Report on

AllyCabs : Book a Cab in a Sec

Submitted as part of

ITE1016 – Mobile Application Development

J-Component

Under the guidance of

Prof. Puviarasi G

Submitted By-

Nishant Mehta(15BIT0233)

Slot – A2+TA2

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

VIT®

UNIVERSITY

(Estd. u/s 3 of UGC Act 1956)

VELLORE ■ CHENNAI

www.vit.ac.in

Fall Semester 2017-2018

UNDERTAKING

This is to declare that the project entitled “AllyCabs : Book a Cab in a Sec” is an original work done by undersigned, in partial fulfillment of the requirements for the degree “BTech in Information Technology – Mobile Application Development” at School of Information Technology and Engineering, VIT University.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Nishant Mehta(15BIT0233)

Abstract

My project entitled “AllyCabs: Book a cab in a sec ” is made from two words i.e. Ally + Cabs where Ally means Friend and cab means taxi. My project aim is to book the taxis at all the fare charges. Manual system that is employed is extremely laborious and quite inadequate. It only makes the process more difficult and hard. The aim of my project is to develop a system that is meant to partially computerize the work performed in the prepaid taxi management system like generating bookings , record of routes available , fare charges of every route; store record of the customer.

Introduction of the Project - AllyCabs :-

A taxicab, also taxi or cab, is a type of vehicle for hire with a driver, used by a single passenger or small group of passengers often for a non-shared ride. A taxicab conveys passengers between locations of their choice. In modes of public transport, the pick-up and drop-off locations are determined by the service provider, not by the passenger, although demand and share taxis provide a hybrid bus/taxi mode. Taxicabs arrived in 1911 to complement horse wagons. According to Government of India regulations, all taxicabs are required to have a fare-meter installed. However, enforcement by authorities is lax and many cabs operate either without fare-meter or with defunct ones. In such cases, fare is decided by bargaining between the customer and the driver. Taxicabs face stiff competition from auto rickshaws but in some cities. In India, most taxicabs, especially those in Delhi and Mumbai, have distinctive black and yellow liveries with the bottom half painted black and upper half painted yellow. In Kolkata, most taxis are painted yellow with a blue strip in the middle.

Taxi Cabs - In cities and localities where taxis are expensive or do not ply as per the government or municipal regulated fares, people use Share taxis. These are normal taxis which carry one or more passengers travelling to destinations either a route to the final destination, or nearby the final destination. The passengers are charged according to the number of people with different destinations. A similar system exists for auto rickshaws, known as Share autos. As one example, "Shared taxis" - and known just as that – have been operating in Mumbai, India, since the early 1970s. These are more like a point to point service that operates only during the peak hours. During off peak hours, they ply just like the regular taxis, can be hailed anywhere on the roads, and passengers are charged by the meter. But in order to bridge the gap between

demand and supply, during peak hours, several of them operate as Shared Taxis, taking a full cab load of passengers to a more or less common destination. The pick-up points for these taxis are fixed, and are marked by a post that says, "Shared Taxis" and cabs line up at this point during peak hours. They display the general destination they are headed for on their windscreens, and passengers just get in and wait for the cab to fill up. As soon as this happens - which takes less than a couple of minutes – the cab moves off. Fares are a fixed amount – fixed between the Taxi Unions and the authorities for the point to point distance - and are far lower than the metered fare to the same destination, but higher than the bus or train fare. Time taken is obviously much less than that by bus. These taxis are very popular because of the lack of waiting time, faster journey speeds, greater comfort, and absence of the crush loads of peak hour commuter traffic in buses and trains.

System Design

FEATURES:

- **User Registration/ Login :** User even either Customer or Driver have to make a login into the app and then proceed with using the desired features. Login would be made available with help of firebase Authentication.
- **Live Map:** This section will consist of live Map loading for different countries.
- **Customer :** Customer can login and even set his details in the Settings and can request for the cab.
- **Driver :** When the Driver logins into the app, he can see the point mark on the app where any customer needs a cab.
- **Pickup Request:** The Customer can make a request for the cab and even he can take the option for the car type.
- **Logout :** If a user wants to logout the app after the work is done , then he will be able to logout from the app and another person may login using the same device.

MODULES:

1) Sign Up Registration Form
2) Live Map
3) Login

4) Customer Request
5) Driver Notification
6) Driver Settings
7) Customer Settings
8) Logout

Source Code -- >

MainActivity.java

```
package com.nishantmehta.allycabs;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    private Button mDriver, mCustomer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mDriver = (Button) findViewById(R.id.driver);
        mCustomer = (Button) findViewById(R.id.customer);

        mDriver.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
DriverLoginActivity.class);
                startActivity(intent);
                finish();
                return;
            }
        });

        mCustomer.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
CustomerLoginActivity.class);
                startActivity(intent);
                finish();
                return;
            }
        });
    }
}
```

DriverLoginActivity.java

```
package com.nishantmehta.allycabs;

import android.content.Intent;
import android.support.annotation.NonNull;
```

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class DriverLoginActivity extends AppCompatActivity {
    private EditText mEmail, mPassword;
    private Button mLogin, mRegistration;

    private FirebaseAuth mAuth;
    private FirebaseAuth.AuthStateListener firebaseAuthListener;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_driver_login);

        mAuth = FirebaseAuth.getInstance();

        firebaseAuthListener = new FirebaseAuth.AuthStateListener() {
            @Override
            public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
                FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
                if(user!=null){
                    Intent intent = new Intent(DriverLoginActivity.this,
DriverMapActivity.class);
                    startActivity(intent);
                    finish();
                    return;
                }
            }
        };

        mEmail = (EditText) findViewById(R.id.email);
        mPassword = (EditText) findViewById(R.id.password);

        mLogin = (Button) findViewById(R.id.login);
        mRegistration = (Button) findViewById(R.id.registration);

        mRegistration.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final String email = mEmail.getText().toString();
                final String password = mPassword.getText().toString();
                mAuth.createUserWithEmailAndPassword(email,
password).addOnCompleteListener(DriverLoginActivity.this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if(!task.isSuccessful()){
                            Toast.makeText(DriverLoginActivity.this, "sign up
error", Toast.LENGTH_SHORT).show();
                        }else{
                            String user_id = mAuth.getCurrentUser().getUid();
                            DatabaseReference current_user_db =
FirebaseDatabase.getInstance().getReference().child("Users").child("Drivers").child
(user_id).child("name");
                            current_user_db.setValue(email);
                        }
```

