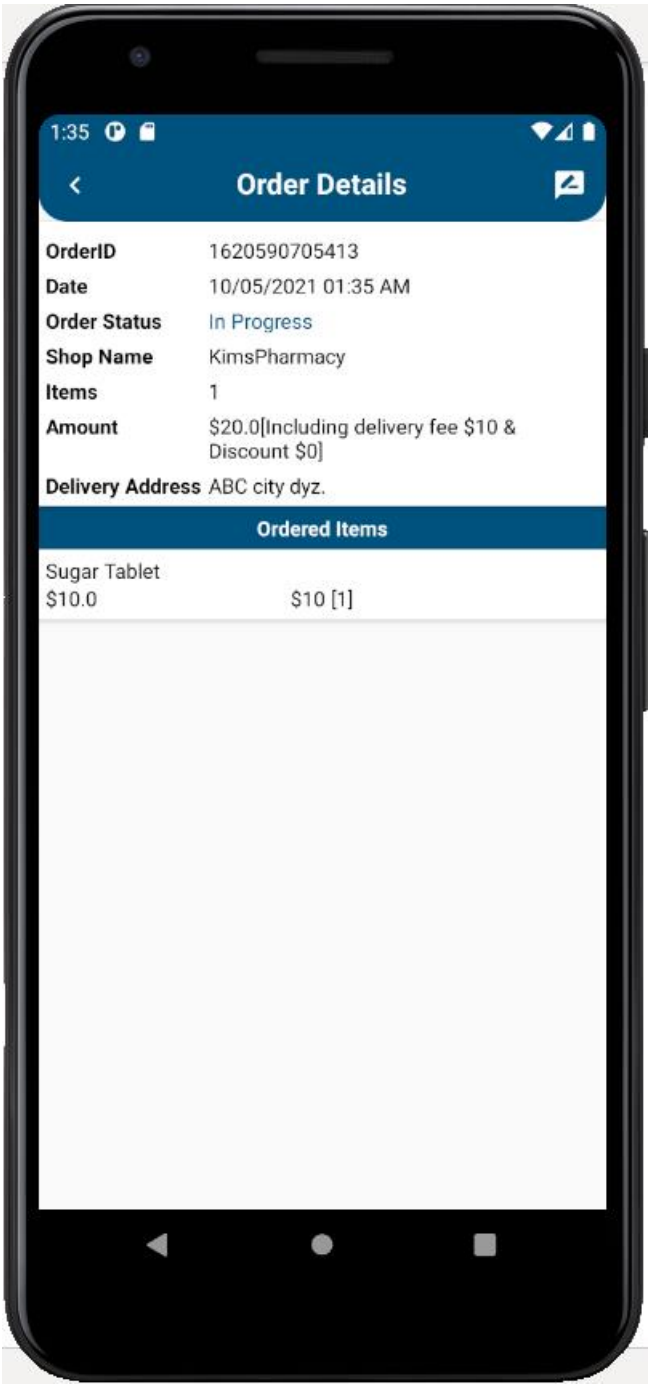
d)Shop page of user



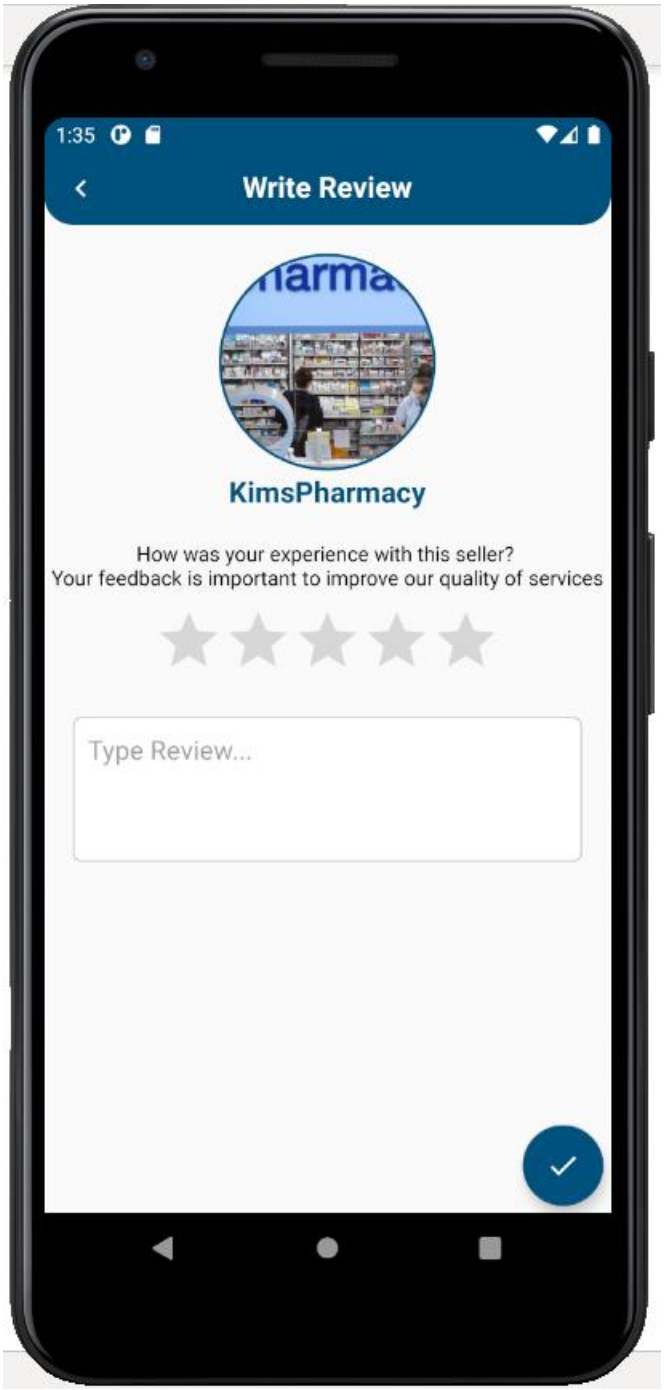
e)Orders page of user



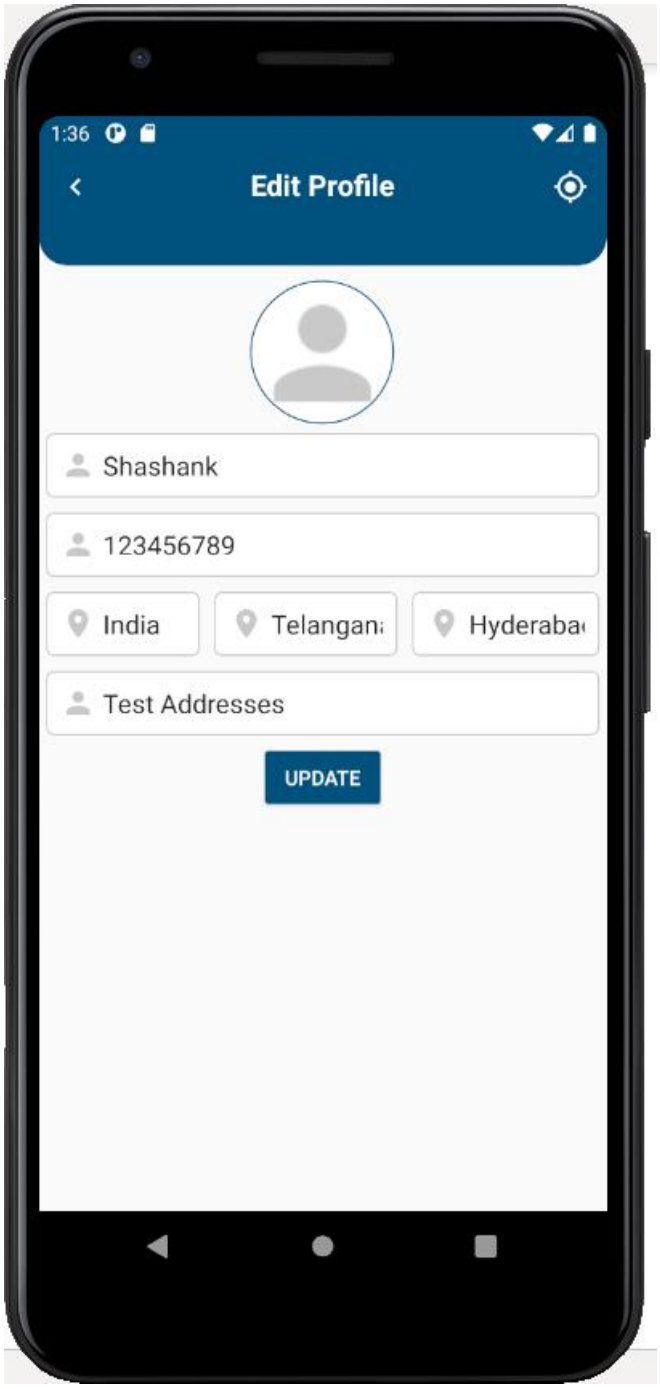
f)Cart page of user



