SMART INDIA
HACKATHON 2019

MIT Academy of Engineering, Pune

GROUP NAME: The_Eklavayas

ABSTRACT

This document is the report on the Mobile application titled as "Fruitful". It consists of all the data related to process of building the mobile application. The application is developed in Android studio. It includes a variety of custom tools that helped us develop mobile application on the android platform.

We all have learn many new things, right from idea building, its virtual implementation on paper that is low fidelity, developing a prototype in proto.io so as to rectify the problem in final application and finally developing an mobile application using android studio and Firebase real-time database system.

SYSTEM REQUIREMENT SPECIFICATION(SyRS)

INTRODUCTION:

- **BACKGROUND:**

  Lot of farmers waste a lot of time in search of tractors and other mechanizations. Also owners are in loss because they aren't able to find customers for their mechanization.
  So application needs to be developed which will rectify this problem of easy communication between owners and farmers.

- **PURPOSE:**

  The application is being developed keeping in mind the needs of the farmers of the rural parts of the country. Each one need tractor and other mechanization related to it very often and have to go searching for them to take it on rent. They often end up with disappointment and a lot of searching work being wasted. This centralized application will end the problem.

- **SCOPE:**

  With digitalization and modernization of agriculture in the country, farmers and other users will find this app very useful at times. They wouldn't have to rush and contact people for tractors and their booking. The scope of the use is very high as the requirement is big and this application provides an easy, effective and dependable way to fulfill the requirements.

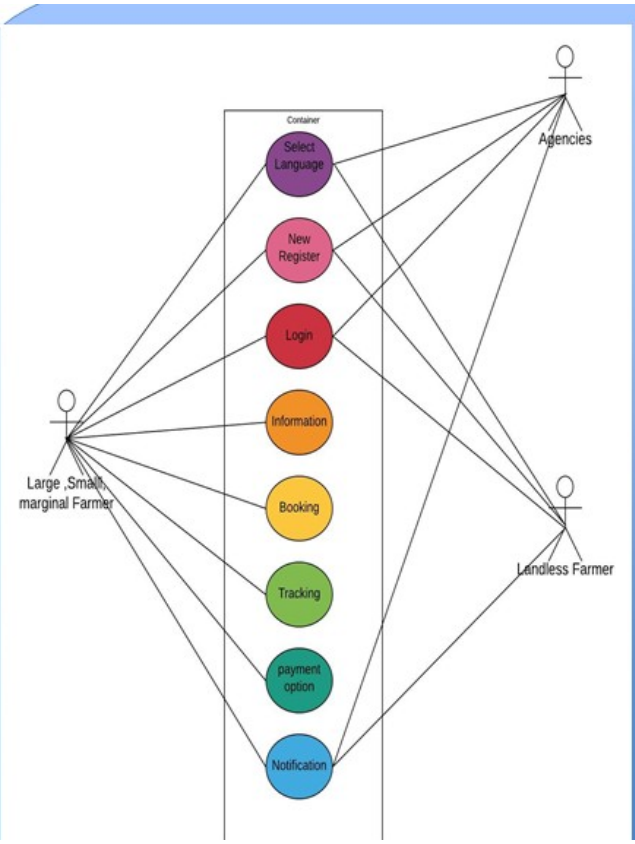
OVERALL DESCRIPTION:

The application will have three main categories of the beneficiers that are the owners ,users and drivers. The owners have to register their vehicles, upload their related documents and receive request for their tractor. The users will have to login, register for new user filter or search the available vehicles according to their requirements. After this they will subsequently book the vehicle and pay online or also offline. Inshort this is simple mobile application which can be easily used by Farmers for efficient farming.

