

# National College of Ireland

<BSc Honours in Computing>

<Software Development>

<2020/2021>

<Sandeep Pulavazhy>

<x16101251>

<x16101251@student.ncirl.ie>

<Salt and Pepper>

Final Technical Report

## Contents

Executive Summary .....	2
1.0 Introduction .....	2
1.1. Background .....	2
1.2. Aims.....	2
1.3. Technology.....	3
1.4. Structure .....	4
2.0 System.....	4
2.1. Requirements.....	4
2.1.1. Functional Requirements.....	4
2.1.1.1. Use Case Diagram .....	4
2.1.1.2. Requirement 1 <Name of requirement in a few words>.....	5
2.1.1.3. Description & Priority.....	6
2.1.1.4. Use Case.....	6
2.1.2. Data Requirements .....	13
2.1.3. User Requirements .....	17
2.1.4. Environmental Requirements .....	21
2.1.5. Usability Requirements.....	21
2.2. Design & Architecture.....	21
2.3. Implementation .....	23
2.4. Graphical User Interface (GUI).....	32
2.5. Testing.....	40
2.6. Evaluation .....	<b>Error! Bookmark not defined.</b>
3.0 Conclusions .....	40
4.0 Further Development or Research .....	41
5.0 References .....	41
6.0 Appendices.....	42
6.1. Project Plan .....	42
6.1. Ethics Approval Application (only if required) .....	<b>Error! Bookmark not defined.</b>
6.2. Reflective Journals .....	<b>Error! Bookmark not defined.</b>
6.3. Other materials used .....	<b>Error! Bookmark not defined.</b>

## Executive Summary

The main idea behind this project report is to describe the project idea that I have developed and pointing out the key features and working of the application. This technical report will provide more information on how my system works, the difficulties that I faced and how I overcome that situations. Salt and pepper is an innovative, smart android application, that can be used for shopping Asian groceries online. This application run as a platform where users and shop owners can do their business by their own. This application have an inbuilt AI voice assistant that will help the users to scroll the pages, direct from one page to another and even help in choosing the products and add them to the cart. The user will get a feeling that this application is talking to them, as this voice assistant will do all these using voice commands from the user. The competitor of this application is “Buymie”. But the android application, that I created works totally different as that of “Buymie”. This report will provide all details based on how this system works. The Introduction part of this report contains more details of project such as the background and aim of this project, the technology used in order to develop the project. The key points are provided under technical requirements, GUI and implementation part of this document.

## 1.0 Introduction

### 1.1. Background

Considering the past decade, it is evident that technology has improved a lot. The invention of computers and internet has changed the life style of people dramatically. The invention of computers/smartphone and internet has even changed the shopping behaviour of people to next level. The recent research conducted by “OBERLO” website says that one in 4 people in this world uses internet and mobile/computers for shopping. This shows that more people are moving to online shopping. The main reason why people shop online is because of the convenience. The better price provided by the ecommerce attracts more people into this online shopping world. From a user point of view, it is hassle free to shop online rather than going to shops physically for buying products. People can save hours of time through online shopping and everything is readily available within few clicks. The ecommerce has proved its strength and capabilities during the pandemic across the world. People were able to order anything they want and get it delivered to home, while the shops were physically closed during this pandemic. This drive more people to depend on ecommerce website.

In Ireland there are no websites/applications that can integrate all Asian groceries under one roof. My idea is to provide a platform to bring all these Asian grocery shops under one roof and thus providing a service where users can shop all the grocery products online. Considering all these facts I came to a conclusion of developing an innovative online Asian grocery application as it has more business scope in this present world.

### Aims

The main aim of this project idea is to develop an android application where users can order Asian groceries online. This application will run as a platform on which the shop owners can register their shop details and add the product details that they wish to sell. Once the shop owner provided all these information, these data will be saved on to the database. When a user register into this application, this app will show all the shops and the products that has been registered by the shop owners. The main feature of this application is an inbuilt voice assistant. This voice assistant will assist both users and shop owners. For users, the voice assistant will help in activities such as scroll, move from one page to another, search

for the products and add the products to the cart. For shop owners, this voice assistant will help in adding the product, move from one page to another and checking the orders. This system will introduce a new world of AI enabled shopping application, and thereby providing even more better shopping experience.

### 1.2. Technology

Choosing technology was quite difficult for this project as there are many technologies available today. The technologies that has been used for this project is given below.

<b>Technologies used for this project</b>	<b>Importance of this technology in this application</b>
Android Studio	Since this application is an android application, android studio is used as an IDE for creating the application
Java	Java is used for as the programming language for developing this application
Firebase Realtime Database	Firebase database is used for storing the details of the users, shop owners, and saving the product details added by the shop owners.
Firebase Cloud Storage	Firebase Cloud Storage is used for saving the profile images of the users, shop owners and the images of the products added by the shop owner
Firebase Authentication	This technology is used for enabling the login/signup feature for the app.
Alan AI SDK	Alan Ai SDK is used in this application to for enabling the AI voice assistant.

#### **Android Studio:**

The official IDE for creating android based application is Android Studio. Android studio is the best IDE available in the market for developing android based applications. Since this my application runs on android operating system, I chose android studio as the IDE.

#### **Java:**

Java programming language is quite easy to use as it has simple syntax. Java is considered as the main programming language in android application development. Java helps to save a lot of time in development as it provides massive collection of open source libraries.

#### **Firebase Realtime Database:**

My application demands a database that can collect the data from one side and show it to another side at the same time. Firebase real time database provides all necessary functions that my application demands. This database in my application will function as an API. For example, when a shop owner register a new shop/add a new item to sell, this database will save this and show the newly registered shop/ newly added item to the user in real time. Firebase database help a lot in the smooth running of this application.

#### **Firebase Cloud Storage:**

Firebase cloud storage are used to save all the images such as profile images of the users, profile images of the shop owners/shops and the images of the item added by the shop owners.

#### **Firebase Authentication:**

Firebase Authentication is used for enabling the feature of sign up, sign in feature. Firebase authentication helps this application to register and sign in a user and even help to recover the password when a user forgot their password by any chance.

### 1.3. Structure

This project Report consists of 4 main parts. They are :

#### 1) Introduction:

This part provides a brief description of the project which includes the background and aim of the project. The technologies that are used for developing this project is also included in this part.

#### 2) System

This part describes about the requirements of the project such as functional requirements, which contains the use case diagram of this application ,the data requirements which explains about the how and where the data are stored, user requirements which describes about the basic requirements that a user must have in order to use this application, Environmental Requirement describes about the basic environmental requirements that are needed the smooth running of this application and finally usability requirements describes how this application can use with ease.

## 2.0 System

### 2.1. Requirements

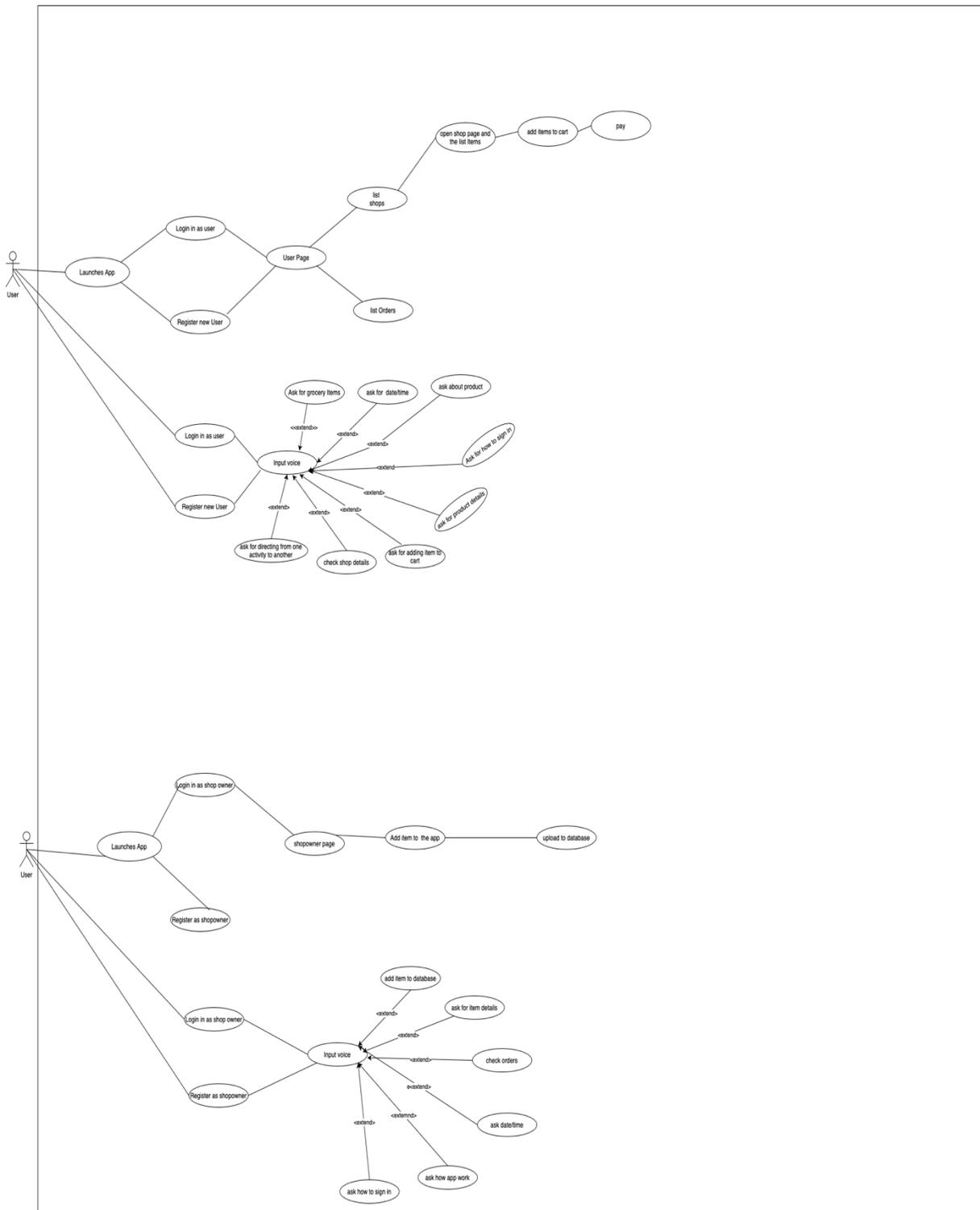
Both functional and non-functional requirement for this application are provided in their respective sections.

#### 2.1.1. Functional Requirements

The below provided is the use case diagram for the this entire system. This use case gives a clear idea about how the system works. The fundamental requirements are well described in this use case diagram.

##### 2.1.1.1. Use Case Diagram

**NOTE: The use Case diagram is given below in the next page for better visibility**



Use diagram of complete application:

2.1.1.2. Requirement 1 <User Registration>

### 2.1.1.3. Description & Priority

In order to use this application, it is very important that a user should register first. There are certain fields that the user needs to fill out, in order to register as user.

### 2.1.1.4. Use Case

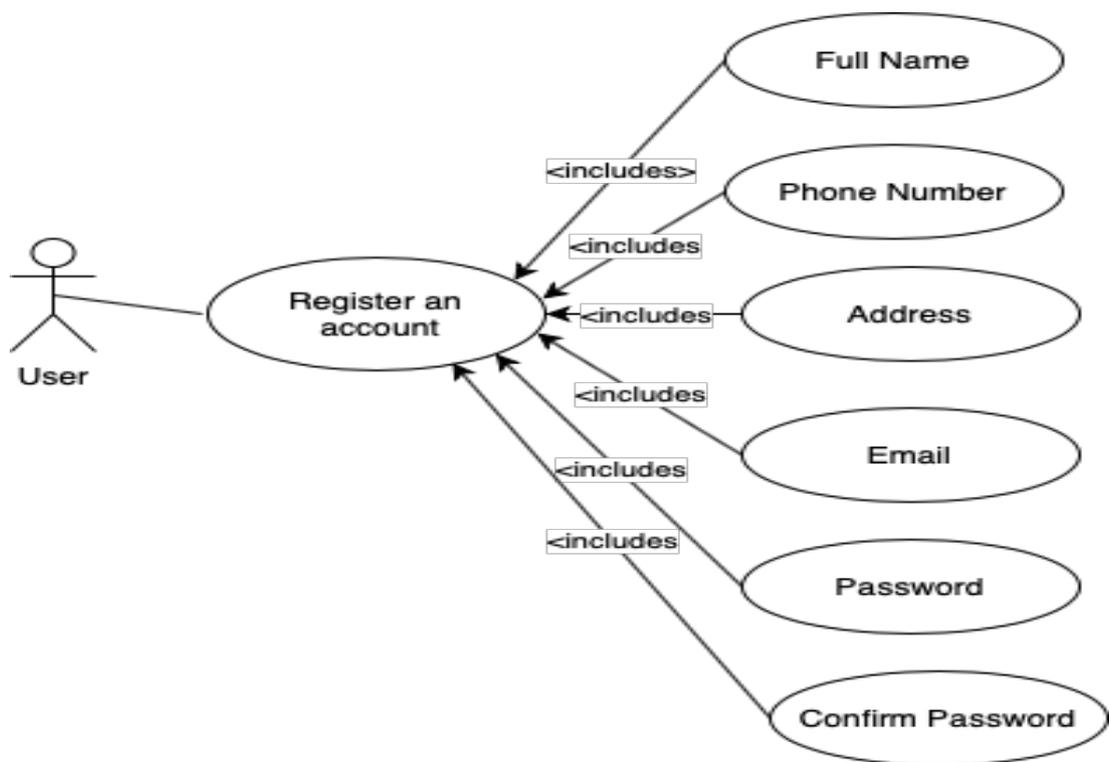
#### Scope

The scope of this use case is to create an account for the user.

