**ONESHOP**

**Mobile Applications** **Sahana Kanaparthi**

**Professor. Bo Sheng** **Pranavi Machinani**

**INTRODUCTION**

The Internet is a massive network that connects even the most remote parts of the globe. This means that one can be in multiple locations at the same time without ever leaving your premises. This takes multi-tasking to whole new level since one can be at work and visit a market located several kilometres away at the same time. This made online shopping become popular.

Online grocery shopping has been a thing that allows customers to buy fresh food or produce via the internet. As a result, people can save time, energy and money by purchasing things online at a discounted price. The offers on the internet can be easily compared, and the consumer can purchase the goods based on their preferred criteria, such as price, quality, and other factors.

1. **Project Statement**

Our online grocery shopping (One Shop) is an android application where users find it convenient and easy to shop online instead of waiting in long queue, carrying heavy bags and struggling with trolleys. Our app will help customers search for nearby grocery stores and select the required produce he/she wants to buy from the selected store and add item to cart. Cart contains product details and the amount. User can make payment through by credit card or debit card. After payment, grocery will be delivered to the specified address This app is useful for everyone but mostly preferred in the following scenarios, when the customers are busy with their work or don’t have transport to go and buy groceries and most importantly in this COVID situation it is high risk to go out and get exposed, so shopping online is easiest and safest way to get groceries delivered to their door step. We got the idea of building an application like ONE SHOP when we faced similar situations where we couldn’t go out and shop. There are similar other applications in the market like Instacart which delivers groceries. Since this system is made in android it is easily available in smart phones. Customers who have phones supporting internet can easily use the application by installing it in their smartphones.

1. **Application Design**

* OneShop Application supports minsdk 21 (Lollipop 5.0) and target 29(Android Q)
* This Application targets smartphones devices

Our Application is mainly built with 4 modules:

* Authentication
* UI
* Data
* Payment.

**Authentication:**

The purpose of this module is to authenticate the user who is trying to login into the application, here we used FirebaseAuth API to authenticate user.

Firebase Auth Provides two functions:

1. Create an entry for user in Firebase when the user is registering for the first time in the application.
2. Authenticating the user who already have entry in Firebase.

**UI:**

The visual representation of the Application is done in this module by using one of the App components i.e. Activity

Several activities are declared for different use cases like

* Zip code search and Store selection
* Show the products based on type, for instance if the type is Vegetables will display respective items.
* Description of selected product and control to add product in to cart.
* Viewing the list of items that have been added into the cart.
* Generating the receipt after successful order.

**Data:**

In this Module we used CRUD operations for retrieval/modifying/inserting the data according to the required functionalities.

Some of the basic functions in this module are:

* Retrieving the stores according to the ZIP code that has been entered by the user.
* Retrieving the Items based on type selected by the user.
* Inserting the data that have been added into cart.
* Updating the Cart based on the items modified by the user.

**Payment:**

Once the user has completed adding items to the cart, they can proceed and place an order. To place an order an external API is required for Payment Integration.

Here we used GooglePay API as an external API for Payment Integration.

UI

Firebase Authentication

SQLite

GooglePay API

Authentication

Data

Payment

Figure1: High level design of our Application

1. **Application Implementation and Evaluation**

**Data:**

* **DatabaseHelper.Java:** we used this class to create a database, define schema and helper functions for CRUD Operations.
* **Cart.java:** This class is used for retrieving the cart Items.
* **Item.java:** This class is used for retrieving the Items
* **Store.java:** This class is used for retrieving the stores.

**Payment:**

* **Constants.java:** This class is used for changing the App environment, defining the currency and country code for payment, defining supported networks and methods.
* **PaymentsUtil.java**: This class contains helper static methods for dealing with Payment API. Google Pay API was used for payment integration.

**UI:**

* **LoginActivity.java:** Authenticates the user for login to the app using FireBaseAuth.
* **RegisterActivity.java:** Registering the user and creating an entry in Firebase using FirebaseAuth.
* **MainActivity.java:** This is the main activity which navigates to all entries and gives user control to search Zip code and select stores.
* **ProductsActivity.java:** we used this class to display the type of product
* **ItemsActivity.java:** we used this class to display list the items based on selected type.
* **ProductDetailsActivity.java:** This class will display the detailed description about item and gives control to the user to add item into the cart.
* **CartActivity.java:** we used this class to show list of items that have been added into the cart by the user, delete an item from the cart and update the quantity for the product.
* **SettingsActivity.java:** We used this class to allow user to save their contact information.
* **CartAdapter.java and CartViewHolder.java:** These classes are used to present the list of cart items in a listview.
* **ProductAdapter.java and ProductViewHolder.java:** These classes are used to present the list of products in a listview.
* **ReceiptAdapter.java and ReceiptViewHolder.java:** These classes are used to present the list of items in a receipt in a listview.
* **StoreAdapter.java and StoreViewHolder.java:** These classes are used to present the list of stores in a listview.