g)Order details page of user

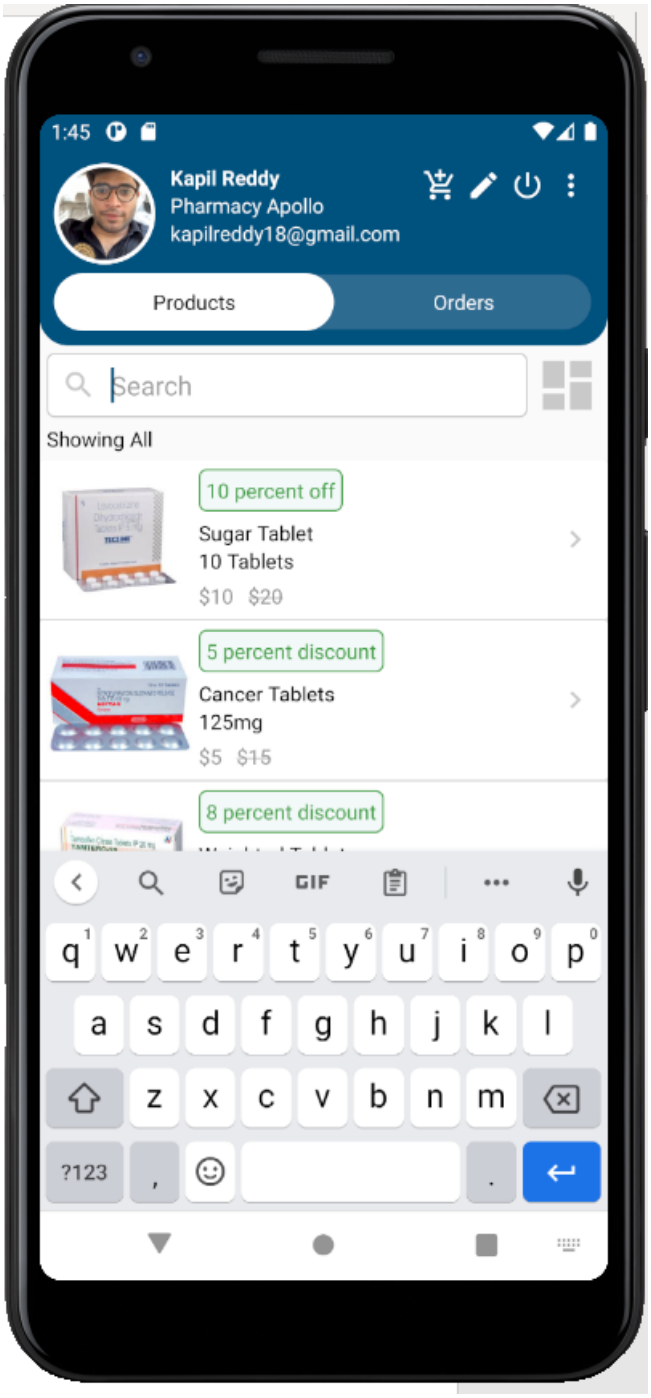


h)ShopReview page of user

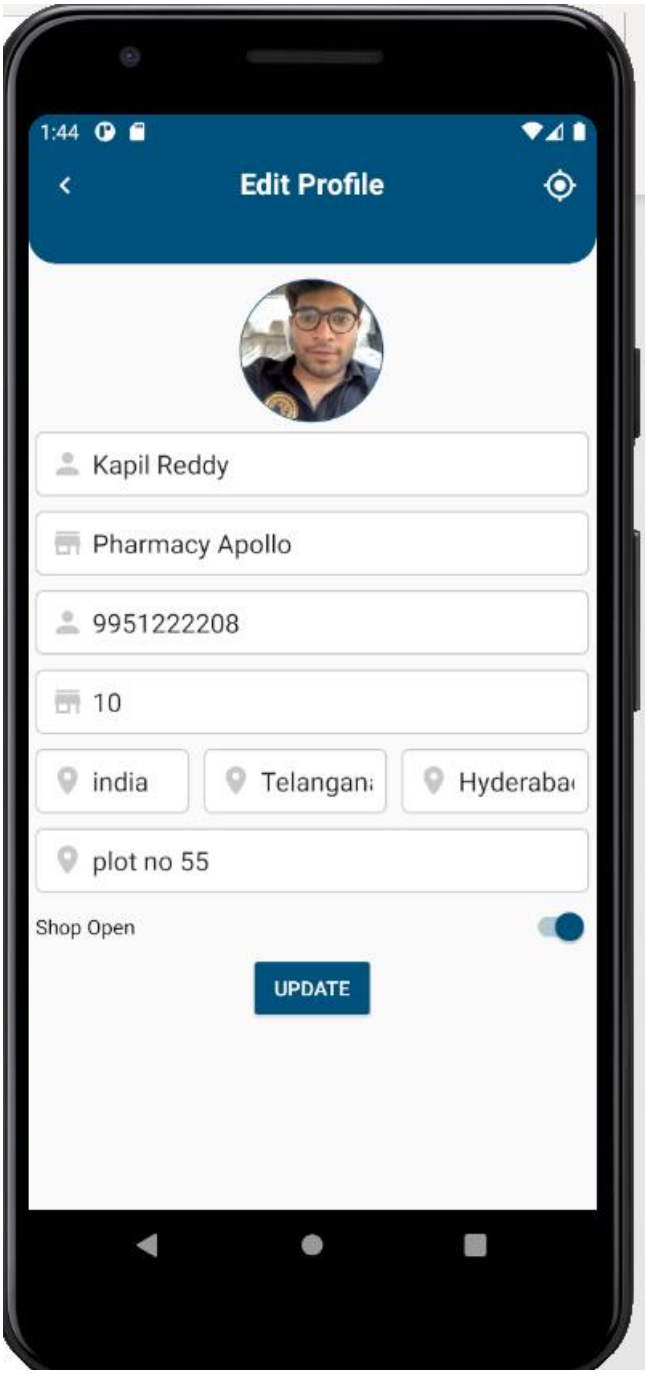


i)Edit Profile page

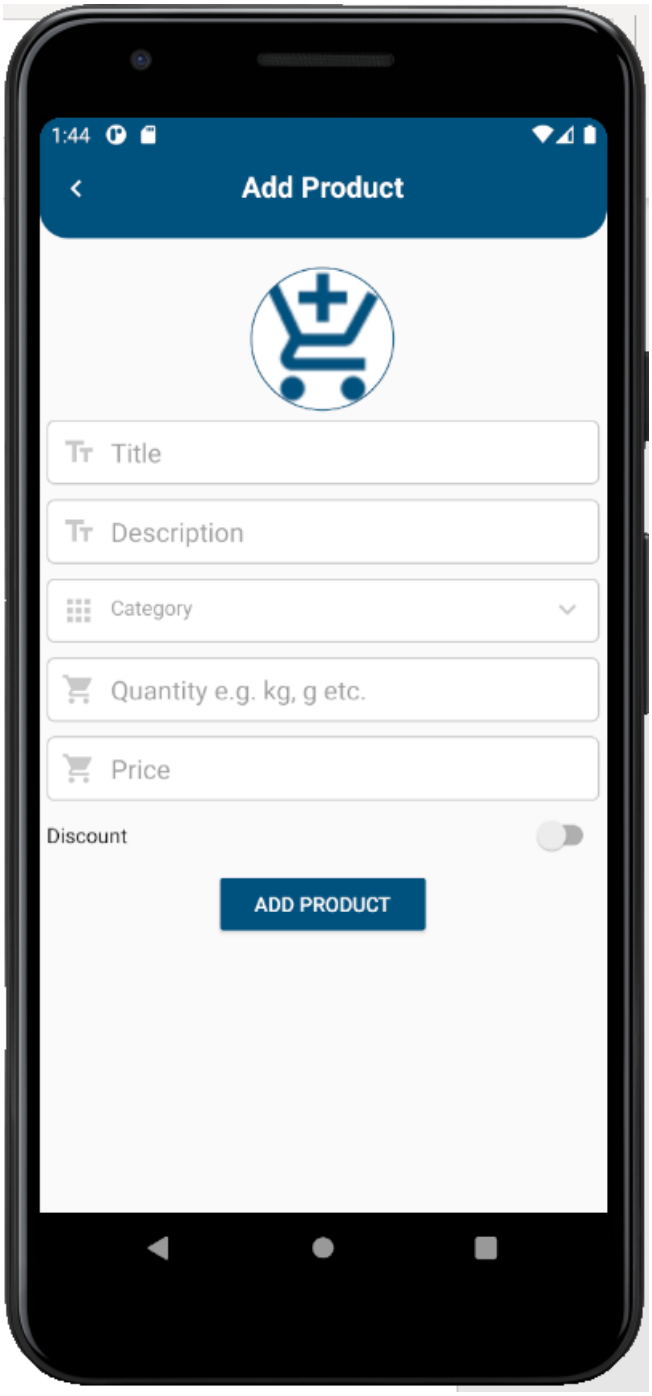
j)Settings page of user