#### Description

This use case describes the how an account is created for user

#### Use Case Diagram



Use Case Diagram for User Registration

#### Flow Description

##### Precondition

Only new users can register. Users with existing account cannot register

##### Activation

This use case starts when an User clicks on the register button

##### Main flow

1. User click on the app icon, the app launches
2. App one the login page
3. User click on the register here button to create an account
4. User enter all relevant information asked in the register form
5. User click on register button, then application will connect to database and save all information.

##### Alternate flow

A1 : <User couldn't complete the required fields>

1. The system will check if the user entered all relevant information
2. The User not input data in certain fields
3. The app will notify by stating, which fields are not complete

### **Exceptional flow**

E1 : <User already exist>

4. The system will check for the user in the database based on the information provided
5. The user will get notified by the application, stating that the account already exists.
6. The application will take user to registration form again.

### **Termination**

The system presents the user page when the user successfully registered account.

### **Post condition**

The system goes into a wait state when the database is not connected to the application.

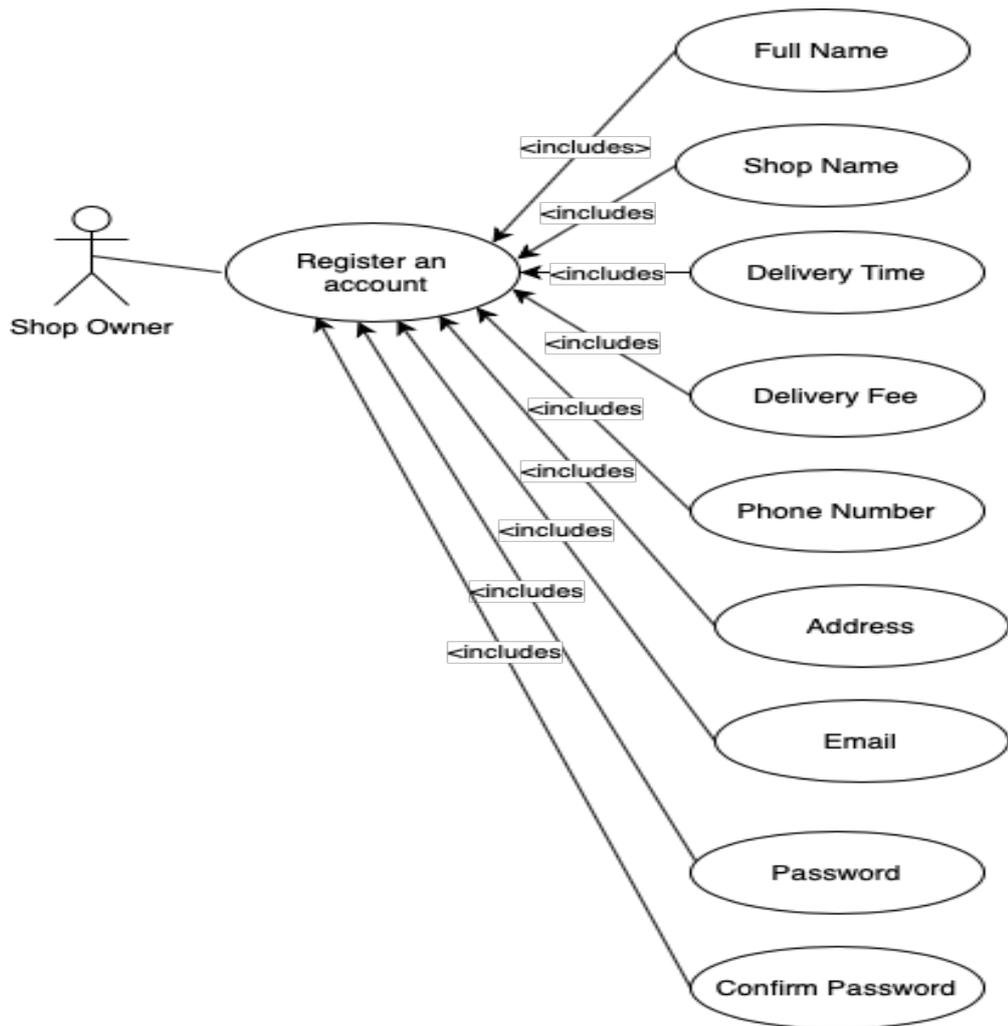
**List further functional requirements here, using the same structure as for Requirement1.**

2.1.1.5. Requirement2<Shop Owner Registration>

2.1.1.6. Description & Priority

The shop owners need to register into this account before they start selling their items. In this application, there are two types of registration. The first one is user registration and the second one is shop owner registration. The shop owner registration is very important as the system will only to use the services for those who created an account in the application.

2.1.1.7. Use Case



Use case of shop Owner Account Registration

**Flow Description**

**Precondition**

Only new shop owners s can register. Shop Owners with existing account cannot register

**Activation**

This use case starts when a <shop owner> clicks on the register button

**Main flow**

6. <shop owner> click on the app icon, the app launches
7. App one the login page
8. <shop owner> click on the “register as shop owner” button to create an account
9. <shop owner>enter all relevant information asked in the register form
10. <shop owner> click on register button, then application will connect to database and save all information.

**Alternate flow**

A1 : <Shop owner couldn't complete the required fields>

7. The system will check if the shop owner entered all relevant information
8. The <Shop owner> not input data in certain fields
9. The app will notify by stating, which fields are not complete

**Exceptional flow**

E1 : <Shop Owner already exist>

10. The system will check for the user in the database based on the information provided
11. The shop owner will get notified by the application, stating that the account already exists.
12. The application will take user to registration form again.

**Termination**

The system presents the shop owner page when the shop owner successfully registered account.

**Post condition**

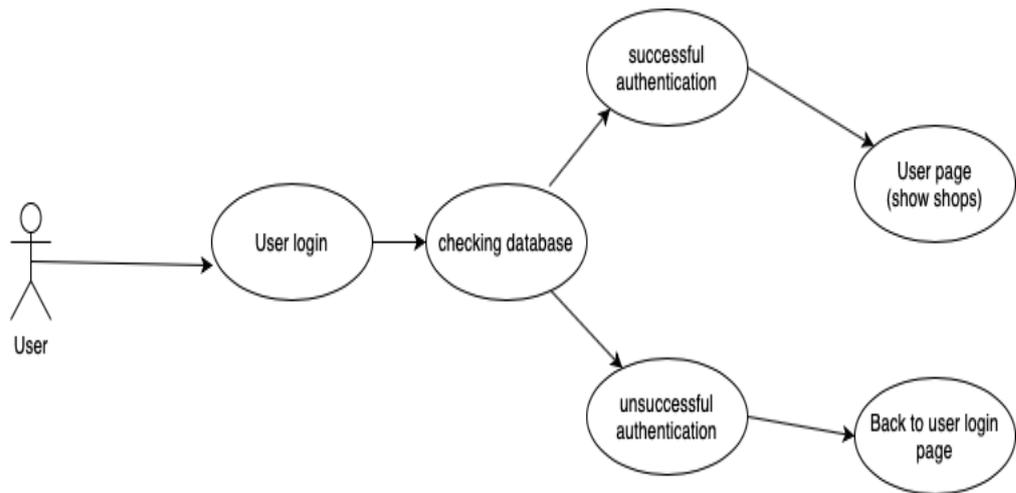
The system goes into a wait state when the database is not connected to the application.

2.1.1.8. Requirement3<User login>

2.1.1.9. Description & priority

Once the user registered an account, the details will be saved on to the firebase database. A user need to login to the system in order to view the list of shops in the user page.

2.1.1.10. Use Case



This is the use case for user sign in

### Flow Description

#### Precondition

Only registered users can sign in

#### Activation

This use case starts when a <User> clicks on the login button

#### Main flow

11. <User> click on the app icon, the app launches
12. App open the login page
13. <User > enter email address and password
14. <User>clicks on login button
15. The application will connect to database and save all information.

#### Alternate flow

A1 : <Invalid email >

13. The system will check if the user entered correct email
14. Connects to database
15. The app will notify by stating, email invalid

#### Exceptional flow

E1 : <Password incorrect >

16. The system will check if the user entered correct password
17. Application connects to database
18. The application will notify the user by stating incorrect password

#### Termination

The system presents the User page when the User successfully logged in to account.

#### Post condition

The system goes into a wait state when the User click on the back button.

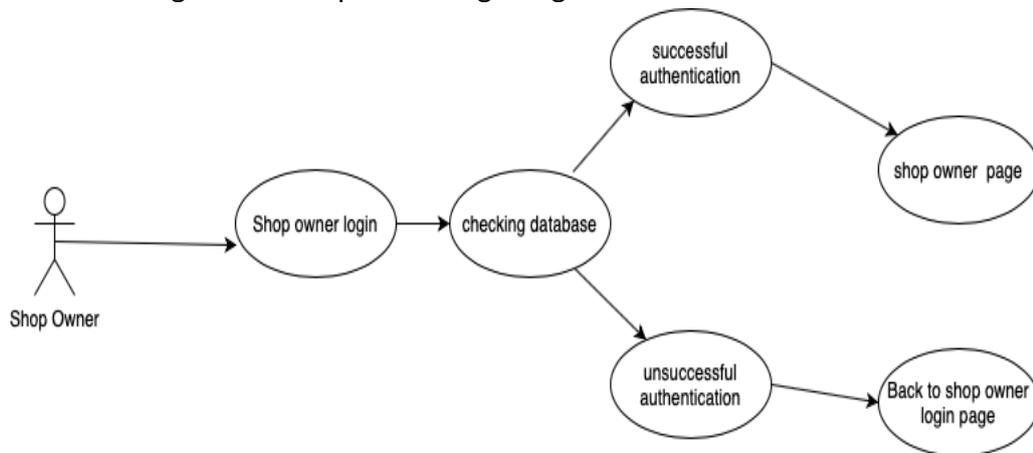
### 2.1.1.11. Requirement4 <Shop Owner Sign in>

### 2.1.1.12. Description & Priority

The shop owner needs to login to the system. When the shop owner enters the email address and password and click on the login button, the system will check if a shop owner with that particular address and password is registered or not. When the database successfully authenticated the shop owner will be able to login in successfully.

### 2.1.1.13. Use Case

Use case diagram for shop owner login is given below:



#### Flow Description

##### Precondition

Only registered shop owners can login in

##### Activation

This use case starts when a <Shop Owner> clicks on the login button

##### Main flow

16. <Shop Owner> click on the app icon, the app launches
17. App open the login page
18. <Shop Owner > enter email address and password
19. <Shop Owner>clicks on login button
20. The application will connect to database and save all information.

##### Alternate flow

A1 : <Invalid email >

19. The system will check if the shop owner entered correct email
20. Connects to database
21. The app will notify by stating, email invalid

##### Exceptional flow

E1 : <Password incorrect >

22. The system will check if the shop owner entered correct password
23. Application connects to database
24. The application will notify the user by stating incorrect password

##### Termination

The system presents the Shop owner page when the User successfully logged in to account.