1. **References:**

* <https://github.com/Gkemon/Android-XML-to-PDF-Generator> (for generating receipt after making a payment).
* <https://developers.google.com/pay/api/android/overview> (Google pay API for payment integration).
* <https://firebase.google.com/docs/auth/android/start> (firebase authentication for login and user registration).

1. **Experiences and Thoughts:**

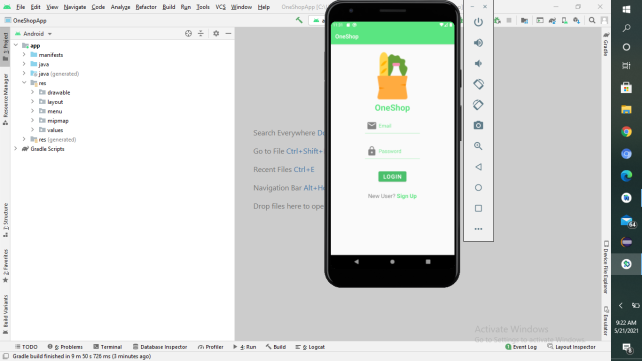
There are many functionalities that can be implemented to make the app more user friendly .Some of these functions would be nice to have in this app are:

* After successfully placing the order, the option to know the status of the order would help the user to keep track of the product.
* Enhancing the UI of the app.
* When 3rd person is delivering the order it would be better to have admin app to keep posting the updates about the delivery to the user app using public server and Restful web services.
* In this app for retrieving the data, local database was used but instead using web services integration would be more optimal.

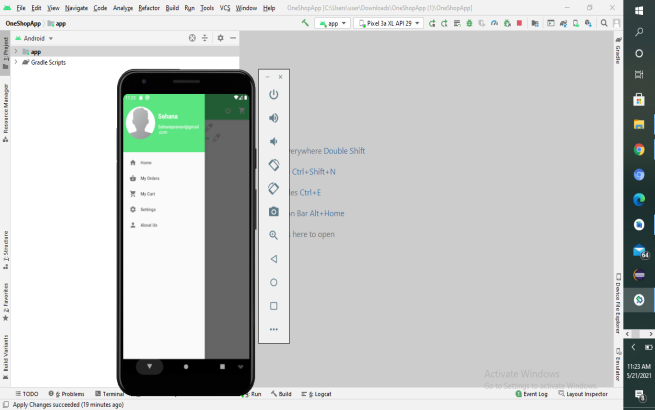
**OUTPUT:**

Below are the output screenshots:

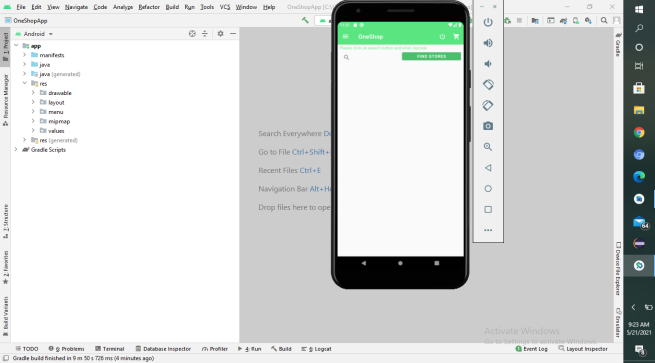
Login page



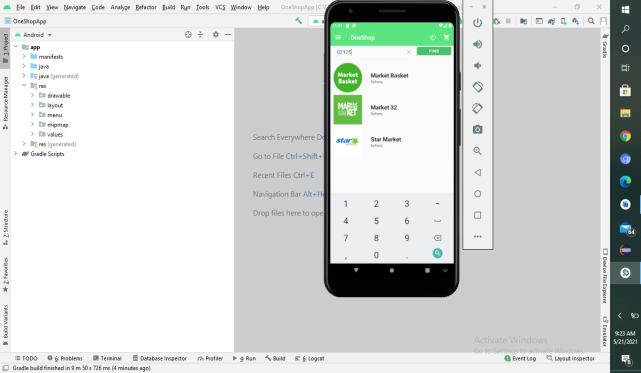
Menu



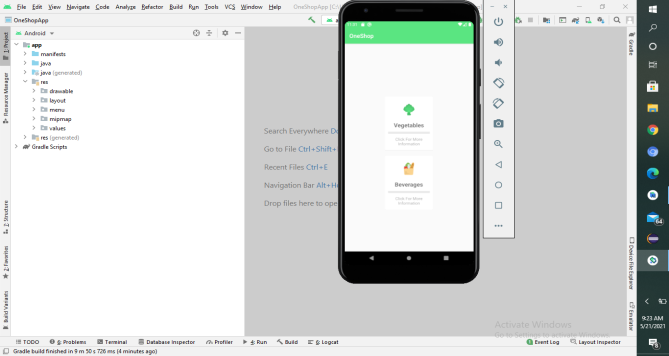
Here, user can enter zipcode and can search for nearby stores