k)Products page of user



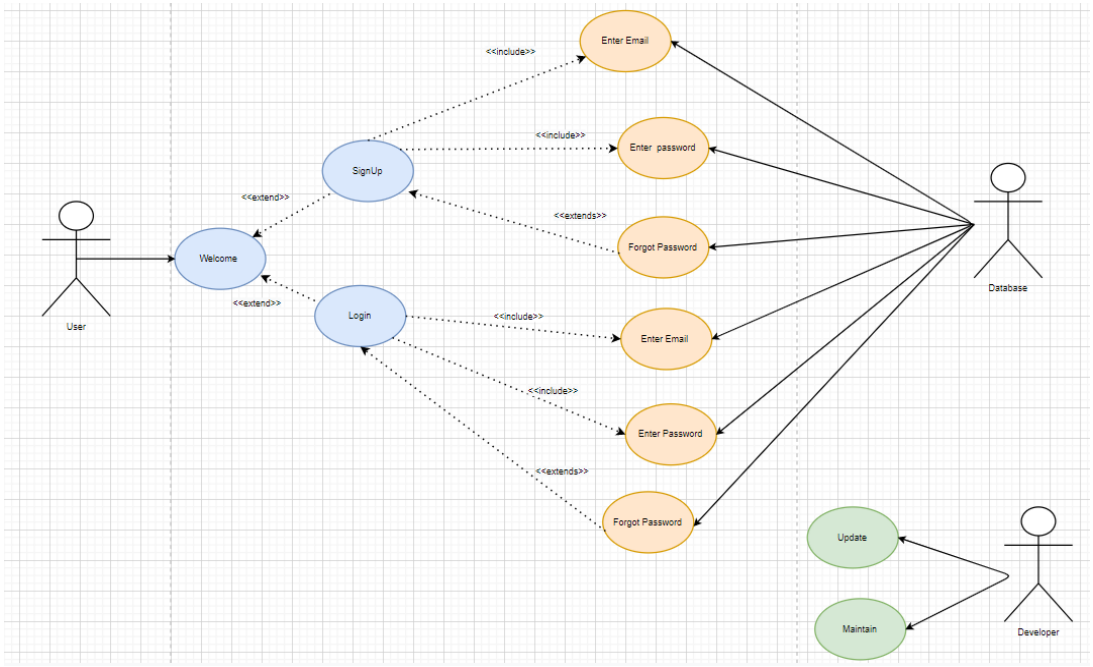
l)Edit profile page



m)Add product page of user

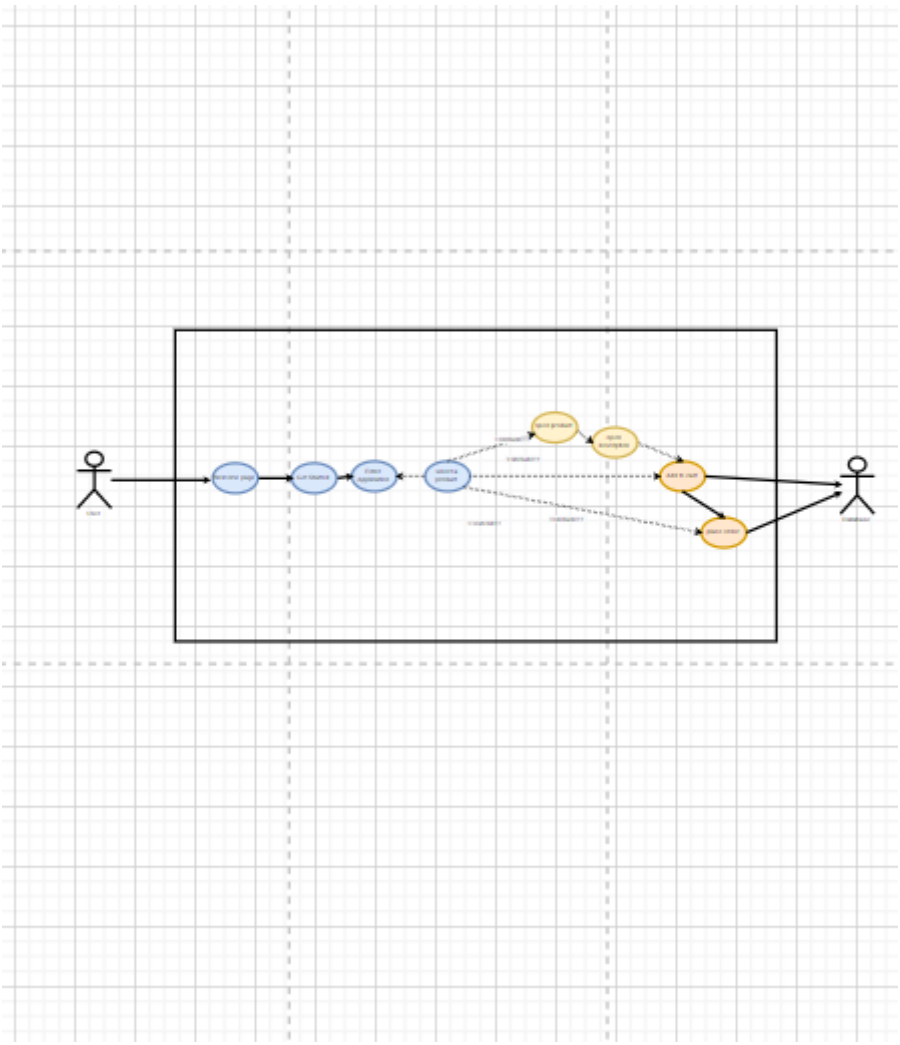
- ***Use Case Designs***