**Post condition**

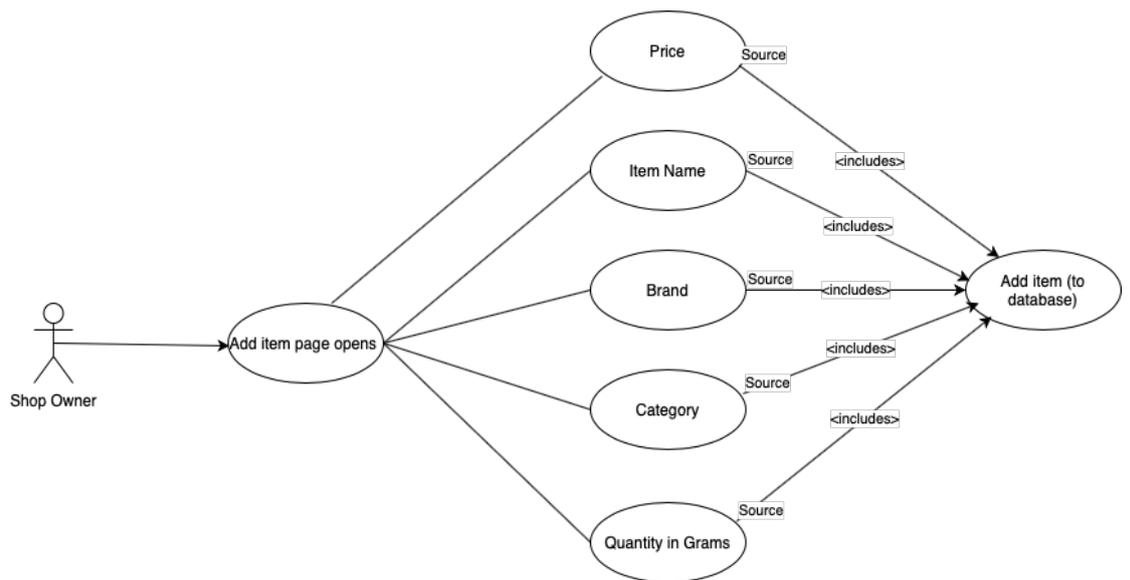
The system goes into a wait state when the Shop owner click on the back button.

2.1.1.14. Requirement5<add item>

2.1.1.15. Description & priority

When the shop owner clicks on the add item icon, add item page opens up. When the shop owner input all the data and click on add item button, the application add all the information provided by the shop owner to the database and save the item as a new product. It is very important that the shop owner provide relevant information about the item.

2.1.1.16. Use Case



**Flow Description**

**Precondition**

Only Signed in shop owners can use this feature

**Activation**

This use case starts when a <Shop Owner> clicks on the add item button

**Main flow**

21. <Shop Owner> sign in to the application
22. Shop owner page opens
23. <Shop Owner > click on the “+” icon
24. <Shop Owner>enter all relevant data asked in the form
25. <Shop Owner> clicks on add item button

26. The application connects to database and save the item as new product

**Alternate flow**

A1 : < Shop owner couldn't complete the required fields >

25. The System will check whether the <shop Owner> enter all relevant fields

26. The <Shop Owner > didn't put data in certain fields

27. The app will notify by stating, the fields are empty

**Termination**

The system clears the add item form, so that the shop owner can add more item without going back.

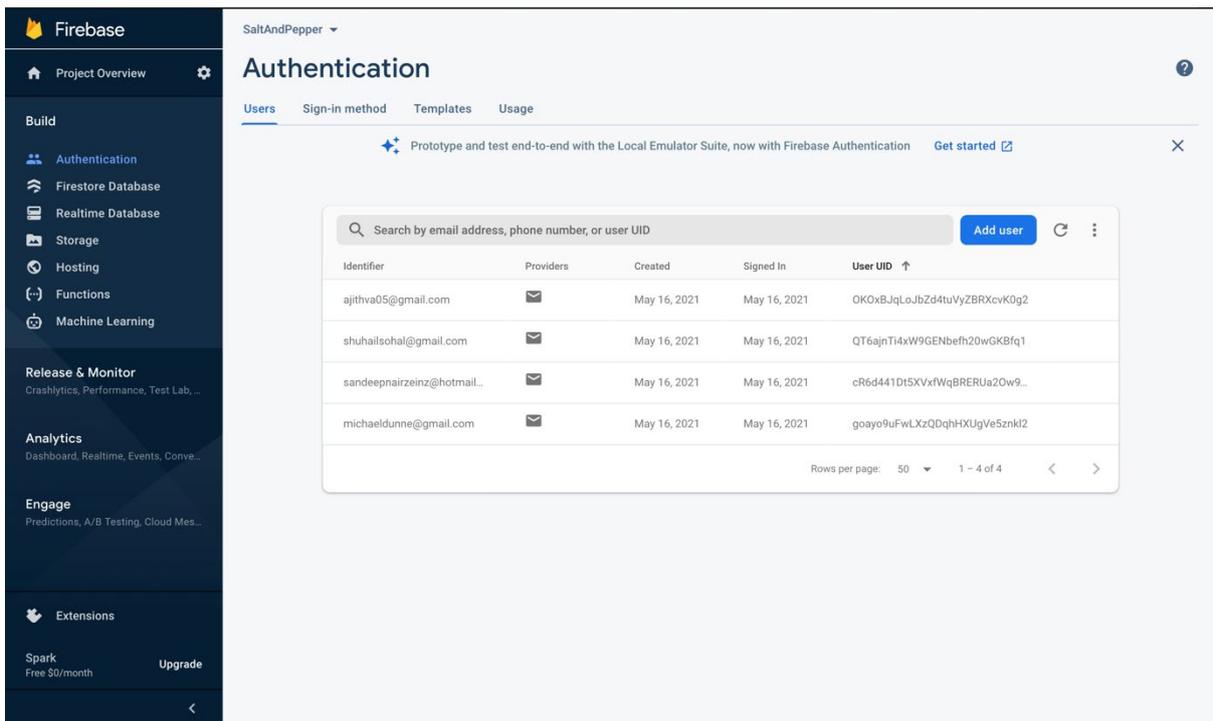
**Post condition**

The system goes into shop owner page when the Shop owner click on the back button.

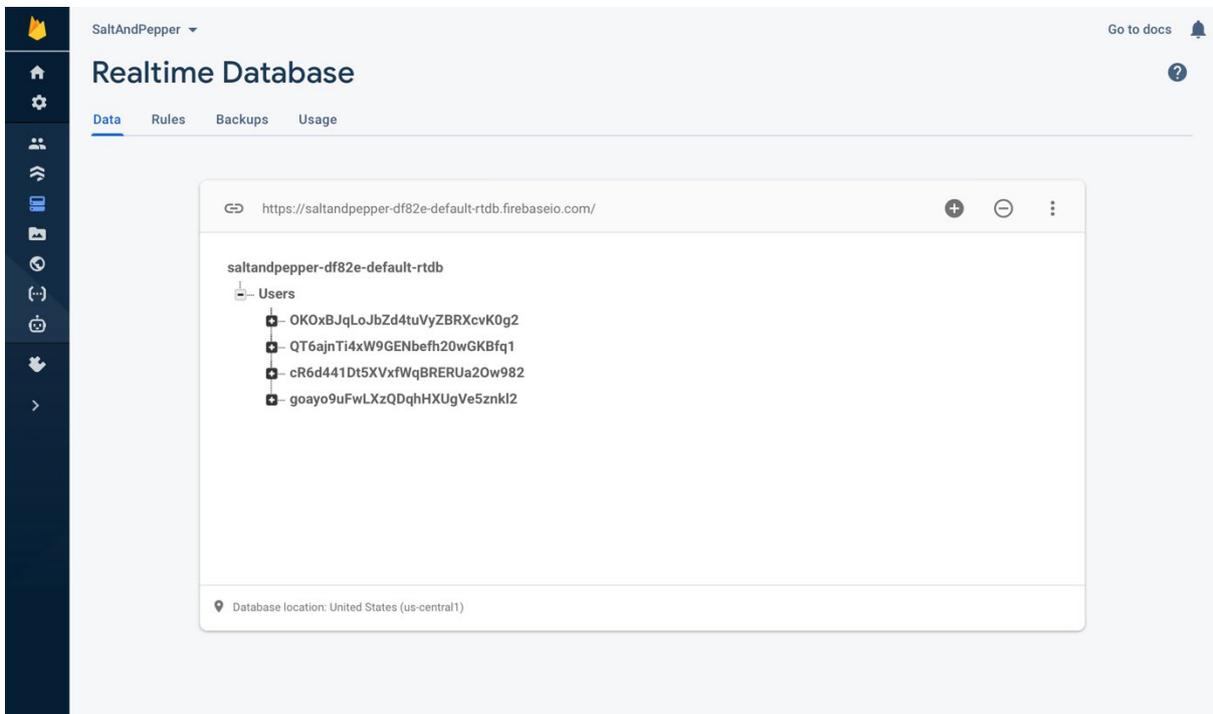
2.1.2. Data Requirements

The application uses Firebase real time database for storing data received from the application. The firebase database will save all the information provided by the user/shop owner. The database in this application will work both as a database and API. When a shop owner register an account, the application will create an account for the shop owner and save it on to the database. The shop owner name, the name of the shop, delivery fee, delivery time location of the shop, the password for the account. The application will receive all this information from the shop owner from the registration activity and save all this information to the database with an account Type classified. The account can be registered in to two account type. They are "User " and "Owner". When a shop owner register his/her account , the system classify he/she as shop owner, since he/she uses shop owner registration activity for registering. During the time of saving the data in to database, this account Type information is also passed. The same rule works in background when he/she register account as user account. Once successful login of shop owner, the shop owner will be able to add items under his account. This information about the items/products will be saved on to the database under "Items section". Shop owner will be asked to provide the item name, item brand name, item category, item price. There is feature in this activity called "offer available" toggle switch. Shop owner can toggle this switch from offer not available to offer available. When offer available toggle switch is turned on, it will display the offer price and the percentage off. Once shop owner input all these relevant information (with photo/ without photo of the item), the add item button will save the photo (if available) in to firebase cloud storage and other data real time database. These data are then combined together and form a new data. The new data formed will be provided to the user at the same time which means that the firebase will receive data from shop owner and at the same time it will convert those received data to new data and display it to the user in real time.

**Please Note: The evidence of how database works in this application are given below as screenshots:**

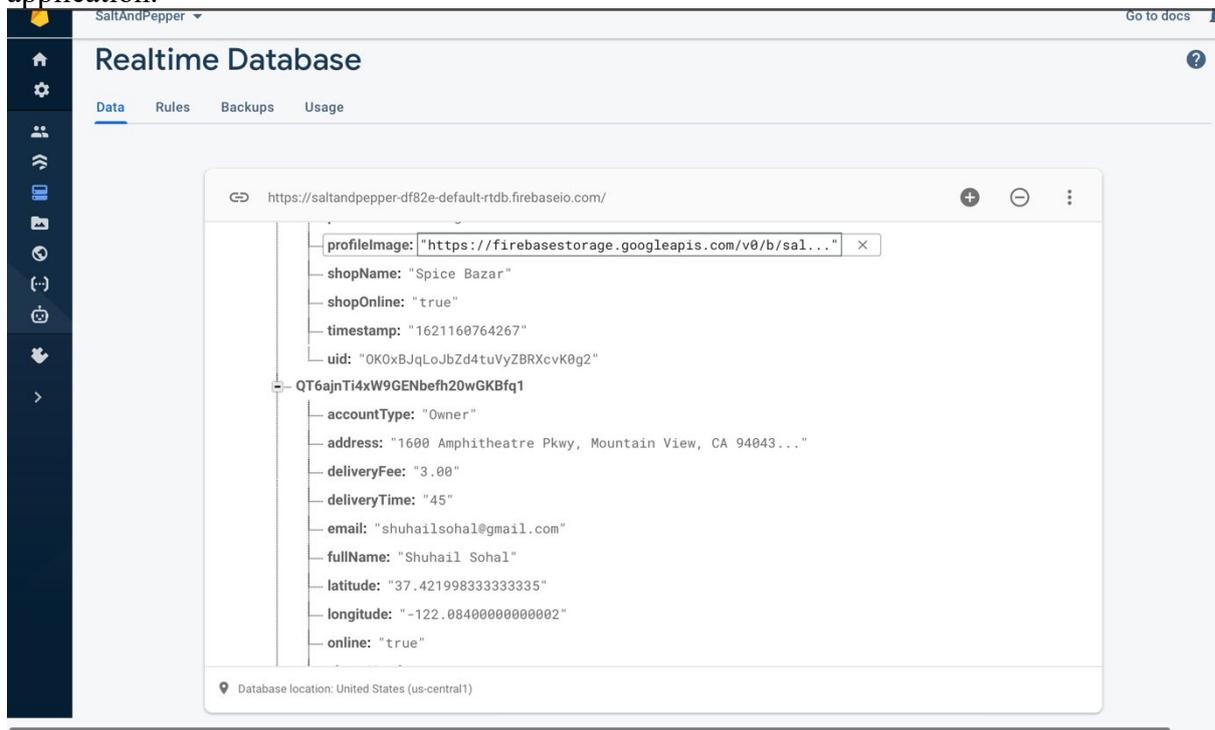


This is the screenshot of firebase authentication. Provided above is the screenshot of the application firebase authentication. By analysing the screenshot it is evident that the application uses firebase authentication for login/sign up feature.



