CSC 392/492: Mobile Applications Development for Android II

Assignment 3 – Special Offer App *(400 pts)*

Uses: Google Maps Activity, Google Location Services, Location Listener, Geocoding, Geofencing, Full-screen, QR Codes, Custom Fonts

A) Special Offer App General Description

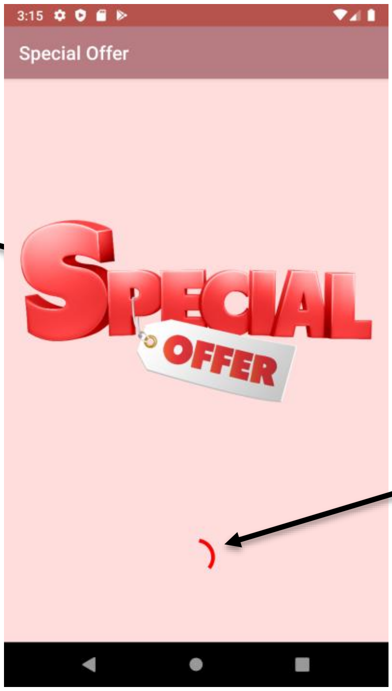
- This app allows users to be alerted to local businesses offering special deals when they are near the business location. Alerts include info on the special offer that is displayed in the app when selected.
- This app will use a Google Maps Activity to display a map of the user's nearby surroundings, to display the Geofence ranges, and to track their movement since the app was opened.

B) Behavior Use Cases and Activities:

- The user's live location is used to update their map position
- The Google Maps Activity will make use of location data provided continuously by a Location Listener to determine the user's current location and to plot their location on the map.
- The map continuously centers itself on the user's current location. The current location is shown via an automobile icon that is oriented to show the users direction of travel. Their path on the map is displayed as a solid line that marks where they have been.
- Geofences are loaded from a cloud-based source (http://www.christopherhield.com/data/fences.json) so they can change and update without requiring app updates. As the user moves, they are alerted if they enter the range of any of the Geofences. Note Geofences can overlap, intersect, etc.
- The deals offered to the user can be viewed in their own activity by tapping on the Special Offer notifications that will appear in the notification bar. Multiple notifications may be present if the user crosses several Geofence boundaries.
- Special Offers content includes the logo of the business, the name and address, the website, and what the special offer includes. A QR code is displayed which is to be scanned when redeeming the offer at the specified business.
- When the app is stopped, the Special Offer notifications should be removed.
- When the app is started, no history of previously displayed offers or any previous route path should be displayed.
- A 3rd party QR code-generation library will be used for display of special offer discount codes.

C) Activity/behavior diagrams

1) Startup “splash” activity:

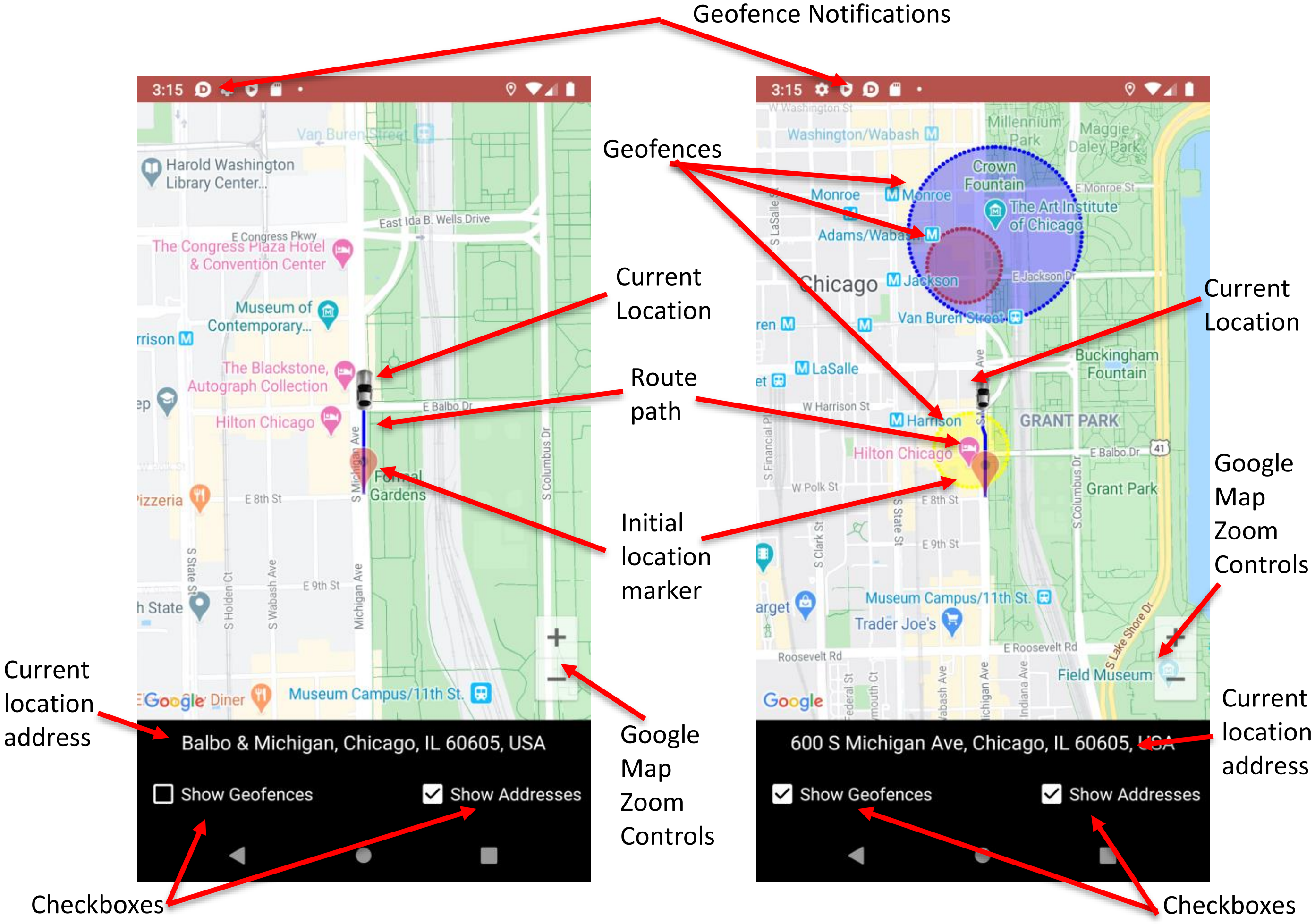


App Logo

Indeterminate ProgressBar

The Splash Activity is used to cover much of the app’s initialization tasks.
Once the critical permissions and data content have been acquired, we transition to the Map Activity.

2) Google Map Activity examples:



Geofence Notifications

Geofences

Current Location

Route path

Initial location marker

Google Map Zoom Controls

Current location address

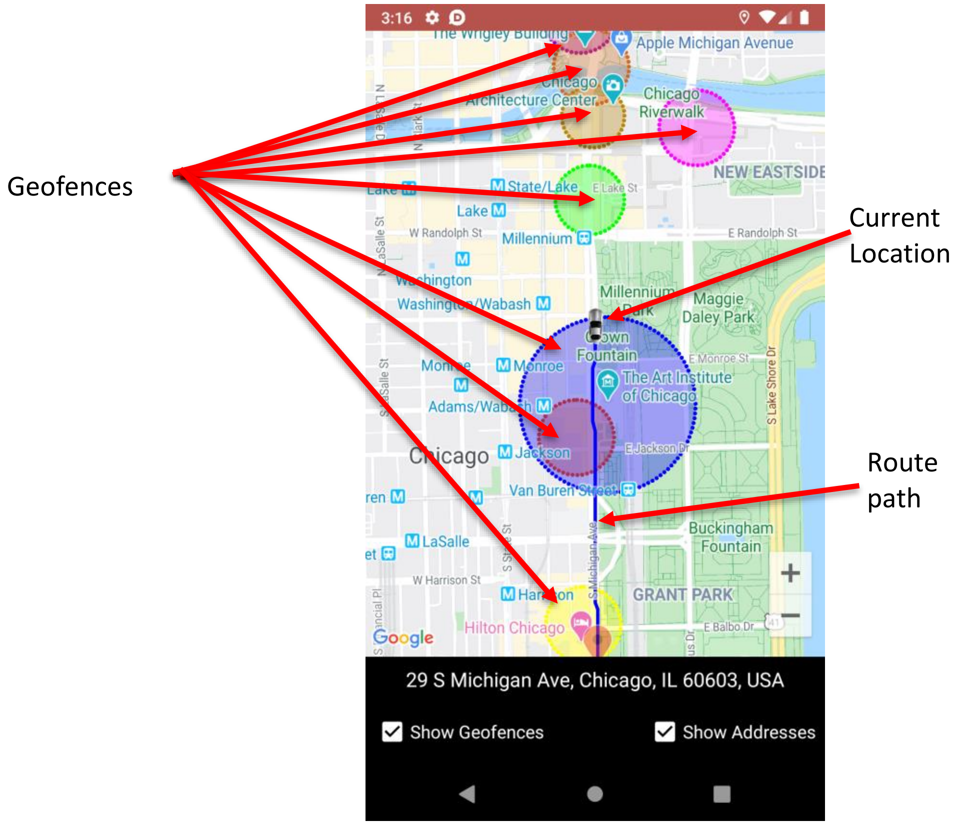
Checkboxes

Balbo & Michigan, Chicago, IL 60605, USA

☐ Show Geofences ☑ Show Addresses

600 S Michigan Ave, Chicago, IL 60605, USA

☑ Show Geofences ☑ Show Addresses

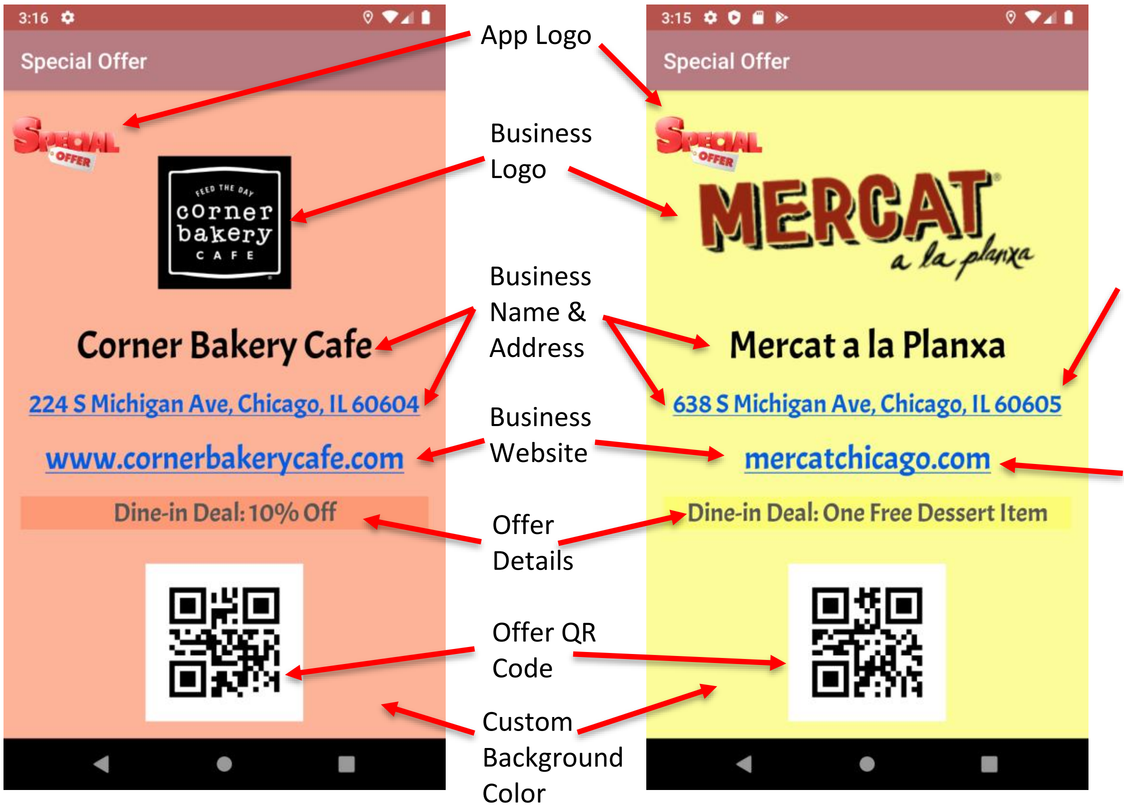


29 S Michigan Ave, Chicago, IL 60603, USA

☑ Show Geofences ☑ Show Addresses

3) Special Offer Activity examples

Special Offer data is specified in a cloud-based JSON file (described in the next section)