```
                    }
                });
            }
        });

        mLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final String email = mEmail.getText().toString();
                final String password = mPassword.getText().toString();
                mAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(DriverLoginActivity.this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if(!task.isSuccessful()){
                            Toast.makeText(DriverLoginActivity.this, "sign in
error", Toast.LENGTH_SHORT).show();
                        }
                    }
                });

            }
        });
    }


    @Override
    protected void onStart() {
        super.onStart();
        mAuth.addAuthStateListener(firebaseAuthListener);
    }
    @Override
    protected void onStop() {
        super.onStop();
        mAuth.removeAuthStateListener(firebaseAuthListener);
    }
}
```

CustomerMapActivity.java

```
package com.nishantmehta.allycabs;

import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.provider.ContactsContract;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;
import com.firebase.geofire.GeoFire;
```

```
import com.firebase.geofire.GeoLocation;
import com.firebase.geofire.GeoQuery;
import com.firebase.geofire.GeoQueryEventListener;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesNotAvailableException;
import com.google.android.gms.common.GooglePlayServicesRepairableException;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.places.Place;
import com.google.android.gms.location.places.ui.PlaceAutocomplete;
import com.google.android.gms.location.places.ui.PlaceAutocompleteFragment;
import com.google.android.gms.location.places.ui.PlaceSelectionListener;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.vision.text.Line;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.text.ParseException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class CustomerMapActivity extends FragmentActivity implements
OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks,
GoogleApiClient.OnConnectionFailedListener,
com.google.android.gms.location.LocationListener {

    private GoogleMap mMap;
    GoogleApiClient mGoogleApiClient;
    Location mLastLocation;
    LocationRequest mLocationRequest;

    private Button mLogout, mRequest, mSettings;

    private LatLng pickupLocation;

    private Boolean requestBol = false;

    private Marker pickupMarker;

    private SupportMapFragment mapFragment;

    private String destination, requestService;

    private LatLng destinationLatLng;

    private LinearLayout mDriverInfo;

    private ImageView mDriverProfileImage;

    private TextView mDriverName, mDriverPhone, mDriverCar;

    private RadioGroup mRadioGroup;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_costumer_map);
        mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);

        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(CustomerMapActivity.this, new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST_CODE);
        }else{
            mapFragment.getMapAsync(this);
        }

        destinationLatLng = new LatLng(0.0,0.0);

        mDriverInfo = (LinearLayout) findViewById(R.id.driverInfo);

        mDriverProfileImage = (ImageView) findViewById(R.id.driverProfileImage);

        mDriverName = (TextView) findViewById(R.id.driverName);
        mDriverPhone = (TextView) findViewById(R.id.driverPhone);
        mDriverCar = (TextView) findViewById(R.id.driverCar);


        mRadioGroup = (RadioGroup) findViewById(R.id.radioGroup);
        mRadioGroup.check(R.id.UberX);

        mLogout = (Button) findViewById(R.id.logout);
        mRequest = (Button) findViewById(R.id.request);
        mSettings = (Button) findViewById(R.id.settings);
        //   mHistory = (Button) findViewById(R.id.history);

        mLogout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FirebaseAuth.getInstance().signOut();
                Intent intent = new Intent(CustomerMapActivity.this,
MainActivity.class);
                startActivity(intent);
                finish();
                return;
            }
        });

        mRequest.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if (requestBol){
                   endRide();


                }else{
                    int selectId = mRadioGroup.getCheckedRadioButtonId();

                    final RadioButton radioButton = (RadioButton)
findViewById(selectId);

                    if (radioButton.getText() == null){
                        return;
                    }

                    requestService = radioButton.getText().toString();
```

```
                    requestBol = true;

                    String userId =
FirebaseAuth.getInstance().getCurrentUser().getUid();

                    DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("pickup_request");
                    GeoFire geoFire = new GeoFire(ref);
                    geoFire.setLocation(userId, new
GeoLocation(mLastLocation.getLatitude(), mLastLocation.getLongitude()));


ref.child("location").setValue(mLastLocation.getLatitude()+"#"+mLastLocation.getLon
gitude());

ref.child("car").setValue(((RadioButton)findViewById(mRadioGroup.getCheckedRadioBut
tonId())).getText());

                    pickupLocation = new LatLng(mLastLocation.getLatitude(),
mLastLocation.getLongitude());
                    pickupMarker = mMap.addMarker(new
MarkerOptions().position(pickupLocation).title("Pickup
Here").icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_pickup)));

                    mRequest.setText("Getting your Driver....");

                    getClosestDriver();
                }
            }
        });
        mSettings.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
               Intent intent = new Intent(CustomerMapActivity.this,
CustomerSettingsActivity.class);
                startActivity(intent);
                return;
            }
        });


        PlaceAutocompleteFragment autocompleteFragment =
(PlaceAutocompleteFragment)

getFragmentManager().findFragmentById(R.id.place_autocomplete_fragment);

        autocompleteFragment.setOnPlaceSelectedListener(new
PlaceSelectionListener() {
            @Override
            public void onPlaceSelected(Place place) {
                // TODO: Get info about the selected place.
                destination = place.getName().toString();
                destinationLatLng = place.getLatLng();
            }
            @Override
            public void onError(Status status) {
                // TODO: Handle the error.
            }
        });

