

**SEAMLESS REALTIME E-COMMERCE  
ANDROID APPLICATION WITH GOOGLE  
CLOUD FIREBASE BACKEND**

Project submitted in fulfillment of the requirements for the degree of  
Master of Computer Applications

Project report submitted

By

**ALIP SAHA**

Registration No: 133684 of 2015-2016

Examination Roll No: MCA186021

Class Roll No: 001510503022

Under the Guidance

Of

**Prof. Mita Nasipuri**

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

2018

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

Kolkata-700032

**To Whomsoever It May Concern**

This is to certify that the project entitled “**Seamless Realtime E-Commerce Android Application With Google Cloud Firebase Backend**” has been carried out by the Alip Saha (Class Roll No: 001510503022, Registration No: 133684 of 2015-2016) under my supervision.

I hereby recommended that the report prepared by him accepted in fulfillment of the degree of “Master of Computer Application”, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2017-2018.

.....

Prof. Mita Nasipuri

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

.....

Prof. Ujjwal Maulik

Head of the department of

Computer Science and Engineering

Jadavpur University, Kolkata-700032

.....

Prof. Chiranjib Bhattacharjee

Dean, Faculty of Engineering and Technology

Jadavpur University, Kolkata-700032

Department of Computer Science and Engineering  
Faculty of Engineering and Technology  
Jadavpur University  
Kolkata-700032

**CERTIFICATE OF APPROVAL**

The forgoing project entitled “**Seamless Realtime E-Commerce Android Application With Google Cloud Firebase Backend**” is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

1.....

(Signature of the Examiner)

2.....

(Signature of the Examiner)

# **Declaration of Originality and Compliance of** **Academic Ethics**

I hereby declare that this project contains original work by the undersigned candidate, as a part of his Master of Computer Application (MCA) studies. All Information in this document have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited referenced all material results that are not original to this work.

Candidate's Name: Alip Saha

Registration No: 133684 of 2015-2016

Examination Roll No: MCA186021

Project title: Seamless Realtime E-Commerce Application With Google Cloud Firebase Backend

.....

(Signature with Date)

# **Acknowledgement**

I am pleased to express my gratitude and regards as my Project Guide Prof. Mita Nasipuri (Computer Science and Engineering, Jadavpur University, Kolkata-700032) for her invaluable guidance, suggestion, encouragement, and inspiration during the period of my project. She has given full support to every part of my work starting from beginning. I am very much grateful to her.

Besides these I must have to mention the name of Dr. Subhadip Basu (Computer Science and Engineering, Jadavpur University, Kolkata-700032) and Biswanath Saha (Director, Dearleaf Technology solution Pvt. Ltd) for providing me fruitful suggestions.

Last but not the least I would like to thank all of my family members for giving me constant encouragement and mental support during my project work.

.....

ALIP SAHA

Registration No: 133684 of 2015-2016

Master of Computer Application (MCA)

Department of Computer Science and Engineering

Jadavpur University

## Table of Contents

Introduction.....	8
Project Objective.....	8
Project Overview.....	9
Project scope.....	9
Platform setup.....	10
Android Studio.....	10
Installation process.....	10
Basics Workflow.....	11
Database Support.....	12
Firebase Real-time Database.....	12
Firebase Authentication.....	13
Firebase Cloud Storage.....	14
Firebase Cloud Messaging.....	15
Node.js.....	16
Installation Steps.....	16
Visual Studio Code.....	17
Integration Firebase With Android Studio.....	17
Project Creation.....	18
Activity.....	20
Navigation Drawer Activity.....	21
Layout.....	22
Working Principle of the App.....	23
Recycler View.....	24
Authentication.....	27
Email and password based authentication.....	27
Federated identity provider integration.....	28

Toast.....	29
Data Fetch From Firebase.....	31
Synchronization with Web version of this application.....	34
Conclusion.....	39
Bibliography.....	41

# LIST OF FIGURES

FIGURE 1 Basics Workflow of the project.....	11
FIGURE 2 Project's database snippet.....	12
FIGURE 3 Firebase console for authentication.....	13
FIGURE 4 Firebase Realtime Database connection with android project using Firebase Assistant.....	17
FIGURE 5 The Create Android project screen.....	18
FIGURE 6 The Target Android Devices screen.....	19
FIGURE 7 Activity lifecycle.....	20
FIGURE 8 Navigation menu options.....	22
FIGURE 9 Home page of the app.....	24
FIGURE 10 Shop By Category page.....	25
FIGURE 11 Login page with progress dialogue.....	28
FIGURE 12 Product details page with toast.....	29
FIGURE 13 Database structure of shopping cart.....	30
FIGURE 14 Shopping cart page.....	31
FIGURE 15 Your order page snippet.....	32
FIGURE 16 Block diagram of FCM.....	33



## **ABSTRACT**

This project is an Android based shopping system for an existing shop. The project objective is to deliver the online shopping application into web platform. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using Android device. Thus the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as Flipcart or Amazon. Since the application is available in the phone it is easily accessible and always available.

## **Introduction**

This project is an online shopping android application. It is a form of electronic commerce. E-commerce is the buying and selling of goods and services, or the transmitting of data, over an electronic network, primarily the internet. E-commerce is fast gaining ground as an accepted and used business paradigm. It is reasonable to say that the process of shopping on the web is becoming commonplace. Online shopping is the process whereby customer directly buy goods from a seller in real-time, without an intermediary service, over the Internet. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favorite shop.

## **Project Objective**

The objective of this project is to develop a general purpose e-commerce store where any product (such as books, foods, computers, mobile phones, electronic items, and home appliances) can be bought from the comfort of home through the Internet. However, for implementation purposes, this application will deal with an online vegetables and seasonings store. An online store is a virtual store on the Internet where customers can browse the product list and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction whereas the checkout process through payment gateway is out of scope of this project and hence not implemented.

## **Project Overview**

The central concept of this android based application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store. The Google Cloud Firebase database is used as backend database support. The information pertaining to the products and other user information are stores on Google Firebase database.

The application was designed for the customers who wish to buy the articles. However the application is synchronized with the web version of this application where there is admin part also and any change made by admin is notified to all the user of this android application.

Data entry into the application can be done by the developers through the real-time database. User security is maintain throughout the application. Only authenticated user can add, remove, modify items into cart, access account information, order items and see order details.

## **Project scope**

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24\*7 and a home delivery system which can make customers happy.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flipcart or ebay. Since the application is available in the Smartphone it is easily accessible and always available. We can also implement online payment option into this project to make customers easy to shop.

## **Platform setup**

Android is the world's most popular mobile operating system, powering billions of devices ranging from phones to watches, tablets, TVs, and more. To setup environment for developing the android application first I install Android Studio.

### **What is Android Studio?**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system designed specifically for Android development.

### **How to install Android Studio?**

#### **Installation process:**

System requirements for installing Android Studio in Windows:

- Microsoft® Windows® 7/8/10 (32- or 64-bit)
- 3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

After downloading the .exe file for Android Studio (version 3.0.1) and installing it in my system (Windows 10, 4 GB RAM) and configuring some basic setting as recommended by setup wizard in Android Studio I was ready to start the project although some others tools are installed later on. Android Studio provides wizards and templates that verify your system requirements, such as the Java Development Kit (JDK) and available RAM, and configure default settings, such as an optimized default Android Virtual Device (AVD) emulation and updated system images.

