



# Augmented Reality Assisted Navigation Application using Google Maps

*Submitted by:*

*Jediael Meshua Sumer **DC2016BTE0139***

*Marlom Bey **DC2016BTE0195***

*Mriganka Shekhar Sarmah **DC2016BTE0202***

# What is *NavAR*?

- NavAR is an application that utilises the phone's camera and augmented reality to assist in navigation.
- The application scans the road/path ahead and overlays directions directly on the road through augmented reality.
- This application aims to provide an extra layer of convenience for complex motorways, tight streets and navigation in general

# Objective

To use the phone's camera and augmented reality along with Google Maps to provide real-time navigation assistance.

# Existing Systems

ARCity by blippAR (only on iOS)

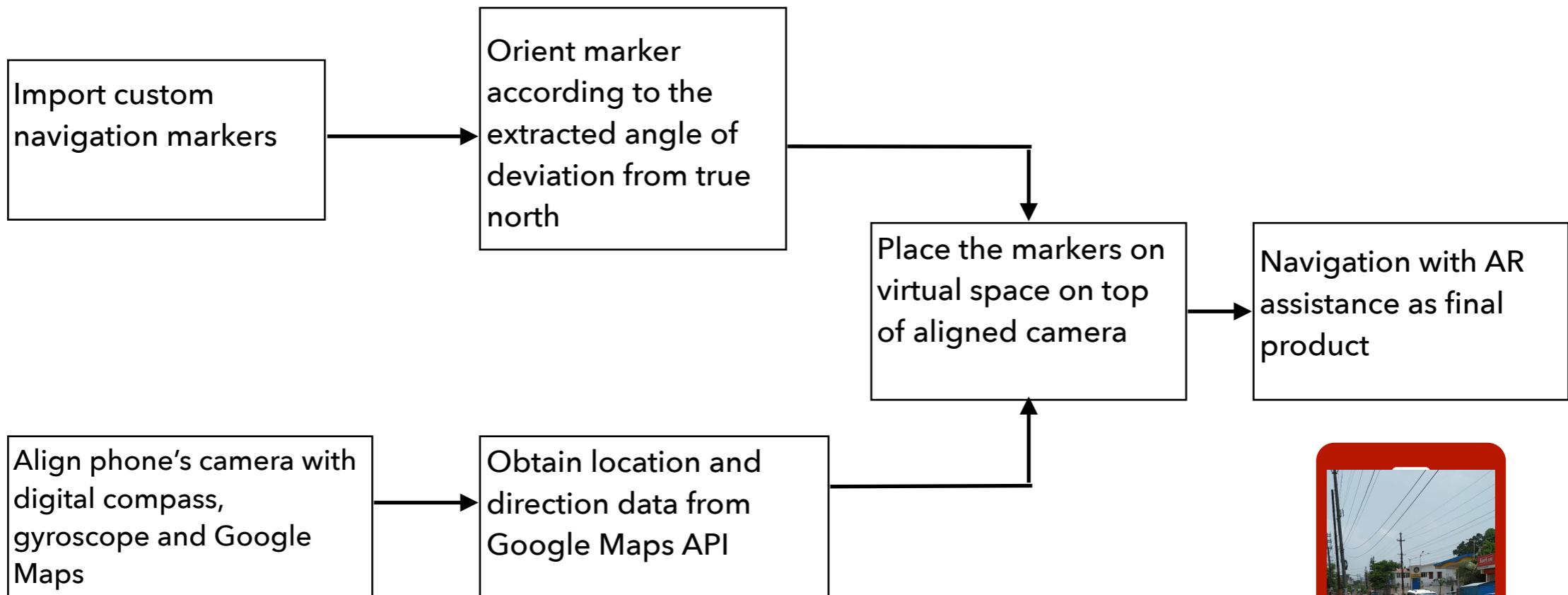


# Existing Systems

AR Navigation & Nearby GPS, AR Walking Navigation by AR Technology | GPS Navigation | Future Apps (only on Google Play Store)



# Proposed Plan



# REQUIREMENT ANALYSIS

# Hardware Requirements

- Processor: Intel Pentium 4 (With SSE2 instruction set support) or later
- RAM: 3 GB or more
- GPU: DirectX 10 (shader model 4.0) support
- Disk Space: 3 GB or more
- Screen Resolution: 1280X800 or more
- Android Phone with Android 7 and above

# Software Requirements

- OS: Windows 7 SP1 or later, MacOS 10.11 or later
- Android Studio 3.3
- ARCore 1.7
- JDK 8
- Programming Languages: JavaScript, Java, C#

# COCOMO Model

For a semi-detached COCOMO model, the coefficients  $a_b$   $b_b$   $c_b$   $d_b$  are given by:

$$a_b = 3.0$$

$$b_b = 1.12$$

$$c_b = 2.5$$

$$d_b = 0.35$$

**Effort Applied,  $E = a_b(KLoC)^{b_b}$**   
[person-months]

$$E = 3.0 * (4K)^{1.12} \text{ PM}$$

$$E = 14.17 \text{ PM}$$

**Development Time, D=**  
 $c_b(\text{Effort Applied})^{d_b}$   
[months]

$$D=2.5*(14.17)^{0.35} \text{ months}$$

$$D=6.32 \text{ months}$$

**People Required, P=Effort**  
Applied/Development time  
[count]

P=14.17/6.32 count

P=2.24 count

P≈3 count

# Technical Feasibility

The requirements for making this application are as follows:

- Android Studio
- ARCore
- Google Maps SDK (free for one year)
- Google Places API (free for one year)
- Google Directions API (free for one year)

The above mentioned utilities are open source.