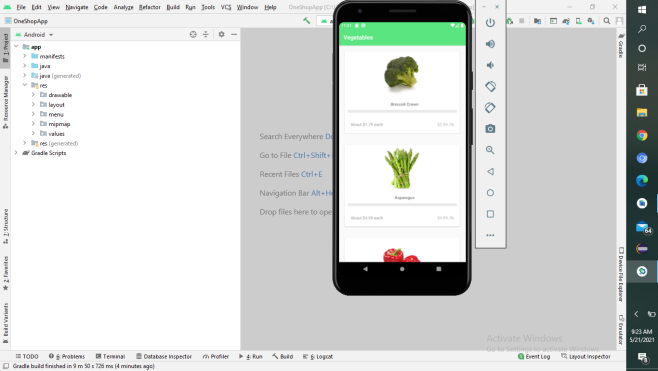
Displaying the list of stores based on zipcode



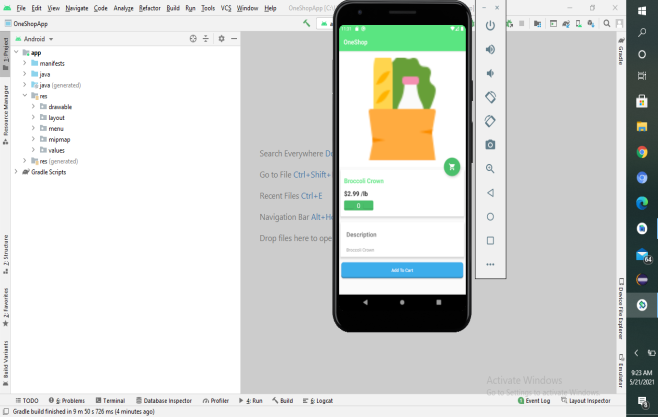
Displaying the type of the product



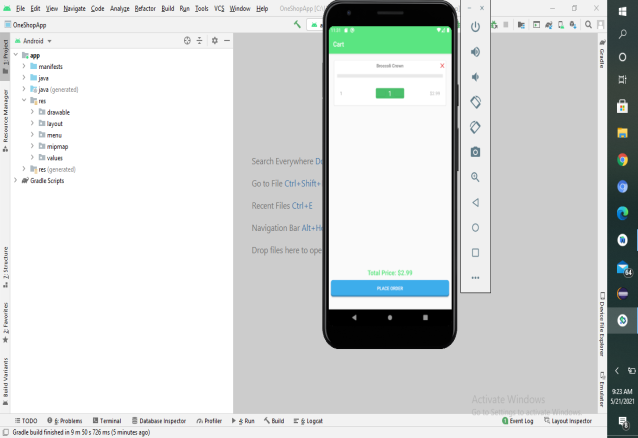
Displaying the list of products(for example vegatables)



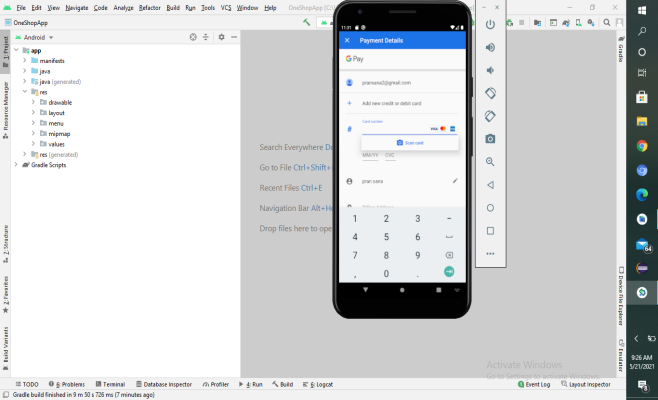
Description about the product



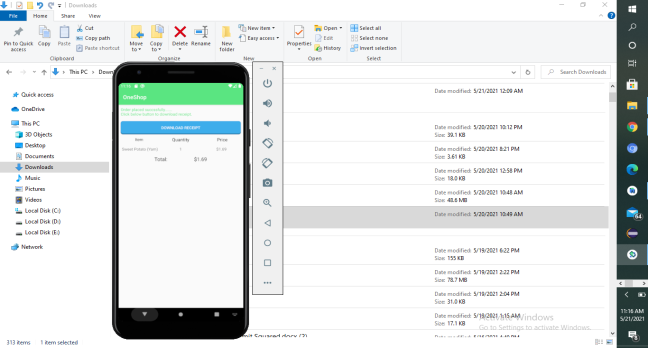
Displays the cart page



Payment page



Option to Download the receipt (generated receipt) page



Displaying the Downloaded receipt

