CS426- Mobile Device Application Development

Class: APCS 2015

Date: June 17-20, 2017 (72+6 hours)

Mini Project 1
Personal Report

Student ID: 1551025
Student's full name: Liêng Thế Phy
Email: ltphy@apcs.vn
Tel: 0901448926

I. Project Self Assessment

In this section, each student should personally evaluate his or her own effort to complete the project.

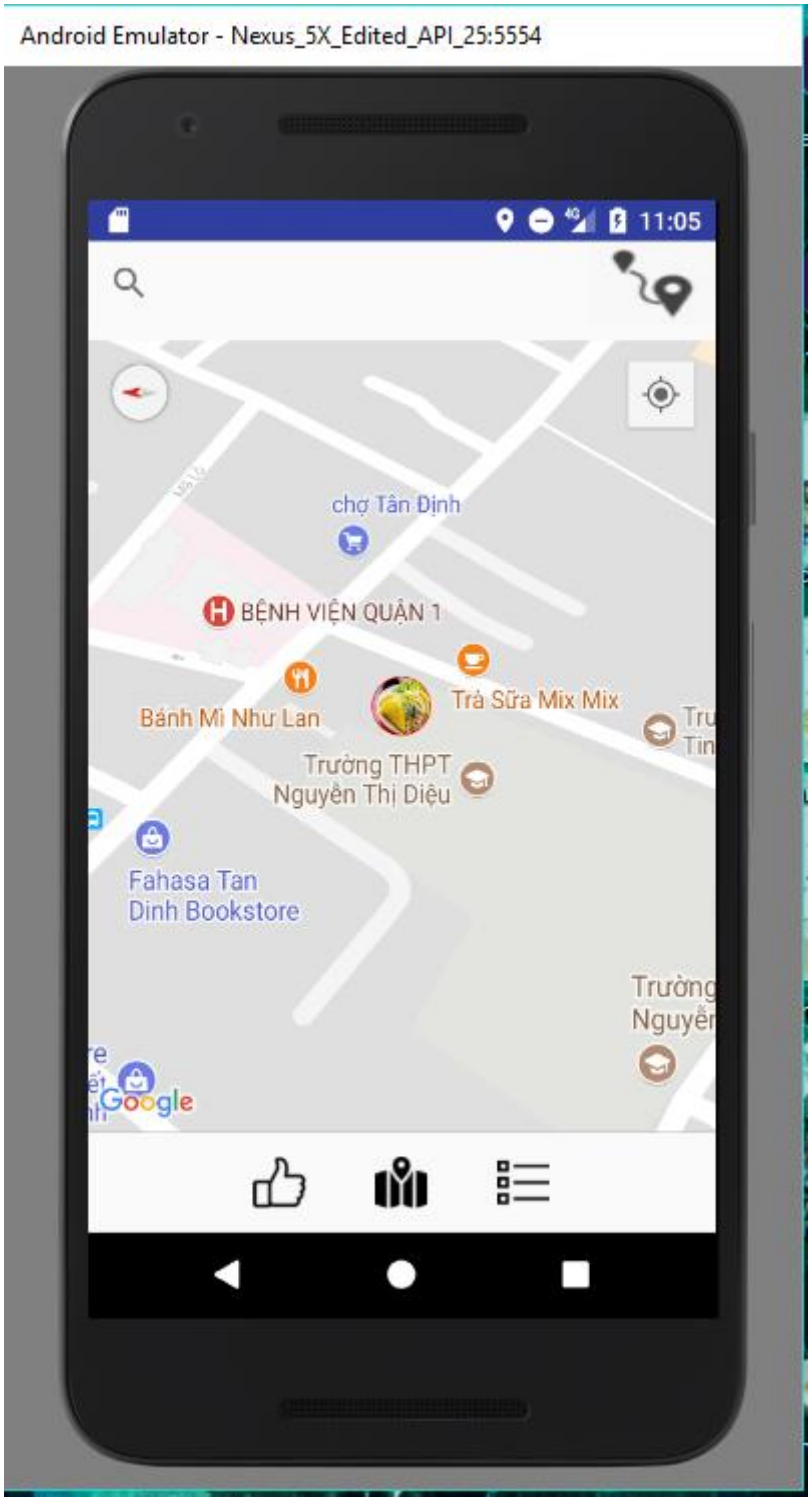
- For each feature, please determine the level of complexity (1-Very simple, 2-Simple, 3-Normal, 4-Difficult, 5-Very difficult) and level of completeness (100%).

Feature	Complexity	Level of completeness (100%)
Multiple List View With UI Design	3	90%
Have A Button to Show Address in Map of Current Item	3	80%
Call Intent	1	100%
Change Different Activity when Clicking Button	3	70%
Show Icon Marker with Round Layout on Map From Detail Item	3	100%
Show Information Window on Map	3	100%
Find Path from Current Location to a Selected Marker Items	4	75%
Find Direction from Path to Path	3	80%
Show Current Position	1	100%
Book Mark Item to My List	4	30%