    }
    private int radius = 1;
    private Boolean driverFound = false;
    private String driverFoundID;

    GeoQuery geoQuery;
    private void getClosestDriver(){
```

```
        DatabaseReference driverLocation =
FirebaseDatabase.getInstance().getReference().child("driversAvailable");

        GeoFire geoFire = new GeoFire(driverLocation);
        geoQuery = geoFire.queryAtLocation(new GeoLocation(pickupLocation.latitude,
pickupLocation.longitude), radius);
        geoQuery.removeAllListeners();

        geoQuery.addGeoQueryEventListener(new GeoQueryEventListener() {
            @Override
            public void onKeyEntered(String key, GeoLocation location) {
                if (!driverFound && requestBol){
                    DatabaseReference mCustomerDatabase =
FirebaseDatabase.getInstance().getReference().child("Users").child("Drivers").child
(key);
                    mCustomerDatabase.addListenerForSingleValueEvent(new
ValueEventListener() {
                        @Override
                        public void onDataChange(DataSnapshot dataSnapshot) {
                            if (dataSnapshot.exists() &&
dataSnapshot.getChildrenCount()>0){
                                Map<String, Object> driverMap = (Map<String,
Object>) dataSnapshot.getValue();
                                if (driverFound){
                                    return;
                                }


if(driverMap.get("service").equals(requestService)){
                                    driverFound = true;
                                    driverFoundID = dataSnapshot.getKey();

                                    DatabaseReference driverRef =
FirebaseDatabase.getInstance().getReference().child("Users").child("Drivers").child
(driverFoundID).child("customerRequest");
                                    String customerId =
FirebaseAuth.getInstance().getCurrentUser().getUid();
                                    HashMap map = new HashMap();
                                    map.put("customerRideId", customerId);
                                    map.put("destination", destination);
                                    map.put("destinationLat",
destinationLatLng.latitude);
                                    map.put("destinationLng",
destinationLatLng.longitude);
                                    driverRef.updateChildren(map);

                                    getDriverLocation();
                                    getDriverInfo();
                                    getHasRideEnded();
                                    mRequest.setText("Looking for Driver
Location....");
                                }
                            }
                        }
                        @Override
                        public void onCancelled(DatabaseError databaseError) {
                        }
                    });
                }
            }

            @Override
            public void onKeyExited(String key) {

            }

            @Override
            public void onKeyMoved(String key, GeoLocation location)
```

```
            }

            @Override
            public void onGeoQueryReady() {
                if (!driverFound)
                {
                    radius++;
                    getClosestDriver();
                }
            }

            @Override
            public void onGeoQueryError(DatabaseError error) {

            }
        });
    }
    private Marker mDriverMarker;
    private DatabaseReference driverLocationRef;
    private ValueEventListener driverLocationRefListener;
    private void getDriverLocation(){
        driverLocationRef =
FirebaseDatabase.getInstance().getReference().child("driversWorking").child(driverF
oundID).child("l");
        driverLocationRefListener = driverLocationRef.addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists() && requestBol){
                    List<Object> map = (List<Object>) dataSnapshot.getValue();
                    double locationLat = 0;
                    double locationLng = 0;
                    if(map.get(0) != null){
                        locationLat = Double.parseDouble(map.get(0).toString());
                    }
                    if(map.get(1) != null){
                        locationLng = Double.parseDouble(map.get(1).toString());
                    }
                    LatLng driverLatLng = new LatLng(locationLat,locationLng);
                    if(mDriverMarker != null){
                        mDriverMarker.remove();
                    }
                    Location loc1 = new Location("");
                    loc1.setLatitude(pickupLocation.latitude);
                    loc1.setLongitude(pickupLocation.longitude);

                    Location loc2 = new Location("");
                    loc2.setLatitude(driverLatLng.latitude);
                    loc2.setLongitude(driverLatLng.longitude);

                    float distance = loc1.distanceTo(loc2);

                    if (distance<100){
                        mRequest.setText("Driver's Here");
                    }else{
                        mRequest.setText("Driver Found: " +
String.valueOf(distance));
                    }
                    mDriverMarker = mMap.addMarker(new
MarkerOptions().position(driverLatLng).title("your
driver").icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_car)));
                }

            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
            }
```

```
        });

    }

    private void getDriverInfo(){
        mDriverInfo.setVisibility(View.VISIBLE);
        DatabaseReference mCustomerDatabase =
FirebaseDatabase.getInstance().getReference().child("Users").child("Drivers").child
(driverFoundID);
        mCustomerDatabase.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists() && dataSnapshot.getChildrenCount()>0){
                    Map<String, Object> map = (Map<String, Object>)
dataSnapshot.getValue();
                    if(map.get("name")!=null){
                        mDriverName.setText(map.get("name").toString());
                    }
                    if(map.get("phone")!=null){
                        mDriverPhone.setText(map.get("phone").toString());
                    }
                    if(map.get("car")!=null){
                        mDriverCar.setText(map.get("car").toString());
                    }
                    if(map.get("profileImageUrl")!=null){

Glide.with(getApplication()).load(map.get("profileImageUrl").toString()).into(mDriv
erProfileImage);
                    }
                }
            }
            @Override
            public void onCancelled(DatabaseError databaseError) {
            }
        });
    }

    private DatabaseReference driveHasEndedRef;
    private ValueEventListener driveHasEndedRefListener;
    private void getHasRideEnded(){
        driveHasEndedRef =
FirebaseDatabase.getInstance().getReference().child("Users").child("Drivers").child
(driverFoundID).child("customerRequest").child("customerRideId");
        driveHasEndedRefListener = driveHasEndedRef.addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists()){

