

## Developing a Novel Android Based Application for Restaurant Management

Himanshu Ojha, Kandepalli Shyam Sundhar, Rithik kumar, Amritpal Singh

School of Computer Science and Engineering,  
Lovely Professional University, Phagwara-144411, Punjab, India.

### Abstract

The brief narration of this research paper is that it deals with the development of a user-friendly Android Applications which will be used for performing several functions for a locally originated Italian restaurant named Apna Country Pizza, which includes applications: Apna Country Pizza and Apna Country Pizza: Merchant Application. Further, for the development purposes JAVA will be used for backend programming, XML (Extensible markup language) will be used for front-end programming, Firebase will be used for Database and Authentication purposes and Razorpay will be used for Payment Control. This paper describes the development, testing and working with a thorough explanation of this application describing some ideas for updates in future.

**Keywords:** Android Studio, Firebase, Razor pay, Java, XML

### INTRODUCTION

APNA Country Pizza and Apna Country Pizza: Merchant Application is a Native applications which uses Android Studio as a Platform to design with JAVA, XML, Firebase and Razor pay to fulfill various purposes as mentioned earlier

We can observe that many large-scale businesses own an online means of contacting, advertising and performing various tasks because it is a very efficient and reliable means. But small-scale business owner cannot get a chance to be introduced with any online platform on which several tasks can be performed. This application is in regard to support a small-scale restaurant start-up named APNA COUNTRY PIZZA located in Bareilly, Uttar Pradesh, India and the applications that are developed are intended to fill the gap and introduce an online platform to a small-scale business.

Apna Country Pizza to be used by the users to perform various functions in which the most basic function is Ordering items. All other functions are to support these basic functions like Online Payment, Order Pickup, Adding Favorite items to favorite list, Adding Information and storing previously processed information to database. All these functions are performed by particular user by Logging in using phone number and password which is Authenticated while Signing up. As, this was mentioned earlier that the application is to introduce Android application to the business this application is provided to the owner free of cost and no terms and condition to give the glimpse that how an android application can be beneficiary to a businesses such as Restaurants itself.

### WHY NATIVE APPLICATION?

In Simple terms A Native Application is a type of application which can be operated only a single type operating system with other such as Hybrid application, Web Application and Cross Platform Application. There are Millions of

Applications available on different platforms, in which Android operating System wins the race with around 3.5 Million applications available on Google Play Store to download. The pie chart below represents the downloads of the applications that had occur worldwide. This clearly represents that 51% of the applications that were downloaded was on Google Play Store.

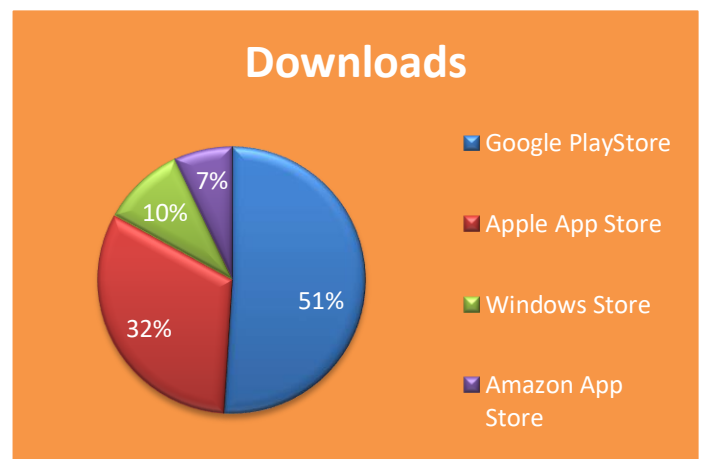


Figure 1 : Representation of downloads over various platform

Given the situation for a Restaurant and food ordering application, the best platform to utilize the application is with a mobile phone. Which leaves the option of Android Operating System and IOS as the best option for the applications?

As, it was mentioned earlier that the application is just an experiment of online platform to the business, native application provides a limited option to Merchant and if the client found the concept to be self beneficiary. He can ask for update so that these applications can be used on different operating systems. So, in conclusion for the best reach to the costumers, and in provided situation. The application is built only for Android users only.

## REQUIREMENTS

In order to advance any type of Application certain set of basic Requirement is a necessity. Following are the requirement with specified versions for development phase of this application.