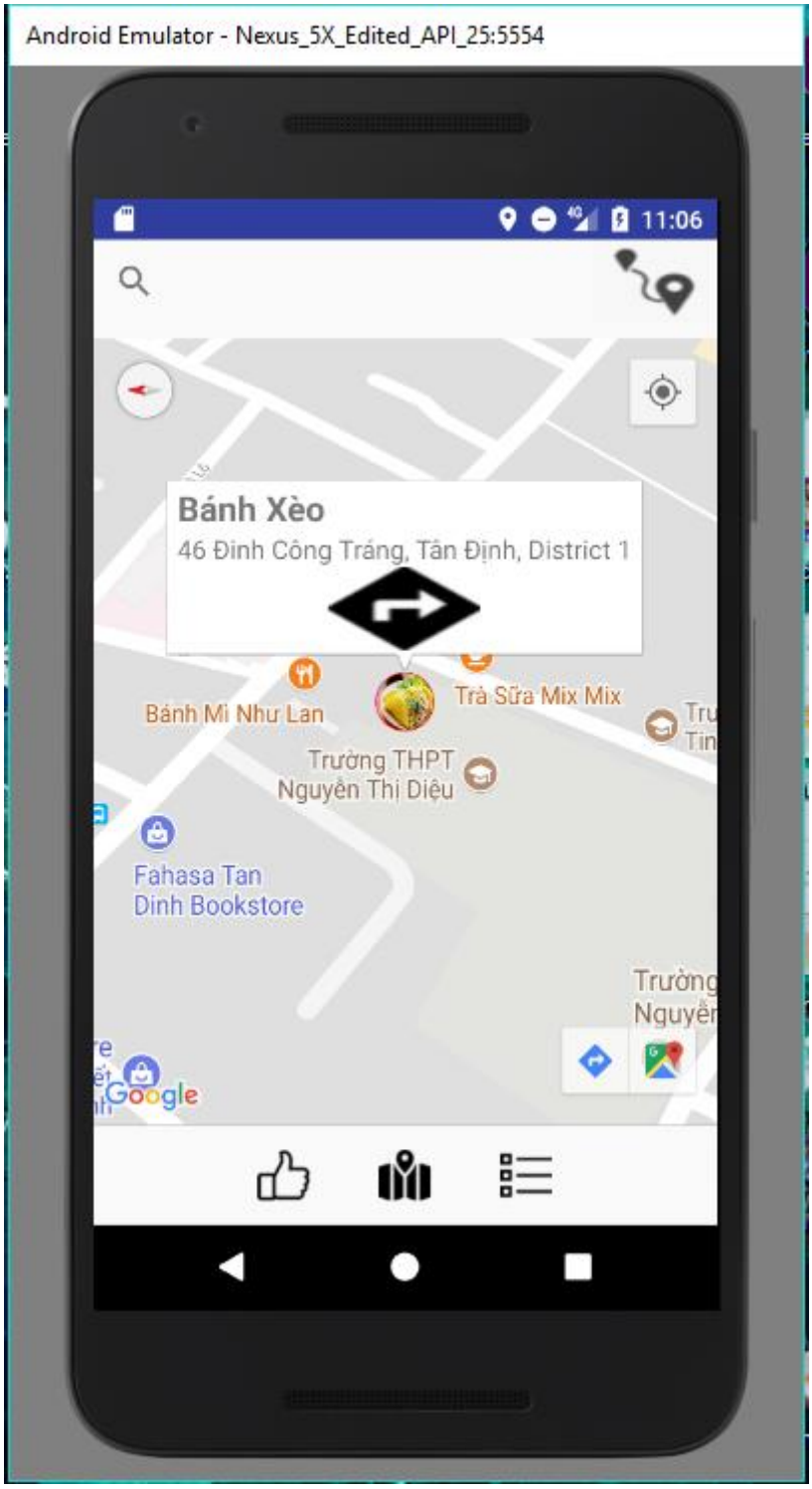
II. Advanced Features

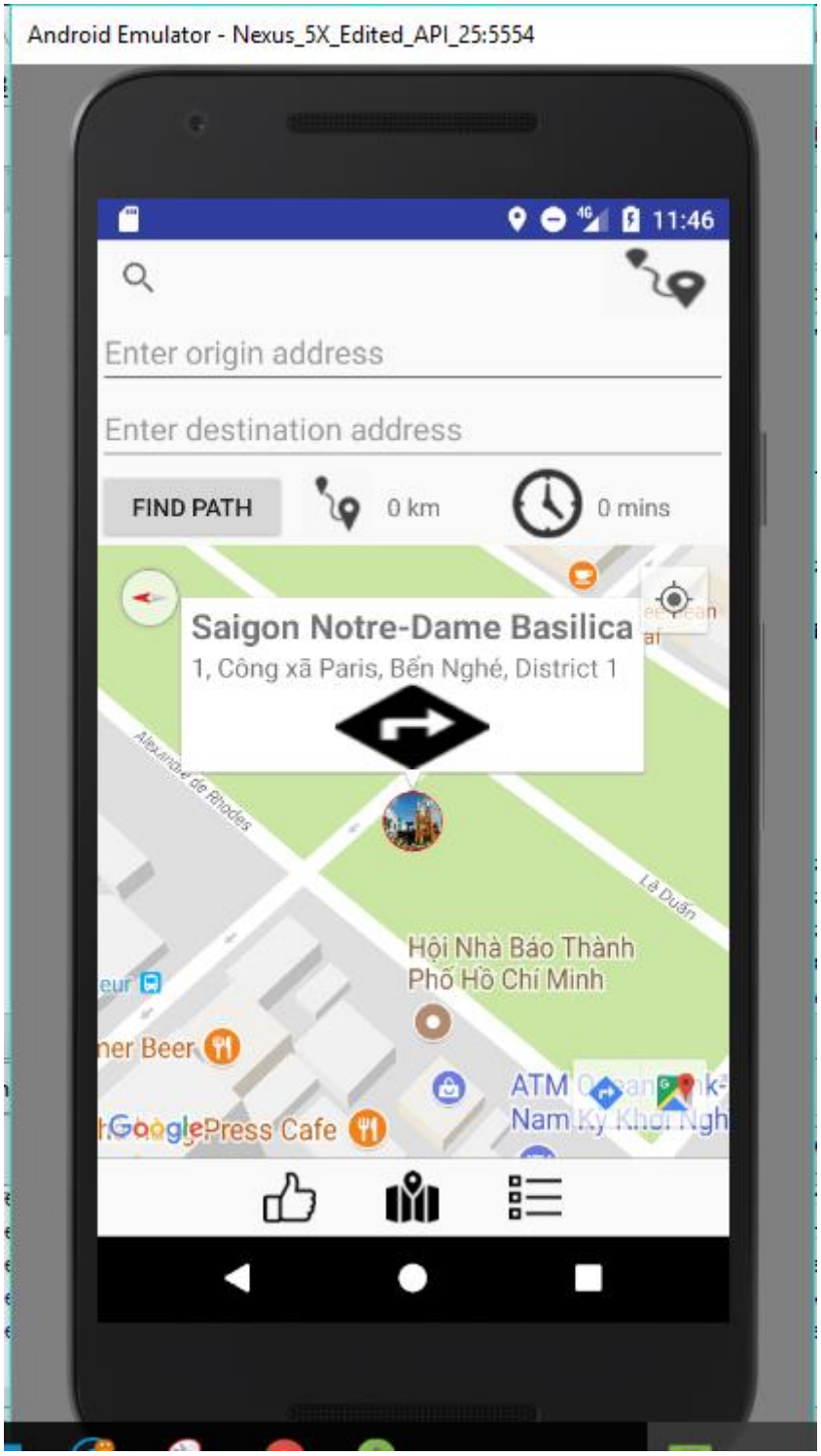
Please list in details all ***advanced features*** in your project.

Screenshots are suitable to illustrate these advanced features.


Android Emulator - Nexus_5X_Edited_API_25:5554
11:05
chợ Tân Định
BỆNH VIỆN QUẬN 1
Bánh Mì Như Lan
Trà Sữa Mix Mix
Trường THPT
Nguyễn Thị Diệu
Fahasa Tan
Dinh Bookstore
Google


Android Emulator - Nexus_5X_Edited_API_25:5554