Special Offer

Corner Bakery Cafe

224 S Michigan Ave, Chicago, IL 60604

www.cornerbakerycafe.com

Dine-in Deal: 10% Off

Special Offer

Mercat a la Planxa

638 S Michigan Ave, Chicago, IL 60605

mercatchicago.com

Dine-in Deal: One Free Dessert Item

The Business Address should be clickable – takes you to that location in the ***google maps*** *app.*

The Business Website should be clickable – takes you to the business' website in a ***web browser****.*

D) Geofence Data Download

The Geofences to be used in this app are defined in a JSON file that should be accessed from within your app at: http://www.christopherhield.com/data/fences.json

The fence JSON file contains one outer JSON Object with one field called "fences", which refers to a JSON Array containing the fences. A sample of that file is shown below (*while all data fields will be used, the highlighted text is for explanation purposes only and does not appear in the JSON file*):

```
{
  "fences": [
    {
      "id":  "Mercat a la Planxa",
      "address": "638 S Michigan Ave, Chicago, IL 60605",
      "website": "mercatchicago.com",
      "radius": 120.0,  ← This is used when creating the Geofence to specify its radius
      "type": 1, ← Used to specify how the fence notifies users, 1 = GEOFENCE_TRANSITION_ENTER
      "message": "Dine-in Deal: One Free Dessert Item",
      "code": "F983HF-340",  ← Used to generate QR code
      "fenceColor":"#FFFF00",  ← Used to set Special Offer Activity background color
      "logo": "https://assets.hospitalityonline.com/brands/employers/logos/000/266/729/logo.png"
    },
    {
      "id":  "Erie Cafe",
      "address": "536 W Erie St, Chicago, IL 60654",
      "website": "eriecafe.com",
      "radius": 130.0,  ← This is used when creating the Geofence to specify its radius
      "type": 1,  ← Used to specify how the fence notifies users, 1 = GEOFENCE_TRANSITION_ENTER
      "message": "Dine-in Deal: One Free Appetizer Item",
      "code": "EC9098",  ← Used to generate QR code
      "fenceColor":"#0000FF",  ← Used to set Special Offer Activity background color
      "logo": "http://food.meganbuttita.com/wp-content/uploads/2008/07/erie.gif"
    },
    ...
    {
      "id":  "Stix n Brix Wood Fired Pizza",
      "address": "220 W 33rd St, Chicago, IL 60616",
      "website": "stixnbrix33.com",
      "radius": 225.0,  ← This is used when creating the Geofence to specify its radius
      "type": 1,  ← Used to specify how the fence notifies users, 1 = GEOFENCE_TRANSITION_ENTER
      "message": "Dine-in Deal: 10% Off Any Appetizer Selection",
      "code": "CKJWGD8",  ← Used to generate QR code
      "fenceColor":"#999900",  ← Used to set Special Offer Activity background color
      "logo": "https://stixnbrix33.files.wordpress.com/2018/11/stixicon.png"
    }
  ]
}
```

Note that android Geofences are created using lat/lon coordinates, not street addresses. You need to convert a street address to lat/lon coordinates using a GeoCoder object (via its *getFromLocationName* method).

E) Custom Font

All TextViews, EditTexts, CheckBoxes, etc in this app should use the provided "Acme-Regular" font. The example code below is provided as a refresher/guide on using custom fonts.

Note, the provided Acme-Regular.ttf font file must be put in the following path:

Project → app → src → main → assets → fonts → Acme-Regular.ttf

To load:

```
   Typeface textFont = Typeface.createFromAsset(getAssets(), "fonts/Acme-Regular.ttf");
```

To use:

```
   myText.setTypeface(textFont);
```

F) QR Codes

We will use the “ZXing” library to generate QR codes from Strings. ZXing (“zebra crossing”) is a barcode image processing library (Google Open Source) that supports bar and QR type codes.