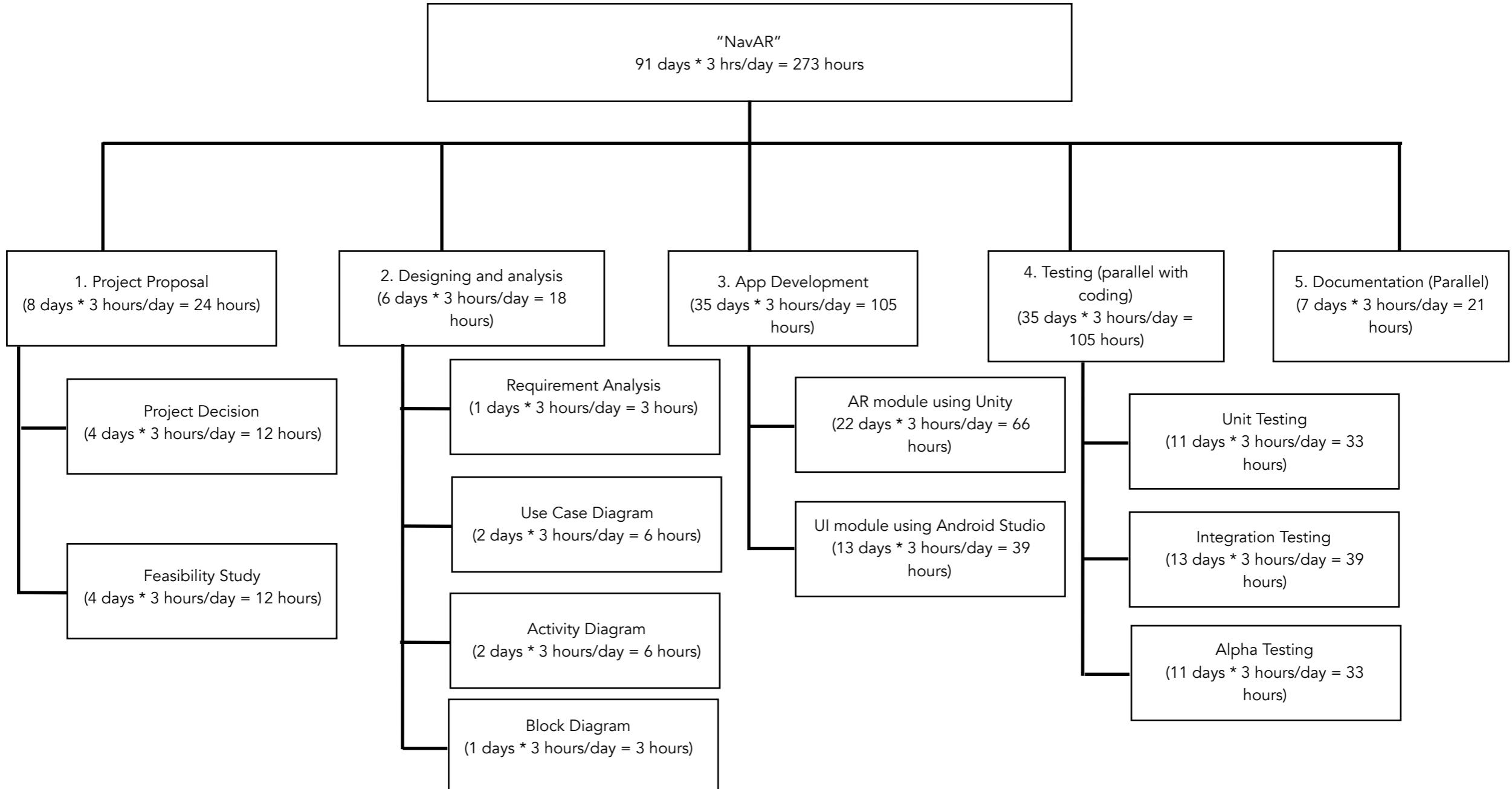
# Operational Feasibility

- The finished product will have an easy-to-use, simple and intuitive interface.
- The end user does not require any special training to use this application.

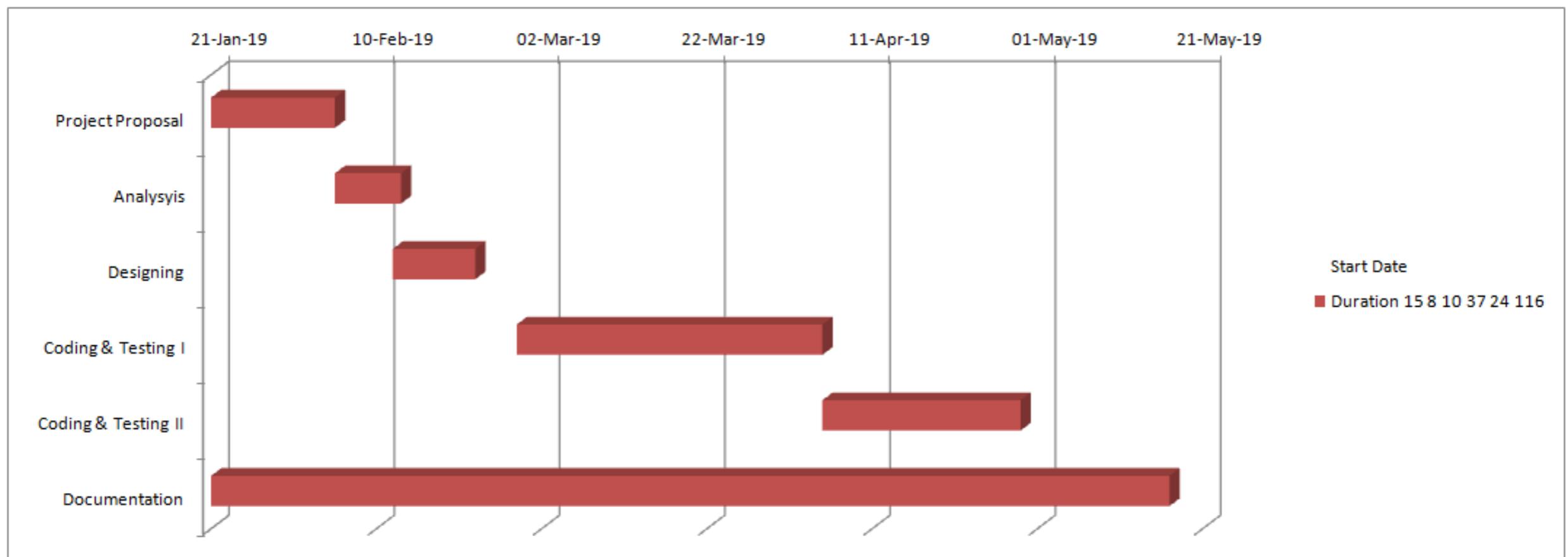
Therefore, the application is operationally feasible.