                }else{
                    endRide();
                }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
            }
        });
    }

    private void endRide(){
        requestBol = false;
        geoQuery.removeAllListeners();
        driverLocationRef.removeEventListener(driverLocationRefListener);
        driveHasEndedRef.removeEventListener(driveHasEndedRefListener);

        if (driverFoundID != null){
            DatabaseReference driverRef =
```

```
FirebaseDatabase.getInstance().getReference().child("Users").child("Drivers").child
(driverFoundID).child("customerRequest");
            driverRef.removeValue();
            driverFoundID = null;

        }
        driverFound = false;
        radius = 1;
        String userId = FirebaseAuth.getInstance().getCurrentUser().getUid();

        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("customerRequest");
        GeoFire geoFire = new GeoFire(ref);
        geoFire.removeLocation(userId);

        if(pickupMarker != null){
            pickupMarker.remove();
        }
        if (mDriverMarker != null){
            mDriverMarker.remove();
        }
        mRequest.setText("call Uber");

        mDriverInfo.setVisibility(View.GONE);
        mDriverName.setText("");
        mDriverPhone.setText("");
        mDriverCar.setText("Destination: --");
        mDriverProfileImage.setImageResource(R.mipmap.ic_default_user);
    }
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(CustomerMapActivity.this, new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST_CODE);
        }
        buildGoogleApiClient();
        mMap.setMyLocationEnabled(true);
    }

    protected synchronized void buildGoogleApiClient(){
        mGoogleApiClient = new GoogleApiClient.Builder(this)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .addApi(LocationServices.API)
                .build();
        mGoogleApiClient.connect();
    }

    @Override
    public void onLocationChanged(Location location) {
        if(getApplicationContext()!=null){
            mLastLocation = location;

            LatLng latLng = new
LatLng(location.getLatitude(),location.getLongitude());

            mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
            mMap.animateCamera(CameraUpdateFactory.zoomTo(11));
        }
    }

    @Override
```

```
    public void onConnected(@Nullable Bundle bundle) {
        mLocationRequest = new LocationRequest();
        mLocationRequest.setInterval(1000);
        mLocationRequest.setFastestInterval(1000);
        mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(CustomerMapActivity.this, new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST_CODE);
        }
        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,
mLocationRequest, this);
    }

    @Override
    public void onConnectionSuspended(int i) {
    }

    @Override
    public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
    }
    final int LOCATION_REQUEST_CODE = 1;
    public void onRequestPermissionsResult(int requestCode, String permissions[],
int[] grantResults) {
        switch (requestCode) {
            case LOCATION_REQUEST_CODE: {
                // If request is cancelled, the result arrays are empty.
                if (grantResults.length > 0
                        && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

                    mapFragment.getMapAsync(this);


                } else {
                    Toast.makeText(getApplicationContext(), "Please provide the
permission", Toast.LENGTH_LONG).show();
                }
                break;
            }
        }
    }
}
```

CustomerSettingsActivity.java

```
package com.nishantmehta.allycabs;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;

import com.bumptech.glide.Glide;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
```

```
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class CustomerSettingsActivity extends AppCompatActivity {

    private EditText mNameField, mPhoneField;

    private Button mBack, mConfirm;

    private ImageView mProfileImage;

    private FirebaseAuth mAuth;
    private DatabaseReference mCustomerDatabase;

    private String userID;
    private String mName;
    private String mPhone;
    private String mProfileImageUrl;

    private Uri resultUri;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_customer_settings);

        mNameField = (EditText) findViewById(R.id.name);
        mPhoneField = (EditText) findViewById(R.id.phone);

        mProfileImage = (ImageView) findViewById(R.id.profileImage);

        mBack = (Button) findViewById(R.id.back);
        mConfirm = (Button) findViewById(R.id.confirm);

        mAuth = FirebaseAuth.getInstance();
        userID = mAuth.getCurrentUser().getUid();
        mCustomerDatabase =
FirebaseDatabase.getInstance().getReference().child("Users").child("Customers").chi
ld(userID);

        getUserInfo();

        mProfileImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_PICK);
                intent.setType("image/*");
                startActivityForResult(intent, 1);
            }
        });

        mConfirm.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
                saveUserInformation();
            }
        });

        mBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
                return;
            }
        });
    }
    private void getUserInfo(){
        mCustomerDatabase.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists() && dataSnapshot.getChildrenCount()>0){
                    Map<String, Object> map = (Map<String, Object>)
dataSnapshot.getValue();
                    if(map.get("name")!=null){
                        mName = map.get("name").toString();
                        mNameField.setText(mName);
                    }
                    if(map.get("phone")!=null){
                        mPhone = map.get("phone").toString();
                        mPhoneField.setText(mPhone);
                    }
                    if(map.get("profileImageUrl")!=null){
                        mProfileImageUrl = map.get("profileImageUrl").toString();

Glide.with(getApplication()).load(mProfileImageUrl).into(mProfileImage);
                    }
                }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
            }
        });
    }



    private void saveUserInformation() {
        mName = mNameField.getText().toString();
        mPhone = mPhoneField.getText().toString();

        Map userInfo = new HashMap();
        userInfo.put("name", mName);
        userInfo.put("phone", mPhone);
        mCustomerDatabase.updateChildren(userInfo);

        if(resultUri != null) {

            StorageReference filePath =
FirebaseStorage.getInstance().getReference().child("profile_images").child(userID);
            Bitmap bitmap = null;
            try {
                bitmap =
MediaStore.Images.Media.getBitmap(getApplication().getContentResolver(),
resultUri);
            } catch (IOException e) {
                e.printStackTrace();
            }

            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            bitmap.compress(Bitmap.CompressFormat.JPEG, 20, baos);
            byte[] data = baos.toByteArray();
```

```
            UploadTask uploadTask = filePath.putBytes(data);

            uploadTask.addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    finish();
                    return;
                }
            });
            uploadTask.addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                    Uri downloadUrl = taskSnapshot.getDownloadUrl();