## Basics Workflow:

The workflow to develop an app for Android is conceptually the same as other app platforms. However, to efficiently build a well-designed app for Android, you need some specialized tools. I also used some tools for this project (such as Firebase, node.js) and those are described later. The following list provides an overview of the process to build an Android app.

1. Set up your workspace
2. Write your app
3. Build and run
4. Debug, profile, and test
5. Publish

However the steps 4 and 5 are out of the scope of this project hence not implemented.

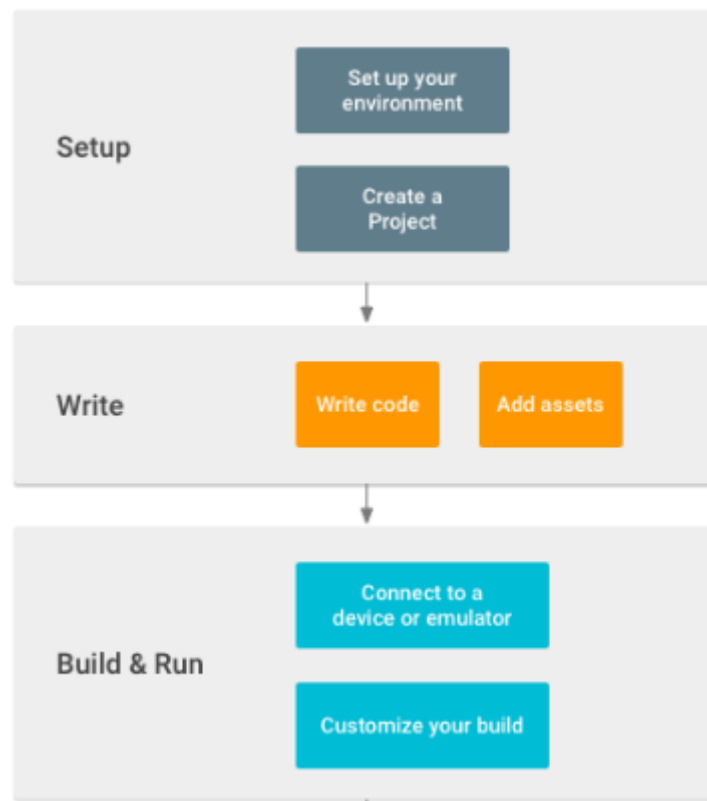


Figure 1: Basics Workflow of the project

To efficiently build a well-designed Android application for this project, I need some specialized tools.

The whole application runs on Firebase Google cloud real-time database as its backend.

## **Database Support:**

### **Why Firebase?**

As far as modern online database platforms are concerned firebase is highly recommended as dynamic database backend with multiple industrial features available through well documented APIs. Firebase provides real-time database triggers, analytics, push notifications, security rules and more technical features in order to help developers built server less applications. It is a JSON based NO-SQL database structure which is very fast and easy on integrations among cross development platforms (Web, Android, iOS etc.) with a single database. For a limited application traffic the usage of firebase database is free, we only need a Google account and this is the major reason that developers are encourage to build on this platform.

There are various benefits of Firebase that relate to the core technology of advancement. In this project I use the following feature of Firebase.

- Firebase Real-time Database
- Firebase Authentication
- Firebase Storage
- Firebase Cloud Messaging

### **Firestore Real-time Database:**

Store and sync data with our NoSQL cloud database. Data is synced across all clients in real-time, and remains available when your app goes offline. The Firestore Realtime Database is a cloud-hosted database. Data is stored as JSON

and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

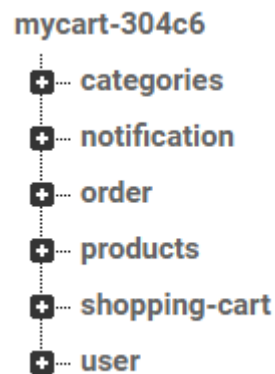


Figure 2: Project's database snippet

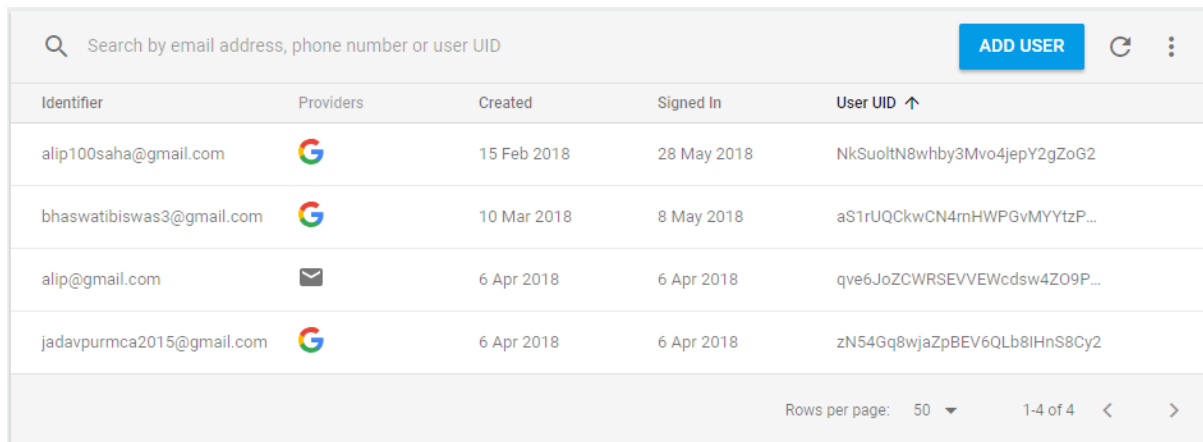
## Firestore Authentication:

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.

Firestore Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firestore Authentication integrates tightly with other Firestore services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

Here in this project I used authentication using email-password and the popular federated identity provider Gmail.







Identifier	Providers	Created	Signed In	User UID ↑
alip100saha@gmail.com		15 Feb 2018	28 May 2018	NkSuoltN8whby3Mvo4jepY2gZoG2
bhaswatibiswas3@gmail.com		10 Mar 2018	8 May 2018	aS1rUQckwCN4rnHWPgVmYYtzP...
alip@gmail.com		6 Apr 2018	6 Apr 2018	qve6JoZCWRSEVVEWcdsw4ZO9P...
jadavpurmca2015@gmail.com		6 Apr 2018	6 Apr 2018	zN54Gq8wjaZpBEV6QLb8IHns8Cy2

Figure 3: Firebase console for authentication.

### Firestore Cloud Storage:

Cloud Storage is built for app developers who need to store and serve user-generated content, such as photos or videos.

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality. We can use SDKs to store images, audio, video, or other user-generated content. On the server, we can use Google Cloud Storage, to access the same files.

I used this feature for storing images of the products and categories.



## **Firestore Cloud Messaging:**

Firestore Cloud Messaging (FCM) is a cross-platform messaging solution that lets you reliably deliver messages at no cost.

Using FCM, you can notify a client app that new email or other data is available to sync. You can send notification messages to drive user re-engagement and retention. For use cases such as instant messaging, a message can transfer a payload of up to 4KB to a client app.

### **How does it work?**

An FCM implementation includes two main components for sending and receiving:

1. A trusted environment such as Cloud Functions for Firestore or an app server on which to build, target, and send messages.
  - o I used Cloud Functions.
2. An iOS, Android, or web (JavaScript) client app that receives messages.