# **SCHEDULE FEASIBILITY**

# Work Breakdown Structure



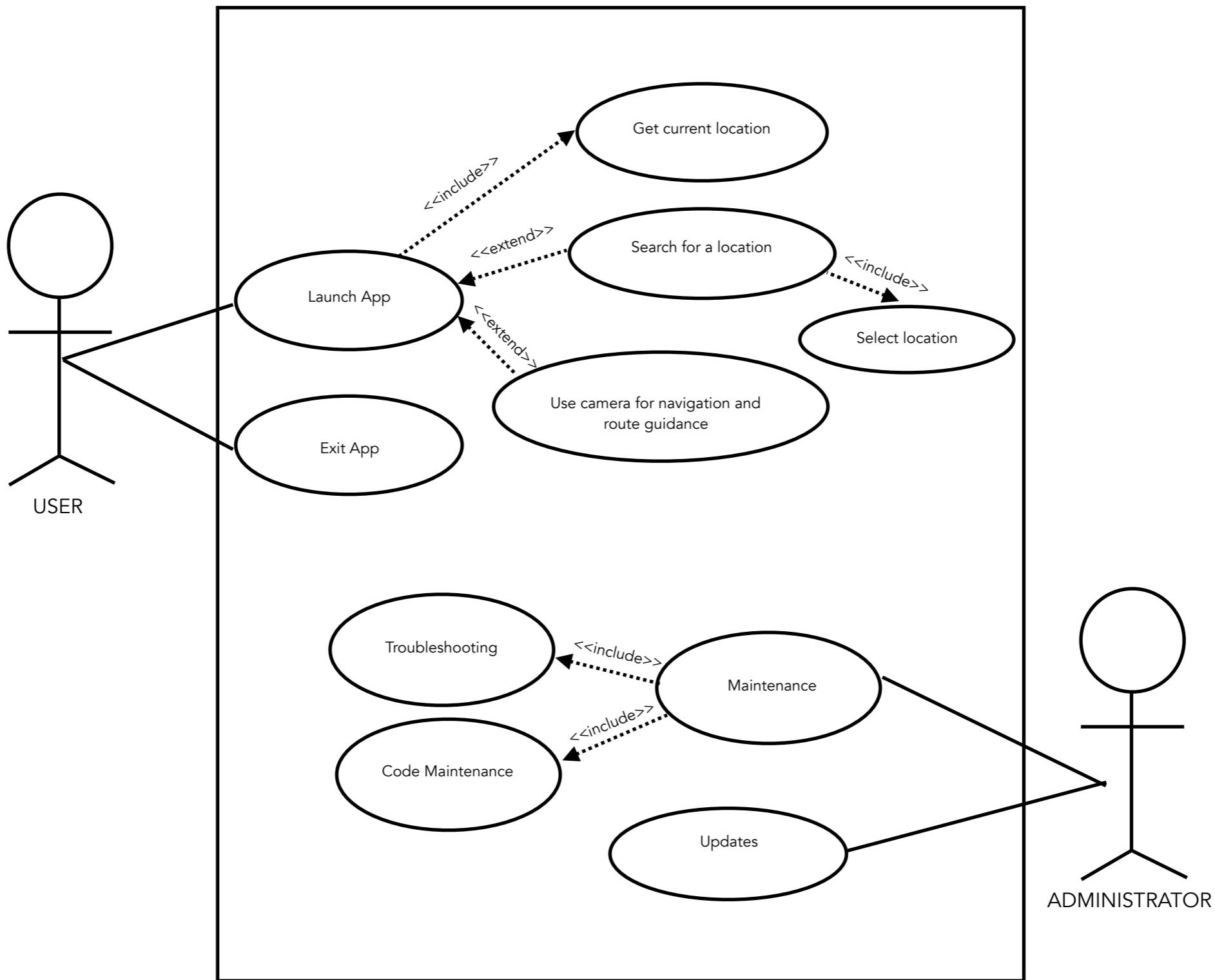
# Gantt Chart



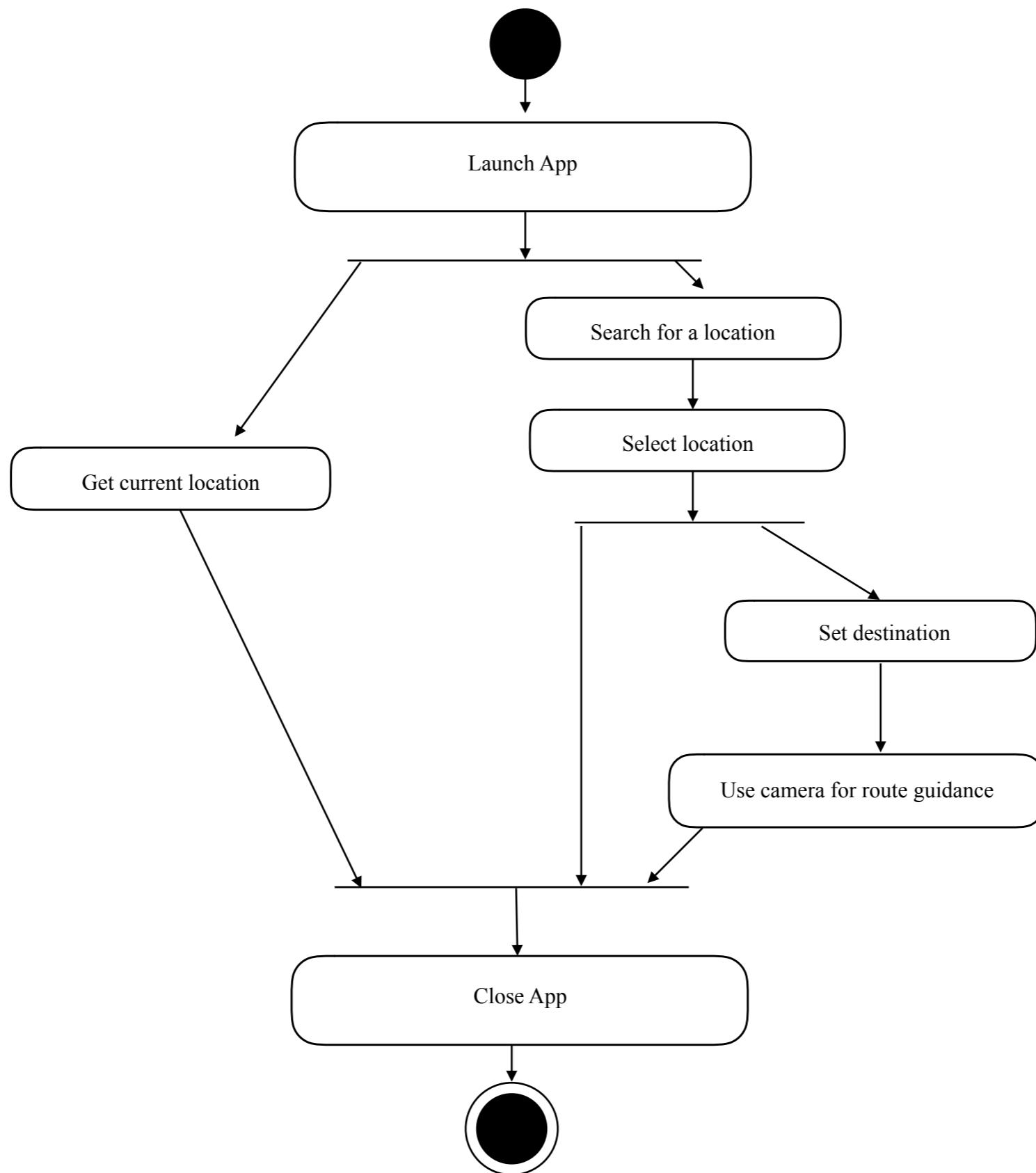
# **DESIGN DIAGRAMS**

# Use Case Diagram

## NavAR

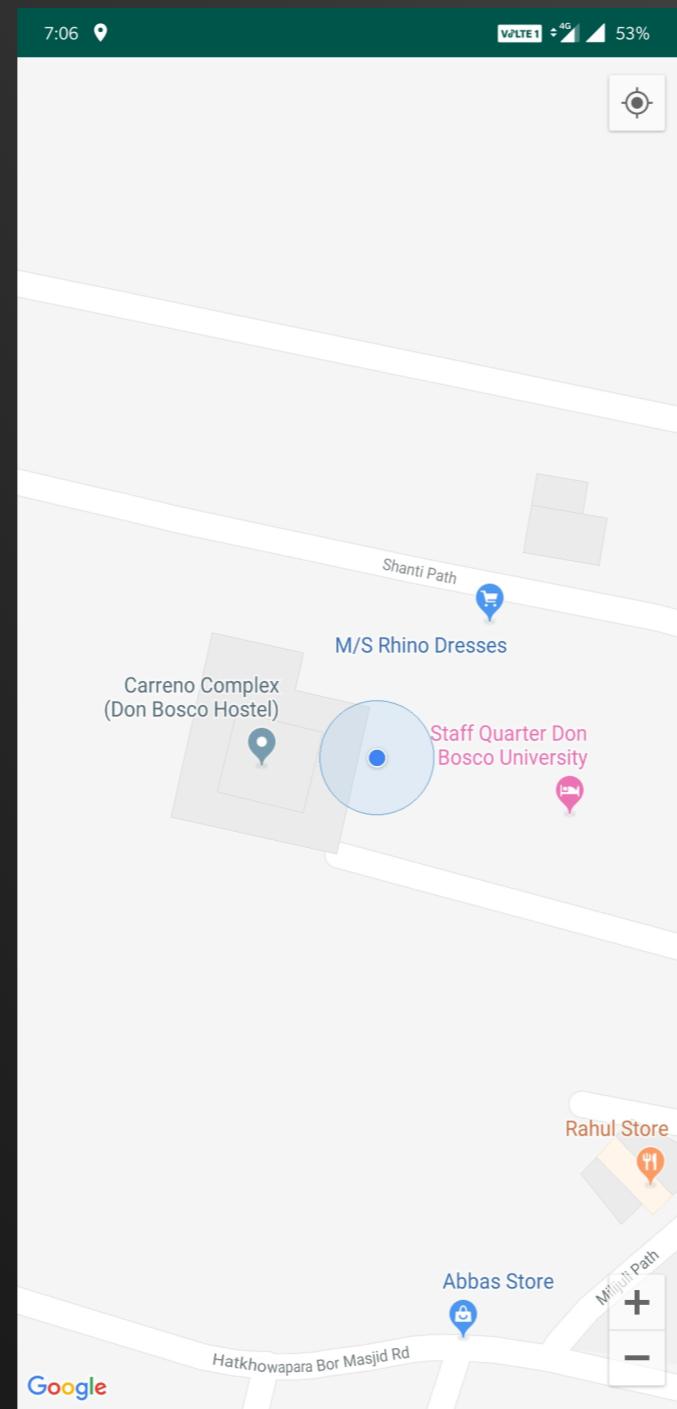


# Activity Diagram

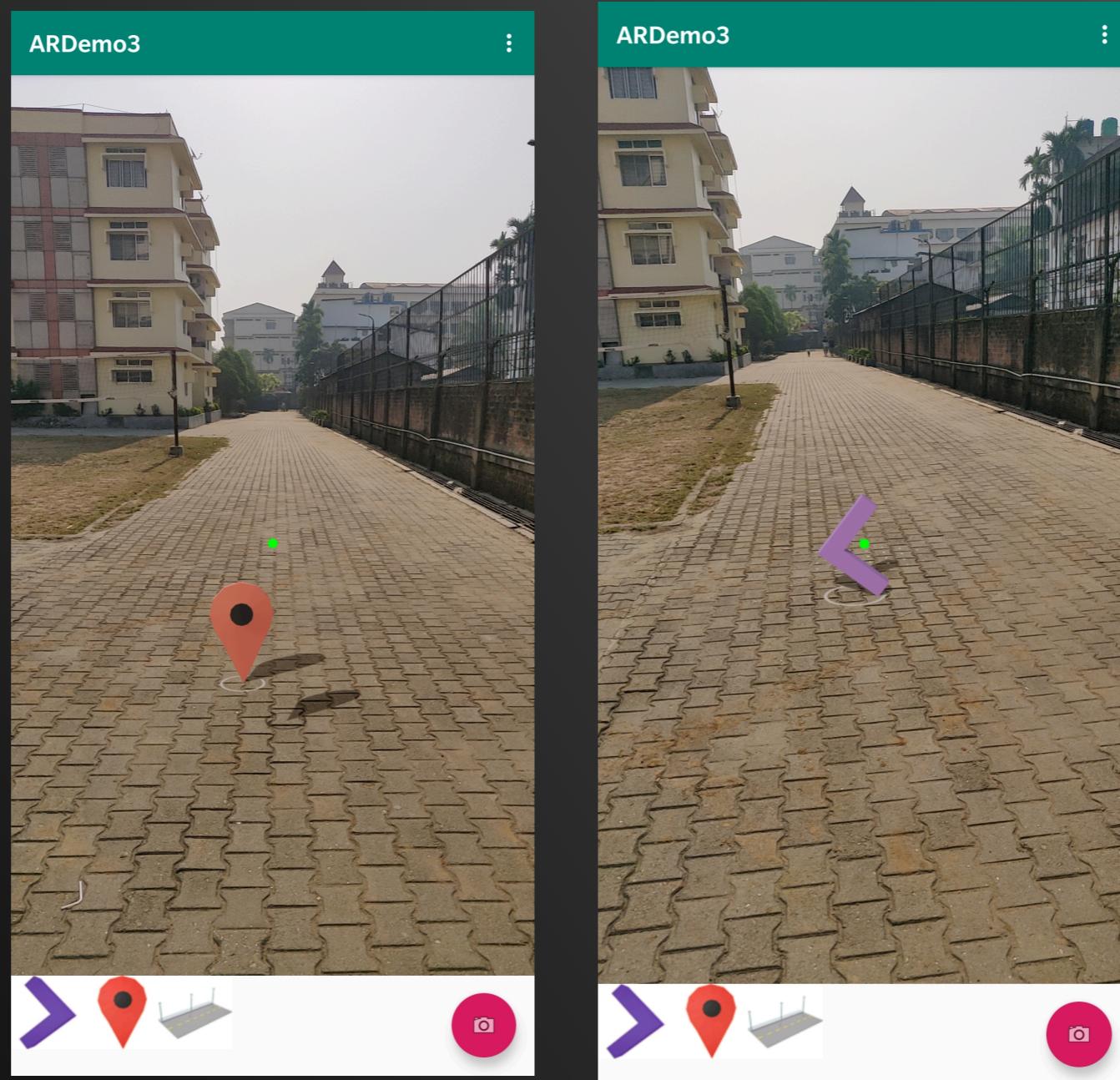


# TESTING

# Map



# Augmented Reality



# IMPLEMENTATION

The screenshot shows the Android Studio interface with the project 'NavAR\_Mrigaya' open. The left sidebar displays the project structure, showing the 'app' module with its Java files. The 'MapsActivity.java' file is currently selected and displayed in the main editor window. The code in the editor is as follows:

```
544    .makeText(this, "" + latLng, Toast.LENGTH_LONG).show();
545 }
546 mMap.addMarker(dest);
547 }
548 flagCheck=1;
549 callUrl();
550 }