                    Map newImage = new HashMap();
                    newImage.put("profileImageUrl", downloadUrl.toString());
                    mCustomerDatabase.updateChildren(newImage);

                    finish();
                    return;
                }
            });
        }else{
            finish();
        }

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode == 1 && resultCode == Activity.RESULT_OK){
            final Uri imageUri = data.getData();
            resultUri = imageUri;
            mProfileImage.setImageURI(resultUri);
        }
    }
}
```

DriverSettingsActivity.java

```
package com.nishantmehta.allycabs;

import android.app.Activity;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Build;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;

import com.bumptech.glide.Glide;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
```

```
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class DriverSettingsActivity extends AppCompatActivity {

    private EditText mNameField, mPhoneField, mCarField;

    private Button mBack, mConfirm;

    private ImageView mProfileImage;

    private FirebaseAuth mAuth;
    private DatabaseReference mDriverDatabase;

    private String userID;
    private String mName;
    private String mPhone;
    private String mCar;
    private String mService;
    private String mProfileImageUrl;

    private Uri resultUri;

    private RadioGroup mRadioGroup;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_driver_settings);


        mNameField = (EditText) findViewById(R.id.name);
        mPhoneField = (EditText) findViewById(R.id.phone);
        mCarField = (EditText) findViewById(R.id.car);

        mProfileImage = (ImageView) findViewById(R.id.profileImage);

        mRadioGroup = (RadioGroup) findViewById(R.id.radioGroup);

        mBack = (Button) findViewById(R.id.back);
        mConfirm = (Button) findViewById(R.id.confirm);

        mAuth = FirebaseAuth.getInstance();
        userID = mAuth.getCurrentUser().getUid();
        mDriverDatabase =
FirebaseDatabase.getInstance().getReference().child("Users").child("Drivers").child
(userID);

        getUserInfo();

        mProfileImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_PICK);
                intent.setType("image/*");
```

```
                startActivityForResult(intent, 1);
            }
        });

        mConfirm.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                saveUserInformation();
            }
        });

        mBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
                return;
            }
        });
    }
    private void getUserInfo(){
        mDriverDatabase.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists() && dataSnapshot.getChildrenCount()>0){
                    Map<String, Object> map = (Map<String, Object>)
dataSnapshot.getValue();
                    if(map.get("name")!=null){
                        mName = map.get("name").toString();
                        mNameField.setText(mName);
                    }
                    if(map.get("phone")!=null){
                        mPhone = map.get("phone").toString();
                        mPhoneField.setText(mPhone);
                    }
                    if(map.get("car")!=null){
                        mCar = map.get("car").toString();
                        mCarField.setText(mCar);
                    }
                    if(map.get("service")!=null){
                        mService = map.get("service").toString();
                        switch (mService){
                            case"UberX":
                                mRadioGroup.check(R.id.UberX);
                                break;
                            case"UberBlack":
                                mRadioGroup.check(R.id.UberBlack);
                                break;
                            case"UberXl":
                                mRadioGroup.check(R.id.UberXl);
                                break;
                        }
                    }
                    if(map.get("profileImageUrl")!=null){
                        mProfileImageUrl = map.get("profileImageUrl").toString();

Glide.with(getApplication()).load(mProfileImageUrl).into(mProfileImage);
                    }
                }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
            }
        });
    }
```

```
    private void saveUserInformation() {
        mName = mNameField.getText().toString();
        mPhone = mPhoneField.getText().toString();
        mCar = mCarField.getText().toString();

        int selectId = mRadioGroup.getCheckedRadioButtonId();

        final RadioButton radioButton = (RadioButton) findViewById(selectId);

        if (radioButton.getText() == null){
            return;
        }

        mService = radioButton.getText().toString();

        Map userInfo = new HashMap();
        userInfo.put("name", mName);
        userInfo.put("phone", mPhone);
        userInfo.put("car", mCar);
        userInfo.put("service", mService);
        mDriverDatabase.updateChildren(userInfo);

        if(resultUri != null) {

            StorageReference filePath =
FirebaseStorage.getInstance().getReference().child("profile_images").child(userID);
            Bitmap bitmap = null;
            try {
                bitmap =
MediaStore.Images.Media.getBitmap(getApplication().getContentResolver(),
resultUri);
            } catch (IOException e) {
                e.printStackTrace();
            }

            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            bitmap.compress(Bitmap.CompressFormat.JPEG, 20, baos);
            byte[] data = baos.toByteArray();
            UploadTask uploadTask = filePath.putBytes(data);

            uploadTask.addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    finish();
                    return;
                }
            });
            uploadTask.addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                    Uri downloadUrl = taskSnapshot.getDownloadUrl();

                    Map newImage = new HashMap();
                    newImage.put("profileImageUrl", downloadUrl.toString());
                    mDriverDatabase.updateChildren(newImage);

                    finish();
                    return;
                }
            });
        }else{
            finish();
        }

    }

    @Override
```

```
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode == 1 && resultCode == Activity.RESULT_OK){
            final Uri imageUri = data.getData();
            resultUri = imageUri;
            mProfileImage.setImageURI(resultUri);
        }
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.nishantmehta.allycabs.MainActivity"
    android:orientation="vertical">


    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="           I'm a Driver            "
        android:id="@+id/driver"
        android:layout_margin="70dp"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="I'm a Customer"
        android:layout_margin="30dp"
        android:id="@+id/customer"/>