You can send messages via the Admin SDK or the HTTP and XMPP APIs. For testing or for sending marketing or engagement messages with powerful built-in targeting and analytics, you can also use the Notifications composer.

My trusted environment is Cloud Functions. I write the sending logic in Node.js and use the Admin Node.js SDK's FCM API and then deploy my code on Cloud Functions for Firestore.

## **Node.js installation:**

### **What is Node.js?**

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript for server-side scripting i.e. running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm unifying web application development around a single programming language, rather than different languages for server side and client side scripts.

Node isn't a program that you simply launch like Word or Photoshop: you won't find it pinned to the taskbar or in your list of Apps. To use Node you must type command-line instructions, so you need to be comfortable with (or at least know how to start) a command-line tool like the Windows Command Prompt.

Installing Node and NPM is pretty straightforward using the installer package available from the Node.js® web site.

### **Installation Steps:**

- Download the Windows installer from the Nodes.js® web site.
- Run the installer (the .msi file you downloaded in the previous step.)
- Follow the prompts in the installer (Accept the license agreement, click the NEXT button a bunch of times and accept the default installation settings).

I write the notification sending logic in JavaScript in Node.js platform and use the Visual Studio Code as JavaScript editor.

## Why Visual Studio Code?

Visual Studio Code provides IntelliSense, debugging, and powerful editor features for JavaScript. VS Code uses the JavaScript language service to make authoring JavaScript easy. In addition to syntactical features like format, format on type and outlining, you also get language service features such as Peek Definition, Go to Definition, Find all References, and Rename Symbol.

Thanks to a feature called Automatic Type Acquisition as a user do not have to worry about these type declaration files. VS Code will install them automatically for you. I use the version 1.21.1

## Integration Firebase With Android Studio:

### How to add firebase to my project?

Prerequisites:

- A device running Android 4.0 (Ice Cream Sandwich) or newer, and Google Play services 15.0.0 or higher.
- The latest version of Android Studio

The Firebase Assistant is the simplest way to connect your app to Firebase. The Assistant can connect your existing project or create a new one for you with all the necessary gradle dependencies.

To open the Firebase Assistant in Android Studio:

- Click **Tools > Firebase** to open the **Assistant** window.
- Click to expand one of the listed features (for example, Analytics), then click the provided tutorial link (for example, Log an Analytics event).

- Click the **Connect to Firebase** button to connect to Firebase and add the necessary code to your app.

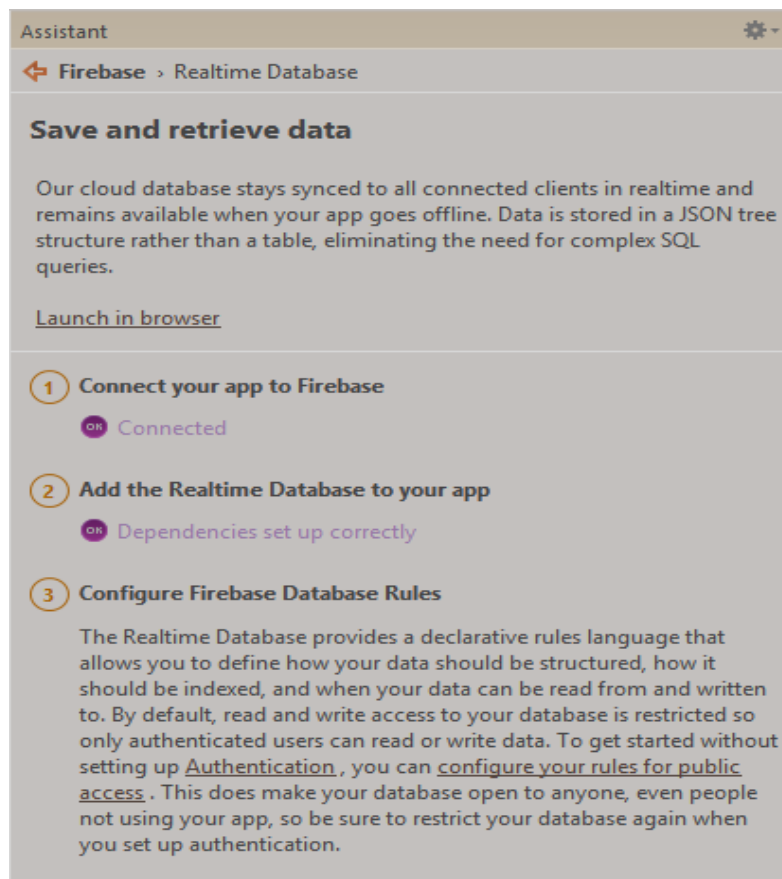


Figure 4: Firebase Realtime Database connection with android project using Firebase Assistant.

## Project Creation:

### How I create the Android project?

After configuring primary development environment I start with creating the project in Android Studio. The following steps describe that in details.

#### Step 1: Start and configure the project

If you don't have a project opened, Android Studio shows the Welcome screen, where you can click start a new Android Studio project.

In the Configure your new project screen that appears, you can set the name of your app, the package name, and the location of your project. The name of my app is **My Shop** and the package name is **com.example.alipsaha.mycart1**. I create a separate folder for the entire project and the location is **B:\MyCart1**.