- Android Studio
- Firebase
- Razor pay account
- Clear Understanding of JAVA
- Clear Understanding of XML
- An Android Operating System

Firestore:2.1.0	Implementation
Junit1.1.1	android Test Implementation
Junit:4.12	Test implementation
Legacy-support-v4:1.0.0	Test implementation

Lib:1.0.2	Implementation
Material:1.1.0	Implementation
Multidex:2.0.1	Implementation
Picasso:2.71828	Implementation
Pinview:1.4.3	Implementation
ViewPager2:1.0.0	Implementation
Firebase-firestore:21.5.0	Implementation

## VERSIONS

For the development of the Applications following is the list of versions of software that are been used.

- Android Gradle plugin Version 4.1.1
- Gradle Version 6.5
- Compile SDK version 3.0
- API 30 : Android 11.0(R)
- Minimum SDK Version 23
- Build tool version 30.0.1

## DEPENDENCIES AND PERMISSIONS

In Android Studio, dependencies allow us to include external library or local jar files or other library modules in our Android project. To add a dependency to your project, specify a dependency configuration such as implementation in the dependencies block of your module's build.gradle file.

### Code Sample to add dependencies:

```

plugins {
    id 'com.android.application'
}
android { ... }
dependencies {
    // Dependency on a local library module
    implementation project(':mylibrary')

    // Dependency on local binaries
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    // Dependency on a remote binary
    implementation 'com.example.android:app-magic:12.3'
}

```

Table 1 : List of all the dependencies used in the applications

Dependencies	Configuration
ImageSlideShow:0.0.7	Implementation
AppCompact:1.2.0	Implementation
ccp:2.4.0	Implementation
Checkout:1.5.10	Implementation
espresso-core:3.2.0	android Test Implementation
Firebase-analytics:17.5.0	Implementation
Firebase-auth:19.4.0	Implementation

Android runtime Permissions, as the name suggests are the Permissions required by application to run.

Without these Permissions the applications may not work. These Permissions are requested by user when the application is used.

### Code Sample to add dependencies:

```

<manifest>
<uses-permission android:name="Permission"/>
</manifest>

```

## COMPONENTS OF APPLICATION

Both Applications are divided into several components which are visible as the user interacts with the applications. In order to create these components ACTIVITY or FRAGMENTS are used. Activities that are currently displayed are then intended to another Activity as programmed. Several Components from Start of the Application to the end creates a fully working Application.

With Activity two files are created, one of which is a JAVA file in which all the Back-end programming is done with other file that is XML file in which all the front-end programming is done. Components of XML files with other Resources are accessed by using R class. R class is generated when the Application is compiled which contains resource IDs for all the resources available in your res/ directory.

Together these things are functioned with the help of programming in the JAVA file which creates a functional component and these components combined creates the application.

The Components required for the Applications are explained below, these have their own functionality and the working of these components will be explained further.

❖ **Splash Screen** : Splash Screen is the Screen

Firestore-core:17.5.0	Implementation
Firestore-database:19.4.0	Implementation
Firestore-firestore:21.5.0	Implementation
Firestore-storage:19.1.1	Implementation

visible every time for 5 seconds every time the application is opened, This screen gives the glimpse of the organisation for which the application is used. This Screen usually contains the name and logo of the organisation and/or developer.

- ❖ **On-Boarding Screen** : Displays just after the Splash Screen is displayed and only be visible when the application is used for the first time. This gives the information about what the application is going to be and how useful.
- ❖ **Log-in Screen** : Use to login to the application to use it, by entering the credentials such as phone number and password. If the credentials provided by the user is right then he/she will be authenticated to use the application further.
- ❖ **Sign-Up Screen** : Use to Sign-Up and initialize the credentials which is further authenticated by sending an OTP to the users phone number. If the credentials were successfully authenticated, new Login credentials are made.
- ❖ **Forgot Password Screen** : Used when user forgot his Password, This Screen can be used to change the password in case user forgets it. This process also needs an authentication in order to process. An OTP will be send to user’s phone number and if the authentication is successful. User can enter a New Password and it will change.
- ❖ **Home Screen** : This screen is visible after successfully logging in to the application. This screen displays the main content of the application which is most essential. In the case of Apna Country Pizza, Home screen displays the items categorically and offers which can be used for further processes.
- ❖ **Menu Screen** : As mentioned, in the home screen the items are marked in the categorical order, when user selects the category, Menu Screen is opened which displays the information of the product such as Price, Image and Description of the Item. User can either add it to Cart or add it to their favorite items.
- ❖ **Cart Screen** : This screen displays all the items that were added to the cart. From here user can either remove the item from cart or can proceed to order. The Final Amount for the items in the cart will be displayed on the Proceed for order button.
- ❖ **Proceed Order Screen** : This consist of the options required to proceed the order for the delivery. Here user can select the address and Payment mode, and proceed to order the items.
- ❖ **Edit Address Screen** : This Screen is used to edit addresses in order to provide it on the proceed order screen.

