

1 Abstract :

Our populations are increasing day by day. For this, we encounter many problems like traffic jam, overcrowd in markets, restaurants. People from different departments are thinking various projects to overcome this problem and make our daily life easier to lead. We from the department of computer science thought of a multipurpose app which can reduce the problems like overcrowd at markets and restaurant, traffic jam out of nowhere very much. In our proposed android app, we can call cars, bikes for transport, order food and deliver our parcels to desired destination while staying home. The main purpose of our app is to do the daily necessities of our life through the app. To build this app, we used open source software named android studio where we could code in java, Firebase database and OneSignal notification service.

2 Introduction :

Now a days people search for comfortness in their daily life. Taking consideration of present population problems and a way to make peoples life easier many developers are thinking of various multipurpose app to build. We are also one of them who wants to make a multipurpose app which could change the way we live until now. This android app would help us to transport one place to another by calling cars, bikes and order our food from restaurants and deliver our parcels to specific destination. We could also pay directly from the app. It will help us choose our favourite company services for parcel delivery. Food delivery would be very fast and smooth with this android app that we will built.



Figure 1: Extreme android app.

3 Methodology :

3.1 Android studio :

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps.

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- 1.Android app modules
- 2.Library modules
- 3.Google App Engine modules

By default, Android Studio displays your project files in the Android project

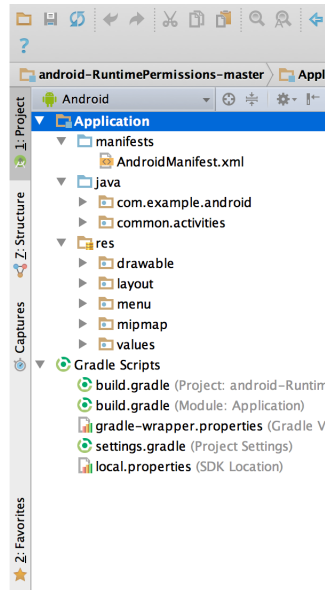


Figure 2: The project files in Android view.

view, as shown in figure 2. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

1. manifests : Contains the AndroidManifest.xml file. Request the various permission of app needs like Location, Internet connectivity, Call Permission etc in declare there. All the Activity also registered in the android Manifest.xml file.

2. java : Contains the Java source code files, including JUnit test code. Here we implement the functionality of our extreme app.
3. res : Contains all non-code resources, such as XML layouts, UI design, strings, and bitmap images etc. Layout is hold layout of all activity, menu is hold menu layout, drawable is hold all the image and vector image resource, values store String.xml, style.xml and color.xml file to develop the complete eXtreme project.

3.2 Firebase Real Time Database :

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.

Firebase Services can be divided into two groups:

I. Develop and App testing :

1. Realtime Database
2. Auth
3. Test Lab
4. Crashlytics
5. Cloud Functions
6. Firestore
7. Cloud Storage
8. Performance Monitoring
9. Crash Reporting
10. Hosting



Figure 3: Firebase services.

II. Grow and Engaging audience :

1. Firebase Analytics
2. Invites
3. Cloud Messaging
4. Predictions
5. AdMob
6. Dynamic Links
7. Adwords
8. Remote Config
9. App Indexing

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync between your users in realtime.

The Realtime Database is really just one big JSON object that the developers can manage in realtime. Our firebase database is given in FIGURE 4 where we stored our all data of the extreme app.

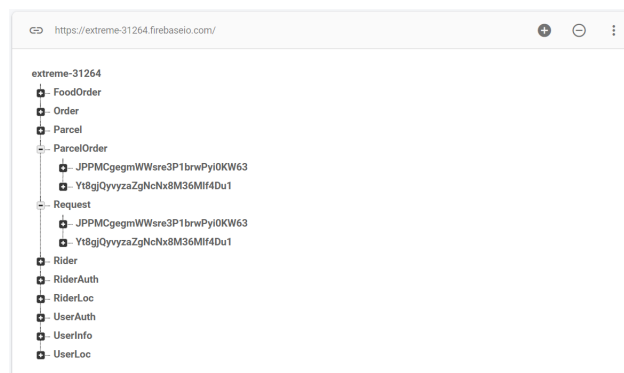


Figure 4: Extreme app main database.