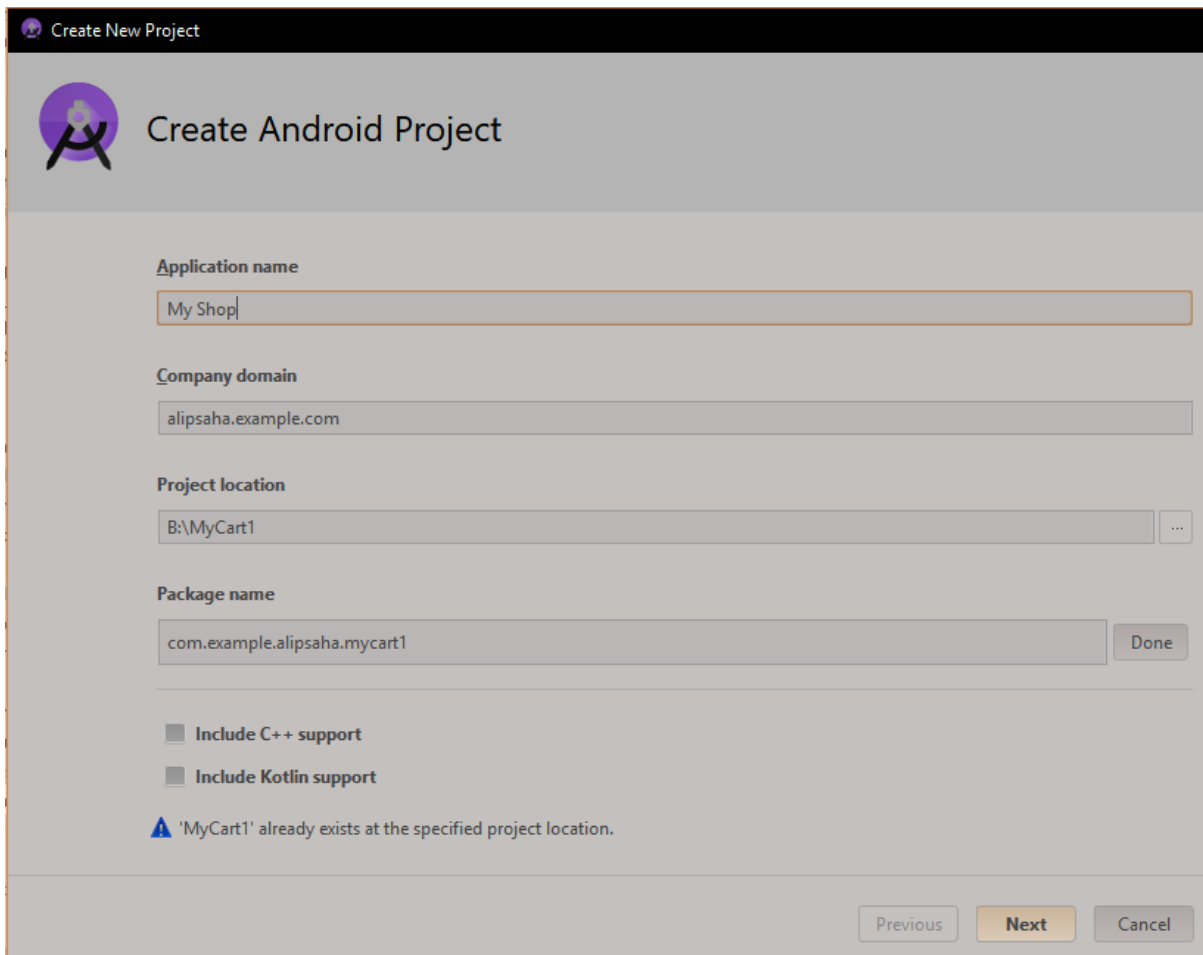


Figure 5. The Create Android project screen.

## Step 2: Select form factors and API level

In the next window I select the device form factors that I want to build for, and the minimum version to support for each. I select the option only for the phone and tablet users having minimum Android version 5.0. i.e. Android Lollipop, which cover 71% users.

Each module contains all the code and resources that will be built into an Android app package (APK) for the corresponding device. If you later decide to add

support for a new device, you can add a module at that time. And you can share code and resources between modules using an Android library.

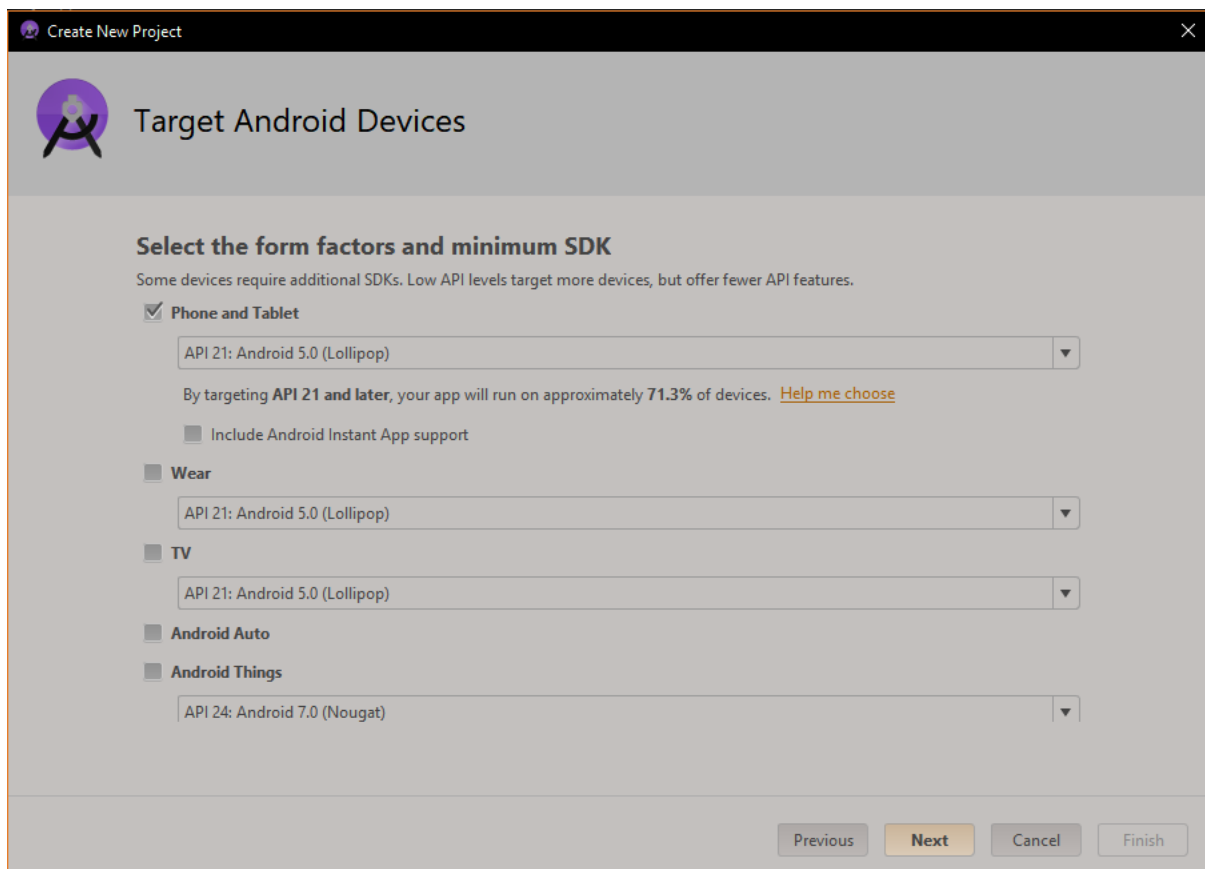


Figure 6. The Target Android Devices screen.

### Step 3: Add an activity

The next screen is for selecting an activity type to add to your app. This screen displays a different set of activities for each of the form factors I selected earlier.

From the set of activity I select Navigation Drawer Activity as my launching activity.

### **What is an activity?**

An activity represents a single screen with a user interface just like window or frame of Java. Android activity is the subclass of ContextThemeWrapper class.

In C, C++ or Java programming language we must have seen that your program starts from main() function. Very similar way, Android system initiates its program with in an Activity starting with a call on onCreate() callback method. There is a sequence of callback methods that start up an activity and a sequence of callback methods that tear down an activity as shown in the below Activity life cycle diagram:

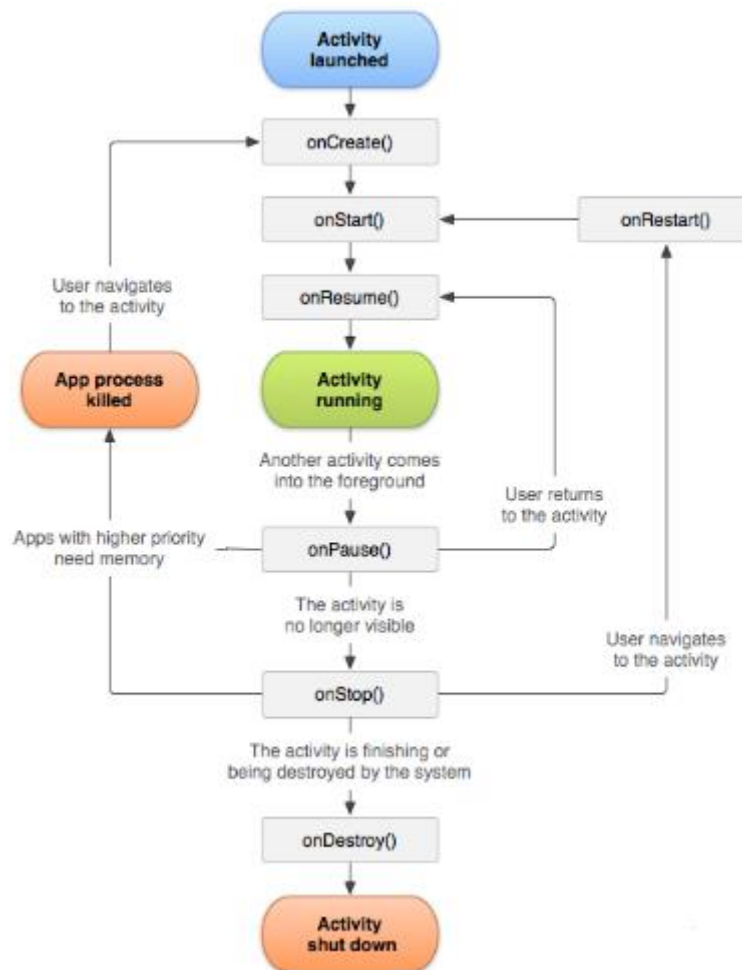


Figure 7: Activity lifecycle

### Navigation Drawer Activity:

As stated before Navigation Drawer is my launching activity. The navigation drawer is a UI panel that shows your app's main navigation menu. It is hidden when not in use, but appears when the user swipes a finger from the left edge of the screen or, when at the top level of the app, the user touches the drawer icon in the app bar.

A project can have several number of activity as needed. We can create an activity any time throughout the project. All the activities are register in AndroidManifest.xml with necessary attributes.

Every Android project must have an AndroidManifest.xml file (with precisely that name) at the root of the project source set. The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play.

#### Step 4: Configure your activity

The next screen lets you configure the activity to add to your app. Here I provide the activity name, the layout name, and the activity title. Then click Finish. Android Studio now sets up your project and opens the IDE. My launching activity name is **MainActivity** and the corresponding layout name is **activity\_main**.

#### **What is a layout?**

A layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects. A View usually draws something the user can see and interact with. Whereas a ViewGroup is an invisible container that defines the layout structure for View and other ViewGroup objects.

Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts. You can also use Android Studio's Layout Editor to build your XML layout using a drag-and-drop interface and most of the time I use this as it is easier. Also I use some customize layout objects or widgets to make my app attractive and user friendly.



From here I was ready to develop my application.

## How this application works?

There are five main navigation menu (Home, Shop by Category, Your Account, Your Order, and Logout) of my application as shown in the following figure.

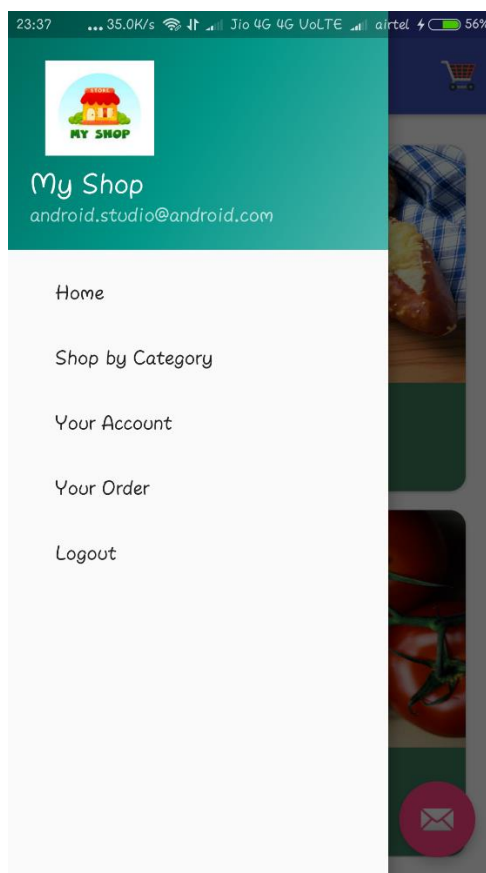


Figure 8: Navigation menu options

The Home menu option shows the product list of my shop which is also launching page of my application. The list is create using Android RecyclerView.

## What is RecyclerView?

If your app needs to display a scrolling list of elements based on large data sets (or data that frequently changes), you should use RecyclerView as described on this page. The RecyclerView widget is a more advanced and flexible version of ListView.

In the RecyclerView model, several different components work together to display your data. The overall container for your user interface is a RecyclerView object that you add to your layout. The RecyclerView fills itself with views provided by a layout manager that you provide. You can use one of our standard layout managers (such as LinearLayoutManager or GridLayoutManager), or implement your own. Here I use LinearLayoutManager.

The views in the list are represented by view holder objects. These objects are instances of a class you define by extending RecyclerView.ViewHolder. Each view holder is in charge of displaying a single item with a view. Each view holder might represent a single product of the list. The RecyclerView creates only as many view holders as are needed to display the on-screen portion of the dynamic content, plus a few extra. As the user scrolls through the list, the RecyclerView takes the off-screen views and rebinds them to the data which is scrolling onto the screen.

The view holder objects are managed by an adapter, which you create by extending RecyclerView.Adapter. The adapter creates view holders as needed. The adapter also binds the view holders to their data. It does this by assigning the view holder to a position, and calling the adapter's onBindViewHolder() method. That method uses the view holder's position to determine what the contents should be, based on its list position.

I use FirestoreRecyclerAdapter, a RecyclerAdapter specially-made for integration with Firebase. All the products information are load from Firebase Real-time database into RecyclerView and shown using CardView which is a container for displaying list of data in similar style.

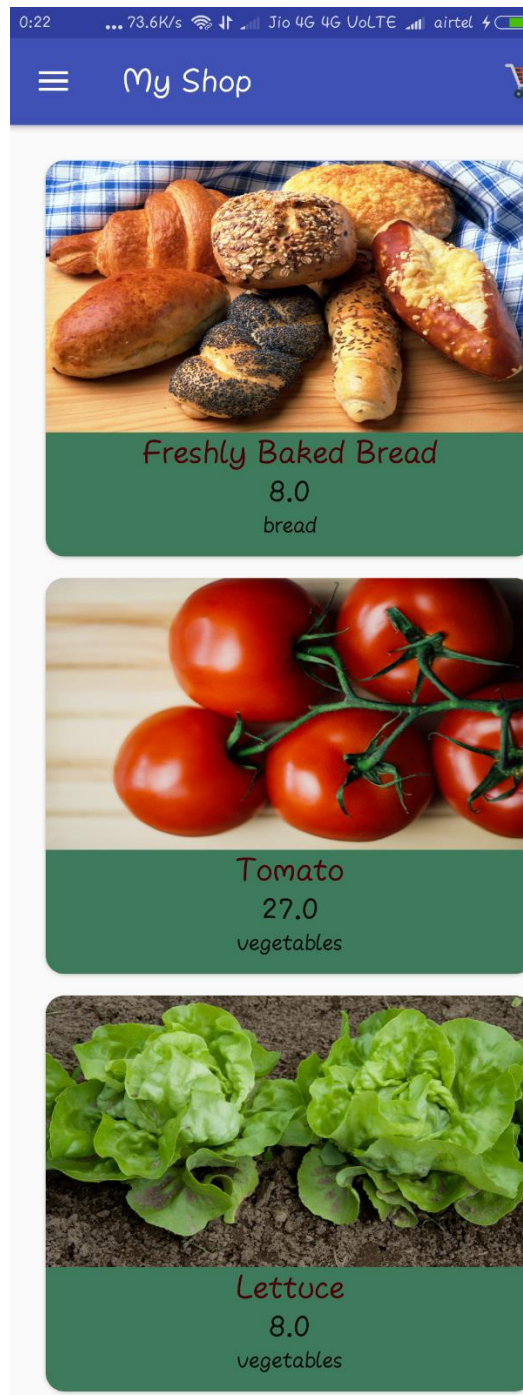


Figure 9: Home page of the app

In the home page the products are listed in arbitrary order. But this is not always an efficient manner because if a user want to browse products of same

category he/she might have to scroll a mile which is not expected. Hence to ensure better user experience it is necessary to provide option for shop product by category. I also implement the Shop by Category menu option. Where customer can show only the products having same category selected by them.