This is screenshot of real time database connected to this application

The above screenshot of real time database proves that the Users details are stored in the database. It is evident from the picture that 4 users are currently using registered in to this application.



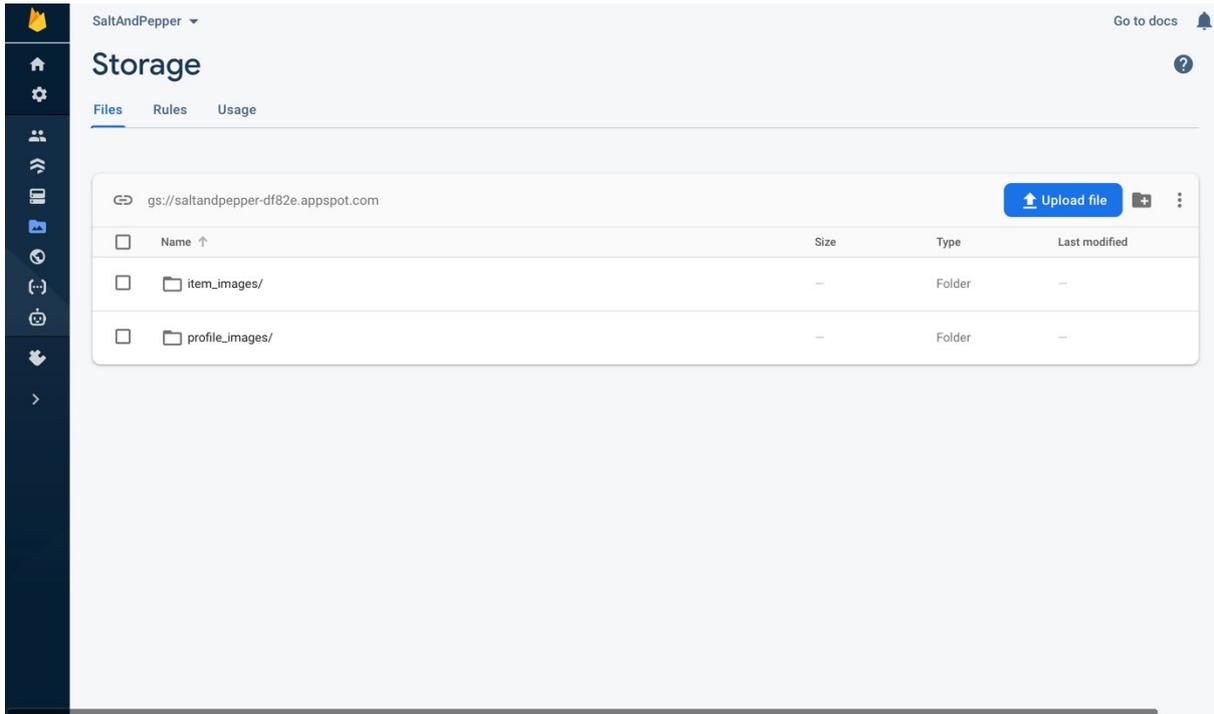
This is the screenshot of account Type information saved on to database

The above screenshot shows that, the a user named “Shuhail Sohal” is currently registered as shop Owner. The “account Type” indicates that Shuhail Sohal is the shop Owner. This is how the database figures out when an account is Owner’s account or not.



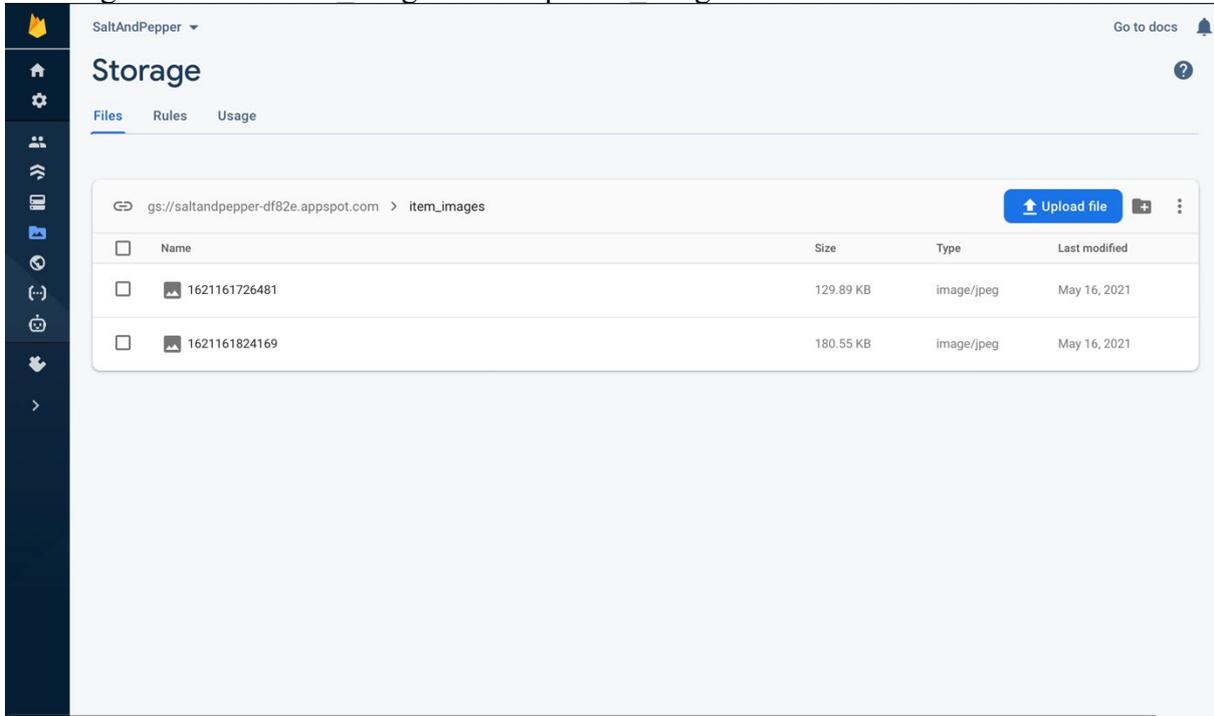
The above screenshot provide shows user named “Sandeep Zan” is registered is as “user”.

In short the firebase used in this application has the capability to figure out whether an account is user or shop owner at the time of registration.

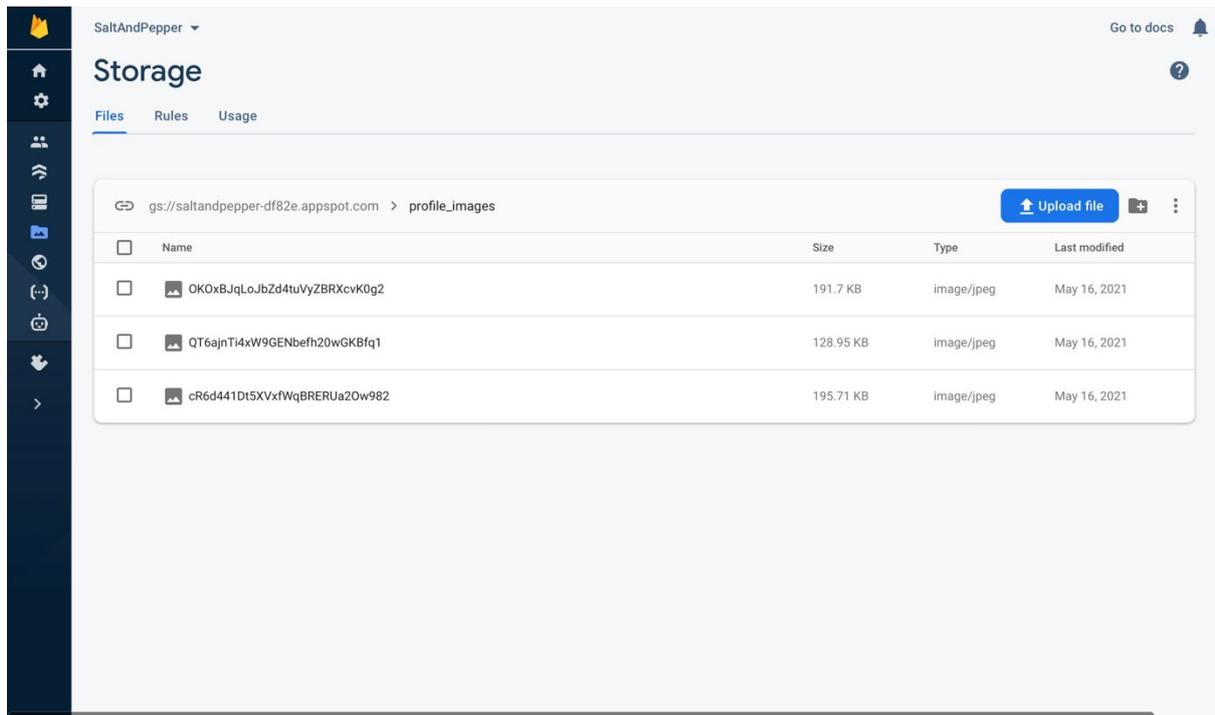


This is the Screenshot of firebase Storage.

From the above screenshot is evident that the images are saved into the firebase storage. Two paths were created in order to save the images. They are “item\_images/” and “profile\_images/”. The path item\_images/” is used to save the images of the products added by the shop owner. All images of products are saved on to this path. The “profile\_images/” is used to save the profile images of both users and Shop owners. The below screenshots show the images saved in “item images/” and “profile images/”.



Images saved in “item\_images/” path



Images saved in “profile\_images/”

### 2.1.3. User Requirements

A user requirement survey was carried out using google forms. 17 users were participated in this survey. The question that were asked in the survey are given below:

A screenshot of a Google Form question. The question is 'what do you expect in Asian Grocery App \*'. There are four multiple-choice options, each with an unchecked checkbox:

- Easy ordering system
- Hassle free sign In/sign Up
- Offers
- All these



What feature would you like to have in an Asian Grocery App? \*

- Contact shop owner directly such as calling or mailing feature
- Fast and efficient application
- inbuilt Voice Assistant
- All of these
- None of these

What type of login feature you like to have in this application \*

- Email login
- social media login

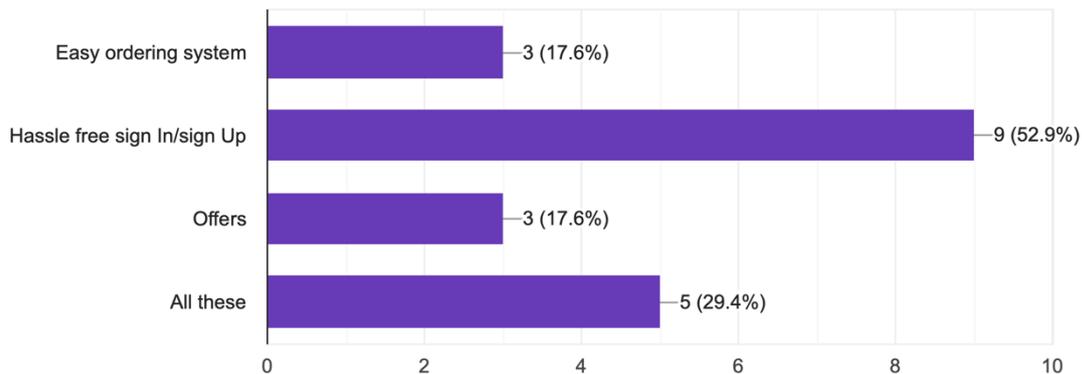
Will you use an application that provides all these feature, if it is available?