//Direct PolyLine from Source to Destination when drawing via FIND ROUTE
551 /*Polyline line = mMap.addPolyline(new PolylineOptions()
552     .add(source.getPosition(), dest.getPosition())
553     .width(2)
554     .color(Color.BLUE));*/
555 }

556 public void callUrl() {
557     String url = getUrl(source.getPosition(), dest.getPosition(),transPref); // "driving";
558     placeARobject(url);
559     new FetchURL(MapsActivity.this).execute(url,transPref); // "driving";
560 }

561 private String getUrl(LatLng origin, LatLng dest, String directionMode) {
562     Toast.makeText(this, "Inside get URL", Toast.LENGTH_SHORT).show();
563     // Origin of route
564     String str_origin = "origin=" + origin.latitude + "," + origin.longitude;
565     // Destination of route
566     String str_dest = "destination=" + dest.latitude + "," + dest.longitude;
567     // Mode
568     String mode = "mode=" + directionMode;
569     // Building the parameters to the web service
570     String parameters = str_origin + "&" + str_dest + "&" + mode;
571     // Output format
572     String output = "json";
573     // Building the url to the web service
574     return "https://maps.googleapis.com/maps/api/directions/" + output + "?" + parameters + "&key=" + getString(R.string.google_maps_key_st);
575 }
576 /**
577 * &avoid=highways" + //before keys parameter in url
578 */
579 //String url = "https://maps.googleapis.com/maps/api/directions/" + output + "?" + parameters + "&avoid=highways" + "&key=" + getString(R.string.
580 //return url; //directionMode
581
582
583
584
585
586
587 }
```

The bottom status bar indicates '2 processes running...', '575:47 CRLF', 'UTF-8', '4 spaces', and various icons for device connection and file explorer.