AEIOU

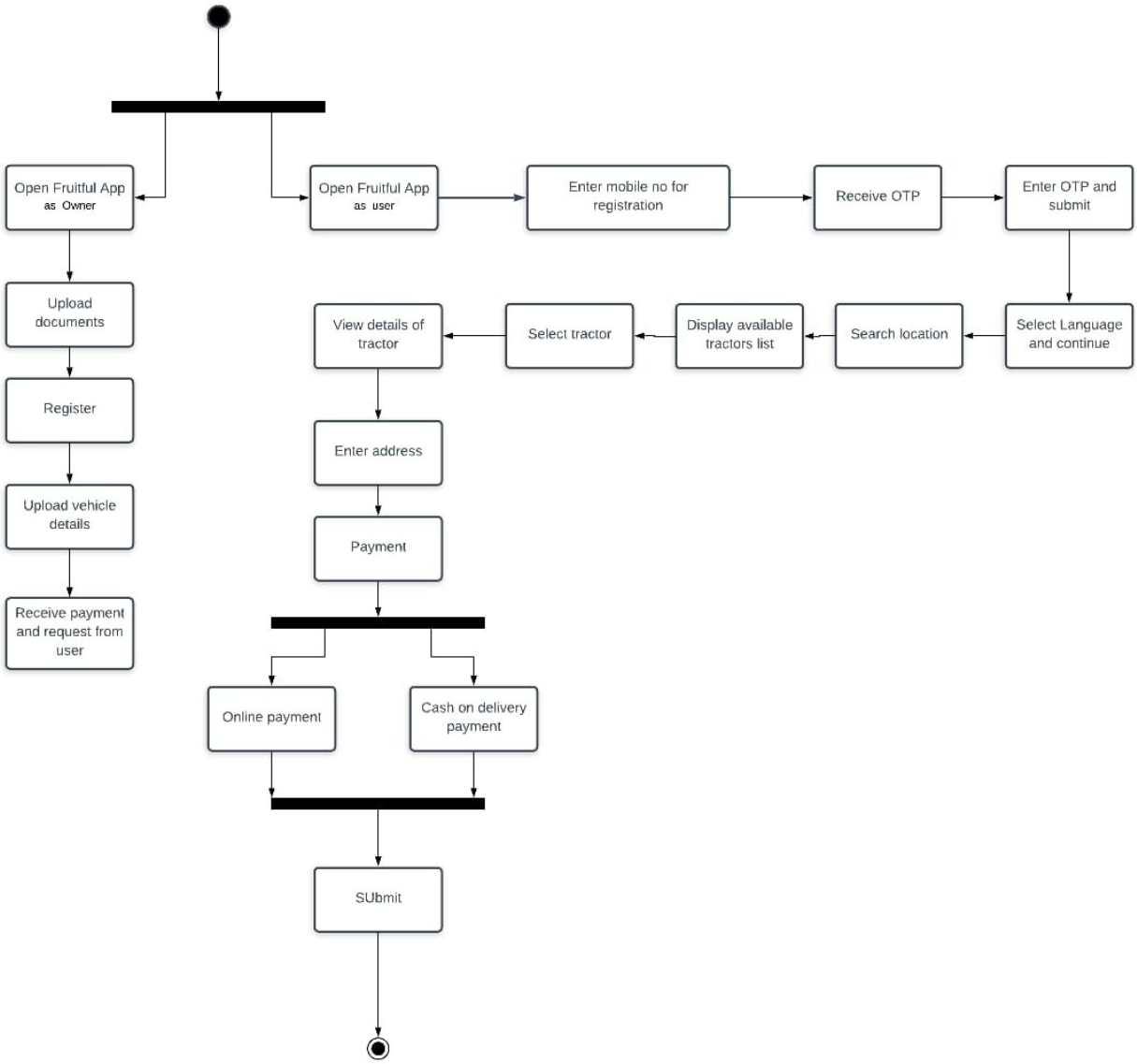
Activity	Renting, borrowing, maintenance, farming
Environment	Rural area-Connectivity problem
Interaction	User and tractor, user and application, owner and application, driver and application, user and owner, user and tractor, user and driver.
Object	Mobile Phone, application, tractors
User	Farmer, owner, driver, server

USE-CASE DIAGRAM



User Scenarios :
Language Selection, registration, Login, Information Gathering, Booking for Tractors and mechanization related to it, Tracking for Place where agencies are available, Payment and notification .
Users:
Large, small, marginal farmers, Landless farmers, Agencies(owner),driver, Tractor and other Mechanization owners.
Rolls:
Agencies will provide availability of mechanization and will be notified after booking process.
Farmers will be able to book, track, login, gather info, get notification and pay for mechanization.