- Yes
- No
- Maybe

**The response for these questions are given below:**

### what do you expect in Asian Grocery App

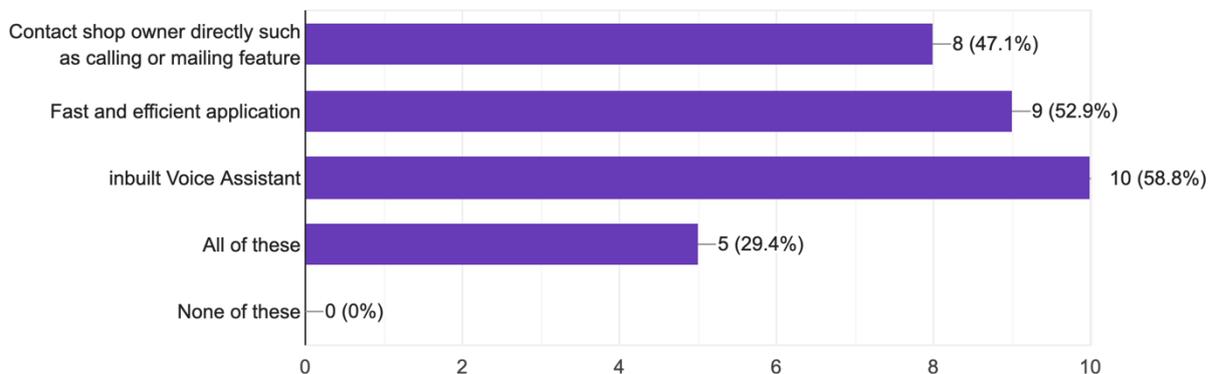
17 responses



The question “What do you expect in Asian Grocery App” was answered by 17 users. By analysing this graph we can see that around 3 people want the system to have an easy ordering feature. 9 people want the system to have hassle free sign in/ sign up, 3 people want the system to provide good offers and out of these around 5 people want this system to have all these features in the application.

### What feature would you like to have in an Asian Grocery App?

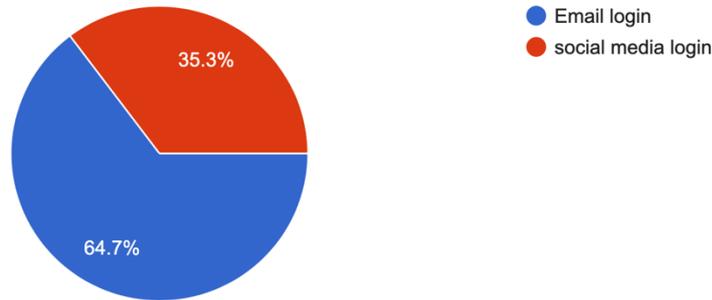
17 responses



The next question was “What feature would you like to have in Asian grocery app”. This graph shows that around 58% of users want to have an inbuilt voice assistant app. This shows that people are more interested in latest technologies. Around 29.4% want to have all these features in this application.

### What type of login feature you like to have in this application

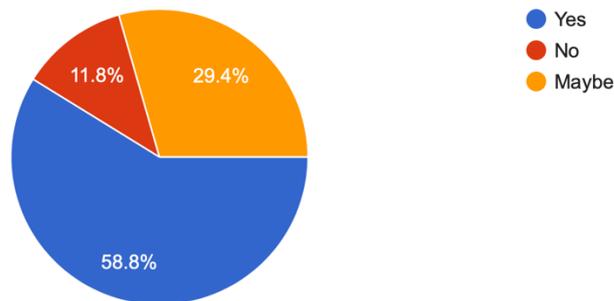
17 responses



This pie diagram shows us what percent of people are interested in email login feature and social media login feature. Out of 17 responses 64.7 % wants the system to have email login feature and only 35.3% of the overall responses want social media login feature. This means that people are giving value for privacy as the users don't want to use their social media account linked to the application.

### Will you use an application that provides all these feature, if it is available?

17 responses



This pie gives a positive feedback from people. The response from these users for this question shows the business opportunities that this application has. 58.8 percent of responses marked as they will use this application. And around 29.4 percent opted they might use this application. And only 11.8% said they will not use this application.

Considering all these requirements from the user side. The application justify user needs as this system have email login feature, fast and efficient ordering system, and inbuilt voice assistant and move over it is very easy to use.

#### 2.1.4. Environmental Requirements

The environmental requirements for this application are given below:

**Android Studio:** In order to build this application Android studio is vital. Android studio is the IDE used for developing this application as it is the official IDE for creating android based applications and this application is programmed in java. Android Studio supports java and its many independent libraries

**Pc/Laptop:** For the smooth development of the application, it is very important that android studio runs efficiently on pc/laptops. For the smooth running of android studio, a high spec pc/laptop is essential.

**Internet:** The devices should have access to internet while developing the application as during development , internet access is. Essential for connecting to database, accessing database, adding SDKs and so on.

#### 2.1.5. Usability Requirements

If a user want to use this application, it is essential that the user needs to have an account registered within this app. User can use the Sign in feature to sign in to the system if the user is already registered. If the user want to create a new account, user can do so by clicking on register account. A test was carried out to find out how long it will take for the users to register in to this account. The result that were found after the testing was absolutely wondering as it only takes less than 1 minute to register a user account and around 1.5 minutes to register shop owner account. This shows that application has hassle free registration form

#### 2.2. Design & Architecture

Salt and Pepper is an smart innovative AI voice assistant enabled android based application. The programming used for creating this application is Java. This application was created using Android Studio IDE. The user needs to download this application to their android phone. After the successful installation, when the user clicks on the app icon that is displayed in the app menu of the phone, the application will launch with a Splash Screen. The splash screen shows the app icon and its motto in an animated view which provide great user experience for the user. The splash screen will take the user to the login page. On this activity he/she can login if they have an account already registered within this application. If he/she is a new user then an account needs to be created. When he/she clicks on the register account button, it will take the user to user sign in page. If he/she wants to buy Asian groceries, then they need to register as a “User”. If he/she want to sell their items then they can do so by creating an account in “register Shop Owner”. Let’s consider he/she is a shop owner. Then he/she should select register shop Owner button that is displayed on register user activity. This will take to the register shop owner page. He/she is required to provide the necessary information asked by the application and click on register button. The application then connects with the firebase real time database that is attached with this application and takes the input and save it to the database with account Type provided as “Owner”. Once the account created he/she will be considered as the shop owner. The application will then take the shop owner to the shop owner page. On this page, there will an app icon displayed on the left top corner and to the to the right corner there will be a log out button. The name and the

shop name of the user will be retrieved from the database and store it on to middle of the toolbar of the activity. There will be a button “+” displayed on right top of the activity laying left to the log out button. This shop owner activity will display all the items/products added by the shop owner using the “+” button. Shop owner will be able to see the voice assistant connected to this application. Shop owner can use voice commands to move from one activity to another activity, search for a particular product that was added to the database and even the shop owner can ask general question to voice assistant which help to increase the shop owner’s interaction with the application. If a shop owner wants to add an item that needs to sell, all he/she want to so is either click on the add button, or use command “add item”. If the shop owner clicks on “+” it will take the owner to a new activity where items can be added. If shop owner use voice command, the voice assistant will search for the command provided by the shop owner with the script provided in Voice assistant server, if the voice command match the script, the voice assistant will open the add item activity followed by a voice stating opening add item page. The voice assistant will also provide information about date and time, about the application and so on with voice commands. Once the user fill the item details in the add item activity the application will save the item image on to firebase storage and the item details on to the database. Once the item is added to the database the application will automatically clear all the fields and get ready to save another item. This will help the shop owner save lot of time. When the items are added, the shop owner activity will display all items that are added under items tabs. The orders tab in the shop owner activity will display all orders that were received from the buyers. If shop owner wants to delete item form the database or if they don’t have that item in the stock, there is an option to delete the item. If the item got deleted, the buyers will no longer be able to see that product.

If he/she wants to register an account as user, all they need to do is to go the user registration page. Once the user provided the user details the system will recognise the user as “User” and save it on to the database. When the accounted is successfully created, the application will take the user to user page, where the shops registered by shop owners will be displayed. As soon as the user login to the application, the voice assistant feature will be available to the user as well. Voice assistant will help the user to select shops, choose items, open activities, direct from one activity to another, add items to the cart. User can do all these by touching on the screen as well. User will be able to see the shops that are online. A green icon will be displayed on all shops that are online. This will help the user to select the shops that are online in much faster way. If a user click on any shop that is displayed, it will take the user to another activity which shows all the items they are selling by the shop. There will be search and filter option available so that the user will be able to get the item faster. User can also use voice assistant for searching products. When the user click in on the item a new dialog box appears on which the user can add the item to the cart. Once the items are selected and moved on to the cart, user will be able to place the order. The order that placed by the user will be received to the corresponding shop. Here the database plays an important role, in this application as it update data in real time and also function as an API for this application. The voice assistant used in application is coded in such a way that the both users and shop owners will be able to use this voice assistant to control the whole application just via voice commands.

## 2.3. Implementation

### Classes used in this application:

- Splash Screen Activity**
- Login Page Activity**
- Register User Activity**
- Register Shop Owner Activity**
- Forgot Password Activity**
- User Page Activity**
- Shop Owner Page Activity**
- Add Item Activity**
- Shop Views Activity**
- Category Constants**
- Item Model**
- Shop Model**
- Items Adapter**
- Item User Adapter**
- Shops Adapter**
- Refine Item Filter**
- Refine Item User Side Filter**

Now let's describe some important classes/codes used in this application:

### Register User Activity:

```
private void setData() {
    //input the data
    fullName = nameEditText.getText().toString().trim();
    phoneNumber = passwordEditText.getText().toString().trim();
    address = addressEditText.getText().toString().trim();
    email = emailEditText.getText().toString().trim();
    password = passwordEditText.getText().toString().trim();
    confirmPassword = cPasswordEditText.getText().toString().trim();

    //now checking the data
    if (TextUtils.isEmpty(fullName)) {
        Toast.makeText(this, "enter your name", Toast.LENGTH_SHORT).show();
        return;
    }
    if (TextUtils.isEmpty(phoneNumber)) {
        Toast.makeText(this, "enter phone Number", Toast.LENGTH_SHORT).show();
        return;
    }
    if (latitude==0 || longitude==0) {
        Toast.makeText(this, "Please turn on Gps", Toast.LENGTH_SHORT).show();
        return;
    }
    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        Toast.makeText(this, "Please check your email", Toast.LENGTH_SHORT).show();
    }
}
```

```

return;
}

if (password.length() < 8) {
    Toast.makeText(this, "your password is less than 8 characters", Toast.LENGTH_SHORT).show();
    return;
}

if (!password.equals(confirmPassword)) {
    Toast.makeText(this, "password doesn't match", Toast.LENGTH_SHORT).show();
    return;
}

setUpAccount();
}

```

This function called setData() will get the data from the input made by the user and assign it to fullName, phoneNumber, address, email, password and confirmPassword. This function will also check if the user haven't entered any relevant field. If the user forgot to add information in any of the fields, this function will let the user know about the field that the user missed.

```

private void saveData() {
    progressDialog.setMessage("Please wait while we save your information onto database...");
    String timestamp = "" + System.currentTimeMillis();
    if (image_uri == null) {
        //save the information without image

        //now setting up the data to be saved
        HashMap<String, Object> hashMap = new HashMap<>();
        hashMap.put("uid", "" + firebaseAuth.getUid());
        hashMap.put("fullName", "" + fullName);
        hashMap.put("phoneNumber", "" + phoneNumber);
        hashMap.put("address", "" + address);
        hashMap.put("email", "" + email);
        hashMap.put("latitude", "" + latitude);
        hashMap.put("longitude", "" + longitude);
        hashMap.put("timestamp", "" + timestamp);
        hashMap.put("accountType", "User");
        hashMap.put("online", "true");
        hashMap.put("profileImage", "");

        //save these data to database
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference("Users");
        reference.child(firebaseAuth.getUid()).setValue(hashMap).addOnSuccessListener(new
    OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            //updated database
            progressDialog.dismiss();
            startActivity(new Intent(RegisterUser.this, UserPage.class));
            finish();
        }
    }).addOnFailureListener(new OnFailureListener() {