# Getting directions from directions API

The screenshot shows the Android Studio interface with the project 'NavAR\_Mrigaya' open. The code editor displays the 'MapsActivity.java' file, which contains Java code for handling location updates and displaying them on a map. The code includes imports for Google API Client, LocationServices, LatLng, MarkerOptions, CameraPosition, and CameraUpdateFactory. It also uses the 'Toast' class for displaying messages. The code is annotated with line numbers from 440 to 492. Below the code editor, the 'Build' tab shows the build status for the project 'NavAR\_Mrigaya'. The log at the bottom indicates that Gradle sync has started and that two processes are running.

```
440     .build();  
441  
442         googleApiClient.connect();  
443     }  
444  
445     //public Location location;  
446     //marks the current location  
447     @Override  
448     public void onLocationChanged(Location location) {  
449  
450         lastLocation = location;  
451  
452         if(currentLocationMarker != null){  
453             currentLocationMarker.remove();  
454         }  
455  
456         LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());  
457         source = new MarkerOptions().position(latLng);  
458         // mMap.addMarker(source);  
459         MarkerOptions markerOptions = new MarkerOptions();  
460         markerOptions.position(latLng);  
461         markerOptions.draggable(false);  
462         markerOptions.title("Current Location");  
463         markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_MAGENTA));  
464         //currentLocationMarker = mMap.addMarker(markerOptions);  
465         mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));  
466         //mMap.animateCamera(CameraUpdateFactory.zoomTo(14));  
467         updateCameraBearing(location.getBearing());  
468         mMap.animateCamera(CameraUpdateFactory.zoomTo(14));  
469         if(googleApiClient != null){  
470  
471             LocationServices.FusedLocationApi.removeLocationUpdates(googleApiClient, this);  
472         }  
473  
474        .makeText(this, "Current: "+ source.getPosition(), Toast.LENGTH_LONG).show();  
475         //callUrl();  
476     }  
477  
478     private void updateCameraBearing(float bearing) {  
479         if ( mMap == null) return;  
480         CameraPosition camPos = CameraPosition  
481             .builder()  
482                 mMap.getCameraPosition() // current Camera  
483             .bearing(bearing)  
484     }  
485  
486     MapsActivity
```

Build: Sync

- NavAR\_Mrigaya: syncing... Running for 29 s
- Starting Gradle Daemon 2 s 827 ms
- Run build /Users/Jedael/AndroidStudioProjects/NavAR\_Mrigaya Running for 12 s

6: Logcat TODO Terminal Build Event Log

Gradle sync started 2 processes running... 625:1 CRLF: UTF-8: 4 spaces: Event Log

# Getting current location

```
public void initAutoComplete(){
    String apiKey = "AIzaSyC-3s1lHSRDKD60zhVNyrMs5WocKT3tax5Q";
    // Initialize Places.
    Places.initialize(getApplicationContext(), apiKey);
    AutocompleteSupportFragment autocompleteFragment = (AutocompleteSupportFragment) getSupportFragmentManager().findFragmentById(R.id.autocomplete_fragment);

    // Specify the types of place data to return.
    autocompleteFragment.setPlaceFields(Arrays.asList(Place.Field.ID, Place.Field.NAME));

    //autocompleteFragment.setTypeFilter(TypeFilter.REGIONS);

    autocompleteFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {
        @Override
        public void onPlaceSelected(Place place) {
           .makeText(MapsActivity.this, ""+ place.getName(), Toast.LENGTH_LONG).show();

            String searchLocation = place.getName();
            List<Address> addressList = null;
            MarkerOptions markerOptions = new MarkerOptions();
            if(!searchLocation.equals("")) {
                Geocoder geocoder = new Geocoder(MapsActivity.this);
                try {
                    addressList = geocoder.getFromLocationName(searchLocation, 1);
                } catch (IOException e) {
                    e.printStackTrace();
                }
                for (int i = 0; i < addressList.size(); i++) {
                    Address myAddress = addressList.get(i);
                    LatLng latLng = new LatLng(myAddress.getLatitude(), myAddress.getLongitude());
                    dest = new MarkerOptions().position(latLng);
                    markerOptions.position(latLng);
                    markerOptions.title("Result");
                    //mMap.addMarker(markerOptions);
                    mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
                    mMap.animateCamera(CameraUpdateFactory.zoomTo(14));
                    Toast.makeText(MapsActivity.this, "" + latLng, Toast.LENGTH_LONG).show();
                }
            }
        }
    });
}
```