11:04
HoChiMinhCity
Bánh Xèo
46 Đinh Công Tráng, Tân Định,
save
SHOW ON MAP
0838241110
9 am – 9 pm daily


Android Emulator - Nexus_5X_Edited_API_25:5554
11:06
Bánh Xèo
46 Đinh Công Tráng, Tân Định, District 1
Bánh Mì Như Lan
Trà Sữa Mix Mix
Trường THPT
Nguyễn Thị Diệu
Fahasa Tan
Dinh Bookstore
Google



III. Screenshots

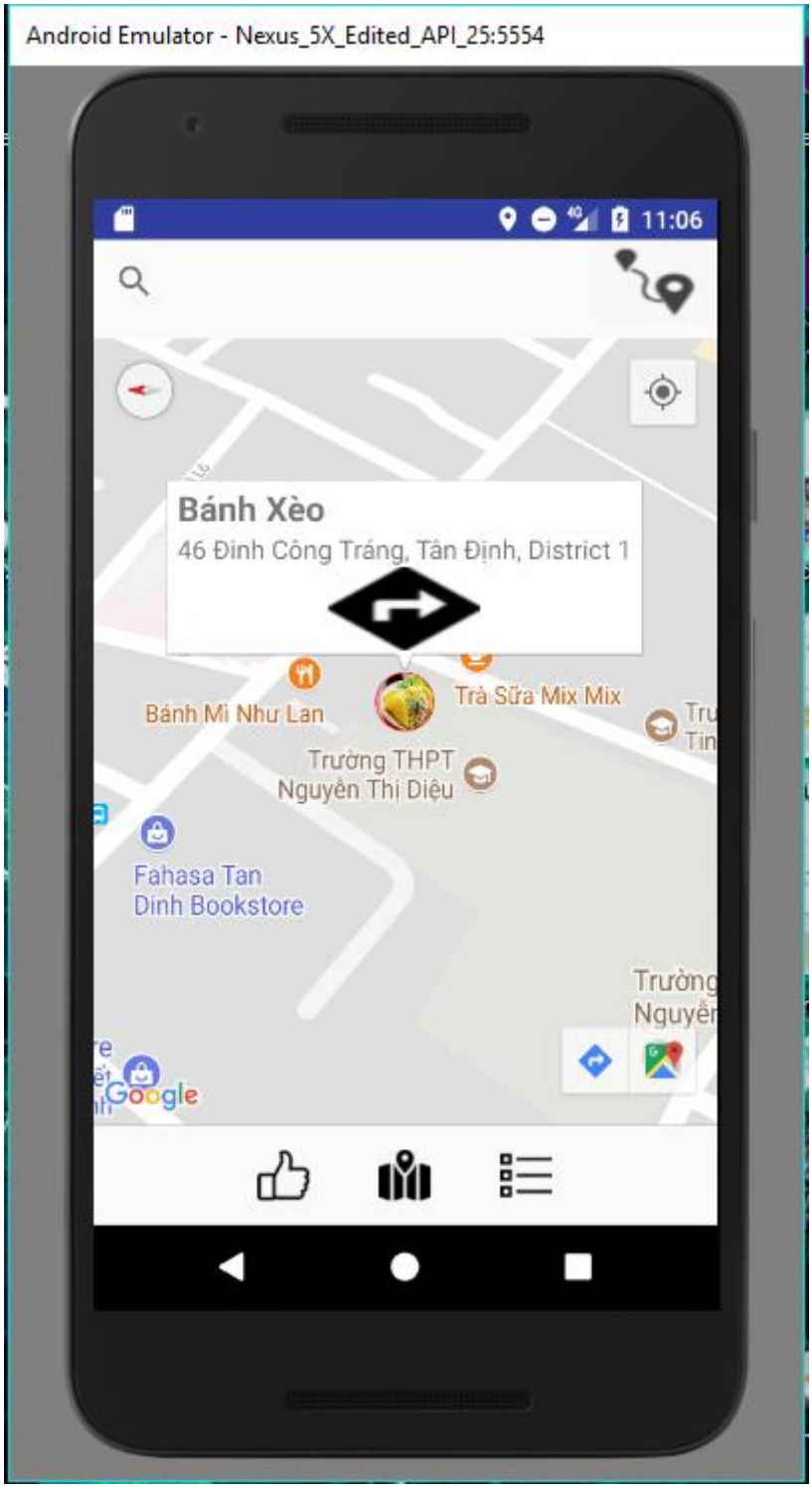
Please insert ***screenshots*** in this section to illustrate ***all main features*** of your project.