</LinearLayout>
```

activity_driver_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.nishantmehta.allycabs.DriverLoginActivity"
    android:orientation="vertical">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="email"
        android:id="@+id/email"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="password"
        android:id="@+id/password"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="login"
        android:id="@+id/login"/>
    <Button
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="registration"
        android:id="@+id/registration"/>
</LinearLayout>
```

activity_customer_map.xml

```
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.nishantmehta.allycabs.CustomerLoginActivity" >



    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/buttons">
            <Button
                android:layout_weight="1"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:id="@+id/logout"
                android:text="logout"/>

            <Button
                android:layout_weight="1"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:id="@+id/settings"
                android:text="Settings"/>
        </LinearLayout>

        <android.support.v7.widget.CardView
            android:layout_below="@+id/buttons"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="20sp">
            <fragment
                android:id="@+id/place_autocomplete_fragment"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

android:name="com.google.android.gms.location.places.ui.PlaceAutocompleteFragment"
                />
        </android.support.v7.widget.CardView>
    </RelativeLayout>



    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_gravity="bottom">
        <LinearLayout
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/driverInfo"
        android:layout_gravity="bottom"
        android:orientation="horizontal"
        android:background="@android:color/white"
        android:visibility="gone">
        <ImageView
            android:layout_width="100sp"
            android:layout_height="100sp"
            android:id="@+id/driverProfileImage"
            android:src="@mipmap/ic_default_user"
            android:padding="20sp"/>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:paddingLeft="40sp">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/driverName"
                android:paddingBottom="10sp"
                android:paddingTop="20sp"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/driverPhone"
                android:paddingBottom="10sp"
                android:paddingTop="20sp"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/driverCar"
                android:paddingBottom="10sp"
                android:paddingTop="20sp"/>
        </LinearLayout>
    </LinearLayout>
    <LinearLayout
        android:background="@android:color/white"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <RadioGroup
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/radioGroup"
            android:orientation="horizontal">
            <RadioButton
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="UberX"
                android:id="@+id/UberX"/>
            <RadioButton
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="UberBlack"
                android:id="@+id/UberBlack"/>
            <RadioButton
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="UberXl"
                android:id="@+id/UberXl"/>
        </RadioGroup>

        <Button
            android:layout_gravity="bottom"
            android:layout_width="match_parent"
```

```
                android:layout_height="wrap_content"
                android:text="call Uber"
                android:id="@+id/request"/>
        </LinearLayout>
    </LinearLayout>

</FrameLayout>
```

activity_customer_settings.xml

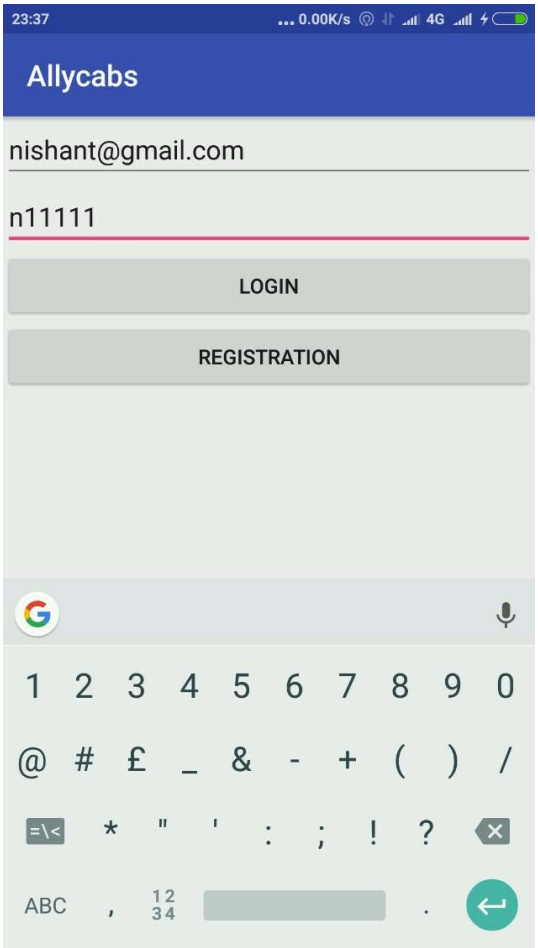
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.nishantmehta.allycabs.CustomerSettingsActivity"
    android:orientation="vertical"
    android:padding="20sp">
    <ImageView
        android:layout_width="100sp"
        android:layout_height="100sp"
        android:id="@+id/profileImage"
        android:src="@mipmap/ic_default_user"
        android:layout_marginBottom="20sp"/>
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@null"
        android:hint="Name"
        android:layout_marginBottom="20sp"/>
    <EditText
        android:id="@+id/phone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@null"
        android:hint="Phone"
        android:layout_marginBottom="20sp"
        android:inputType="number"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/confirm"
        android:text="confirm"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/back"
        android:text="back"/>