To use this library of classes in your project, add the following 2 lines to your app-level build.grade file in the “dependencies” section along with the other library dependencies:

```
implementation "com.google.zxing:core:3.3.0"
implementation 'com.google.zxing:javase:3.3.0'
```

The example code below is provided as a refresher/guide on using the XZing library:

```
import android.graphics.Bitmap;
import com.google.zxing.BarcodeFormat;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.qrcode.QRCodeWriter;

QRCodeWriter writer = new QRCodeWriter();
try {
    BitMatrix bitMatrix = writer.encode(stringToEncode, BarcodeFormat.QR_CODE, 512, 512);

    int width = bitMatrix.getWidth();
    int height = bitMatrix.getHeight();

    // The below Bitmap is what will be displayed in an ImageView in the Activity
    Bitmap bmp = Bitmap.createBitmap(width, height, Bitmap.Config.RGB_565);
    for (int x = 0; x < width; x++) {
         for (int y = 0; y < height; y++) {
                bmp.setPixel(x, y, bitMatrix.get(x, y) ? Color.BLACK : Color.WHITE);
         }
    }
    // The below line uses the Bitmap just created as the ImageView’s image bitmap
    ((ImageView) findViewById(R.id.myImageView)).setImageBitmap(bmp);

} catch (WriterException e) {
    e.printStackTrace();
}
```

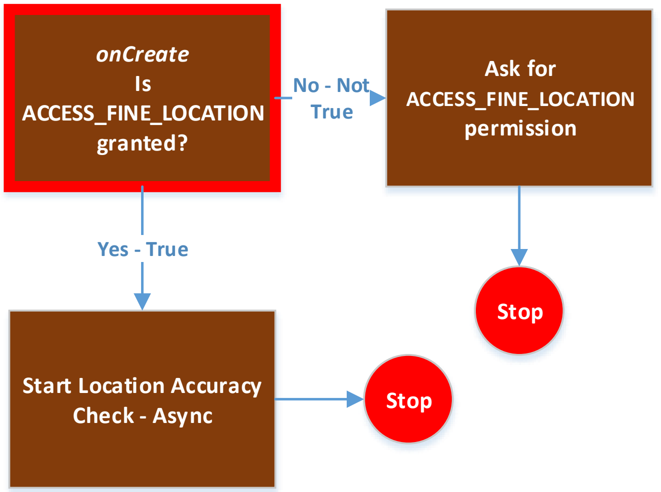
QR String Encoding Examples:

“CSC 392/492” ➔

“Mobile Applications Development for Android II” ➔

G) App Startup Behavior Diagrams

SplashActivity:



onCreate Is ACCESS_FINE_LOCATION granted?