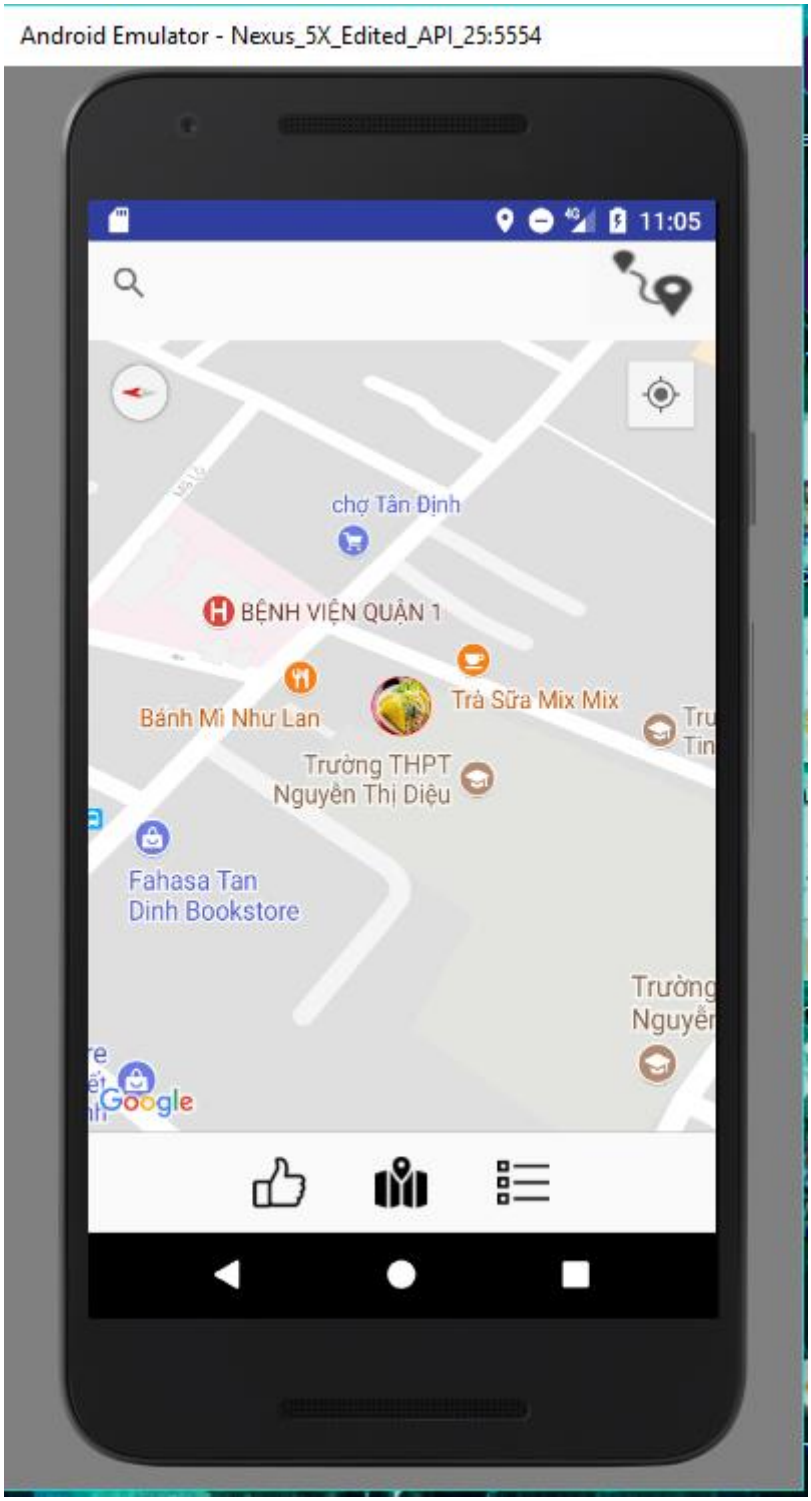
Several screenshots and features ***can be the same*** as in Section II. Advanced Features.