Userinfo, Riderinfo, Foodorder, parcel database are given in the FIGURE 5, 6, 7, 8.



Figure 5: Userinfo database.

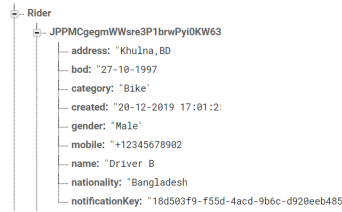


Figure 6: Riderinfo database.



Figure 7: Foodorder database.

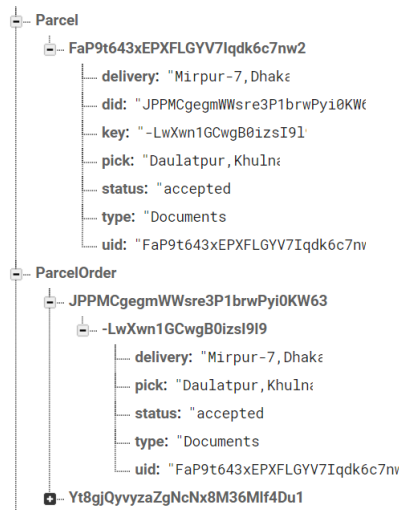


Figure 8: Parcel database.

3.3 Google Map API :

Google map api has been used for adding map features to our extreme app. This api is free version so advanced map functions couldn't be used in the extreme app. Using map in our app for biker, foodorder, parcel is shown in following FIGURE 9 and for car driver FIGURE 10.

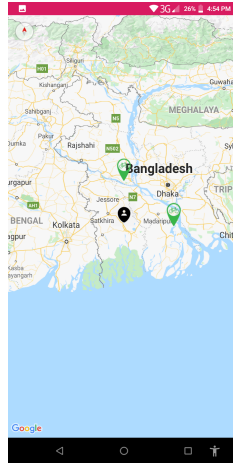


Figure 9: Google map for bike transportation.

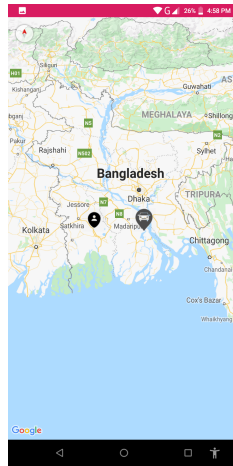


Figure 10: Google map for car transportation.

3.4 OneSignal Notification Service API :

OneSignal is a high volume and reliable push notification service for websites and mobile applications. We support all major native and mobile platforms by providing dedicated SDKs for each platform, a RESTful server API, and an online dashboard for marketers to design and send push notifications. It is used for notifications to biker and car driver for sending request and also for showing that the request has been accepted. Notification is shown in FIGURE 11, 12, 13.

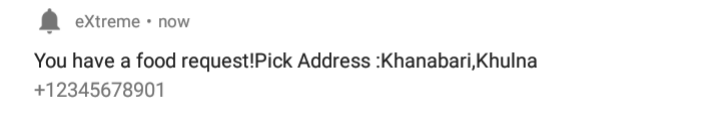


Figure 11: Food Request Notification.

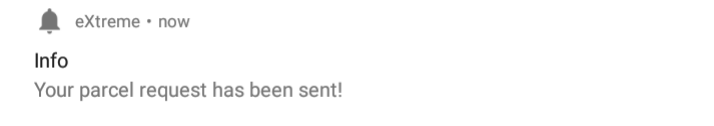


Figure 12: Parcel Request Notification.

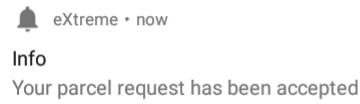


Figure 13: Parcel Accept Notification.