- ❖ **Payment Screen** : If the online mode of payment is selected, this screen is used to accept the payment which uses the Razorpay interface to do so. If the Payment is successful the order will be placed. Otherwise, the order will be cancelled.
- ❖ **Order Screen** : This Screen is used to display all the Previous, Past , Cancelled and Current Order with the essential information and delivery status.
- ❖ **Favourite Screen** : This screen displays the favourite items which are marked by the user on item Screen. User can either further process it or can remove the item from the favourite screen
- ❖ **Profile Screen** : This screen includes some functions that are used to edit the profile of user such as, Change phone number, change name, Edit Address, Logout, etc.

Below mentioned components are specifically use for the application Apna Country Pizza: Merchant Application

- ❖ **Dashboard** : Includes buttons to access screen such as Live Orders, Accepted Orders, Delivered Order, Cancelled Order.
- ❖ **Live Orders** : This screen is used to display the items ordered by the screen. Merchant can accept the order or cancel it.
- ❖ **Accepted Orders** : All the orders accepted by the Merchant can be accessed through this screen.
- ❖ **Cancelled Orders** : All the orders that are cancelled by the merchant can be accessed by this screen.

## WORKING OF APPLICATIONS

This Topic deals with all the explanation required to understand the working of the Applications.

Apna Country Pizza, as mentioned earlier deals with the user-end, which will provide an interface to the user on which user can perform operations. The fundamental operation of both the application is Ordering and Delivering the items provided by the organization. All other operations are to support this this fundamental operation which by the help of this applications processed systematically.

On Opening the Application a Splash Screen appears for 5 seconds which displays the logo and name of the brand, After that an On-Boarding Screen appears if the user is using the application for the first time. Otherwise, On-Boarding Screen will be skipped and the user directly reaches to Login-Screen.

Here user can Log-in, Sign-up or change password accordingly. After a successful Login Home Screen will appear which displays various offers and categories of item user can select, After selecting the category user can select an item and add it to cart and/or add the item to favorite which displays in the favorite-screen. To order the items user has to access Cart, from there user can order the items mentioning various information required such as Quantity and Address. The mode of Payment can be online

or Cash on Delivery according to user. If all the provided information is correct and the payment is successful the order will be live and Merchant can access by Merchant application. All the live, delivered and Cancelled orders are displayed in the Order tab. User can edit or review the info in profile. The detailed working of the application is displayed in the flowchart below.