```

```

@Override
public void onFailure(@NonNull Exception e) {
    //update failed
    progressDialog.dismiss();
    startActivity(new Intent(RegisterUser.this, UserPage.class));
    finish();
}
});

}
else{
    //save with image
    //setting the name and path of image
    String filepath = "profile_images/" + ""+firebaseAuth.getUid();
    //uploading image
    StorageReference storageReference = FirebaseStorage.getInstance().getReference(filepath);
    storageReference.putFile(image_uri).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
        //getting the url of the image that is uploaded
        Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
        while (!uriTask.isSuccessful());
        Uri imageDownload = uriTask.getResult();

        if (uriTask.isSuccessful()){
            //now setting up the data to be saved
            HashMap<String, Object> hashMap = new HashMap<>();
            hashMap.put("uid", "" + firebaseAuth.getUid());
            hashMap.put("fullName", "" + fullName);
            hashMap.put("phoneNumber", "" + phoneNumber);
            hashMap.put("address", "" + address);
            hashMap.put("email", "" + email);
            hashMap.put("latitude", "" + latitude);
            hashMap.put("longitude", "" + longitude);
            hashMap.put("timestamp", "" + timestamp);
            hashMap.put("accountType", "User");
            hashMap.put("online", "true");
            hashMap.put("profileImage", "" + imageDownload);

            //save these data to database
            DatabaseReference reference = FirebaseDatabase.getInstance().getReference("Users");
            reference.child(firebaseAuth.getUid()).setValue(hashMap).addOnSuccessListener(new
OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    //updated database
                    progressDialog.dismiss();
                    startActivity(new Intent(RegisterUser.this, UserPage.class));
                    finish();
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    //update failed
                    progressDialog.dismiss();
                    startActivity(new Intent(RegisterUser.this, UserPage.class));

```



```

1 package com.example.saltandpepper;
2
3 public class CategoryCon {
4     //This will display the categories that needs to be related to each product
5
6     public static final String[] lists = {
7         "Rice/Flour",
8         "Bread",
9         "Confectionery",
10        "Drinks",
11        "Frozen Food",
12        "Noodles",
13        "Spices",
14        "Vegetables",
15        "Sauces",
16        "Others"
17    };
18
19
20
21    public static final String[] lists1 = {
22        "All",
23        "Rice/Flour",
24        "Bread",
25        "Confectionery",
26        "Drinks",
27        "Frozen Food",
28        "Noodles",
29        "Spices",
30        "Vegetables",
31        "Sauces",
32        "Others"
33    };
34
35 }
36
37

```

The category const class is used to categorise the items added by the shop owner. This class is also used to filter the items for both shop owner and users. This application allows filter by category with the help of refine Items filter class and the category const class.

### Item Model:

Item Model class are created to save and manage the data that are going to present in the UI. This class contains the constructor, getters and setters.

### Shop Model:

The shop model class are created to store the details of the data of the shop that are going to present in the UI. Same as like Item Model class this class consists of the constructors, getters and setters.

### Adapter Class:

Adapter classes function as a bridge between the UI component and data. Three adapters are created in this project. They are:

#### a) Items Adapter

This adapter is used to show the data of items in the layout file that is created to show the items added by the shop owner to the database. When the users go back to user page after adding the item to the database this adapter invokes.

```

@Override
public itemsHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

```

```
//this will inflate the layout
View view = LayoutInflater.from(context).inflate(R.layout.item_owner_view, parent, false);
return new itemsHolder(view);
}
```

This code is used to inflate the data to the UI. Here “item\_owner\_view” is recycler view layout. This will indulge the data to this view and show it to the shop owner in the shop owner page. The holder class will inflate the layout

#### b) Shops Adapter

This adapter is used to show the details of shops that needs to be represented in a layout file (UI). This shops adapter is used to list the shops in the users page. When the users login to the system, the shops adapter invokes and display the shops data in a UI layout. Shop adapter uses shop model to show the data.

```
@Override
public ShopHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    //inflating the layout
    View view = LayoutInflater.from(context).inflate(R.layout.shop_user_view, parent, false);
    return new ShopHolder(view);
}
```

This code is used to inflate the layout “shop\_user\_view”. “shop\_user\_view” is a recycler view that is used to show the list of shops with necessary details. The shopHolder class will inflate the shop\_user\_view layout.

#### c) Item User Adapter

Item User adapter is used to show the items to the user when the user clicks on any shop listed in the user page. When a user click on any shop, the items related to that shop will be displayed. Item User Adapter uses Item model.

```
@NonNull
@Override
public HolderItemsUser onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    //inflate the views
    View view = LayoutInflater.from(context).inflate(R.layout.item_user_view, parent, false);
    return new HolderItemsUser(view);
}
```

This code is used to inflate the layout called “item\_user\_view”. The holder class called HolderItemUser will inflate the “item\_user\_view”. Layout.

### Refine Item Filter

The refine item filter is created in order to enable filtering in the shop owner items page. The Refine Items Filter uses the items Adapter and Item model array list.

```
@Override
protected FilterResults performFiltering(CharSequence constraint) {
    FilterResults results = new FilterResults();
    //checking the data
    if (constraint !=null && constraint.length()>0){
        //if the user search something on the search button perform this function
        //this will change the data to uppcase to make case sensitive
        constraint = constraint.toString().toUpperCase();

        //save the refined items
        ArrayList <ItemModel> refinedList = new ArrayList<>();
        for (int i =0; i<refineItems.size();i++){
            //this will check the data by item name, item category and item brand
            if (refineItems.get(i).getItemName().toUpperCase().contains(constraint)||refineItems.get(i).getItemCategory().t
                ||refineItems.get(i).getItemBrand().toUpperCase().contains(constraint)){

                //this will add the refined data to the list
                refinedList.add(refineItems.get(i));
            }
        }
        results.count = refinedList.size();
        results.values = refinedList;
    }
    else{
        //if the user is not searching anything, show the items on the array list
        results.count = refineItems.size();
        results.values = refineItems;
    }
    return results;
}

@Override
protected void publishResults(CharSequence constraint, FilterResults results) {

    itemsAdapter.itemList = (ArrayList<ItemModel>)results.values;
}
//refresh adapter
```

This filter class will help the shop owner to search for the items they were added by typing the item name, item brand. This filter class will also help the shop owner to search the item by category. This filter class will change the data to uppcase to make it case sensitive.

### Refine Item User Side Filter

This filter class is used to filter the items that can be viewed from the user side. This filter class will help the user to search for the products/items they are searching. Users will be able to search the item by item name, item brand and even by category they belongs to. The same code technique that is used to create the Refine Item Filter class is implemented in the refine Item User Side Filter.

### Shop Views Activity:

```

private void showShopsInfo() {
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users");
    reference.child(shopId).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            //this will get the shop details that needs to be displayed on the users shop details activity
            String profileImage = ""+snapshot.child("profileImage").getValue();
            String name = ""+snapshot.child("fullName").getValue();
            String deliveryFee = ""+snapshot.child("deliveryFee").getValue();
            shopName = ""+snapshot.child("shopName").getValue();
            shopEmailAddress = ""+snapshot.child("email").getValue();
            shopFullAddress = ""+snapshot.child("address").getValue();

            //setting the data
            shopNameEditText.setText(shopName);
            emailEditText.setText(shopEmailAddress);
            addressEditText.setText(shopFullAddress);
            deliveryFeeEditText.setText("Delivery Fee: €"+deliveryFee);

            try{
                //loading shop image
                Picasso.get().load(profileImage).into(shopImageView);
            }catch (Exception e){

            }

        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }

    });
}
}

```

The showShopsInfo() will connect with the database and get the shop details and the set the shop details to the ids of the layout file. Picasso is used to load the image from the database.

### Alan AI:

Alan AI is the Voice assistant that is used to implement on this application.

```
implementation 'app.alan:sdk:4.7.17'
```

The above shown is the dependency for Alan SDK.

This is the code that have been used in the layout file to show the button

```

//adding voice assistant
<com.alan.alansdk.button.AlanButton
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:id="@+id/alan_button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

```

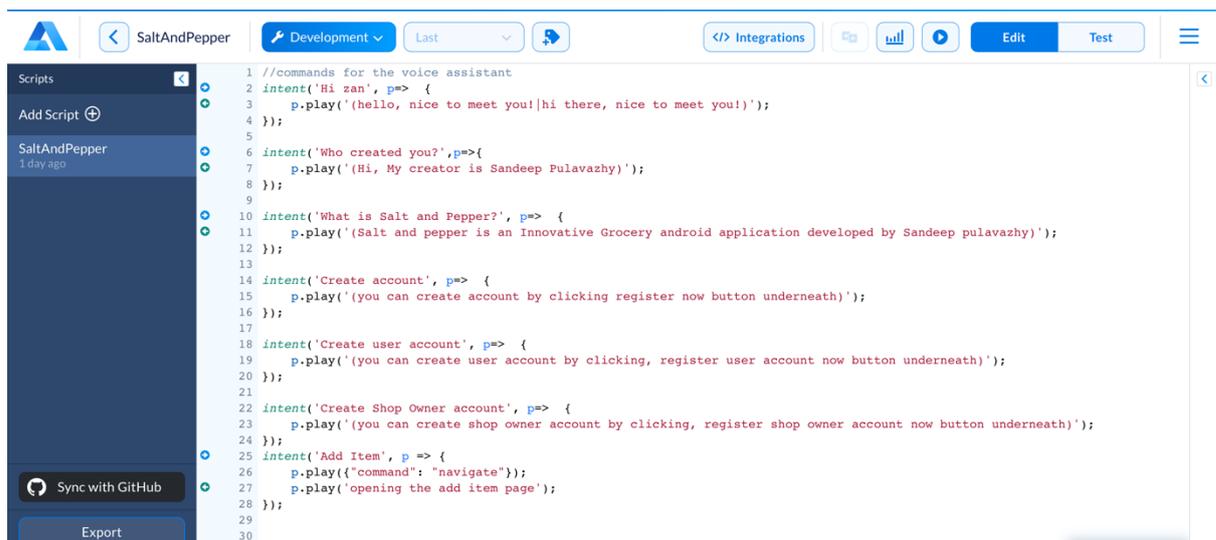
This code id used to initialise the Alan button

```
AlanButton alanButton = findViewById(R.id.alan_button);
```

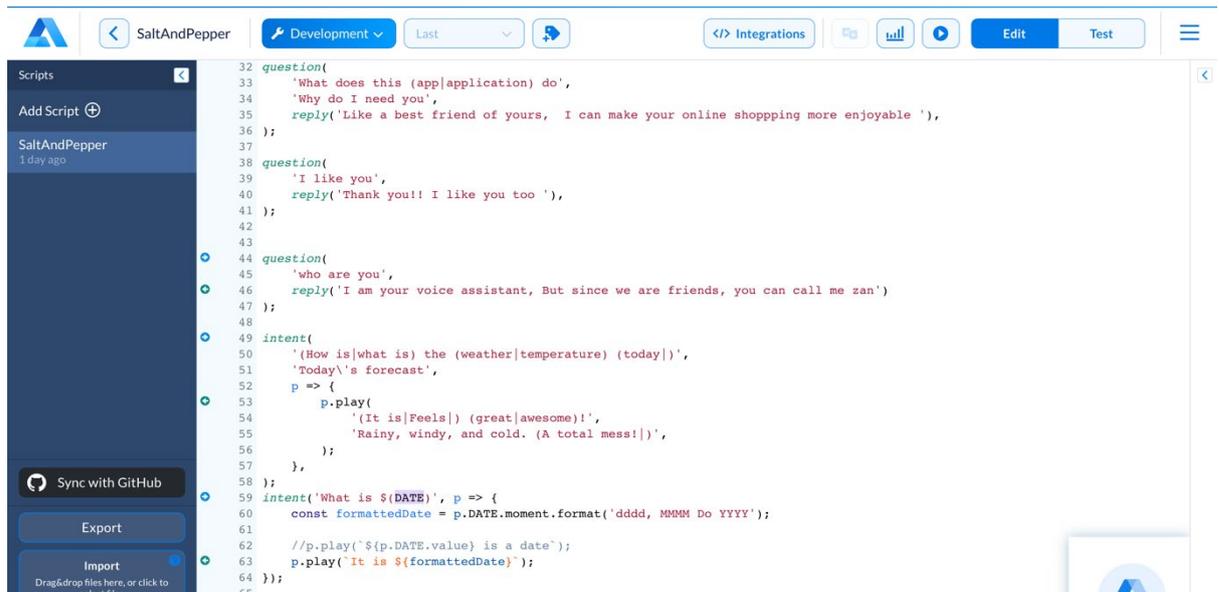
The below is used to config Alan AI to the application. Project id is the project id of Salt and Pepper app created in Alan AI server.

```
AlanConfig alanConfig = AlanConfig.builder()
    //build id of the project SaltAndPepper in Alan AI
    .setProjectId("3e43b24ae6cca427a6d516dc5ffc735e2e956eca572e1d8b807a3e2338fdd0dc/stage")
    .build();
alanButton.initWithConfig(alanConfig);
```