ACTIVITY DIAGRAM



TECHNOLOGY USED:

Android Studio:

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

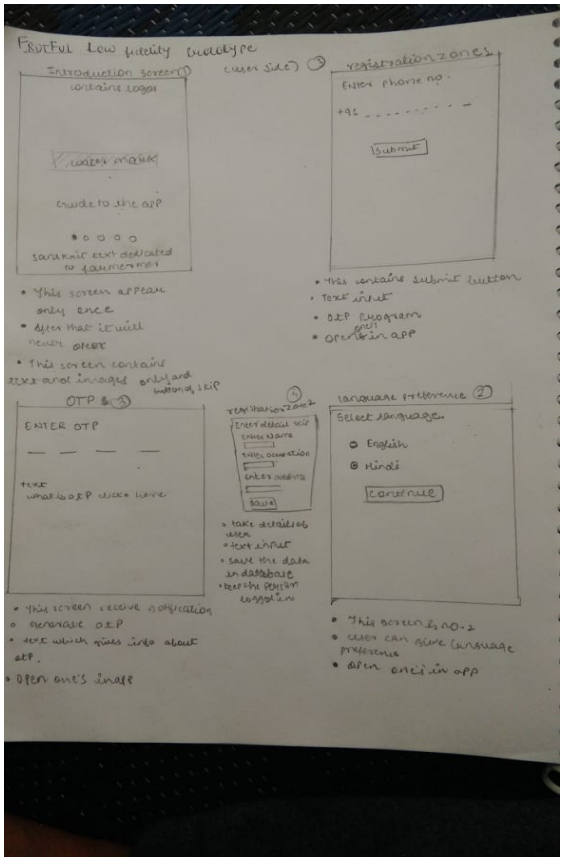
Proto.io :

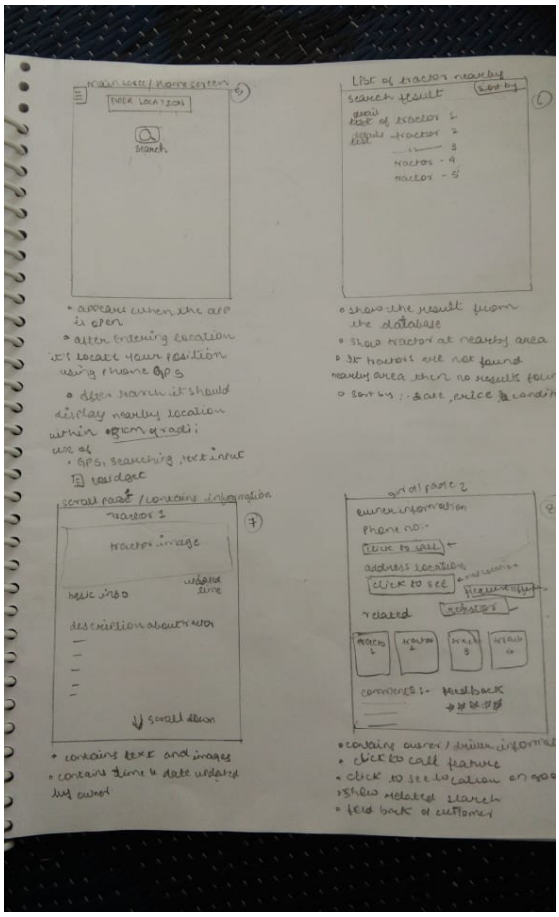
Proto.io is an application prototyping platform launched in 2011 and developed by the Labs Division of SNQ Digital. Originally designed to prototype on mobile devices, Proto.io has expanded to allow users to prototype apps for anything with a screen interface, including Smart TV's, digital camera interfaces, cars, airplanes, and gaming consoles. Proto.io utilizes a drag and drop user interface (UI) and does not require coding.

Firebase Realtime Database:

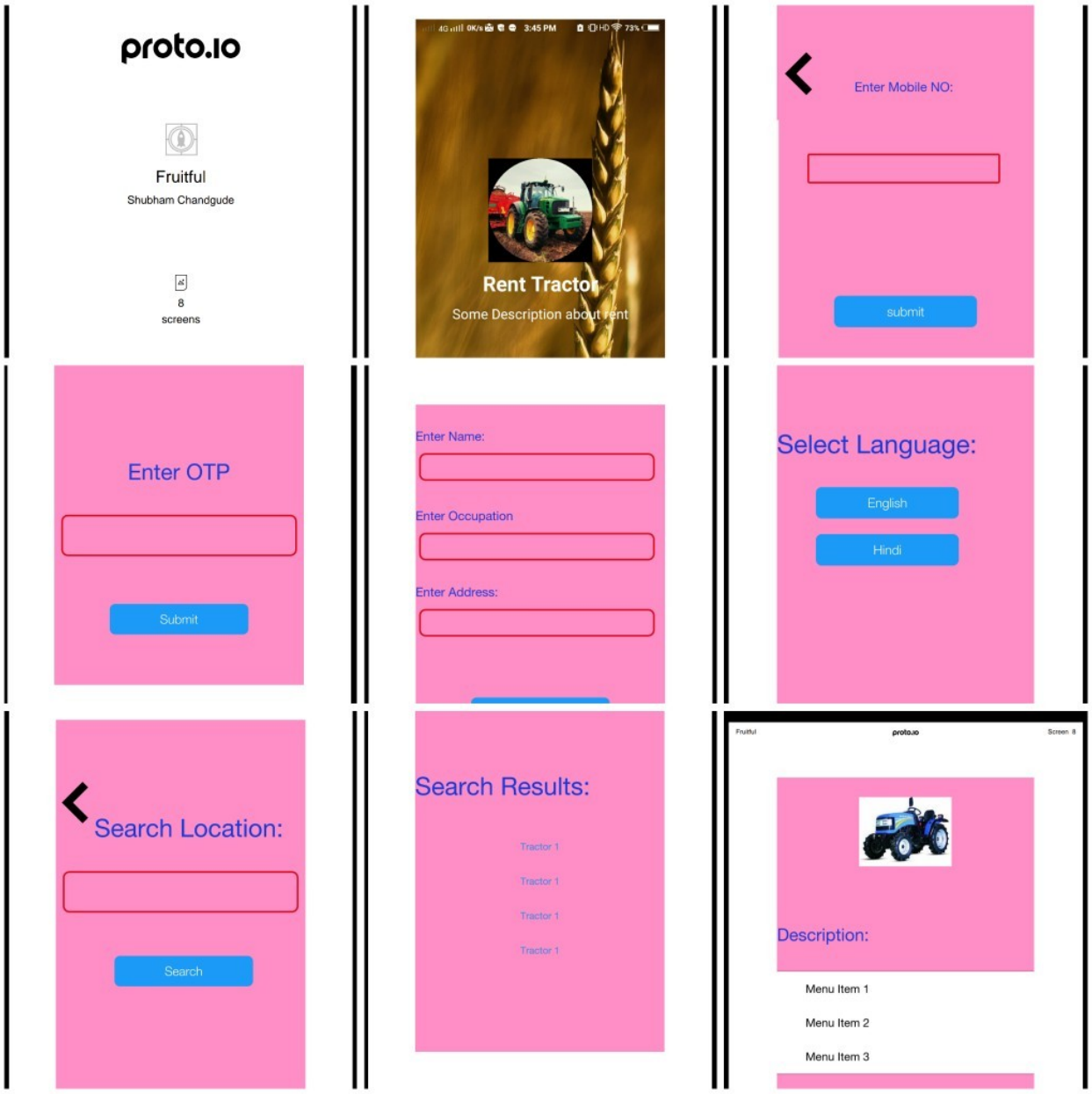
It Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.
The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build 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.

Low fidelity:

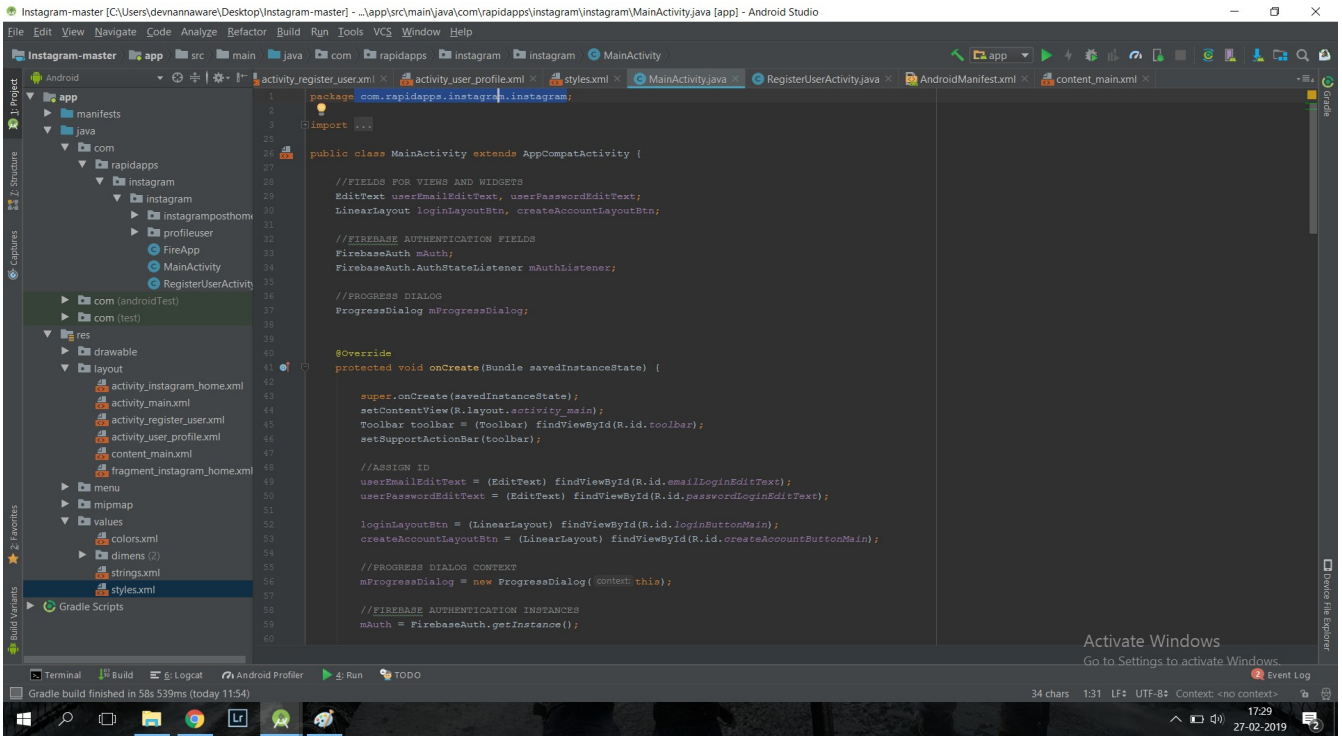


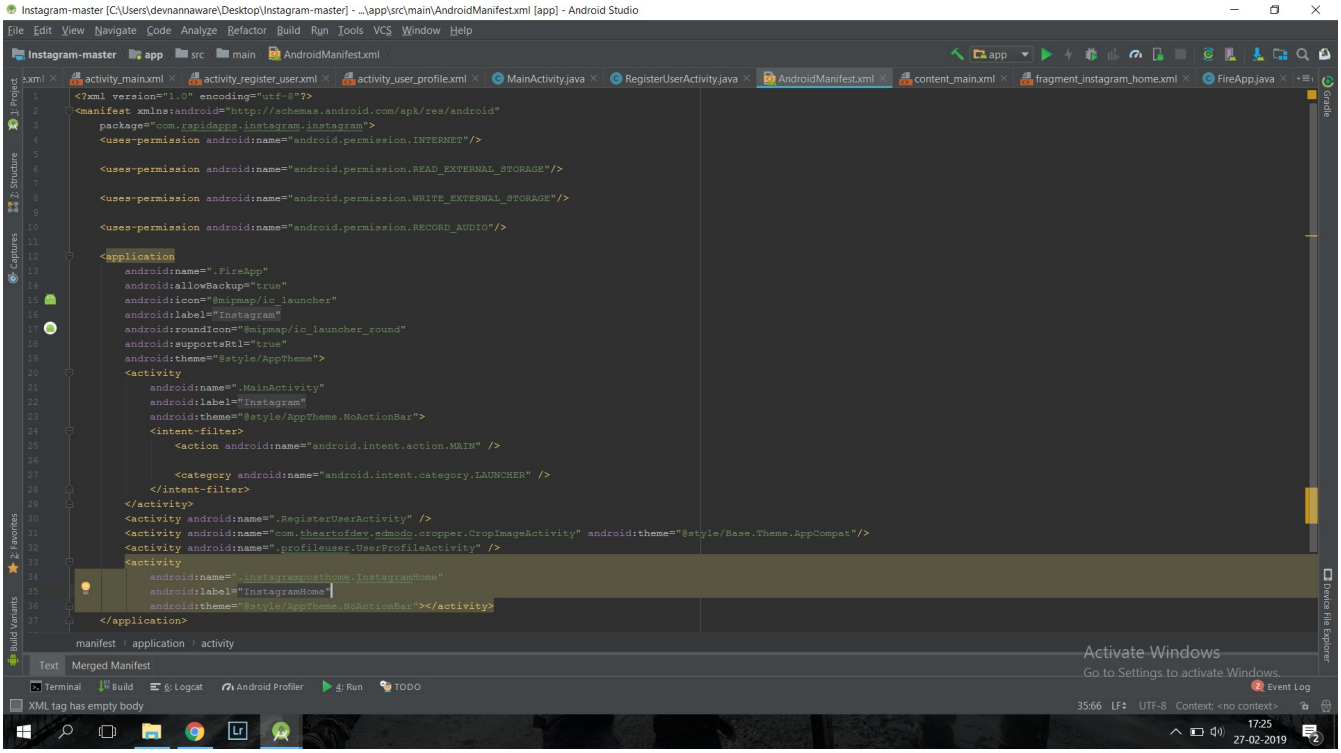


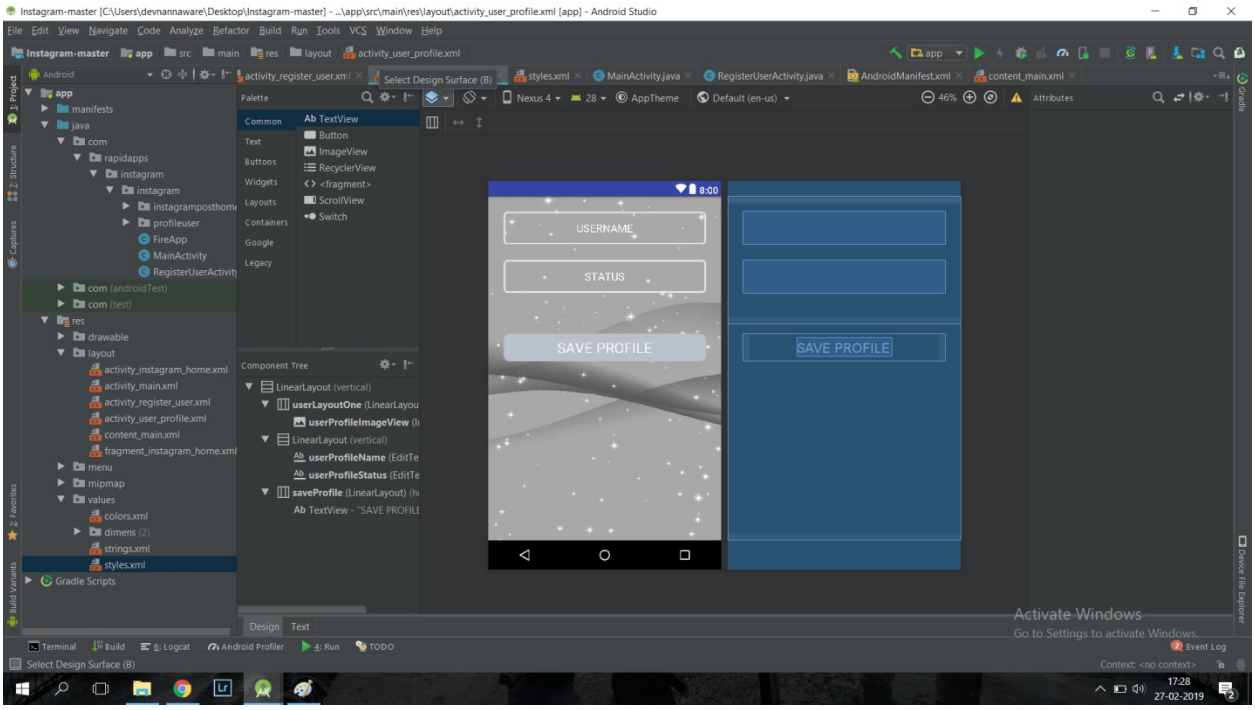
Proto.io:


proto.io
Fruitful
Shubham Chandgude
8
screens
Rent Tractor
Some Description about rent
Enter Mobile NO:
submit
Enter OTP
Submit
Enter Name:
Enter Occupation
Enter Address:
Select Language:
English
Hindi
Search Location:
Search
Search Results:
Tractor 1
Tractor 1
Tractor 1
Tractor 1
Fruitful
proto.io
Screen 8
Description:
Menu Item 1
Menu Item 2
Menu Item 3