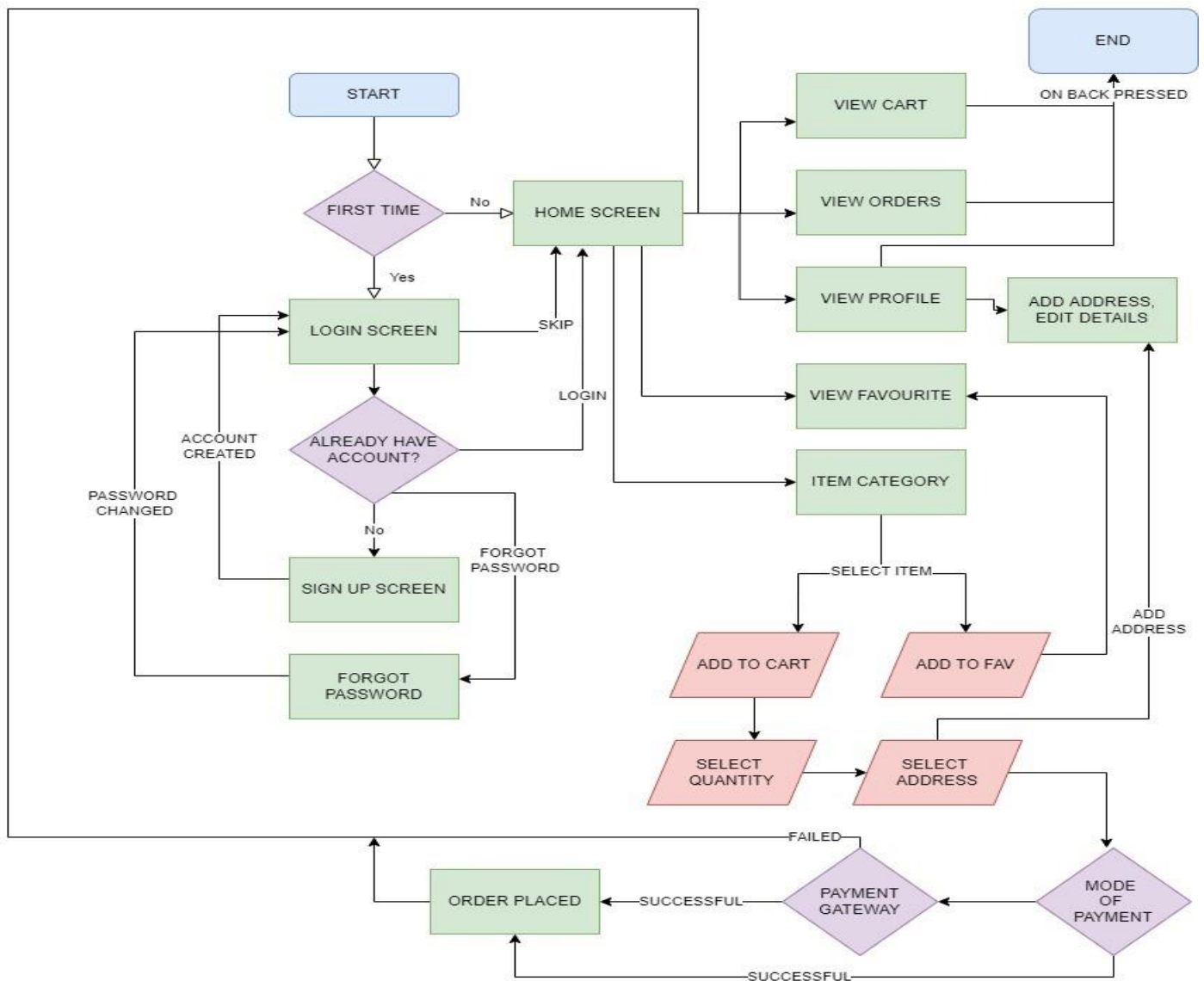


Figure 3: Flowchart of Application: Apna Country Pizza

Apna Country Pizza: Merchant Application is the application which concerns the merchant side of the application. This Application displays all the orders in four Category.

- ❖ **Live Orders:** These are the orders which is just ordered by the user, It will be displayed in the
- ❖ **Accepted orders :** These are the orders which are accepted and are ready to deliver. If by any concern the order cannot deliver, Merchant can cancel it. All the orders that are delivered will be transferred to delivered order and all the orders which are cancelled are transferred to cancelled order.
- ❖ **Delivered order:** All the order which are successfully delivered.
- Cancelled order: All the orders that are cancelled.
- ❖ **Cancelled order:** All the orders that are cancelled.

Live Order Screen of the Apna Country Pizza: Merchant Application . These orders can be Accepted or Declined by the will of Merchant, All the orders that are accepted will be then transferred to Accepted orders and the orders that are cancelled will be transferred to Cancelled order .

All the accepted orders can further be cancelled or delivered , all the delivered orders are visible in delivered order and all the Cancelled order will be visible in cancelled orders. The detailed workflow of the application is described in flowchart below.

After a successful login in the application the Merchant can see the orders which are live, orders can be accepted or cancelled accordingly