The script for the Alan AI are given below:



```
1 //commands for the voice assistant
2 intent('Hi zan', p=> {
3     p.play('hello, nice to meet you!|hi there, nice to meet you!');
4 });
5
6 intent('Who created you?', p=>{
7     p.play('Hi, My creator is Sandeep Pulavazhy');
8 });
9
10 intent('What is Salt and Pepper?', p=> {
11     p.play('Salt and pepper is an Innovative Grocery android application developed by Sandeep pulavazhy');
12 });
13
14 intent('Create account', p=> {
15     p.play('you can create account by clicking register now button underneath');
16 });
17
18 intent('Create user account', p=> {
19     p.play('you can create user account by clicking, register user account now button underneath');
20 });
21
22 intent('Create Shop Owner account', p=> {
23     p.play('you can create shop owner account by clicking, register shop owner account now button underneath');
24 });
25 intent('Add Item', p => {
26     p.play({"command": "navigate"});
27     p.play('opening the add item page');
28 });
29
30
```



```
32 question(  
33   'What does this (app)application do',  
34   'Why do I need you',  
35   reply('Like a best friend of yours, I can make your online shopping more enjoyable '),  
36 );  
37  
38 question(  
39   'I like you',  
40   reply('Thank you!! I like you too '),  
41 );  
42  
43  
44 question(  
45   'who are you',  
46   reply('I am your voice assistant, But since we are friends, you can call me zan')  
47 );  
48  
49 intent(  
50   '(How is|what is) the (weather|temperature) (today|)',  
51   'Today\'s forecast',  
52   p => {  
53     p.play(  
54       '(It is|Feels|) (great|awesome|)',  
55       'Rainy, windy, and cold. (A total mess!|)',  
56     );  
57   },  
58 );  
59 intent('What is $(DATE)', p => {  
60   const formattedDate = p.DATE.moment.format('dddd, MMMM Do YYYY');  
61  
62   //p.play(`${p.DATE.value} is a date`);  
63   p.play(`It is ${formattedDate}`);  
64 });  
65
```

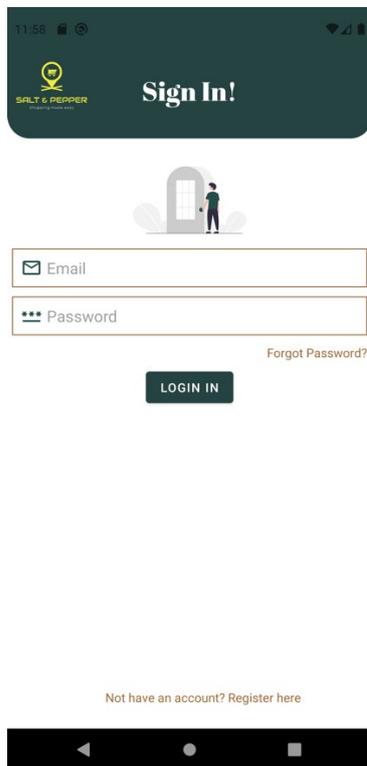
## 2.4. Graphical User Interface (GUI)

### Splash Screen:



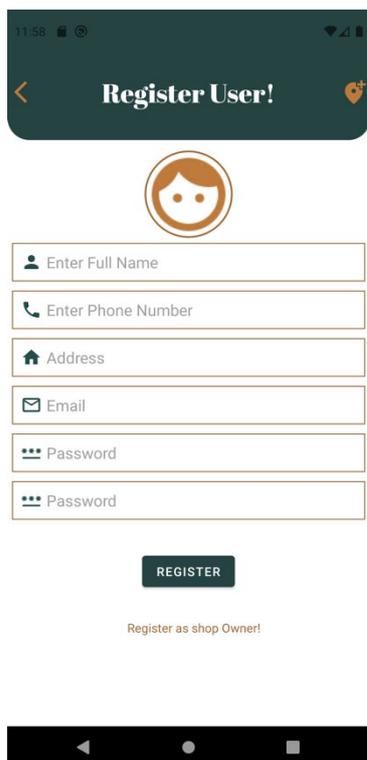
This is the splash Screen of this application. This splash screen is provided with an animated view, which provides good user experience while launching application.

## Login Page Activity:



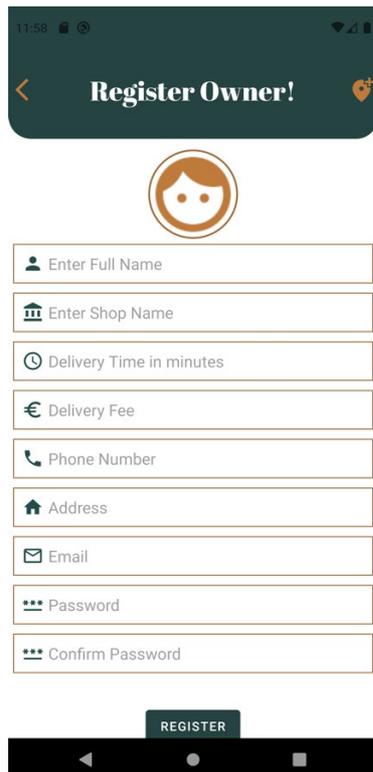
This is the Login page activity. Users will be able to sign in through this activity once they create an account with in this application.

## Register User Activity:



This is the Register User Activity. This activity allows the users to register as “Users” in this application. The users account will be able to see shops and items and buy items. There is a back button which enables the user to go back to login activity and the location button on the top right corner will detect the location automatically. The register as shop owner button will take the user to register as shop owner registration Activity.

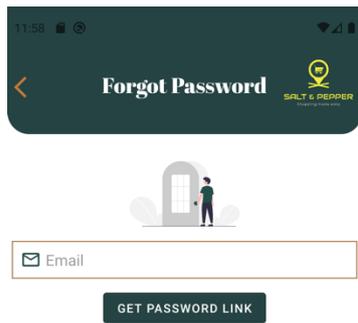
### Register Shop Owner Activity



The screenshot shows a mobile application interface for registering as a shop owner. The title bar is dark green with a white back arrow on the left and a location pin icon on the right. The main title is "Register Owner!". Below the title is a circular icon of a shop owner's face. The form consists of several input fields, each with a small icon on the left: "Enter Full Name" (person icon), "Enter Shop Name" (shop icon), "Delivery Time in minutes" (clock icon), "Delivery Fee" (euro symbol icon), "Phone Number" (phone icon), "Address" (house icon), "Email" (envelope icon), "Password" (password icon), and "Confirm Password" (password icon). At the bottom, there is a dark green button labeled "REGISTER" and a black navigation bar with a white back arrow, a white dot, and a white square.

This Activity helps the user to register as shop owner. The back button the to left corner will help the user to go back to previous activity and the location button will find the user’s current location and display it on the address field.

### Forgot Password Activity



The Forgot Password activity will help the user to get password reset link via email. The back button will take the user back to login page.

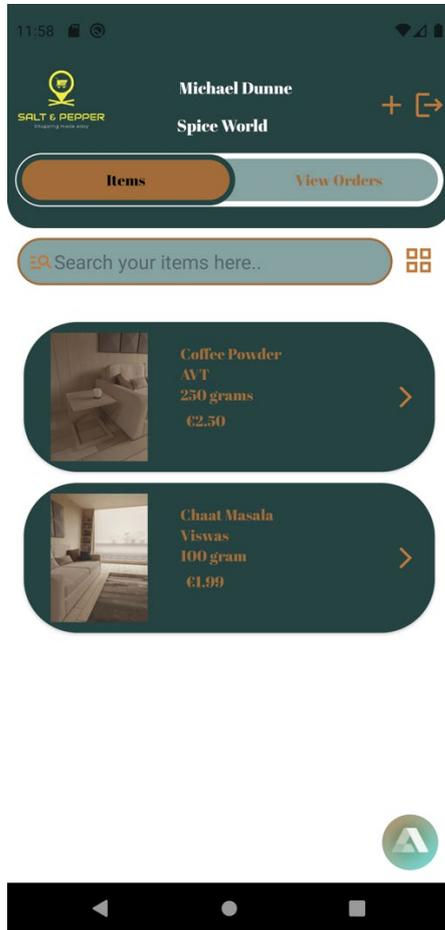
### User Page Activity



This is the User page activity. This page will show the name and the email address of the user. This page has two tabs. They are Shops and View Orders. The shops Tab will show all the shops that are registered with salt and pepper. The name of the shop, the contact email of the shop, the delivery time of the shop and the delivery fee is mentioned on this

view. The logout button will help the user to logout from the application. The voice assistant can be seen on the bottom left corner.

### Shop Owner Page Activity



This page will display the products/ items added by the shop owner. Here we can see that the shop owner name is Michael Dunne and the shop name is Spice World. The items added by shop owner is displayed below. This activity contain two tabs. They are items and View Order. The items tab displays the items and the view Orders tab will show the orders received from the users who ordered online. The “+” button on the top right will enable the users to add the items to the database and eventually display it on to the shop owner page, once successfully uploaded it to the database. The logout button will enable the shop owner to logout from the application. There is a search bar and filter icon button which help the shop owner to search the product by name, brand and category. The Voice assistant is included in this activity.

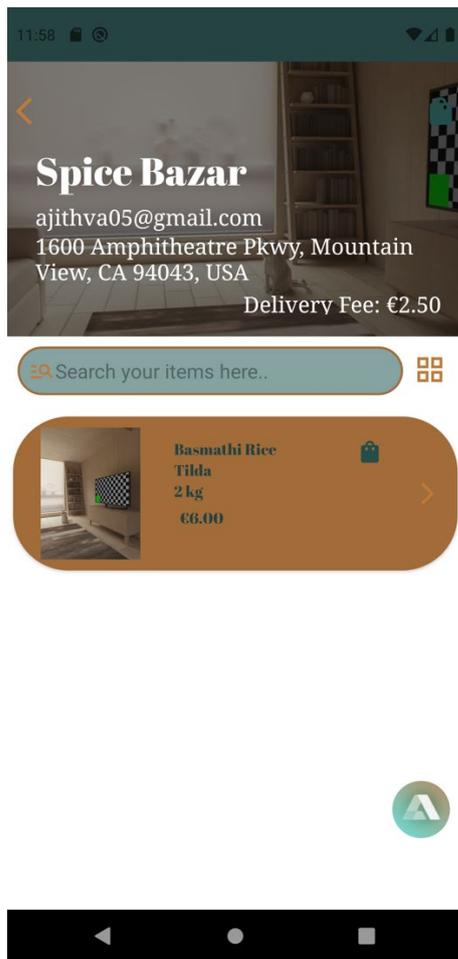


### Item Details Bottom Page:



This is a bottom page that activates, when the shop owner selects any one item. In this bottom page the shop owner will be able to delete the item if it is not in stock or if the item want to be removed by any chance. The delete button on the top right will delete the item and the back button on the top left will take back to the shop owner page.

## Shop Details Activity:



This is the shop details activity. If the user select any of the shops that is displayed on user page, the application will take the user to shop views activity. In this activity the shops details are provided. On the top the Images of the shop is provided, the name of the shop, the name of the shop owner, the contact mail address and the location of the shop is also provided. The delivery fee of the shop is also included. Top left is the back button, if the user click in the back button, the application will take the user to previous activity. The products/items that are available in then shop are provided in this activity. The voice assistant button is also located at the bottom right.

## Testing

The testing was carried out by installing the .apk file on to an android smart phone.

#	Requirement	tested	passed	Comments
1	User able to register	yes	yes	User registration allows the user to create an account
2	Shop Owner able to register	yes	yes	Shop owners can register into the system
3	User able to login	yes	yes	User is able to login, and the system identifies it as a user and divert to user page
4	Shop owner able to login	yes	yes	Shop Owner is able to login and the system identifies it as a shop owner and divert to shop owner page
5	Shop Owner able to add items/products to database	yes	yes	Shop Owner can add item to the database
6	Shop owner pages displays the items that were added into database	yes	yes	Shop owner page is displaying item once the item are successfully uploaded to database
7	Delete item from database	yes	yes	Shop owner is able to delete item, the bottom sheet appears if the shop owner click on the item, and the system is allowing to delete the item form the database.
8	Shop owner able to filter products in search	yes	yes	The shop owner can search items using item name, item brand and category.
9	Users able to see shops list	yes	yes	Users can see the shops that are registered, soon after they log in.
10	Users can choose shops and items	yes	yes	Users can choose the shops listed and it is taking the user to a new activity showing the items listed by that particular shop.
11	Voice Assistant	yes	yes	Voice assistant is working as it is responding to voice commands.