a)Login System Use Case



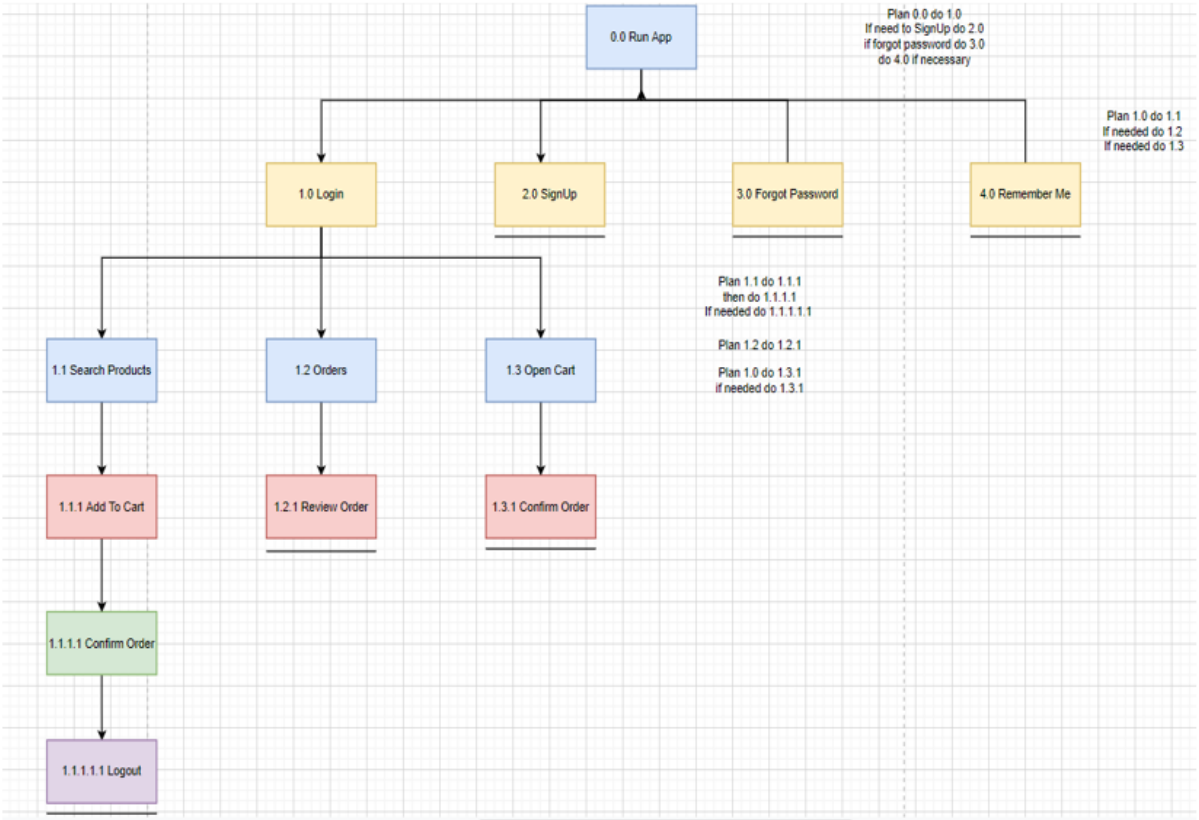
- This is the Use case Diagram of the LogIn and Sign Up of the system which shows the different use cases between database, developer and the user.