Android Emulator - Nexus_5X_Edited_API_25:5554
11:02
HoChiMinhCity
Top Things in SaiGon
TOP 10
MUST-EAT FOOD
in SaiGon
© Mark Wiens
Recommended Spots
in SaiGon

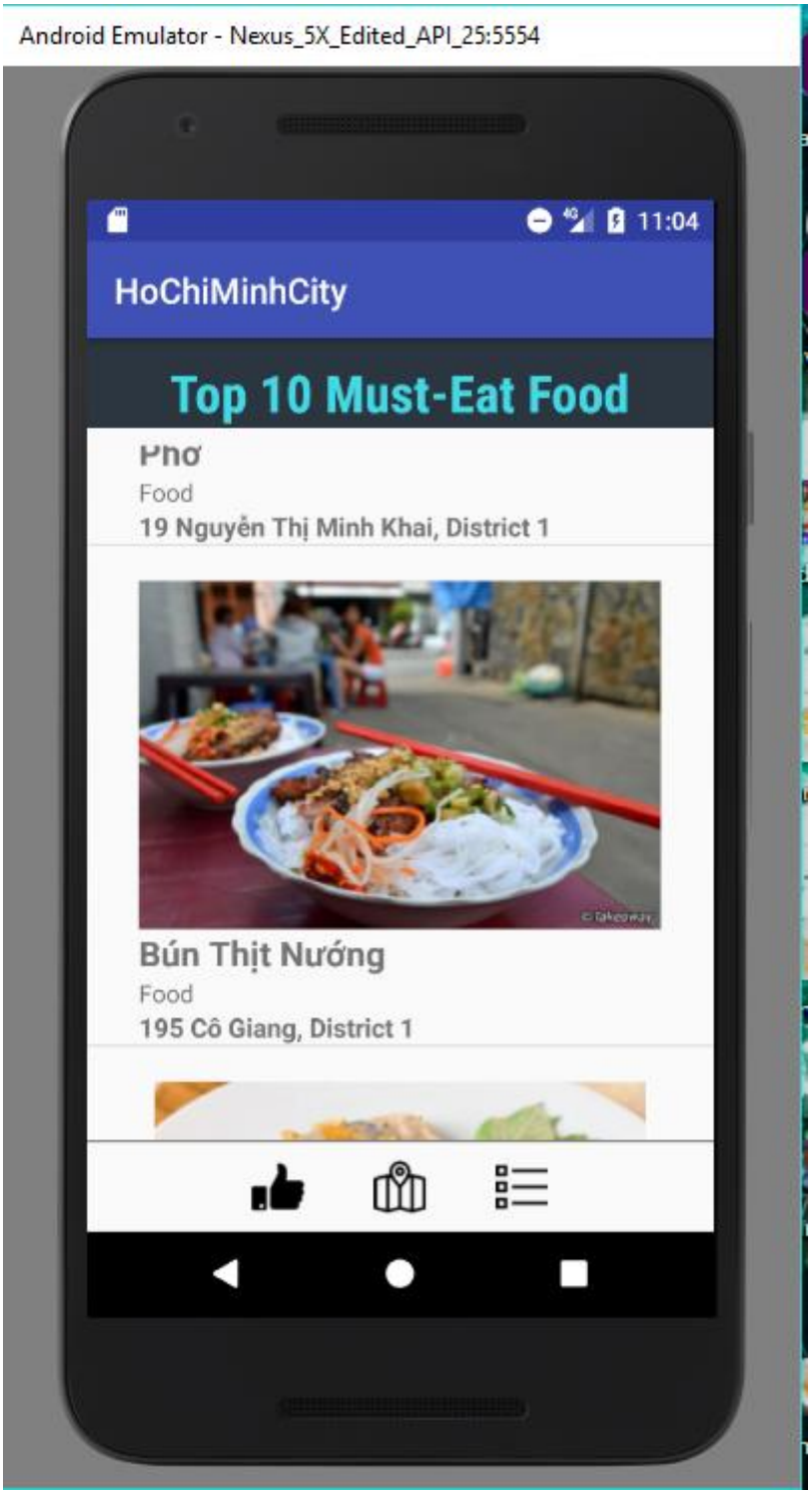

Android Emulator - Nexus_5X_Edited_API_25:5554
11:03
HoChiMinhCity
Top Things in SaiGon
Recommended Spots
in SaiGon
SHOPS
in SaiGon