4 Work Flow :

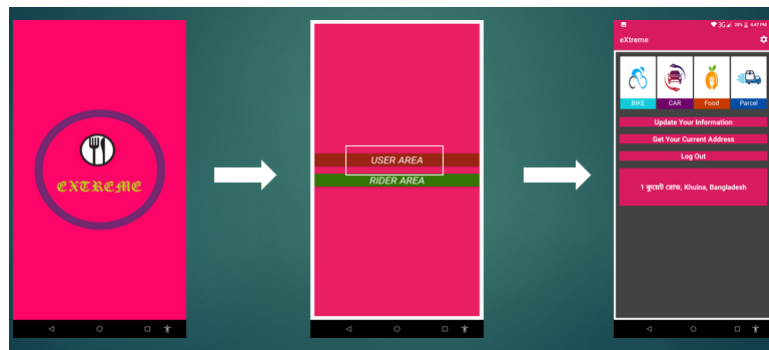


Figure 14: Work Flow 1.

In Our Application,After the loading screen it shows two module.Ex: User Area and Rider Area.In User Area,there are four sub-module like Bike,Car,Food and Parcel.And also included some basic functionality for user information.

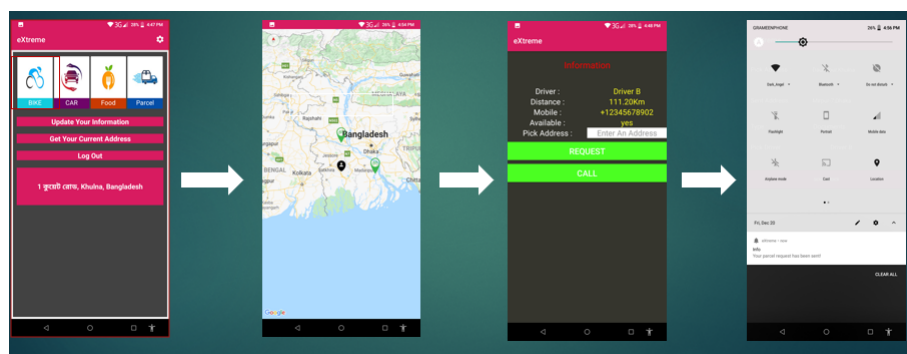


Figure 15: Work Flow 2.

Inside of Bike Sub-module it is showing up a map with Biker Rider marker and also user's current address marker.If user select any rider marker then we

consider the user may select the rider and Map activity transferred to request rider activity. Request Rider Activity have two option one is directly call the rider and other is request the rider through our application by clicking Request. After clicking Request a notification send to the rider phone and store the request in database for rider confirmation. If rider accept or reject the request then user again notified by a notification.

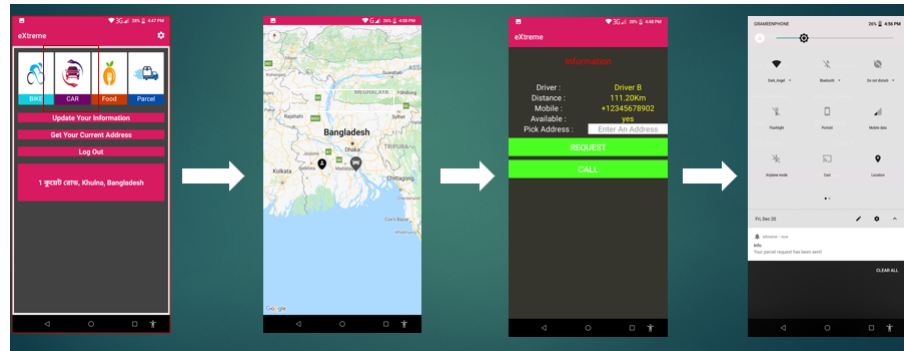


Figure 16: Work Flow 3.

Inside of Car Sub-module it is showing up a map with Rider marker and also user's current address marker. If user select any rider marker then we consider the user may select the rider and Map activity transferred to request rider activity. Request Rider Activity have two option one is directly call the rider and other is request the rider through our application by clicking Request. After clicking Request a notification send to the rider phone and store the request in database for rider confirmation. If rider accept or reject the request then user again notified by a notification.