Filtering and sorting can be expensive, especially when done on the client so I decided to implement that in server side. And I use the Firebase Realtime Database Query class to retrieve data sorted by value of a child (category). You can also filter the sorted result to a specific number of results or a range of keys or values.

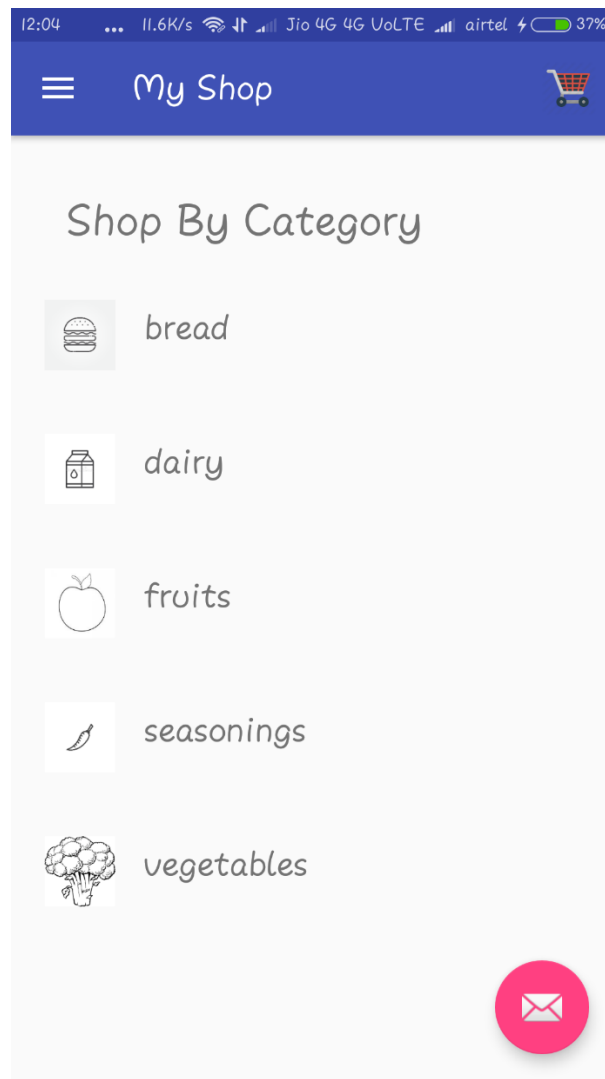


Figure 10: Shop By Category page

Authentication is a major part of this project. To provide better security to the customer I spent lots of time in this section. For this purpose again I use the feature of Firebase.

For the first time customer have to register and later on he/she can login to get user depended services. In two activities RegisterActivity and LoginActivity I implemented whole authentication process.

In this application I provide two options for user authentication.

## **1. Email and password based authentication:**

Authenticate users with their email addresses and passwords. The Firebase Authentication SDK provides methods to create and manage users that use their email addresses and passwords to sign in. Firebase Authentication also handles sending password reset emails.

### **How I implement this?**

The first step is Add the dependency for Firebase Authentication to the app-level build.gradle file:

```
implementation 'com.google.firebase:firebase-auth:16.0.1'
```

Next I enable Email/Password sign-in from Firebase console.

### **Create a password-based account:**

To create a new user account with a password, complete the following steps in your app's RegisterActivity activity:

- A. In Register activity's onCreate method, get the shared instance of the FirebaseAuth object.
- B. When initializing the Activity, check to see if the user is currently signed in.

- C. When a new user signs up using your app's sign-up form, complete any new account validation steps that your app requires, such as verifying that the new account's password was correctly typed and meets your complexity requirements. For this project password should be more than 5 characters.
- D. Create a new account by passing the new user's email address and password to `createUserWithEmailAndPassword`.

If the new account was created, the user is also signed in. In the callback, we can use the `getCurrentUser` method to get the user's account data.

### **Sign in a user with an email address and password:**

The steps for signing in a user with a password are similar to the steps for creating a new account. The only difference is in step D. Here in this step we have to pass the user's email address and password to `signInWithEmailAndPassword` when a user sign-in to this app.

## **2. Federated identity provider integration:**

Authenticate users by integrating with federated identity providers. The Firebase Authentication SDK provides methods that allow users to sign in with their Google, Facebook, Twitter, and GitHub accounts.

I use Google as identity provider and authenticate user against their Gmail account. Implementation of this is quite similar to Email and Password authentication.

When the user is logging in to the app a progress dialogue is shown for better user experience and after completing each process (such as register, login) I provide toast like “Successfully Logged In” to make the app responsive to the user.

## What is a toast?

A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout.

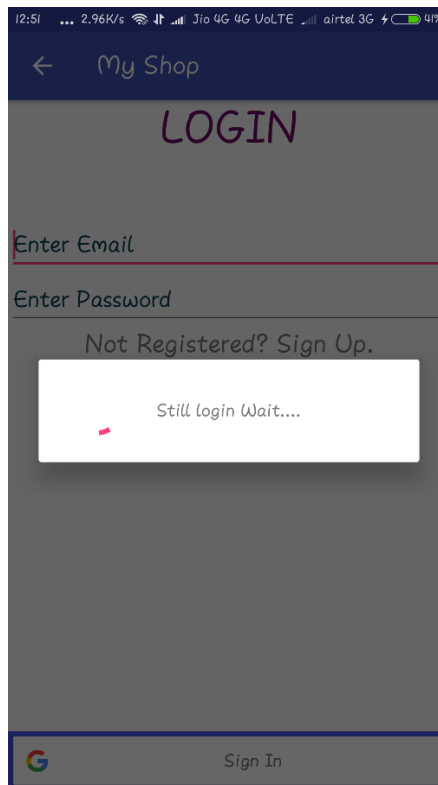


Figure 11: Login page with progress dialogue

To logout from the app user have to go to the navigation drawer menu item Logout, on clicking that menu user can logged out that account.

When a user register into this app some basic information's (such as email, name) are stored in Firebase database under the unique key of the user. And the logged in user can see those information from Your Account menu option.

By clicking on a product from products list user can see the details of this item and can select the quantity. If the user is logged in Here he/she can add the product

into his/her shopping cart or can ordered the item. If adding into cart or ordering item is successful or failed user can show the corresponding toast (such as “Item added to Cart “).