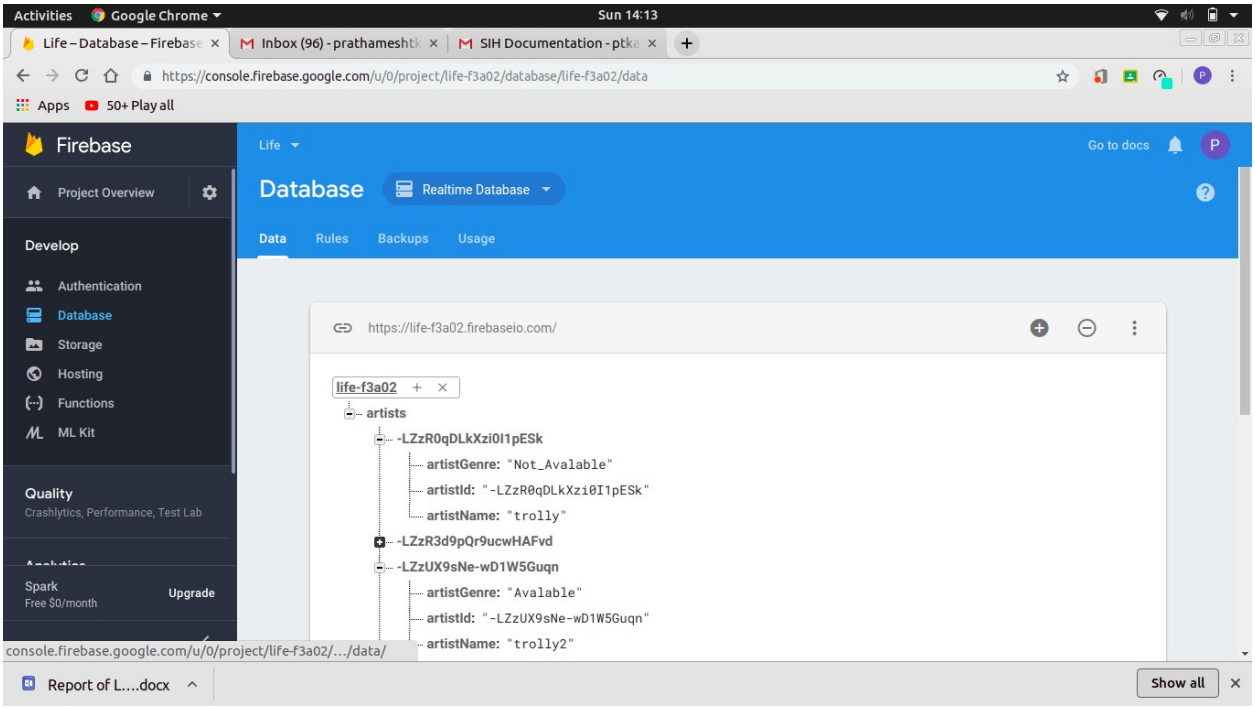
OVERVIEW:


Instagram-master [C:\Users\devnannaware\Desktop\Instagram-master] - ...\app\src\main\java\com\rapidapps\instagram\instagram\MainActivity.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_register_user.xml
activity_user_profile.xml
styles.xml
MainActivity.java
RegisterUserActivity.java
AndroidManifest.xml
content_main.xml
package com.rapidapps.instagram.instagram;
import ...
public class MainActivity extends AppCompatActivity {
//FIELDS FOR VIEWS AND WIDGETS
EditText userEmailEditText, userPasswordEditText;
LinearLayout loginLayoutBtn, createAccountLayoutBtn;
//FIREBASE AUTHENTICATION FIELDS
FirebaseAuth mAuth;
FirebaseAuth.AuthStateListener mAuthListener;
//PROGRESS DIALOG
ProgressDialog mProgressDialog;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
//ASSIGN ID
userEmailEditText = (EditText) findViewById(R.id.emailLoginEditText);
userPasswordEditText = (EditText) findViewById(R.id.passwordLoginEditText);
loginLayoutBtn = (LinearLayout) findViewById(R.id.loginButtonMain);
createAccountLayoutBtn = (LinearLayout) findViewById(R.id.createAccountButtonMain);
//PROGRESS DIALOG CONTEXT
mProgressDialog = new ProgressDialog(context: this);
//FIREBASE AUTHENTICATION INSTANCES
mAuth = FirebaseAuth.getInstance();
Activate Windows
Go to Settings to activate Windows.
Gradle build finished in 58s 539ms (today 11:54)
17:29
27-02-2019