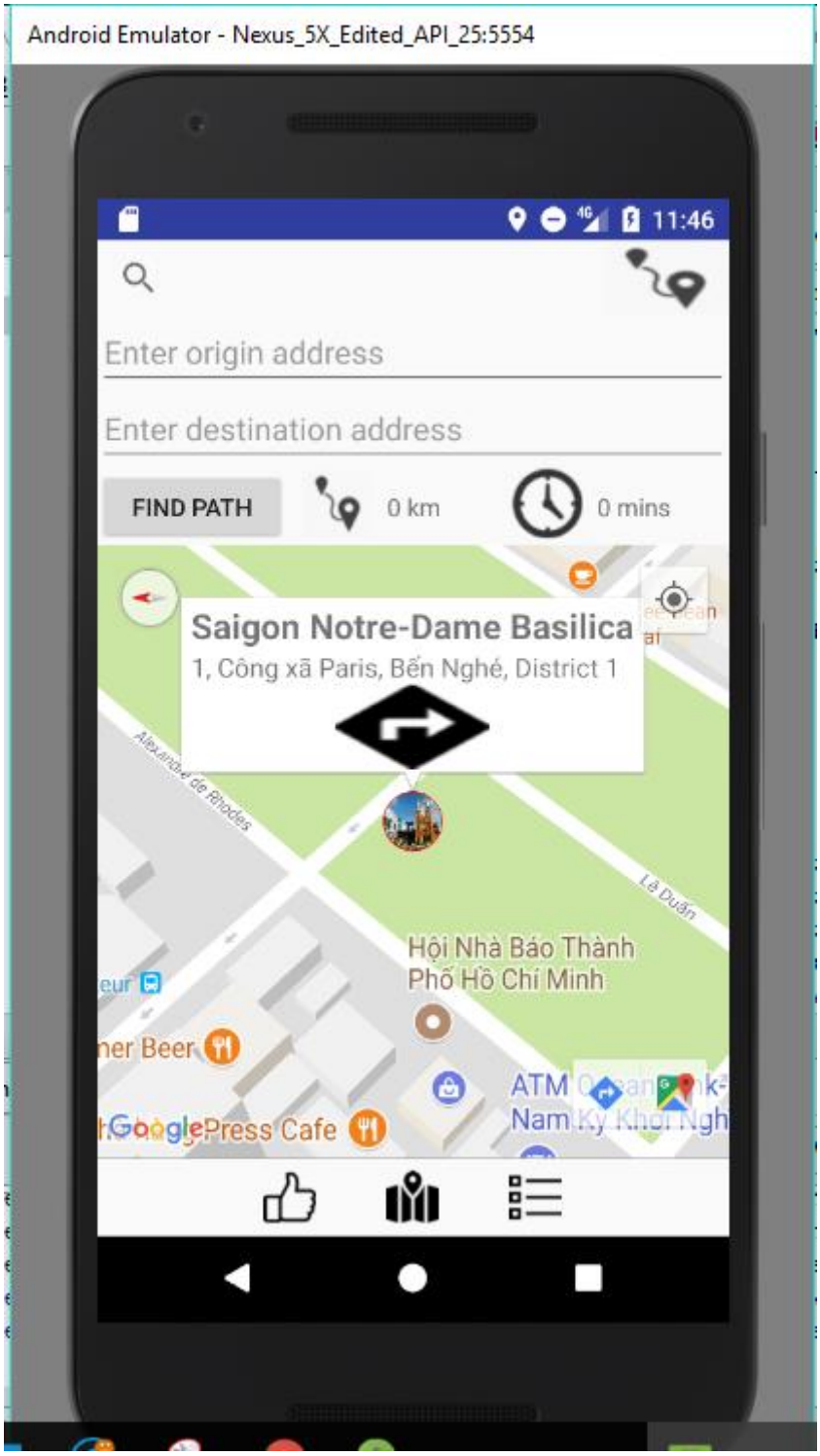

Android Emulator - Nexus_5X_Edited_API_25:5554
11:14
HoChiMinhCity
Bún Thịt Nướng
195 Cô Giang, District 1
save
SHOW ON MAP
0908538079
7 am – 10 pm daily


Android Emulator - Nexus_5X_Edited_API_25:5554
11:06
Bánh Xèo
46 Đinh Công Tráng, Tân Định, District 1
Bánh Mì Như Lan
Trà Sữa Mix Mix
Trường THPT
Nguyễn Thị Diệu
Fahasa Tan
Dinh Bookstore
Google


Android Emulator - Nexus_5X_Edited_API_25:5554
11:05
chợ Tân Định
BỆNH VIỆN QUẬN 1
Bánh Mì Như Lan
Trà Sữa Mix Mix
Trường THPT
Nguyễn Thị Diệu
Fahasa Tan
Dinh Bookstore
Google


Android Emulator - Nexus_5X_Edited_API_25:5554
11:04
HoChiMinhCity
Bánh Xèo
46 Đinh Công Tráng, Tân Định,
save
SHOW ON MAP
0838241110
9 am – 9 pm daily


Android Emulator - Nexus_5X_Edited_API_25:5554
11:04
HoChiMinhCity
Top 10 Must-Eat Food
Phở
Food
19 Nguyễn Thị Minh Khai, District 1
Bún Thịt Nướng
Food
195 Cô Giang, District 1


Android Emulator - Nexus_5X_Edited_API_25:5554
11:46
Enter origin address
Enter destination address
FIND PATH
0 km
0 mins
Saigon Notre-Dame Basilica
1, Công xã Paris, Bến Nghé, District 1
Alexandre de Rhodes
Lê Duẩn
Hội Nhà Báo Thành
Phố Hồ Chí Minh
ner Beer
GooglePress Cafe
ATM
Nam Kỳ Khởi Ngh

IV. References

You should list all source codes and/or references that you use in your project in this section.

Any code fragments or libraries that are not yours ***MUST be explicitly declared*** in this section.