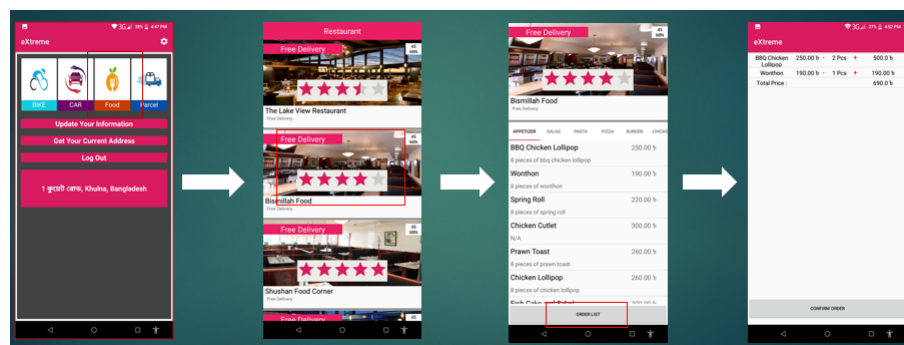


Figure 17: Work Flow 4.

Inside of Food Sub-module, There are around twenty hotel list and each of hotel list has many more items of food with various category. In food list after

clicking the item it is added to add cart option. After choosing a hotel, only one hotel can be ordered at a time. If any of pending food order stay in your order then no more request can be perform. After clicking Order List in below button then it's show up your all of add cart food order. You can also maximize or minimize item and and remove the item from list. Finally in Confirm order Section, It is forwarded to a map for selecting driver to delivery. After selecting the driver, you can now input your delivery address and confirm your order. Then the order will be sent to the selecting driver and the driver will be notified by a notification. After confirming the order by driver, user is also notified as the way.

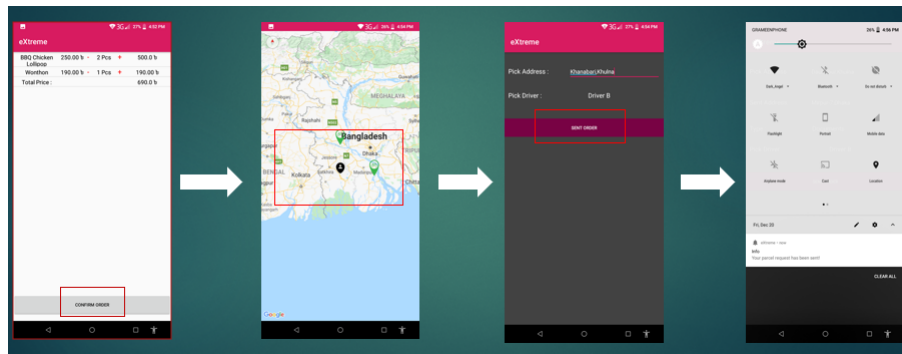


Figure 18: Work Flow 5.

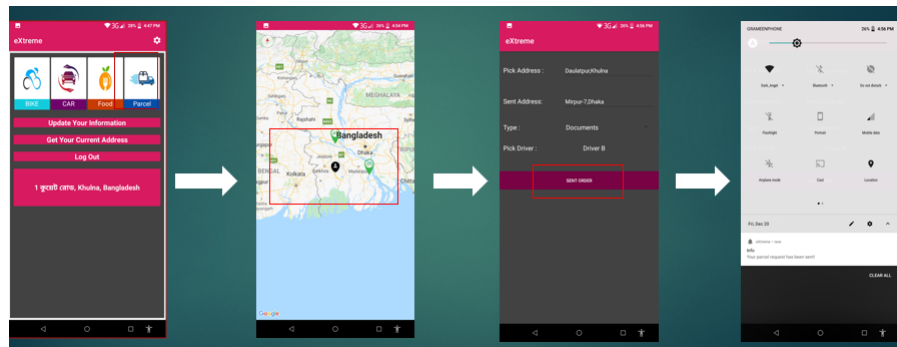


Figure 19: Work Flow 6.