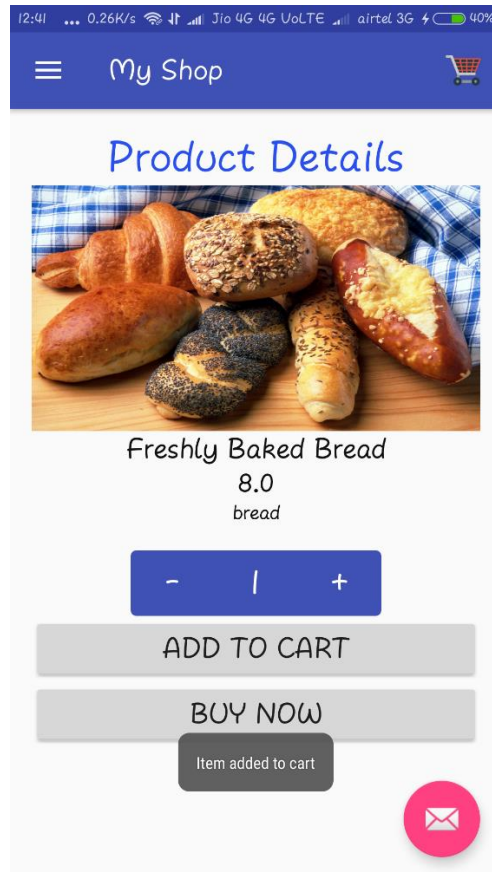


Figure 12: Product details page with toast

The logic of the product details page i.e. store the cart or ordered item with quantity under the corresponding node with respect to the user in Firebase Real-time database are written within a fragment called ProductFragment. There are 3 other fragments in this application.



## What is fragment?

A Fragment represents a behavior or a portion of user interface in a FragmentActivity. You can combine multiple fragments in a single activity to build a multi-pane UI and reuse a fragment in multiple activities. You can think of a fragment as a modular section of an activity, which has its own lifecycle, receives its own input events, and which you can add or remove while the activity is running (sort of like a "sub activity" that you can reuse in different activities).

A fragment must always be hosted in an activity and the fragment's lifecycle is directly affected by the host activity's lifecycle.

## Data Fetch From Firebase:

A logged in user can show his cart item anytime by clicking the cart icon on the right side of the action bar. Shopping cart shows the cart items of the user in a recycler list view and FirestoreRecyclerAdapter load the data from firebase Realtime database. Firebase store data in tree structure. There is a node shopping-cart in my database where I store the cart items details of all user corresponding to their unique key as shown in the following figure.



Figure 13: Database structure of shopping cart

From the above database structure we can see that user having unique key NkSuoltN8whby3Mvo4jepY2gZoG2 added two item in his cart named Freshly Baked Bread and Banana. There is child node of the node shopping-cart named total and whose value is 26. This is the cart subtotal i.e. sum of price of all item in the cart. From the shopping cart page user can modify quantity of items, remove items from cart and checkout all cart item together. And any change made here update the database accordingly also that reflect the page immediately.

Following figure is the shopping cart page of the user whose database structure under shopping-cart node is given in above figure.

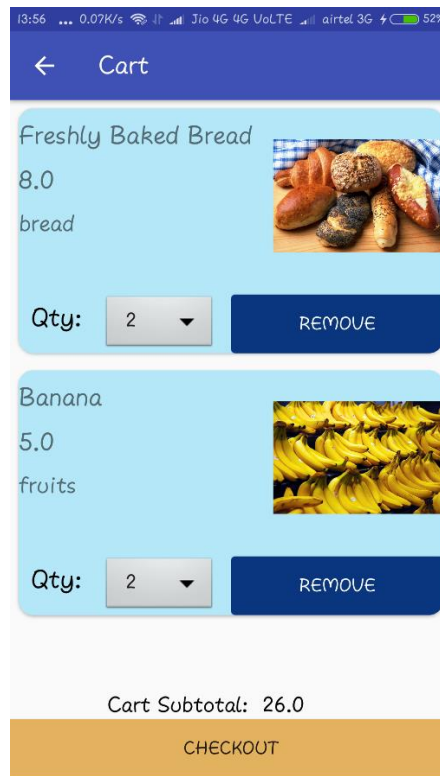


Figure 14: Shopping cart page

The Your Order menu item is for listing all the items order by the user. And those order information with the time of order are also saved in Firebase under the order node. This is also a RecyclerView but the only difference is instead of using the FirestoreRecyclerViewAdapter here I use my own customize adapter RecyclerViewAdapter to fetch data from firebase.

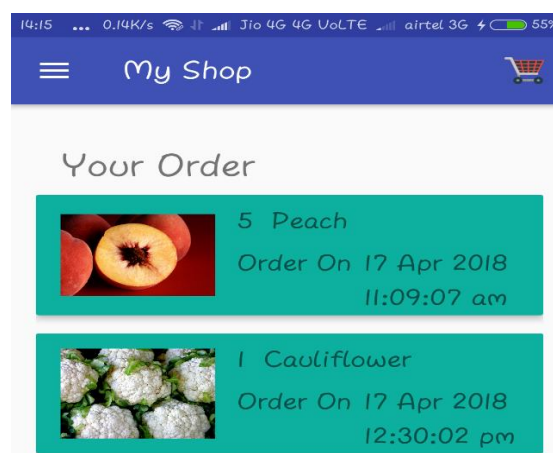


Figure 15: Your order page snippet

## **Synchronization with Web version of this application:**

There is a Web version of this application and my android app is synchronized with that platform. Android app is only for the user but the Web app also maintain the admin part where admin can manage products, order etc. Any user can browse both application with a single account. And this is implemented easily using Firebase. Both web and Android use same Firebase project as their backend.

If any new products are added or existing products are modified by admin a push notification related to that change is send to each Android user. Firebase Cloud Messaging is used in my project for sending notification.

## **Process of sending Firebase cloud message to Android device using Firebase function:**

First we need to add internet permission to the manifest file.

```
<uses-permission android:name="android.permission.INTERNET" />
```