</LinearLayout>
```

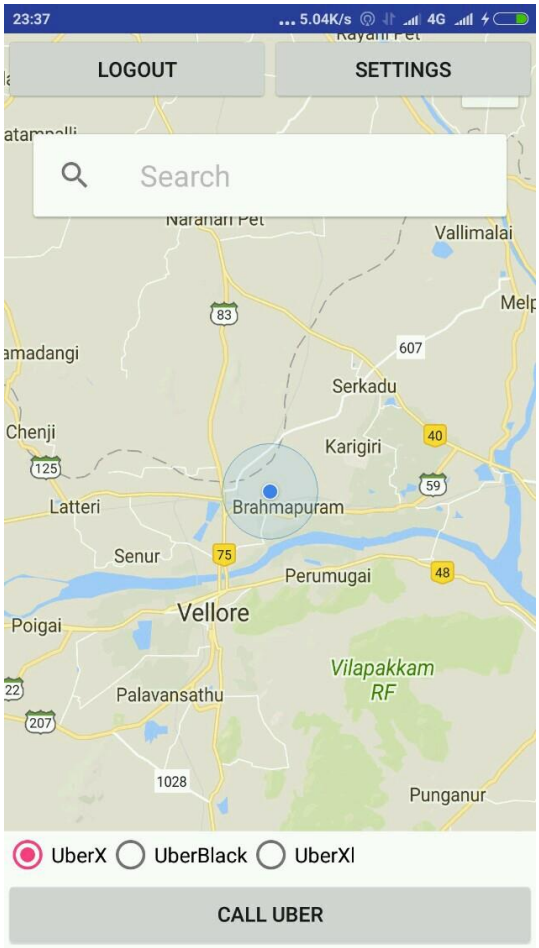
activity_driver_settings.xml

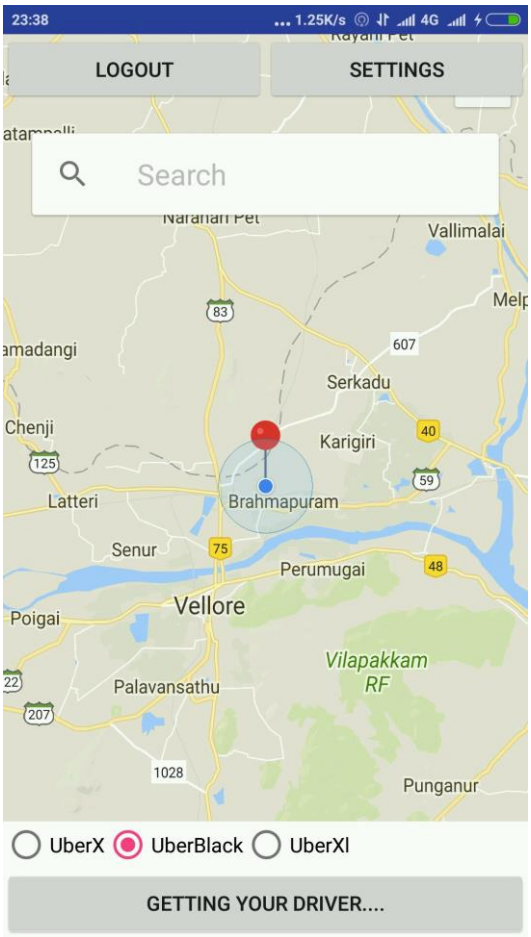
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.nishantmehta.allycabs.CustomerSettingsActivity"
    android:orientation="vertical"
    android:padding="20sp">
    <ImageView
        android:layout_width="100sp"
```

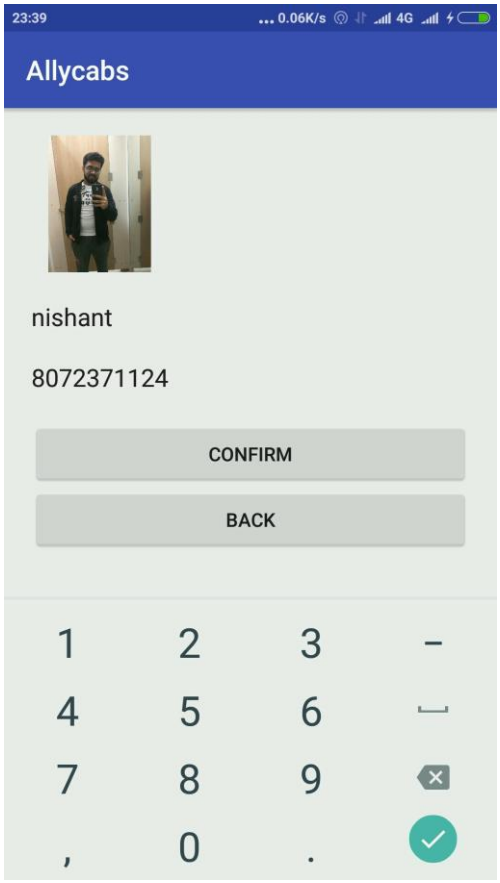
```
            android:layout_height="100sp"
            android:id="@+id/profileImage"
            android:src="@mipmap/ic_default_user"
            android:layout_marginBottom="20sp"/>
        <EditText
            android:id="@+id/name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@null"
            android:hint="Name"
            android:layout_marginBottom="20sp"/>
        <EditText
            android:id="@+id/phone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@null"
            android:hint="Phone"
            android:layout_marginBottom="20sp"
            android:inputType="number"/>
        <EditText
            android:id="@+id/car"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@null"
            android:hint="Car"
            android:layout_marginBottom="20sp"/>
        <RadioGroup
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/radioGroup"
            android:orientation="horizontal">

            <RadioButton
                android:id="@+id/UberX"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="UberX" />

            <RadioButton
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="UberBlack"
                android:id="@+id/UberBlack"/>
            <RadioButton
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="UberXl"
                android:id="@+id/UberXl"/>
        </RadioGroup>
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/confirm"
            android:text="confirm"/>
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/back"
            android:text="back"/>

    </LinearLayout>
```