Inside of Parcel Sub-module, it was forwarded to a map for selecting driver and after selecting the driver it will be forwarded to Request page. Now You can select your parcel type, pick address and delivery address to confirm the request. Then the request has been sent to the specific driver for picking your parcel and deliver to it your entered destination. Also notification system has been same as well as bike and car sub-module.

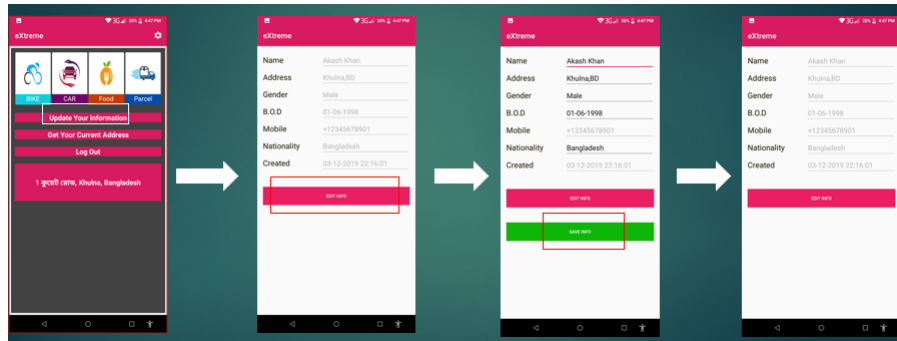


Figure 20: Work Flow 7.

In Update Your Information, user can update and save their information for clearing the identification of his/her own. After clicking the Edit Button, it will be editable of his/her basic information but in the number section it can't be changed and created date also auto generated. After saving the information, it will be stored in Firebase database.

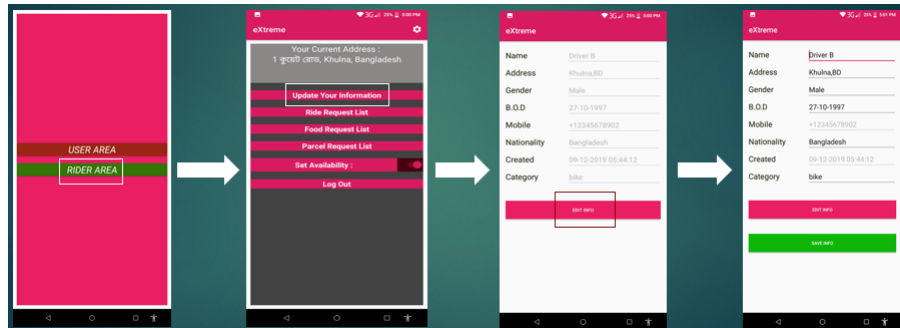


Figure 21: Work Flow 8.

In Rider Area, there are update information of rider as the same as User Information Section.

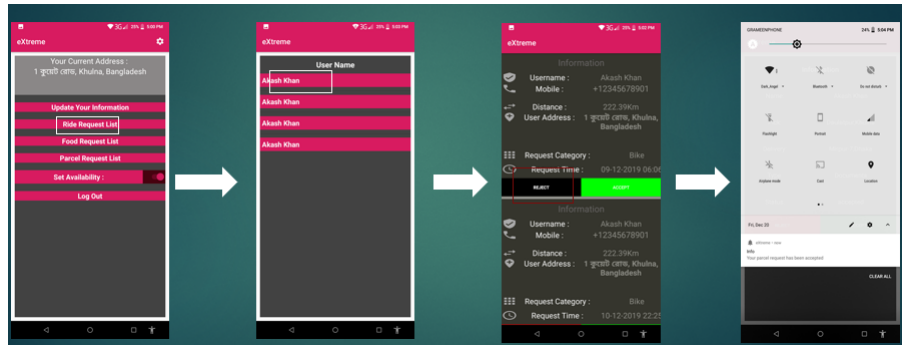


Figure 22: Work Flow 9.