If you fail/forget to declare those "inherited" resources, you will be considered "cheating"!

```
**
 * Created by Mai Thanh Hiep on 4/3/2016.
 */
public class DirectionFinder {
```

```
    private static final String DIRECTION_URL_API =
"https://maps.googleapis.com/maps/api/directions/json?";
    private static final String GOOGLE_API_KEY = "AIzaSyCNQSLnHqg-DBLfY8DZ3kmh02hT1sOMpm4";
    private DirectionFinderListener listener;
    private String origin;
    private String destination;
    private LatLng startPos;
    public DirectionFinder(DirectionFinderListener listener, String origin, String destination)
{
        this.listener = listener;
        this.origin = origin;
        this.destination = destination;
        this.startPos = new LatLng(-1,-1);
    }
    public DirectionFinder(DirectionFinderListener listener,LatLng startPos, String destination)
    {
        this.listener = listener;
        this.destination = destination;
        this.startPos = startPos;

    }
    public void execute() throws UnsupportedEncodingException {
        listener.onDirectionFinderStart();
        new DownloadRawData().execute(createUrl());
    }

    private String createUrl() throws UnsupportedEncodingException {
        String urlOrigin = URLEncoder.encode(origin, "utf-8");
        String urlDestination = URLEncoder.encode(destination, "utf-8");

        return DIRECTION_URL_API + "origin=" + urlOrigin + "&destination=" + urlDestination +
"&key=" + GOOGLE_API_KEY;
    }

    private class DownloadRawData extends AsyncTask<String, Void, String> {

        @Override
        protected String doInBackground(String... params) {
            String link = params[0];
            try {
                URL url = new URL(link);
                InputStream is = url.openConnection().getInputStream();
                StringBuffer buffer = new StringBuffer();
                BufferedReader reader = new BufferedReader(new InputStreamReader(is));

                String line;
                while ((line = reader.readLine()) != null) {
                    buffer.append(line + "\n");
                }

                return buffer.toString();

            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(String res) {
            try {
                parseJSon(res);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }

    private void parseJSon(String data) throws JSONException {
        if (data == null)
            return;

        List<Route> routes = new ArrayList<Route>();
```

```
        JSONObject jsonData = new JSONObject(data);
        JSONArray jsonRoutes = jsonData.getJSONArray("routes");
        for (int i = 0; i < jsonRoutes.length(); i++) {
            JSONObject jsonRoute = jsonRoutes.getJSONObject(i);
            Route route = new Route();

            JSONObject overview_polylineJson = jsonRoute.getJSONObject("overview_polyline");
            JSONArray jsonLegs = jsonRoute.getJSONArray("legs");
            JSONObject jsonLeg = jsonLegs.getJSONObject(0);
            JSONObject jsonDistance = jsonLeg.getJSONObject("distance");
            JSONObject jsonDuration = jsonLeg.getJSONObject("duration");
            JSONObject jsonEndLocation = jsonLeg.getJSONObject("end_location");

            route.distance = new Distance(jsonDistance.getString("text"),
jsonDistance.getInt("value"));
            route.duration = new Duration(jsonDuration.getString("text"),
jsonDuration.getInt("value"));
            route.endAddress = jsonLeg.getString("end_address");
            if(startPos.latitude>=0&& startPos.longitude>=0)
            {
                route.startLocation = new LatLng(startPos.latitude,startPos.longitude);
            }
            else {
                JSONObject jsonStartLocation = jsonLeg.getJSONObject("start_location");
                route.startAddress = jsonLeg.getString("start_address");
                route.startLocation = new LatLng(jsonStartLocation.getDouble("lat"),
jsonStartLocation.getDouble("lng"));
            }
            route.endLocation = new LatLng(jsonEndLocation.getDouble("lat"),
jsonEndLocation.getDouble("lng"));
            route.points = decodePolyLine(overview_polylineJson.getString("points"));

            routes.add(route);
        }

        listener.onDirectionFinderSuccess(routes);
    }

    private List<LatLng> decodePolyLine(final String poly) {
        int len = poly.length();
        int index = 0;
        List<LatLng> decoded = new ArrayList<LatLng>();
        int lat = 0;
        int lng = 0;

        while (index < len) {
            int b;
            int shift = 0;
            int result = 0;
            do {
                b = poly.charAt(index++) - 63;
                result |= (b & 0x1f) << shift;
                shift += 5;
            } while (b >= 0x20);
            int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
            lat += dlat;

            shift = 0;
            result = 0;
            do {
                b = poly.charAt(index++) - 63;
                result |= (b & 0x1f) << shift;
                shift += 5;
            } while (b >= 0x20);
            int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
            lng += dlng;

            decoded.add(new LatLng(
                    lat / 100000d, lng / 100000d
            ));
        }

        return decoded;
    }
}
```

```java
package com.example.phy.hochiminhcity;

import android.content.Context;
import android.graphics.Point;
import android.util.AttributeSet;
import android.view.MotionEvent;
import android.view.View;
import android.widget.LinearLayout;

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.Marker;

/**
 * Created by Phy on 6/5/2017.
 */

class MapWrapperLayout extends LinearLayout{
    private GoogleMap map;

    private int bottomOffsetPixels;
    private Marker marker;

    private View infoWindow;

    public MapWrapperLayout(Context context) {
        super(context);
    }

    public MapWrapperLayout(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public MapWrapperLayout(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
    }


    public void init(GoogleMap map, Context context) {
        this.map = map;
        this.bottomOffsetPixels = getPixelsFromDp(context, 30+ 29);
    }