Instagram-master [C:\Users\devnannaware\Desktop\Instagram-master] - ...\app\src\main\AndroidManifest.xml [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_main.xml
activity_register_user.xml
activity_user_profile.xml
MainActivity.java
RegisterUserActivity.java
AndroidManifest.xml
content_main.xml
fragment_instagram_home.xml
FireApp.java
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.rapidapps.instagram.instagram">
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<application
android:name=".FireApp"
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="Instagram"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity
android:name=".MainActivity"
android:label="Instagram"
android:theme="@style/AppTheme.NoActionBar">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity android:name=".RegisterUserActivity" />
<activity android:name="com.theartofdev.edmodo.cropper.CropImageActivity" android:theme="@style/Base.Theme.AppCompat"/>
<activity android:name=".profileuser.UserProfileActivity" />
<activity
android:name=".instagramposthome.InstagramHome"
android:label="InstagramHome"
android:theme="@style/AppTheme.NoActionBar"></activity>
</application>
manifest application activity
Text Merged Manifest
Terminal Build 6: Logcat Android Profiler 4: Run TODO
Event Log
XML tag has empty body
35:66 LF UTF-8 Context: <no context>
Activate Windows
Go to Settings to activate Windows.
17:25
27-02-2019


Instagram-master [C:\Users\devnannaware\Desktop\Instagram-master] - ...\app\src\main\res\layout\activity_user_profile.xml [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Instagram-master app src main res layout activity_user_profile.xml
Android
app
manifests
java
com
rapidapps
instagram
instagram
instagramposthome
profileuser
FireApp
MainActivity
RegisterUserActivity
com (androidTest)
com (test)
res
drawable
layout
activity_instagram_home.xml
activity_main.xml
activity_register_user.xml
activity_user_profile.xml
content_main.xml
fragment_instagram_home.xml
menu
mipmap
values
colors.xml
dimens (2)
strings.xml
styles.xml
Gradle Scripts
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Component Tree
LinearLayout (vertical)
userLayoutOne (LinearLayou
userProfileImageView
LinearLayout (vertical)
userProfileName
userProfileStatus
saveProfile (LinearLayout)
Ab TextView - "SAVE PROFILE
Select Design Surface (B)
styles.xml
MainActivity.java
RegisterUserActivity.java
AndroidManifest.xml
content_main.xml
Nexus 4
28
AppTheme
Default (en-us)
46%
Attributes
8:00
USERNAME
STATUS
SAVE PROFILE
SAVE PROFILE
Design Text
Terminal Build 6: Logcat Android Profiler 4: Run TODO
Event Log
Select Design Surface (B)
Context: <no context>
Activate Windows
Go to Settings to activate Windows.
17:28
27-02-2019

Firebase Database:-



ADVANTAGES OF THIS APPLICATION:

- **The application is easy to use by both farmers and owner.**
- **Time saving as farmers don't need to roam around in search of tractors.**
- **Easy access to the application as regional languages are available. So users can use the language in which they are comfortable to understand the working of application.**
- **Farmers can hire tractors from nearby places. According to location of farmer the application provides nearby locations where tractors are available.**
- **Full description about tractors and owners are provided in the application. Owners can easily add new mechanization in the list or menu.**
- **According to the crops chosen by farmers they will get additional Information about seeds,fertilizers and temperature. Also other mechanizations which are needed for that crop are available for renting.**
- **If the price of tractor is not in budget of the farmers we have an option of negotiatiation where farmers can request the owners to rent the tractor in their budget.**

CONTRAINTS:

- **Basic handling of mobile and application should be known.**
- **Net availability.**

SURVEY QUESTIONS:

Farmers:

1)What problems do you face while taking tractors on rent?

2)Do you use any online application for this purpose?

3)What problems do you face using an online application for this purpose?

4)Do you get tractors of your choice and in your budget?

Owners:

1)Is it difficult for you to find customers for renting your tractor?

2)How do you handle the the misuse or damage caused to the tractor or what care do you take while giving tractor on rent?

CONCLUSION:

In this journey of designing an application we all went through various steps of development which made us aware of various processes involved in developing mobile application. We were exposed to learn about android studio and also we used firebase real-time database system.