b)Place Order Use Case

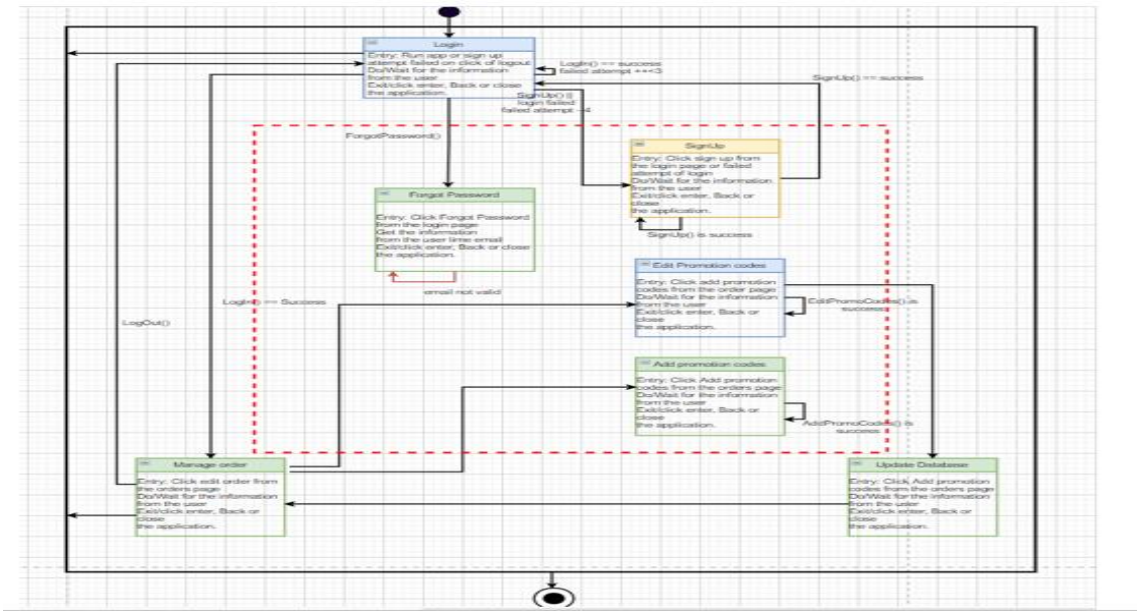


- This is the Use case Diagram of the Product picking and placing a order to the database which shows the different use cases between database, developer and the user.

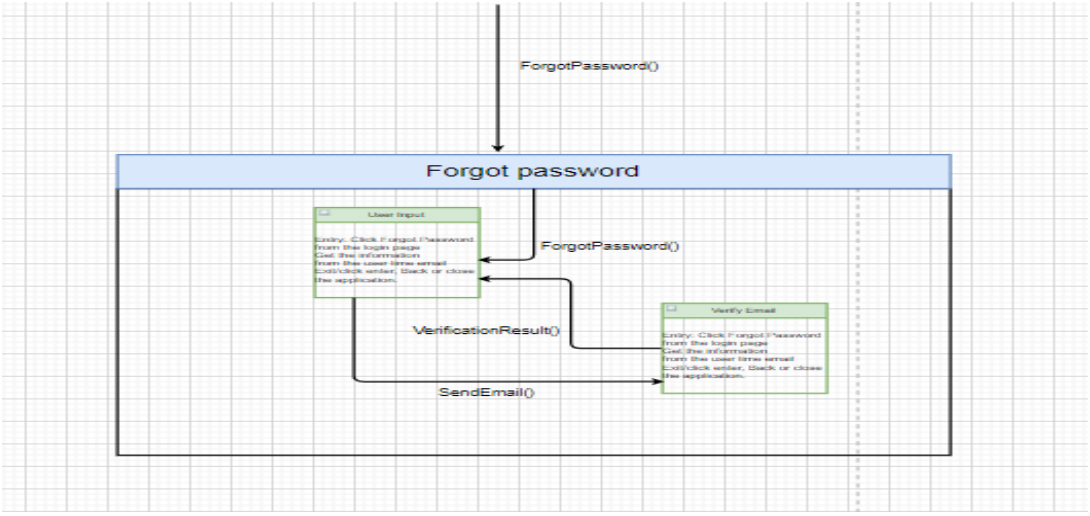
- **Hierarchical Task Analysis :**



- ***Finite State Diagrams :***



Forgot Password Level 2 Diagram :



- This is Diagram of the Finite State Machine which explains the logic and states of the app events and transitions of the different states that interact with user. The apps FSM is divided into two levels the first level displays the states and transitions of Login, Sign Up, forgot password and Remember Me main dashboard transitions including the About Us state. The second level explains the states when the user is logged in to the app and includes state transitions of Add to cart, Review Order and Place order.

State	Description
Login	Prompts user for existing user info
Register	Prompts user for new user info
Forgot Password	Prompts user for existing email and sends reset link
About Us	Displays the info about company/app
View Shops	Displays the Shops based on location
View Orders	Displays the order related details
Categories	Prompts user to select a Category
View order Details	Displays the details of the Order placed
Add promotion Code	Prompts user to add the promo code

- This is the table that shows and explains the definitions that were used in the FSM and a description of each state and transition.

- This table describes the transition and state of each events that are used in the App. Condition shows the functionality of the events taken by the user

About Us State Chart

Current State	Condition	State Transition
About Us	Back()	Login
About Us	Save() && CloseApp()	Exit

Login State Chart

Current State	Condition	State Transition
Login	Login()!=success	Login
Login	Login()==success	View Articles
Login	SignUp()	Sign Up
Login	ForgotPass()	Forgot Password
Login	About()	About Us
Login	Save() && CloseApp()	Exit

Register State Chart

Current State	Condition	State Transition
Sign Up	SignUp()!=success	Sign Up
Sign Up	SignUp()==success	Login
Sign Up	Login()	Login
Sign Up	Save() && CloseApp()	Exit