In Rider Request List, there are multiple user name list who are requested for ride. After selecting the user name, rider can see their information and also accept and reject request button to accept the request either reject the request and this event is also for only one action and can't be undone.

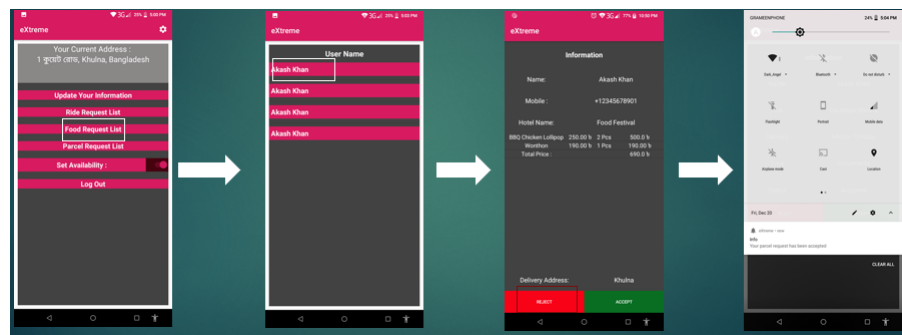


Figure 23: Work Flow 10.

In Food Request List, there also a user name list and after clicking the name rider can preview the food item list, total cost, hotel name and user phone number for contact. And also request accept and reject capability for rider and notification system as before mentioned in User Area.

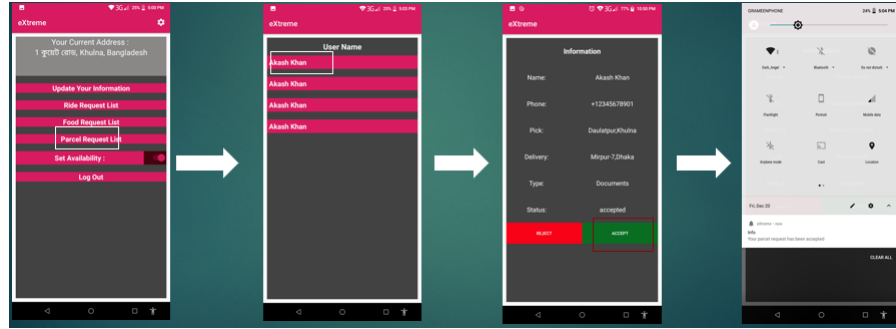


Figure 24: Work Flow 11.

In Parcel Request List, there is also a user name list and after clicking the name rider can preview the parcel type, source and destination, user mobile number for contact. And also request accept and reject capability for rider and notification system as before mentioned in User Area.

5 Discussion :

5.1 Limitation :

One of the main limitations is that we couldn't implement a direction path in Google Maps because it is a paid API. Then, the payment method is not implemented in our extreme app. We could not measure distance because we could not buy the Google API. For this, we could not calculate the fare of the transport system. In Restaurant, we could not add all the restaurants because of lacking the JSON for restaurants. That's why there are many restaurants that could not be added. Food's in all restaurants also need to be updated manually. In parcel, we also have lacking in adding particular company service.

5.2 Future Plan :

We want to add the direction path in Google Maps so that we can measure the distance and can calculate the fare. We want to add all restaurants' JSON data so that we can be active all the time and users can pay directly from the app. We want to add all the parcel company services to the app so that users can choose their desired company while using parcel delivery. With this, we can make our app better user friendly.

5.3 Conclusion :

We could resolve our population problem with this app. If we could achieve our goals and gain funds to launch this app in the Google App Store. We hope many people will use this app to get comfort in their daily life. Because an app

which could call transportation any time, order food in restaurants open time and deliver parcel through users favourite company services will surely change the life of a person and make their life easier if it is only a little bit.

6 References :

- * Course : Advanced JAVA Programming Laboratory
- * Firebase Documentation
- * OneSignal Documentation
- * Google Map API Documentation