Build: Sync

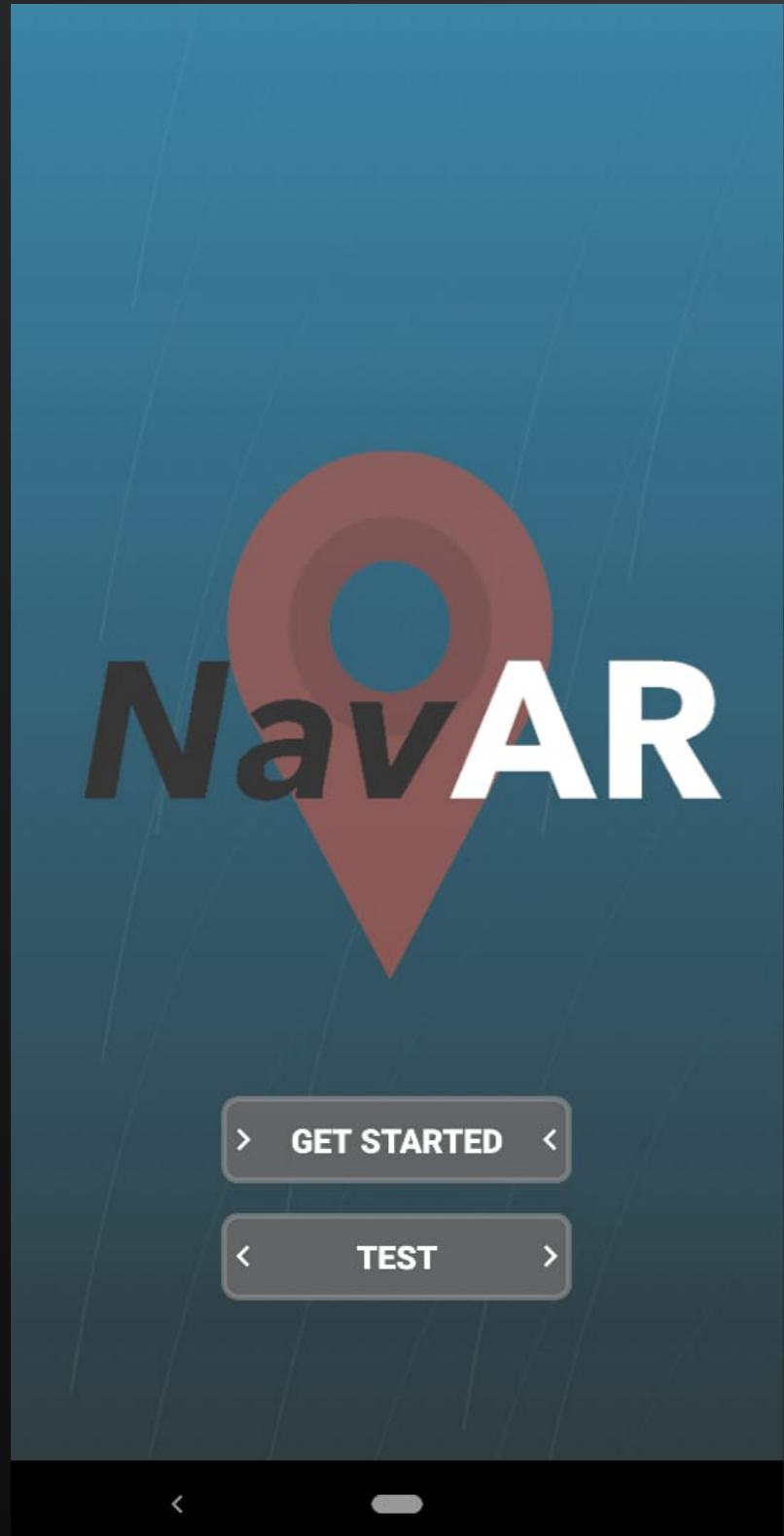
- NavAR\_Mrigaya: syncing...
  - Starting Gradle DaemonRunning for 3 m 37 s
- Run build /Users/Jedael/AndroidStudioProjects/NavAR\_Mrigaya2 s 277 ms
- Running for 3 m 23 s

6: Logcat TODO Terminal Build Event Log

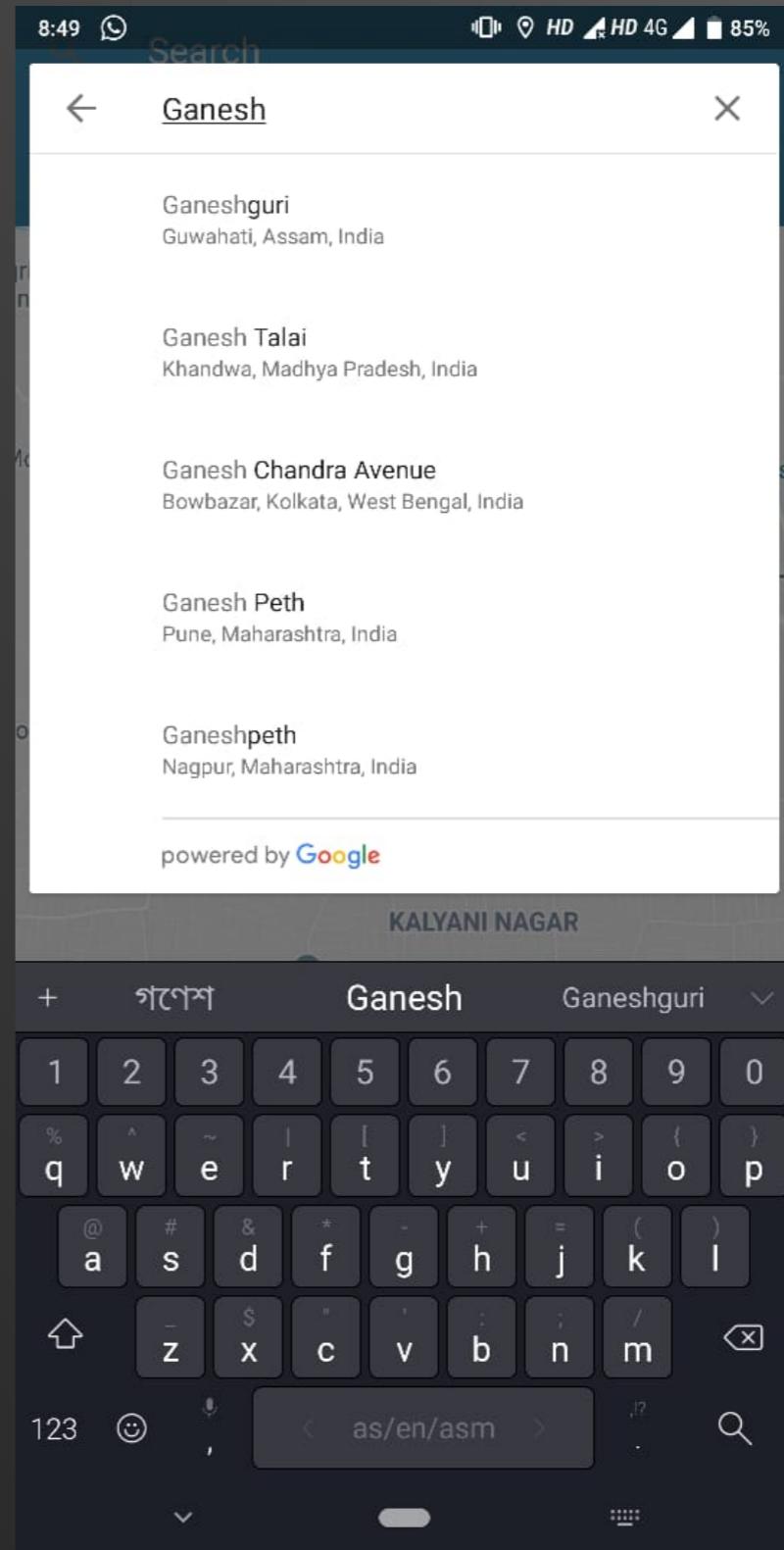
Gradle sync started (4 minutes ago)