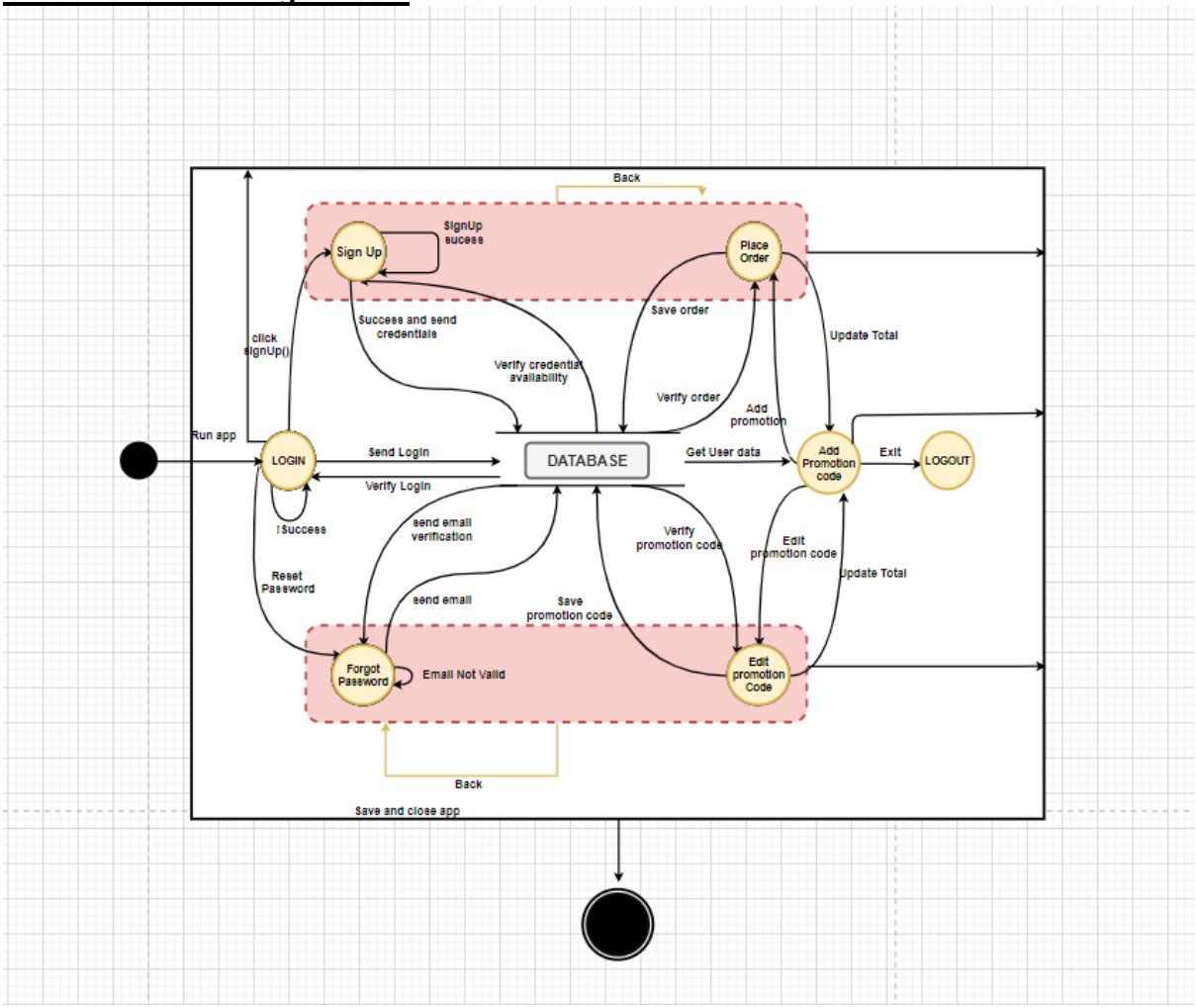
Forgot Password State Chart

Current State	Condition	State Transition
Forgot Password	EmailInValid()	Forgot Password
Forgot Password	Back()	Login
User Input	SendEmail()	Verify Email
Verify Email	VerifyResult()	User Input
Forgot Password	Save() && CloseApp()	Exit

View Shops and Medicines

Current State	Condition	State Transition
View Shops	UpdateShops ()	Shops
View Product Categories	Categories()	Categories
View AboutUs	AboutUs()	About Us
Logout	LogOut()	Login
Update Shops	Save() && CloseApp()	Exit

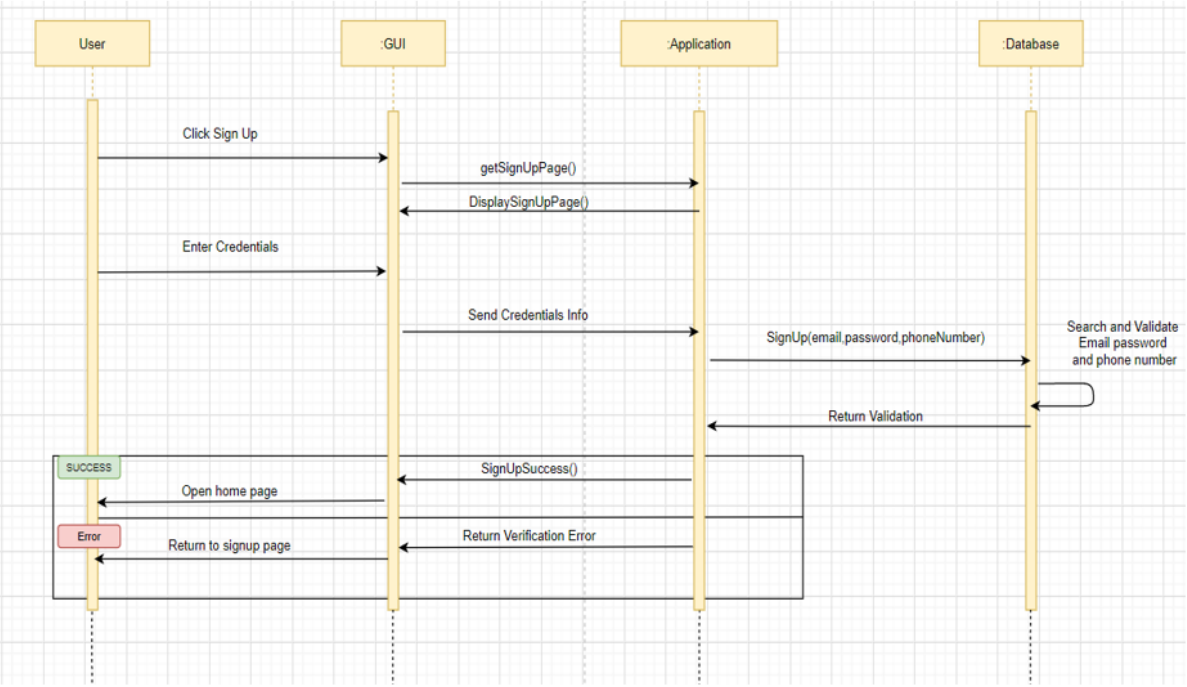
- ***Data Flow Diagram :***



a) *Data Flow Diagram*

This is the data flow diagram which describes the flow of the functionalities and events through the apps system. It represents what activities request or respond in the form of data flows.