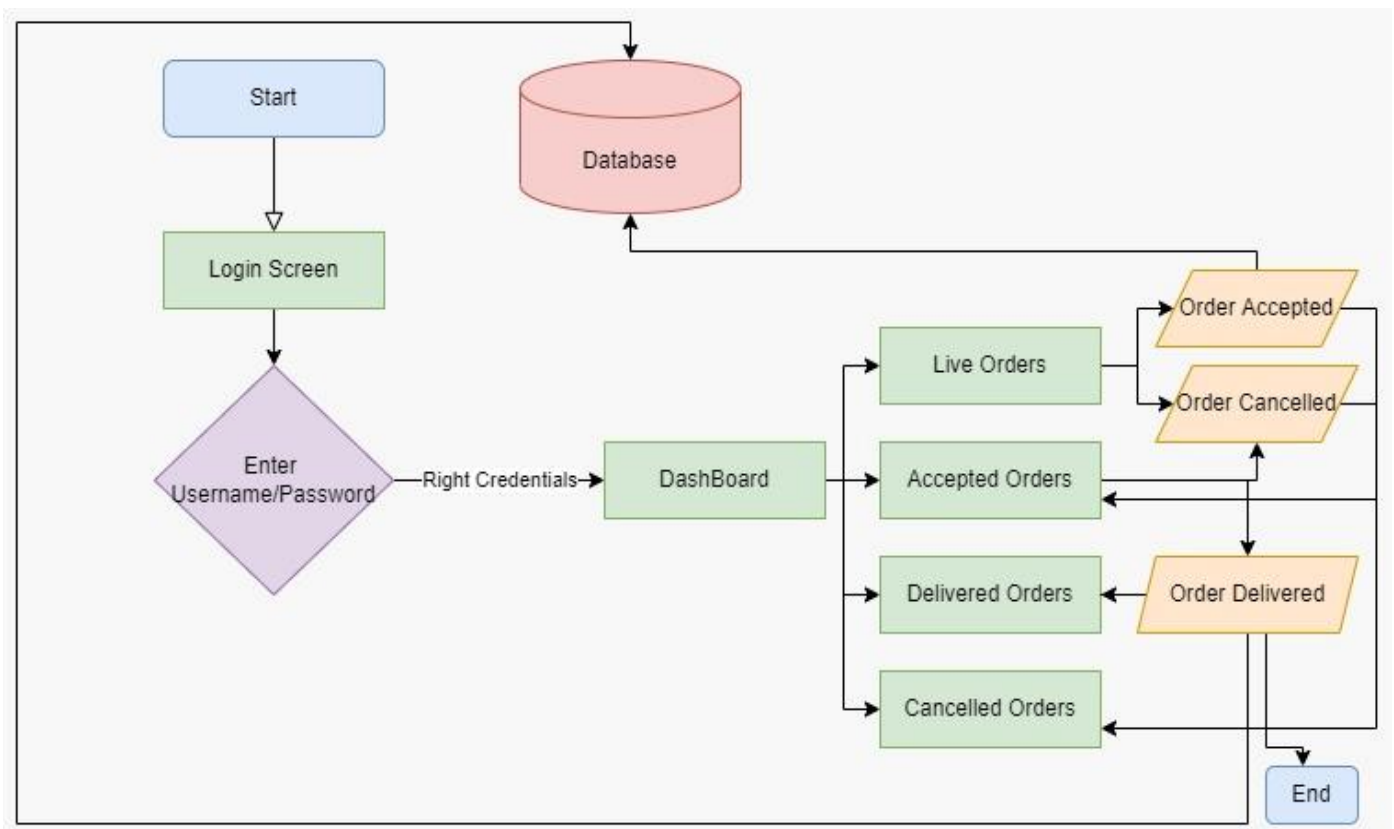


Figure 4 : Flowchart for Application APNA COUNTRY PIZZA : MERCHANT APPLICATION

**POSSIBLE UPDATES**

With all the research related to develop an application. There must be a topic which reflects some of the ideas which should be kept in mind for further updates. An update is an essential part of keeping an Android Application alive. Below mentioned are some update that could be added to the applications for better performance.

in order to make it more reliable and efficient.

- ❖ **Review System :** Review system could be added to provide feedback for the item purchased, this will be a better option if the quantity of user increases and user cannot provide Review to the Merchant personally.
- ❖ **30 Minute delivery System:** With proper

- ❖ **Introducing to cross-platform** : If the performance of the application is satisfactory to the merchant, then in order to increase the reach of the application. The application can be developed so that it can be used on multiple operating system.
- ❖ **Google Maps** : As initially the application is locally used, the Google Maps could be overrated for the application, but further if the application is used widely, Google Maps could be introduced

## CONCLUSION

With all the things mentioned about the research and requirements to develop the application. This research paper concludes with making a strong base to develop the application which intends to fulfil the need to expand businesses with online means. The application will deliver a great convenience to users as they can order Italian cuisines to their door steps. This paper also expands the knowledge about the use and development of Android Applications

discussion with Merchant, a 30 Minute delivery system could be introduced. This ensures the delivery of items in 30 minutes otherwise the amount will not be taken or returned if already paid.

## REFERENCES

- [1] Neil Smith (2017). Android Studio 3.0 Development Essentials - Android 8 Edition: O'Reilly Media'.
- [2] Megan Dennis (2018), "Native vs. Cross-Platform Apps – You'll Be the Winner" (March 18, 2022) from <https://www.zeolearn.com/magazine/native-vs-crossplatform-apps-youll-be-the-winner>.
- [3] David Griffiths and Dawn Griffiths (2015), Head First Android Development: A Brain-Friendly Guide: O'Reilly Media.
- [4] Ivan Demoski "Tips for Developing an Android App: My Lessons Learned" from.
- [5] [5]. N. Litayem, B. Dhupia, and S. Rubab, "Review of CrossPlatforms for Mobile Learning Application Development," from <https://www.toptal.com/android/making-an-android-applessons-learned>.
- [6] Anupam Chugh, Android Activity Lifecycle – 7 Stages and Functions from <https://www.journaldev.com/53/androidactivity-lifecycle>.