No - Not True

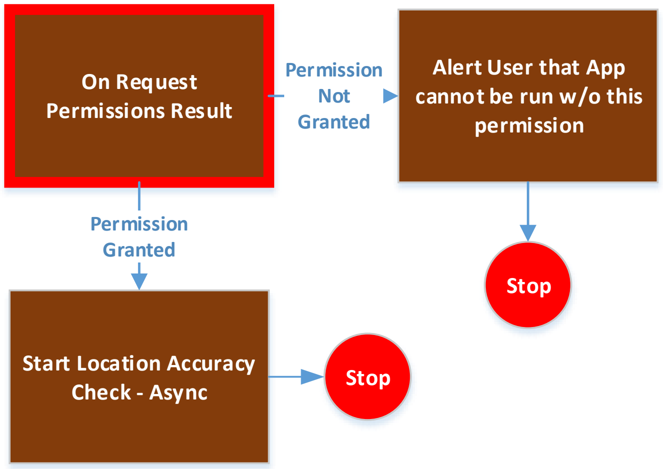
Ask for ACCESS_FINE_LOCATION permission

Stop

Yes - True

Start Location Accuracy Check - Async

Stop



On Request Permissions Result

Permission Not Granted

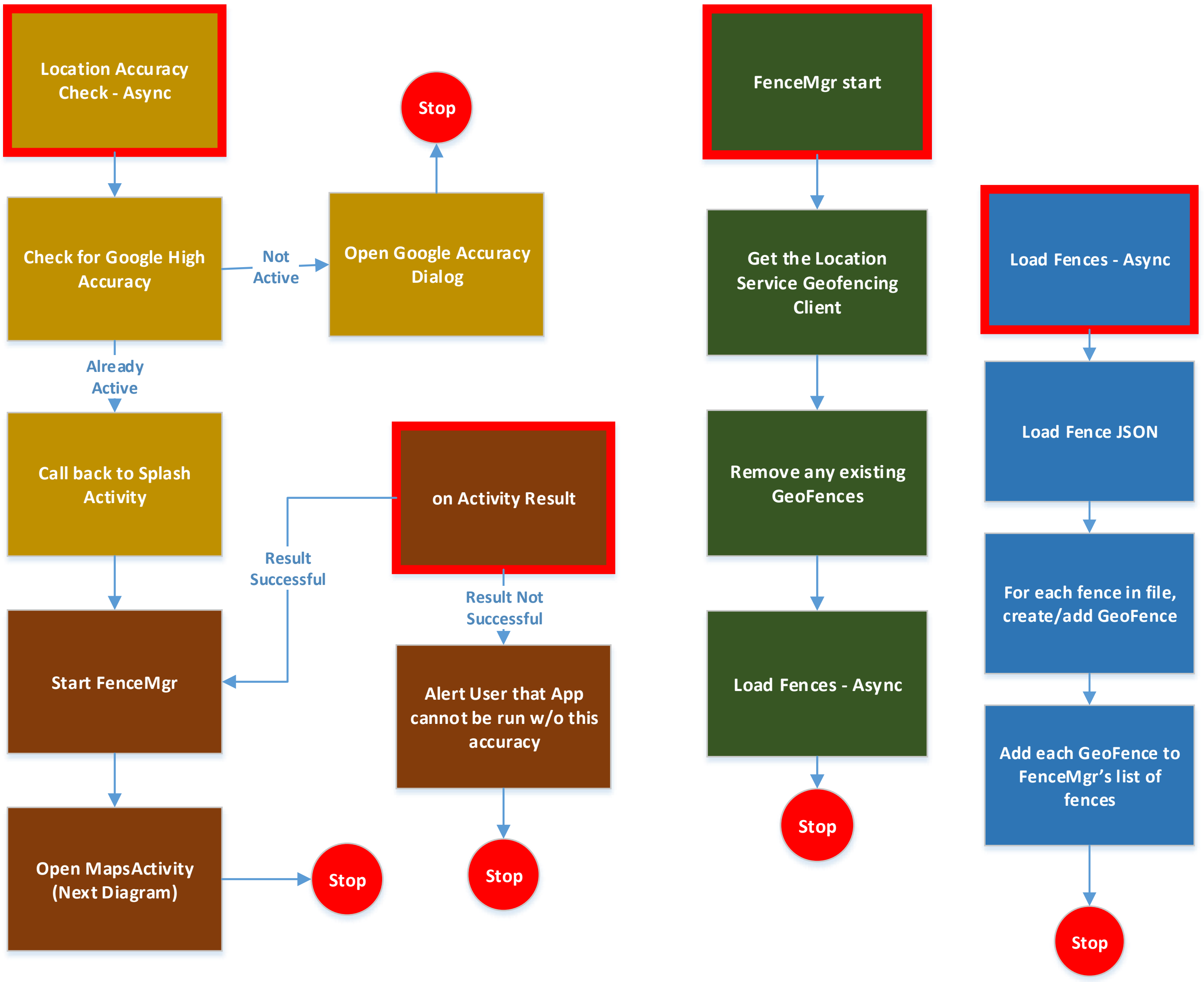
Alert User that App cannot be run w/o this permission

Stop

Permission Granted

Start Location Accuracy Check - Async

Stop



Location Accuracy Check - Async

Check for Google High Accuracy

Not Active

Open Google Accuracy Dialog

Stop

Already Active

Call back to Splash Activity

Start FenceMgr

Open MapsActivity (Next Diagram)