## 3.0 Conclusions

Salt and pepper is my final year project. This is an android based mobile application. The main idea behind this project is that, there are no platforms that link all Asian grocery shop under one roof. The application that I developed will bring all the Asian grocery shop under one roof, so that users those who want to order Asian groceries can use the platform called "Salt and Pepper". This application allows the users to find all Asian shops and find all items and place the order within few clicks. The inbuilt AI voice assistant linked with this application unleash a new user experience for the user. From the research that I carried out, there are no such platform that links all Asian groceries in this country. I would like to call this project as innovative because, there are no ecommerce application, that has voice

assistant which help the user to control the app using voice commands. The database in this application will work as an API. The best part in the technical part is that, the database is automatically updated. When a user register new shop account, that shop will be visible to all buyers straight away. The items added by the shop owner keeps the database updated as this system will allow the shop owner to add and remove item form the database, using their account. This reduce stress level in the development side because the database are real time database. I consider the database used in this application is more efficient as compared to other ecommerce websites. Developing this application helped me to learn more about java programming language.

This application is only available on android platforms. I consider this a major limitation for the app. The project is still on developing stage as I need to pursue more knowledge in developing this application.

#### 4.0 Further Development or Research

I would like to fully develop this project. The next thing that I want make is that the voice assistant should be able to fully control the application. The Voice assistant should be able to deal everything so that those who have vision impairment can also use this application. Those people can use this application with the voice commands. For the further development, I am planning to develop this application for ios platform which will make this application more familiar and thus improve business.

#### 5.0 References

le.oberlo.com. 2021. *How Many People Shop Online in 2021? [Feb 2021 Update]*. [online] Available at: <<https://ie.oberlo.com/statistics/how-many-people-shop-online>> .

Ofni Systems. 2021. *User Requirement Specifications (User Specs, URS) | Ofni Systems*. [online] Available at: <<http://www.ofnisystems.com/services/validation/user-requirement-specifications/>>.

www.javatpoint.com. 2021. *Java Adapter Classes - javatpoint*. [online] Available at: <<https://www.javatpoint.com/java-adapter-classes>>.

Class, H. and Srivastava, V., 2021. *Holder Class in Adapter Class*. [online] Stack Overflow. Available at: <<https://stackoverflow.com/questions/11559664/holder-class-in-adapter-class/11560095>>.

Visual-paradigm.com. 2021. *What is Use Case Diagram?*. [online] Available at: <<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>>.

## 6.0 Appendices

This section should contain information that is supplementary to the main body of the report.

### 6.1. Project Plan

# PROJECT PROPOSAL

## 7.0 Objectives

The Main objective of this project is to create a **Smart Asian grocery App** that helps the user to order Asian grocery products and deliver it to user's doorsteps. Unlike other Grocery apps, this app will have a voice assistant. This voice assistant will help the user to search for a product, move a product to the shopping cart and even help the user to place an order, with the help of this **Smart App**. The user will be able to interact with the voice assistant enabled within the app, and thus shopping online will be more easier and flexible. Even a user who is blind can use my mobile application using this smart voice assistant. Another great technology that is going to implement on this app is Augmented Reality. This Mobile application will have AR enabled so that the users can see the products in a 3d view on the surface they choose. This will give the user a clear idea about the product that they are looking for. The AR feature that use in this application not only provide the 3D view of the product, it will also displays the details of the products such as the price of the product, the calories of the products if available and so on.

The other objectives are:

- ◆ Fast and easy signup/sign in to the application.
- ◆ Fast and high quality UI experience.
- ◆ Hassle free checkout.
- ◆ Notifies customers regarding the latest offers available.
- ◆ Completely free to download.
- ◆ Will be available on play store to download.

## 8.0 Background

There are many Android based mobile applications that are available in the market, especially for grocery shopping. Buymie is one of the top Irish company that delivers groceries from selected stores to the users nationwide. I started to study a lot about Buymie app. Buymie was founded in 2015 and now Buymie has become one of the top Irish start-up company to deliver groceries at door steps. After learning about Buymie, an idea came into my mind on creating a mobile application, that delivers Asian grocery products to the people out here in Ireland. I started to research about the percentage of non-nationals living in Ireland. According to rte news that published on august 20 2020, it is said that the number of non- Irish nationals has increased and now they add up to 12.9% of the total population. I came to realise that there is a good market for such kind of app, that delivers the products at door steps to these people.

The next step was to search for the Asian Grocery shops available in the market. There are many Asian grocery shops in Dublin. Some of them are listed below:

- ◆ Oriental pantry Supermarket
- ◆ Asia Market
- ◆ Ingredients Finglas
- ◆ Kquality Food Asian Grocery Store
- ◆ Spice Bazar
- ◆ Eurasia

Those listed above are the top supermarkets available on the market. There are 100s of big and small Asian supermarkets other than those listed above. The problem that I figured out with these shops are, they don't have a very good or functional websites. Most of these shops don't have a website. The only way to purchase from these shops are to go to these shops in person and buy the products, which is not feasible for all customers. Eurasia is one of the largest Asian supermarket located in Lucan. But when it comes to their website, it is not that

impressive. They have online shopping facilities enable on their website, but it seems like it is not working properly as the it is not displaying the whole products in the shops. Then I made a conclusion that there is a huge scope in this area, as people will be highly impressed if there is a mobile application that is fully functional and that will help the users to order Asian groceries and deliver it to their door steps.

The next step was to confirm that whether there is any mobile applications that is currently available in the market. I came to know about a mobile application called “**kalabasa**”. This application is delivering almost all Asian groceries and deliver it to door steps. This app has a functionality that has a barcode reading functionality. But my application is quite different. My application uses the latest available technologies. My application will have a voice assistant which can be used by the user to search for a product and even just talk to voice assistant for adding that product into the cart. The augmented reality will help the user to see the product on surface and gives complete information about that product through AR.

## 9.0 Technical Approach

The first approach to do carrying out some research that helps to get some fresh project ideas. I started to discuss it with my friends through phone and in person and took their advice on how to develop the idea that I have because I considered my friends as my customers. After discussing with my friends, I came to the idea of creating a Mobile application for delivering Asian groceries that would be helpful for the users which help them save time.

The second approach was how to make these ideas into innovative so that there will be a good scope for this application in the market. I started to research about the mobile application and websites that provide door step delivery of Asian groceries. I came to realise that there are many mobile applications and websites that has the same idea as that of mine. This led me to develop my idea into something more innovative. I planned to implement a voice assistant and thereby helps the users to interact with my app and control the app by voice command. The second innovation that I am bringing into my app is implementing AR. This feature will introduce a new world where a user can view their product and the product details in 3D on the surface they choose.

The third approach is to gather more information such as collect the feedback, carrying out surveys in order to make the most effective and innovative Asian grocery app. In order to do that, I will carry out some surveys within the college students, friends and to the Asian community in Ireland so that I will be able to gather users requirements.

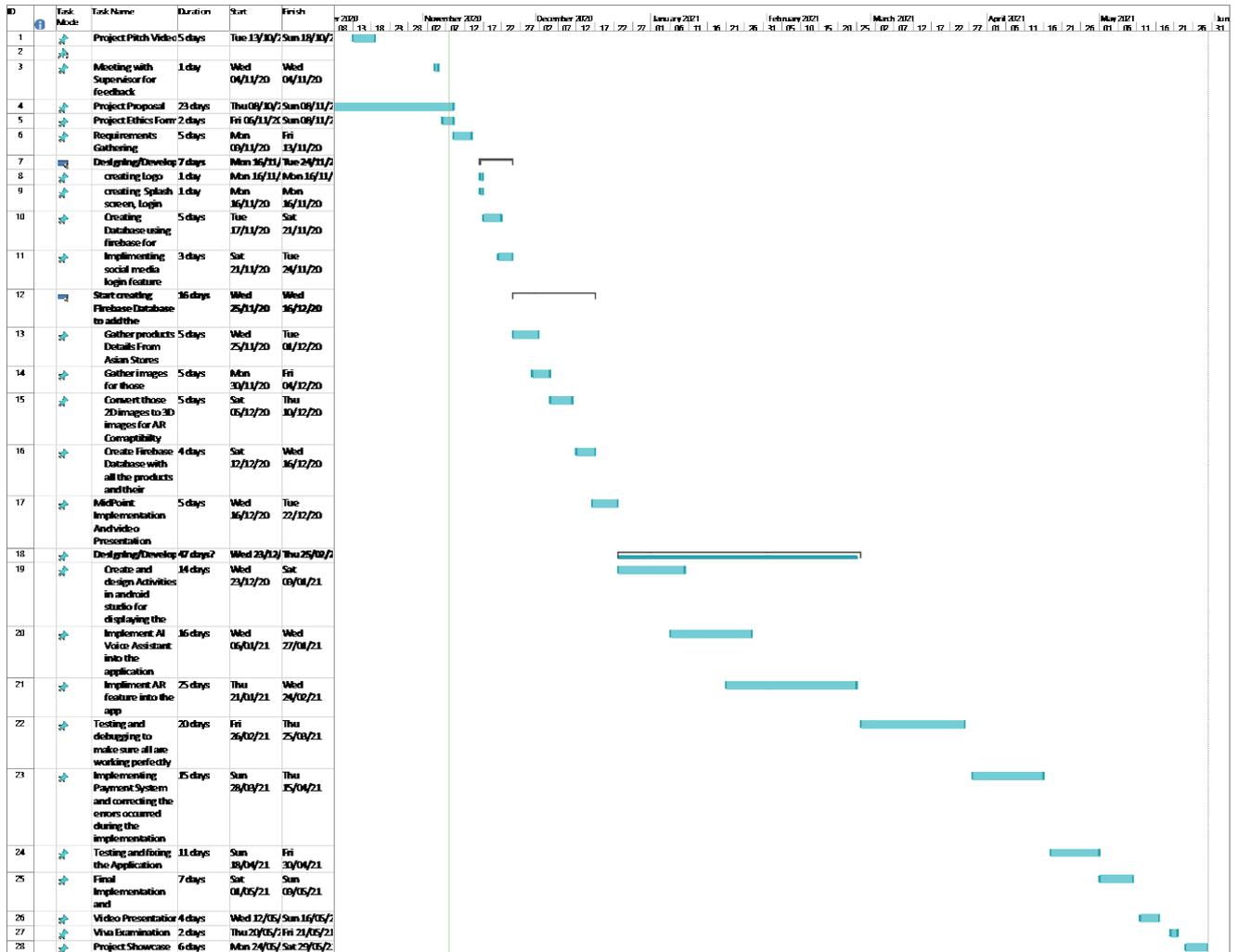
The fourth approach is to create a design and start developing. Once the users requirements and feedbacks are collected, the next step is to get into the design and development stage.

The fifth approach is to carry out testing and get some feedback from the selected users. Once the design and development part is done, the last and the final step is to test the app. More about testing is described in Evaluation section.

The last approach is to deploy this app, so that the user will be able to download the application and use it. This app will be presented to the supervisor for more feedback and it will be deployed to play store.

# 10.0 Project Plan

## Gantt chart



# 11.0 Technical Details

Smart Asian grocery App (name is not finalized yet) is a mobile application that will run on android devices. This application will be created using Programming languages such as java, kotlin, and the IDE that will be used is Android studio. Adobe photoshop needs to be used in order to convert the 2d images to 3d and there by using those 3D object to use for AR in the application. I will be using firebase for creating the database for this mobile application. The login details will also be stored in firebase database. The benefits of using firebase Realtime database is that any changes that made to the database, it will be reflective to those connected users within millisecond. In order to enable the voice assistant feature in my app, I am integrating a Voice AI platform called Alan AI into my application. This will help me to design and code the Alan AI in a way, the grocery application is demanding. In order to carry out the payment function, I will integrate Payrexx into my mobile application. Payrexx is one of the world’s most straightforward Payment platform.

The functionalities includes:

- ◆ A user – friendly Signup/Login system (Will also include social media login feature)

- ◆ Providing platform where user can save their time by ordering groceries online.
- ◆ Providing Same day delivery feature.
- ◆ Smart AI enabled Voice assistant, where a user can interact and ask the voice assistant to order a product without even touching the screen.
- ◆ User will have the option to see the products and the details using Augmented Reality feature.
- ◆ Fast, secure and effective payment system.

## 12.0 Evaluation

Once the Design and Development stage is over, the next step is to test this application with the users. This is the most crucial phase because, in this phase the application will be tested for any bugs, efficiency and so on. Once the application is ready for testing, it will be send to the supervisor and also to some selected users to make sure the app is working efficiently and meet the expectations.