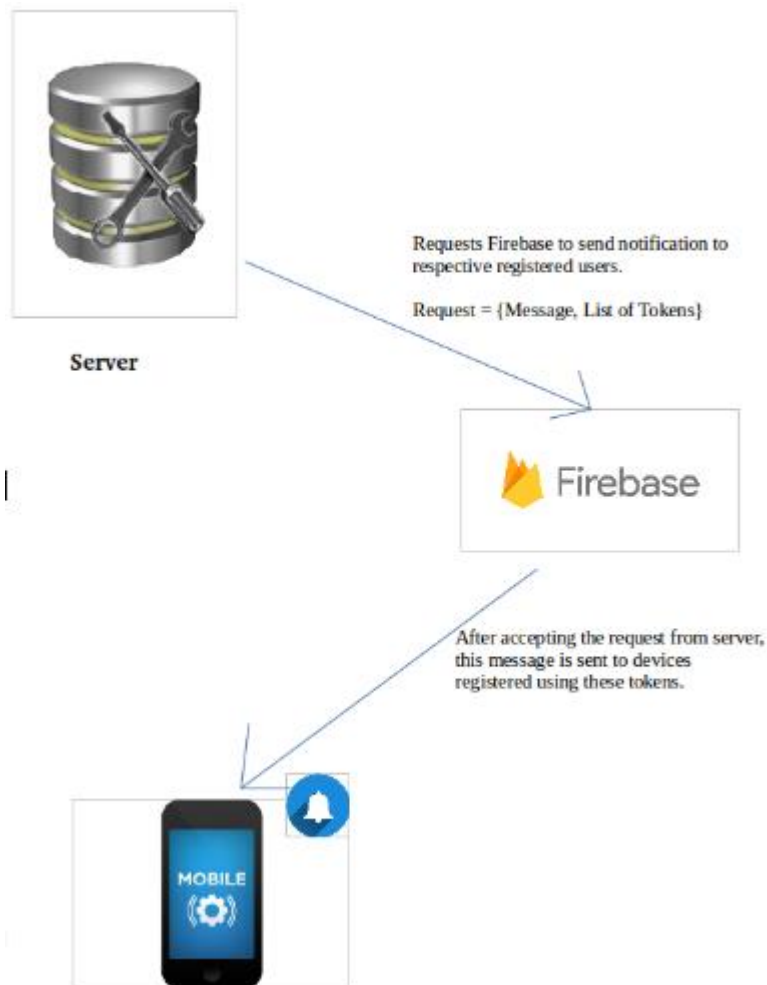


Figure 16: Block diagram of FCM

Now we need to add a service that extends FirebaseMessagingService. This is the Base class for communicating with Firebase Messaging. This FirebaseMessagingService extends Service class. This class provides various functionalities:

- Automatically displaying notifications.
- Message handling beyond receiving notifications on apps in the background.
- Receive notifications in foregrounded apps
- Receiving data payload
- Sending upstream messages

To handle any type of events required by the application, we need to override this base class methods. These methods are invoked on a background thread. To add this service include following to your manifest file:

```
<service android:name=".MyFirebaseMessagingService">
  <intent-filter>
    <action
      android:name="com.google.firebase.MESSAGING_EVENT" />
    </intent-filter>
  </service>
```

Now we need to add a service that extends FirebaseInstanceIdService. This is the Base class to handle Firebase Instance ID token refresh events. This FirebaseInstanceIdService extends Service class. It provides various functionalities:

- Creation of registration tokens
- Rotation of registration tokens
- Update process of registration tokens
- Sending notification to a single device or group of devices
- Allows devices to subscribe to various topics with the server subscription management API.

Since my app uses Instance ID tokens, hence I extend this class and implement onTokenRefresh() to receive token refresh events.

On initial startup of the app, the FCM SDK will generate a registration token for the client app instance. Since I want to target a device groups, I need this token which can be retrieved from our FirebaseInstanceIdService class. This token could be rotated or changed after initial startup. So, it is necessary fetch the latest updated registration token using FirebaseInstanceId.getInstance().getToken() ;

If the system determines that the token needs to be refreshed, then this callback is called. Once the token is retrieved, I save it to Firebase database under the node named notification .This will not be called very frequently. It is needed for key rotation and to handle Instance ID changes; which happens in following cases:

- When app deletes Instance ID
- When app is restored on a new device
- When user re-installs or uninstalls the app
- When user deletes the app data

Once this is set up we are ready to start sending downstream messages with the Firebase Function.

### **What is Firebase Function?**

Firebase functions are the set of JavaScript functions that you can run directly in Google cloud and integrated with your firebase application. It can interactive your database, files, users, and also has the ability to perform other firebase operations like triggering push notification.

The coolest part is we can do more things as you can power it with npm modules. Some of the things are like resizing the image, slack integrations, complex calculations and many more.

### **How to integrate function to the project?**

1. Run `firebase login` to log in via the browser and authenticate the firebase tool.
2. Go to your Firebase project directory.
3. Run `firebase init functions`. The tool gives you an option to install dependencies with npm. It is safe to decline if you want to manage dependencies in another way.

4. The tool gives you two options for language support:

- Javascript
- TypeScript.

After these commands complete successfully, the project structure looks like this:

```
+-.firebasesrc
|
|
+- firebase.json
|
+- functions/
  |
  +- .eslintrc.json
  |
  +- package.json
  |
  +- index.js
  |
  +- node_modules/
```

### How to write a function?

- firebase-admin (For listening to changes in database, one can also use firebase module for it, but firebase-admin gives you more control on the data)
- firebase-functions (For initializing and defining the functions)

The name of my function is onProductAdded().To deploy and execute the onProductAdded() function, follow these steps:

1. Run this command to deploy your functions:

```
$ firebase deploy --only functions
```

After you deploy, the Firebase CLI outputs the URL for any HTTP function endpoints.By default, the Firebase CLI looks in the functions/ folder for the source code.



2. Add a text query parameter to the onProductAdded() URL, and open it in a browser:

The function executes and redirects the browser to the Firebase console at the database location where the text string is stored.

After deploying and executing functions, we can view logs in the Firebase console for Cloud Functions.

## **Conclusion:**

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur's but also from the customer's point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. "Application design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other application". Hence I have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible.

In this project, the user is provided with an e-commerce application that can be used to buy vegetables and fruits online. To implement this as a mobile application I used Android as the Technology. Android has several advantages such as enhanced performance, scalability and simplicity. To build any web application using Android I need a programming language such as JAVA, JavaScript.

Firestore was used as back-end database since it is one of the most popular open source real-time databases, and it provides fast data access, easy installation and simplicity. Also I used Firestore for user authentication, cloud storage and cloud messaging.

A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides a number of features that are designed to make the customer more comfortable.

This project helps in understanding the creation of an interactive app layout and the technologies used to implement it. The design of the project which includes Data Model and Process Model illustrates how the database is built with different nodes, how the data is accessed and processed from the database.

The building of the project has given me a precise knowledge about how Android Studio is used to develop an Android application, how it connects to the Firestore database to access the data and how the data and app layouts are modified to provide the user with a shopping cart application.

## **Bibliography:**

### **Articles:**

1. Abelson, Hal (July 31, 2009). "App Inventor for Android". Google Research Blog. Archived from the original on August 7, 2010. Retrieved July 12, 2010.
2. Diwakar, H., Marathe, M. (2000). The architecture of a one-stop web-windowshop. December, ACM SIGecom Exchanges, Volume 2 Issue 1.

### **Books:**

1. Head First Android Development by Jonathan Simon. Copyright © 2011 Jonathan Simon. United States of America. Published by O'Reilly Media.
2. Head First JavaScript Programming by Eric T. Freeman and Elisabeth Robson Copyright © 2014 Eric Freeman, Elisabeth Robson. All rights reserved. United States of America. Published by O'Reilly Media.

### **Websites:**

1. <https://developer.android.com/training/implementing-navigation/nav-drawer>
2. [https://www.tutorialspoint.com/android/android\\_fragments.htm](https://www.tutorialspoint.com/android/android_fragments.htm)
3. <http://www.vogella.com/tutorials/AndroidDatabinding/article.html>
4. <https://stackoverflow.com/questions/27187725/retrieving-specific-data-from-firebase-android>
5. <https://firebase.google.com/docs/auth/android/start>
6. [https://github.com/mmazzarolo/firebaserecyclerview/blob/master/app/src/main/java/com/example/matteo/firebase\\_recycleview/FirebaseRecyclerAdapter.java](https://github.com/mmazzarolo/firebaserecyclerview/blob/master/app/src/main/java/com/example/matteo/firebase_recycleview/FirebaseRecyclerAdapter.java)
7. <https://docs.npmjs.com/getting-started/installing-node>
8. <https://www.w3schools.in/category/android-tutorial>