Stop

on Activity Result

Result Successful

Result Not Successful

Alert User that App cannot be run w/o this accuracy

Stop

FenceMgr start

Get the Location Service Geofencing Client

Remove any existing GeoFences

Load Fences - Async

Stop

Load Fences - Async

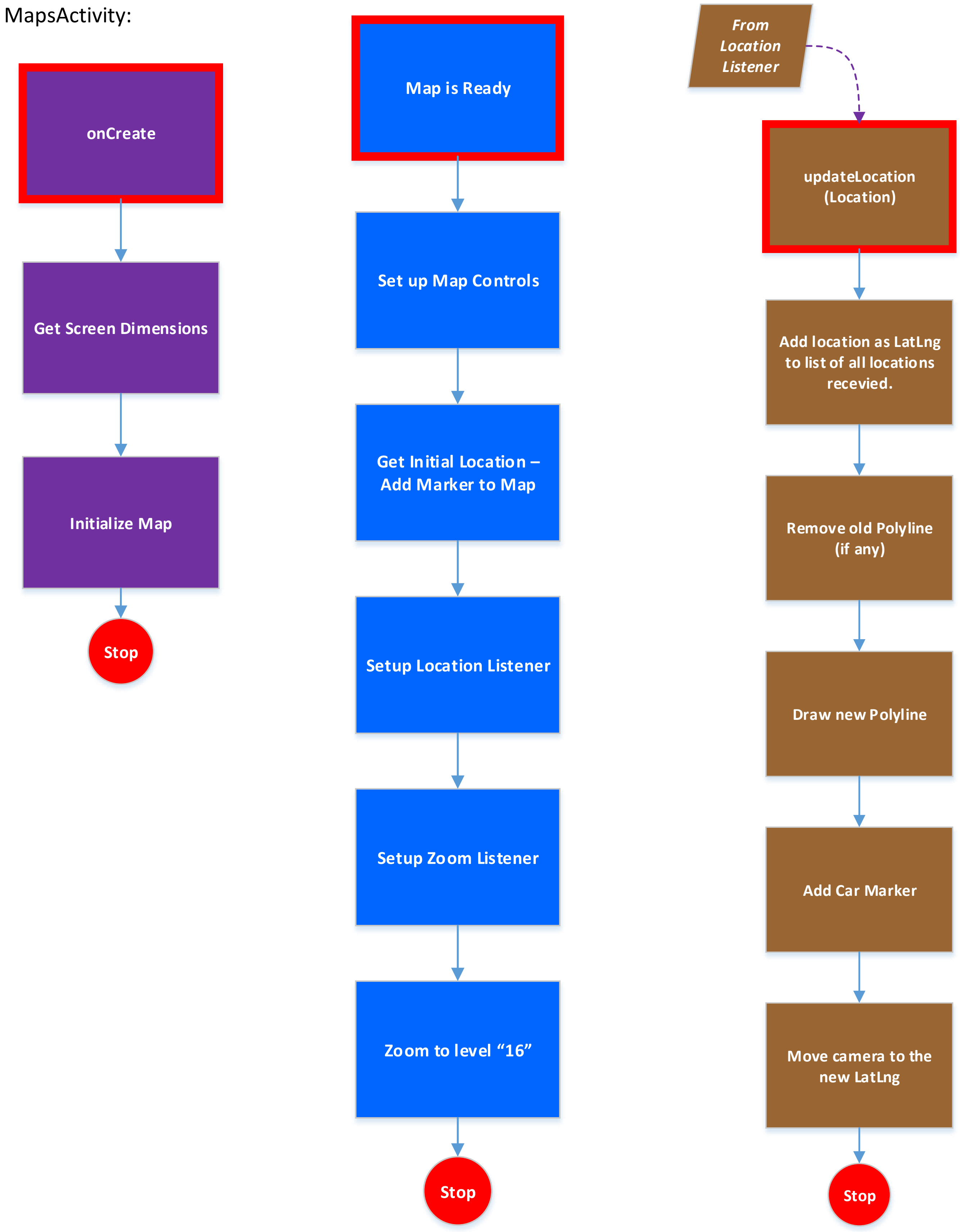
Load Fence JSON

For each fence in file, create/add GeoFence

Add each GeoFence to FenceMgr's list of fences

Stop

MapsActivity:



Assignment Assistance

If you are stuck on an assignment problem that you have exhaustively researched and/or debugged yourself, you can email me a ZIP file of your entire project so that I can examine the problem. All emailed assistance requests must include a detailed description of the problem, and the details of what steps you have already taken in trying to determine the source of the problem.

Note: To make your submission zip file smaller, before zipping your project file you must remove the ".gradle" folder (found in your project's root directory), and remove the "build" folder (found in the "app" folder in your project's root directory).

Submissions & Grading

1) Submissions must consist of your zipped project folder. For submission - before zipping your project file you *must* remove the ".gradle" folder (found in your project's root directory), and remove the "build" folder (found in the "app" folder in your project's root directory). Submissions not following these requirements will be penalized.
2) Submissions should reflect the concepts and practices we cover in class, and the requirements specified in this document.
3) Late submissions will be penalized by 10% per week late. (i.e., from one second late to 1 week late: 10% penalty, from one week late to 2 weeks late: 20% penalty, etc.).
4) Grading will be based upon the presence and proper functionality of all features and behaviors described in this document.

NOTE

This assignment is worth 400 points. This means (for example) that if you get 89% on this assignment, your recorded score will be:

(89% * 400 points = 356 points)

If you do not understand anything in this handout, please ask.

Otherwise the assumption is that you understand the content.

Unsure? Ask!