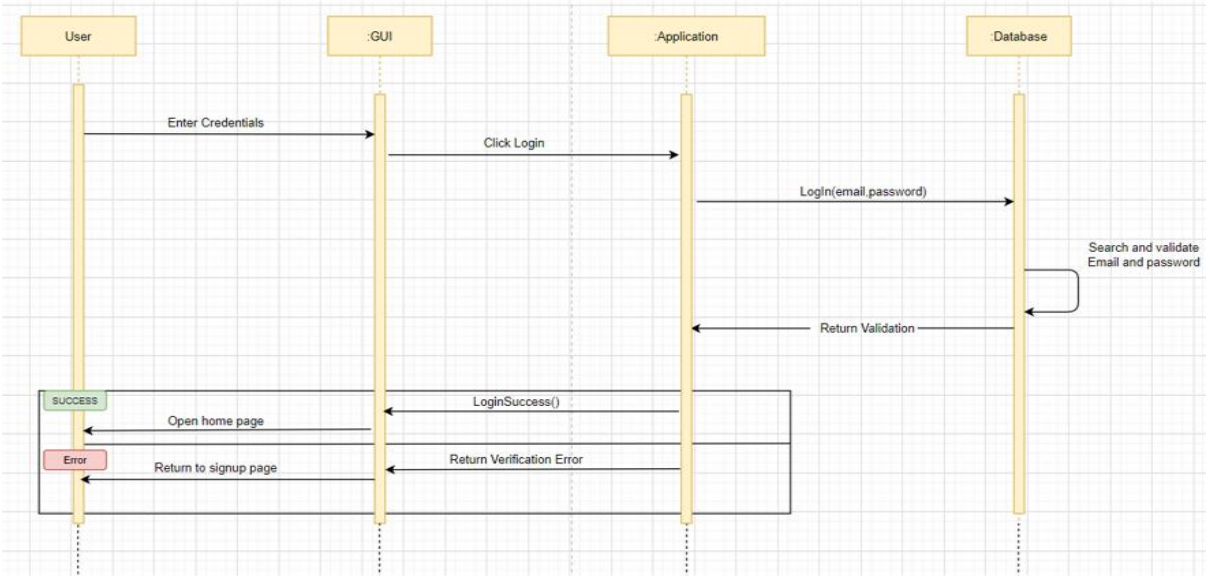
- ***Sequence Diagrams:***



a) *Sign Up Sequence Diagram*

This is the Sequence diagrams that describe how—and in what order—a group of objects works together. These diagrams are important to understand requirements for a new system existing process. First one is the Sequence diagram of the Login systems that shows the sequence of activities in order thru the Login process.

This is the Sequence diagram of the Sign Up systems that shows the sequence of activities in order thru the Sign Up process.

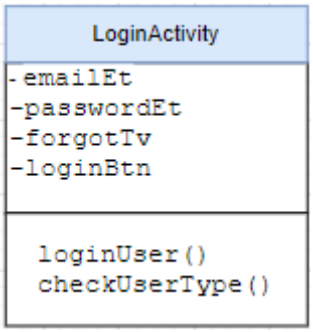


b) *Login Sequence Diagram*

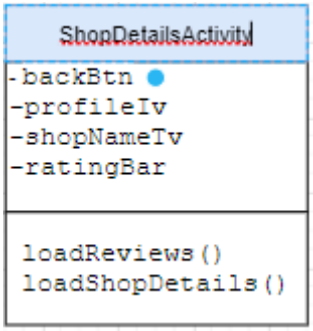
CLASSES

class explanations:

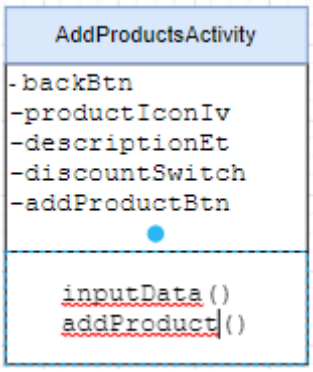
- *LoginActivity.java*
  - *Here the login functionality of the user is taken care by the code.*
  - *This gets invoked on the click of the LOGIN button from the GUI.*



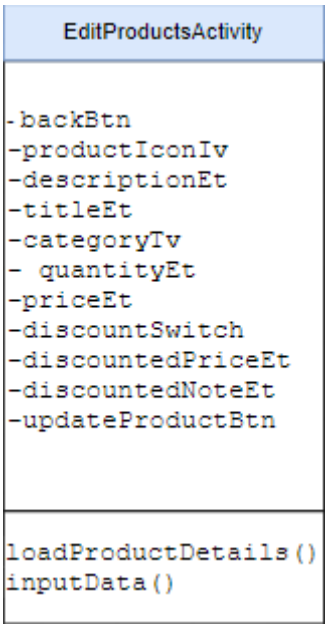
- *ShopDetailsActivity.java*
  - *Here the creation and processing of the shop takes place by the code.*
  - *This gets invoked on the click of the Shop icon button from the GUI.*



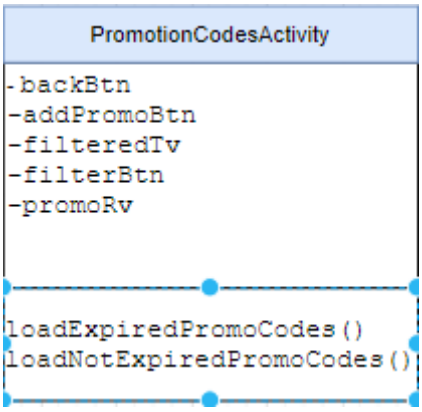
- *AddProductActivity.java*
  - *Here the adding and saving of the product takes place by the code.*
  - *This gets invoked on the click of the Add Product button from the GUI.*



- *EditProductActivity.java*
  - *Here the editing and saving of the product takes place by the code.*
  - *This gets invoked on the click of the Edit Product button from the GUI.*



- *PromotionCodesActivity.java*
  - *Here the editing and saving of the promotion codes takes place by the code.*
  - *This gets invoked on the click of the promotion codes button is clicked from the GUI.*



- *AdapterCartItem.java*
  - *Here the adding and saving of the cart items takes place by the code.*
  - *This gets invoked on the click of the Add to cart button from the GUI.*

TESTING

Test Used:

We have used the Black-box White-box testing in our code base.

- *Black-Box :*
  - **Black Box Testing** is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.
  - Initially, the requirements and specifications of the system are examined.
  - Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
  - Tester determines expected outputs for all those inputs.

- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.
-
-

Test case #	Test goal	Expected result	Actual result
1.	Readability of the GUI elements of the project	Clear and visible color schemes for GUI elements	The GUI has good design implementation and coloring of each element by making them visible to the user
2.	Identifying errors in the functionality of the buttons and functions	Each button and functionalities need to be designed and described clearly and have a matching behavior	Every button and function of any page clearly states the functionality logic behind it and functions accordingly
3.	Identifying any errors of the input/output data fields	Output functionalities of the project have to operate according to the function logic and input formats have to be described accordingly	Input fields were tested for proper entries and output functionalities display relevant data according to the input

- *White-Box :*
  - **White Box Testing** is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.
  - It is one of two parts of the Box Testing approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or end-

user type perspective. On the other hand, White box testing in software engineering is based on the inner workings of an application and revolves around internal testing.

- The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.
- White box testing involves the testing of the software code for the following:
- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis
- The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

Following are important WhiteBox Testing Techniques used while testing:

- Statement Coverage
- Decision Coverage
- Branch Coverage
- Condition Coverage
- Multiple Condition Coverage
- Finite State Machine Coverage
- Path Coverage

Test case #	Test goal	Expected result	Actual result
1.	Identifying any errors and correctness of	Method calls have to be formatted according to the user	Method calls and responses are set

	method calls and the use of proper classes	and seller requirements with the use of proper classes	according the Requirements
2.	Identifying errors in the Firebase authentication, Firebase storage and Firebase database connections	Firebase dependencies have to be imported and connected to the original project libraries	Firebase dependencies are imported properly to Gradle.module settings and the project is connected to the Firebase console
3.	Identifying the correctness of data storage and retrieving all the information from the database without hard coding	Every data type should have its own entity in the database with proper relevance of entity elements	All the information of users and Posts are stored as a separate entities and data is retrieved directly from the Firebase database
4.	Check for proper use of function activities and navigation menus	Individual activities have to be created as a separate class activity with proper design structure including xml design elements	Activities are created properly as separate class and with its own xml, and design elements as coloring, style and menu have its own xml files

CONCLUSION

Mostly people buy the product from the medical store that means people need to go there physically to buy the medicine. Due to dependency of the customer on medical store shopkeepers takes benefit of that. They charge more than the actual cost of product from the Customer. Medical website will provide offers the customer which they usually not get from the shopkeepers. Most of the Website do not show the product like wheelchair, surgery products, necessary blood requirement for the person(A-), which is difficult to find any website or place.

