7.2 Application code

Code Location: https://github.com/c3ko/Parts-Crib-Android

Demo

/1 Hardware present?

/1 Memo by student A

/1 Login activity

/1 Data visualization activity

/1 Action control activity

Report

/1 Login activity

/1 Data visualization activity

/1 Action control activity

/1 Modified Code Files in Appendix

/1 Link to Complete Code in Repository

The Login Activity (along with a registration and forgotten password activity) were already implemented in the CENG 318. Those activities along with the ability to create requests was also demonstrated during the previous android application status update. In the intervening time we’ve further added an activty to input the six digit pin required to retrieve requested parts instead of using the keypad. The acitvity updates and sets the current pin being required by the parts crib through the Firebase database so that each key press on either the keypad or the phone will be updated throughout the system in real time. Furthermore, we’ve implemented an activity to display the current location of the part-crib in real time by reading the GPS coordinates from the GPS sensor from the Firebase database through the android application.

7.2.1 KeypinActivity.java

```
package com.partscrib.partscribmanagementsystem;


import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import com.davidmiguel.numberkeyboard.NumberKeyboard;
import com.davidmiguel.numberkeyboard.NumberKeyboardListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
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 com.partscrib.partscribmanagementsystem.model.ExpandableListPartData;
import com.partscrib.partscribmanagementsystem.model.PartExpandableListAdapter;
import com.partscrib.partscribmanagementsystem.model.PartModel;

import android.os.Bundle;
import android.service.autofill.UserData;
import android.util.Log;
import android.widget.TextView;

import org.w3c.dom.Text;
import java.util.ArrayList;
import static
com.partscrib.partscribmanagementsystem.Login.USER_NAME_MESSAGE;

public class KeypinActivity extends AppCompatActivity {

    NumberKeyboard numberKeyboard;
    TextView textView;
```

```
String keypin = "";

private FirebaseDatabase db;

private DatabaseReference dbRef;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_keypin);

    numberKeyboard = (NumberKeyboard) findViewById(R.id.numberKeyboard);

    textView = (TextView) findViewById(R.id.keypin_text);

    numberKeyboard.showRightAuxButton();

    db = FirebaseDatabase.getInstance();

    String user = getIntent().getStringExtra(USER_NAME_MESSAGE);



    String path = "userdata/" + user + "/currentPinEntry";

    dbRef = db.getReference(path);

    numberKeyboard.setListener(new NumberKeyboardListener() {

        @Override

        public void onNumberClicked(int number) {
            if (keypin.length() < 6){
                setCurrentPin(number);
                textView.setText(keypin);
            }
            else {

            }

        }
        @Override
        public void onLeftAuxButtonClicked() {
        }
        @Override
        public void onRightAuxButtonClicked() {
        }
    });
    getCurrentPin();
}
```

```
    public void setCurrentPin(int number){

        keypin = keypin + number;

        // Submit pin to firebase

        dbRef.setValue(keypin);

    }

    public void getCurrentPin(){

        dbRef.addValueEventListener(new ValueEventListener() {

            @Override

            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

                keypin = dataSnapshot.getValue(String.class);

                textView.setText(keypin);

            }

            @Override

            public void onCancelled(@NonNull DatabaseError databaseError) {

            }

        });

    }

}
```

7.2.2 MapActivity.java

```
package com.partscrib.partscribmanagementsystem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapView;
import com.google.android.gms.maps.MapsInitializer;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
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 static
com.partscrib.partscribmanagementsystem.Login.USER_NAME_MESSAGE;


public class MapActivity extends AppCompatActivity {


    class Location {

        private String latitude;

        private String longitude;


        public Location(String latitude, String longitude) {

            this.latitude = latitude;

            this.longitude = longitude;
        }

        public String getLatitude() {
            return latitude;
        }

        public void setLatitude(String latitude) {
            this.latitude = latitude;
        }

        public String getLongitude() {
            return longitude;
        }

        public void setLongitude(String longitude) {
            this.longitude = longitude;
        }
    }
```

```
Location location;
MapView mMapView;
GoogleMap googleMap;

FirebaseDatabase db;
DatabaseReference dbRef;



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

    db = FirebaseDatabase.getInstance();

    String path = "newsBulletin/location";

    dbRef = db.getReference(path);
```

```
mMapView = (MapView) findViewById(R.id.mapView2);
mMapView.onCreate(savedInstanceState);


mMapView.onResume(); // needed to get the map to display immediately
dbRef.addValueEventListener(new ValueEventListener() {


    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        location = dataSnapshot.getValue(Location.class);
    }
    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {


    }


});
try {
    MapsInitializer.initialize(getApplicationContext());
```

```
        } catch (Exception e) {
            e.printStackTrace();
        }
        mMapView.getMapAsync(new OnMapReadyCallback() {
            @Override
            public void onMapReady(GoogleMap mMap) {
                googleMap = mMap;

                // For dropping a marker at a point on the Map
                //LatLng Humber = new LatLng(43.724330436,-79.605497578);
                LatLng Humber = new LatLng(Float.parseFloat(location.longitude),
Float.parseFloat(location.latitude));
                googleMap.addMarker(new MarkerOptions()
                        .position(Humber)
                        .icon(BitmapDescriptorFactory.fromResource(R.drawable.humber))
                        .title("Welcome to Humber College"));

                googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(Humber,
(float) 6.0));
```

```
            googleMap.setMinZoomPreference(2.0f);
            googleMap.setMaxZoomPreference(30.0f);
            googleMap.setTrafficEnabled(true);
        }

    });

}
    @Override
    public void onResume() {
        super.onResume();
        mMapView.onResume();
    }

    @Override
    public void onPause() {
        super.onPause();
        mMapView.onPause();
```

```
}


@Override
public void onDestroy() {
    super.onDestroy();
    mMapView.onDestroy();
}


@Override
public void onLowMemory() {
    super.onLowMemory();
    mMapView.onLowMemory();
}


private double distance(double lat1, double lon1, double lat2, double lon2) {
    double theta = lon1 - lon2;
    double dist = Math.sin(deg2rad(lat1))
            * Math.sin(deg2rad(lat2))
```

```
                + Math.cos(deg2rad(lat1))

                * Math.cos(deg2rad(lat2))

                * Math.cos(deg2rad(theta));

            dist = Math.acos(dist);

            dist = rad2deg(dist);

            dist = dist * 60 * 1.1515;

            return (dist);

        }



        private double deg2rad(double deg) {

            return (deg * Math.PI / 180.0);

        }



        private double rad2deg(double rad) {

            return (rad * 180.0 / Math.PI);

        }

    }
```