    public void setMarkerWithInfoWindow(Marker marker, View infoWindow) {
        this.marker = marker;
        this.infoWindow = infoWindow;
    }

    @Override
    public boolean dispatchTouchEvent(MotionEvent ev) {
        boolean ret = false;
        // Make sure that the buildingInfoWindow is shown and we have all the needed references
        if (marker != null && marker.isInfoWindowShown() && map != null && infoWindow != null) {
            // Get a marker position on the screen
            Point point = map.getProjection().toScreenLocation(marker.getPosition());

            // Make a copy of the MotionEvent and adjust it's location
            // so it is relative to the buildingInfoWindow left top corner
            MotionEvent copyEv = MotionEvent.obtain(ev);
            copyEv.offsetLocation(
                    -point.x + (infoWindow.getWidth() / 2),
                    -point.y + infoWindow.getHeight() + bottomOffsetPixels);

            // Dispatch the adjusted MotionEvent to the buildingInfoWindow
            ret = infoWindow.dispatchTouchEvent(copyEv);
        }
        // If the buildingInfoWindow consumed the touch event, then just return true.
        // Otherwise pass this event to the super class and return it's result
        return ret || super.dispatchTouchEvent(ev);
    }
```

```
    private int getPixelsFromDp(Context context, float dp) {
        final float scale = context.getResources().getDisplayMetrics().density;
        return (int)(dp * scale + 0.5f);
    }
}
abstract class OnInterInfoWindowTouchListener implements OnTouchListener {

    private final View view;
    private final Handler handler = new Handler();

    private Marker marker;
    private boolean pressed = false;

    public OnInterInfoWindowTouchListener(View view) {
        this.view = view;
    }

    public void setMarker(Marker marker) {
        this.marker = marker;
    }

    @Override
    public boolean onTouch(View vv, MotionEvent event) {
        if (0 <= event.getX() && event.getX() <= view.getWidth() &&
                0 <= event.getY() && event.getY() <= view.getHeight())
        {
            switch (event.getActionMasked()) {
                case MotionEvent.ACTION_DOWN: startPress(); break;

                // We need to delay releasing of the view a little so it shows the pressed state
on the screen
                case MotionEvent.ACTION_UP: handler.postDelayed(confirmClickRunnable, 150);
break;

                case MotionEvent.ACTION_CANCEL: endPress(); break;
                default: break;
            }
        }
        else {
            // If the touch goes outside of the view's area
            // (like when moving finger out of the pressed button)
            // just release the press
            endPress();
        }
        return false;
    }

    private void startPress() {
        if (!pressed) {
            pressed = true;
            handler.removeCallbacks(confirmClickRunnable);
            if (marker != null)
                marker.showInfoWindow();
        }
    }

    private boolean endPress() {
        if (pressed) {
            this.pressed = false;
            handler.removeCallbacks(confirmClickRunnable);
            if (marker != null)
                marker.showInfoWindow();
            return true;
        }
        else
            return false;
    }

    private final Runnable confirmClickRunnable = new Runnable() {
        public void run() {
            if (endPress()) {
                onClickConfirmed(view, marker);
            }
        }
```

```
    };

    /**
     * This is called after a successful click
     */
    protected abstract void onClickConfirmed(View v, Marker marker);
}



super.onCreate(savedInstanceState);
setContentView(R.layout.activity_maps);

intentThatCalled = getIntent();
voice2text = intentThatCalled.getStringExtra("v2txt");
getLocation();



View.OnClickListener x = new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        sendRequest(0);
    }
};
private void sendRequest(int id) {
    String origin = edtOrigin.getText().toString();
    String destination = edtDestination.getText().toString();
    LatLng startPos = new LatLng(latitude,longitude);
    if (origin.isEmpty()&&id==0) {
        Toast.makeText(this, "Please enter origin address!", Toast.LENGTH_SHORT).show();
        return;
    }

    if (destination.isEmpty()&&id==0) {
        Toast.makeText(this, "Please enter destination address!", Toast.LENGTH_SHORT).show();
        return;
    }

    try {
        if(id == 0) {

            new DirectionFinder(this, origin, destination).execute();
        }
        else
        {
            new DirectionFinder(this, startPos, destination).execute();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
private void getLocation() {
    if (isLocationEnabled(MapsActivity.this)) {
        locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);
        criteria = new Criteria();
        bestProvider = String.valueOf(locationManager.getBestProvider(criteria,
true)).toString();

        //You can still do this if you like, you might get lucky:
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
!= PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        Location location = locationManager.getLastKnownLocation(bestProvider);
        if (location != null) {
            Log.e("TAG", "GPS is on");
```

```
            latitude = location.getLatitude();
            longitude = location.getLongitude();
            Toast.makeText(MapsActivity.this, "latitude:" + latitude + " longitude:" +
longitude, Toast.LENGTH_SHORT).show();
            searchNearestPlace(voice2text);
        }
        else{
            //This is what you need:
            locationManager.requestLocationUpdates(bestProvider, 1000, 0, this);
        }
    }
    else
    {
        //prompt user to enable location....
        //.................
    }
}

@Override
public void onLocationChanged(Location location) {
    //Hey, a non null location! Sweet!

    //remove location callback:
    locationManager.removeUpdates(this);

    //open the map:
    latitude = location.getLatitude();
    longitude = location.getLongitude();
    Toast.makeText(MapsActivity.this, "latitude:" + latitude + " longitude:" + longitude,
Toast.LENGTH_SHORT).show();
}
```