The aim of the site is reliability, availability, security, portability. There can be scope of improvement in the website. After implementation of this website it will not only

helpful for the user but also signifies the study content to everyone.

Personal mentions:

- Special thanks to Dr. G for his continuous guidance throughout – Sincere gratitude

- [1] Laurent, M. R.; Vickers, T. J. (2009). "Seeking Health Information Online: Does Wikipedia Matter?". Journal of the American Medical Informatics Association 16 (4): 471–479. doi:10.1197/jamia.M3059. PMC 2705249. PMID 19390105. Nimishapanchalhttp://www.slideshare.net/NIk_Panchal/online-shopping-system
- [2]http://cysonline.org/article.asp?issn=2229-5186;year=2010;volume=1;issue=1;spage=16;epage=25;aulast=Anand;type=3
- [3]http://www.consumersresearchcncl.org/Pharmacists/pharmacist_chapters.html

C O D E

java code :

FilterProduct.java

```
package com.pharmac.buyer;

import android.widget.Filter;

import com.pharmac.buyer.adapters.AdapterProductSeller;
import com.pharmac.buyer.models.ModelProduct;

import java.util.ArrayList;

public class FilterProduct extends Filter {

    private AdapterProductSeller adapter;
    private ArrayList<ModelProduct> filterList;

    public FilterProduct(AdapterProductSeller adapter, ArrayList<ModelProduct> filterList) {
        this.adapter = adapter;
        this.filterList = filterList;
    }

    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        FilterResults results = new FilterResults();
        //validate data for search query
        if (constraint != null && constraint.length() > 0){
            //search filed not empty, searching something, perform search

            //change to upper case, to make case insensitive
            constraint = constraint.toString().toUpperCase();
            //store our filtered list
            ArrayList<ModelProduct> filteredModels = new ArrayList<>();
            for (int i=0; i<filterList.size(); i++){
                //check, search by title and category
                if (filterList.get(i).getProductTitle().toUpperCase().contains(constraint) ||
                        filterList.get(i).getProductCategory().toUpperCase().contains(constraint) ){
                    //add filtered data to list
                    filteredModels.add(filterList.get(i));
                }
            }

            results.count = filteredModels.size();
            results.values = filteredModels;

        }
```

```
        else {
            //search filed empty, not searching, return original/all/complete list

            results.count = filterList.size();
            results.values = filterList;
        }
        return results;
    }

    @Override
    protected void publishResults(CharSequence constraint, FilterResults results) {
        adapter.productList = (ArrayList<ModelProduct>) results.values;
        //refresh adapter
        adapter.notifyDataSetChanged();
    }
}
```

Constants.java

```
package com.pharmac.buyer;

public class Constants {

   public static final String FCM_KEY = "AAAA9xV4t7Y:APA91bFh-
pZ0yrLPu1BXU8NBwECFdSSvRTA51xgiB75FdZ1hh2Cmrxt_l-
ND2oLmrefa6BFSV8myIhtIxqIdI9XOJYsk-Kg--u9Ty1e4rgANfWzEna-
SdVaj9sIKSNyoaqJ5SNjFNMA8";
   public static final String FCM_TOPIC = "PUSH_NOTIFICATIONS";

   //product categories
   public static final String[] productCategories = {
         "Beverages",
         "Beauty & Personal Care",
         "Baby Kids",
         "Biscuits Snacks & Chocolates",
         "Breakfast & Dairy",
         "Cooking Needs",
         "Frozen Food",
         "Fruits",
         "Pet Cate",
         "Pharmacy",
         "Vegetables",
         "Others"
   };

   public static final String[] productCategories1 = {
         "All",
         "Beverages",
         "Beauty & Personal Care",
         "Baby Kids",
         "Biscuits Snacks & Chocolates",
         "Breakfast & Dairy",
         "Cooking Needs",
         "Frozen Food",
         "Fruits",
         "Pet Cate",
         "Pharmacy",
         "Vegetables",
         "Others"
   };
```

FilterProductUser.java

```
package com.pharmac.buyer;

import android.widget.Filter;

import com.pharmac.buyer.adapters.AdapterProductUser;
import com.pharmac.buyer.models.ModelProduct;

import java.util.ArrayList;

public class FilterProductUser extends Filter {

    private AdapterProductUser adapter;
    private ArrayList<ModelProduct> filterList;

    public FilterProductUser(AdapterProductUser adapter, ArrayList<ModelProduct>
filterList) {
        this.adapter = adapter;
        this.filterList = filterList;
    }


    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        FilterResults results = new FilterResults();
        //validate data for search query
        if (constraint != null && constraint.length() > 0){
            //search filed not empty, searching something, perform search

            //change to upper case, to make case insensitive
            constraint = constraint.toString().toUpperCase();
            //store our filtered list
            ArrayList<ModelProduct> filteredModels = new ArrayList<>();
            for (int i=0; i<filterList.size(); i++){
                //check, search by title and category
                if (filterList.get(i).getProductTitle().toUpperCase().contains(constraint) ||
                        filterList.get(i).getProductCategory().toUpperCase().contains(constraint) ){
                    //add filtered data to list
                    filteredModels.add(filterList.get(i));
                }
            }

            results.count = filteredModels.size();
            results.values = filteredModels;

        }
        else {
            //search filed empty, not searching, return original/all/complete list
```

```
            results.count = filterList.size();
            results.values = filterList;
        }
        return results;
    }

    @Override
    protected void publishResults(CharSequence constraint, FilterResults results) {
        adapter.productsList = (ArrayList<ModelProduct>) results.values;
        //refresh adapter
        adapter.notifyDataSetChanged();
    }
}
```

Code References:

- [1] http://www.dallaspharmacies.com/
- [2] http://www.positiveway.com/recovery/internet_pharmacy.htm

ABOUT ME

Biography:

I have joined Masters in CSC in the year 2019 and loved it since then. I have worked on many projects during my course period and learned a lot of new stuff.

My Goals:

My goal is to get a good job in a big Multi National Company(MNC), and make my parents proud.

My CS Experience:

I have learned many new technologies during my course period. Special thanks to my mentor DR. G without him I would not be able to complete this project. He has motivated and guided me a lot. I like the way how courses were build and taught.

END PAGE