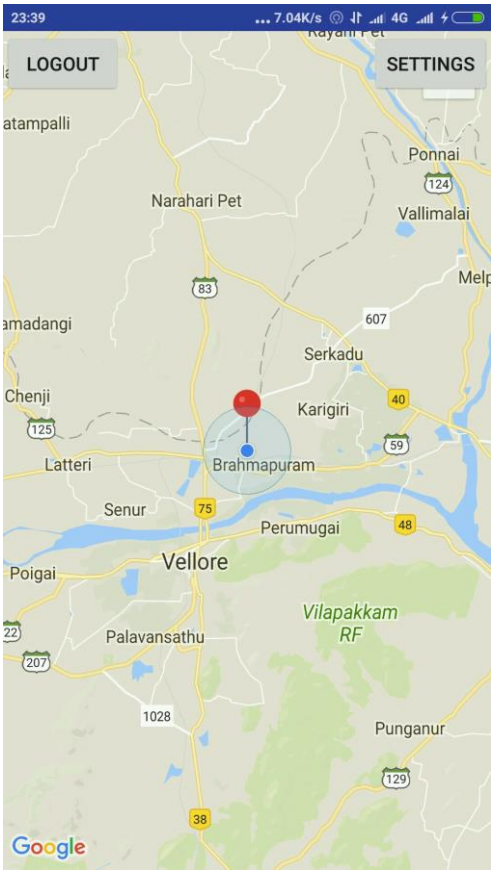
Screenshots of the App -- >

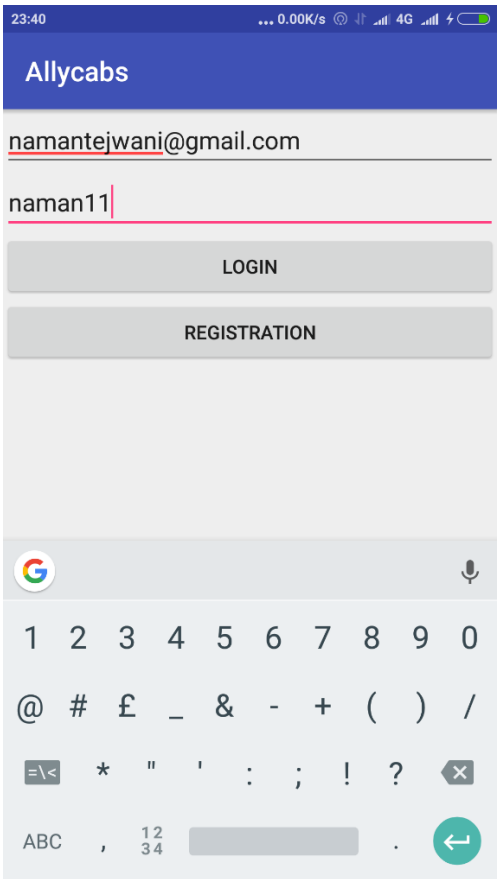


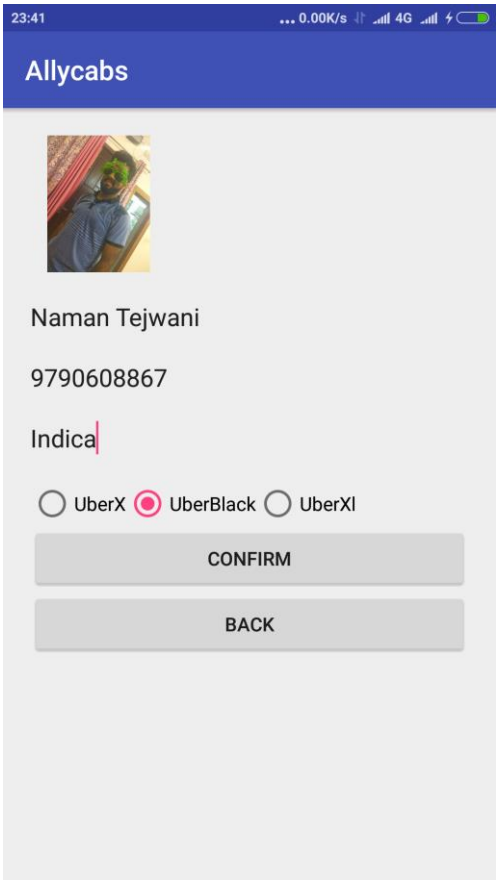

23:37
5.04K/s
4G
LOGOUT
SETTINGS
Search
Vallimalai
Melp
amadangi
Serkadu
Chenji
Karigiri
Latteri
Brahmapuram
Senur
Perumugai
Vellore
Poigai
Vilapakkam
RF
Palavansathu
Punganur
1028
UberX
UberBlack
UberXl
CALL UBER


23:38
1.25K/s
4G
LOGOUT
SETTINGS
Search
Vallimalai
Melp
amadangi
Serkadu
Chenji
Karigiri
Latteri
Brahmapuram
Senur
Perumugai
Vellore
Poigai
Vilapakkam
RF
Palavansathu
Punganur
1028
UberX
UberBlack
UberXl
GETTING YOUR DRIVER....


23:39
0.06K/s
4G
Allycabs
nishant
8072371124
CONFIRM
BACK


23:39
7.04K/s
4G
LOGOUT
SETTINGS
Narahari Pet
Ponnai
Vallimalai
Serkadu
Karigiri
Chenji
Latteri
Brahmapuram
Senur
Perumugai
Vellore
Poigai
Palavansathu
Vilapakkam
RF
Punganur
Google


23:40
0.00K/s
4G
Allycabs
namantejwani@gmail.com
naman11
LOGIN
REGISTRATION



Conclusion

This project is about the designing a Cab booking Android Application using Android Studio and Firebase.
This project presents an investigative view of presenting the cab booking application including live map loading and current position of the customer. Present system of taxi management system is having some shortcoming on which I have tried to work, on that to eliminate the disadvantages. I have used Firebase for a real time database for customers and taxis and connecting the customer and driver.
This project was a small attempt to make a simple and efficient android application that could be used by normal people and solve their problem of travelling.

Future Enhancement

I need to talk with some of the employees of each state's cab service about the features and shortcoming of present taxi management system.
There is a need to do more efficient coding for the application so that the app size should decrease so that more and more people could use this app.
There is a need to do a research and after the research with the associated people and other sources i need to found out some of the major facts regarding the taxi management system like cab fares, cab time efficiency, other problems faced by people in India to book cabs and try to eliminate the shortcoming of system. In the last I would like to conclude that Indian taxi is having a strong IT Infrastructure and a well-equipped taxi management system but there is some shortcoming in the system on which i have tried to work and successfully completed our project.

References

[1] **https://developer.android.com/training/index.html**

[2] **https://www.tutorialspoint.com/android/**

[3] **https://stackoverflow.com/**

[4]https://www.youtube.com/watch?v=Z3HdOTQ8YHI&list=PLxabZQCAe5fgXx8cn2iKOtt0VFJrf5bOd

[5]https://doc.lagout.org/programmation/Android/Android%20Programming%20for%20Beginners%20%5BHorton%202016-01-06%5D.pdf