2 processes running... 582:6 CRLF UTF-8 4 spaces

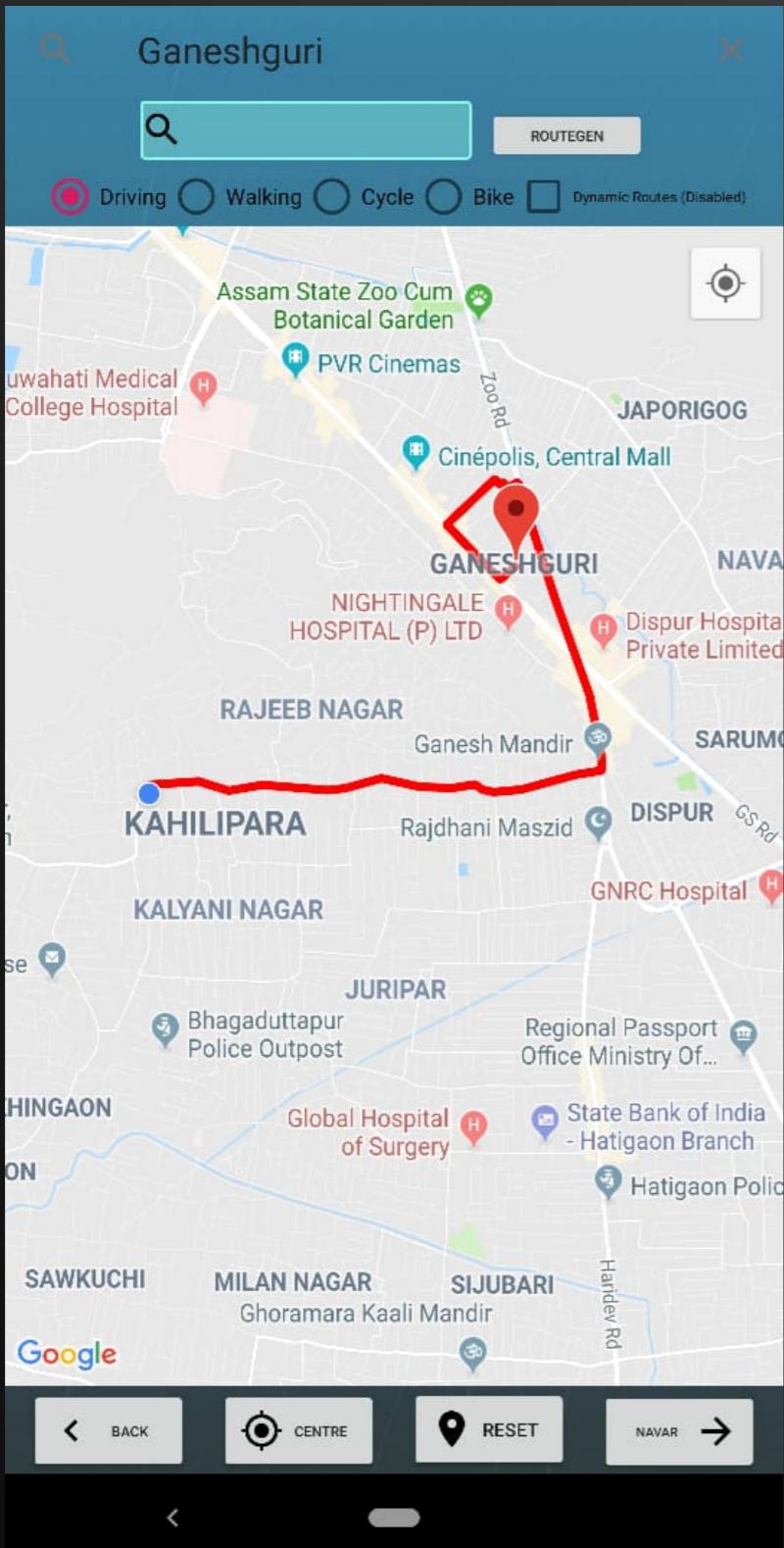
# Getting Autocomplete suggestions from Places API



Start Screen



Searching with  
Autocomplete



Driving directions



Augmented  
Reality navigation

# Conclusion

With this application, we hope to incorporate Augmented Reality in making our everyday lives easier and more convenient. Our aim for this application is to service navigation in complex roadways, and therefore, help the users to make correct decisions while driving or navigating.

# References

[1] Unity System Requirements:

<https://unity3d.com/unity/system-requirements> (22/02/2019)

[2] Android Studio System Requirements:

<https://developer.android.com/studio> (02/02/2019)

[3] COCOMO Model:

<https://www.geeksforgeeks.org/software-engineering-cocomo-model> (02/02/2019)

[4] ARCity by blippAR:

<https://www.blippar.com/blog/2017/11/06/welcome-ar-city-future-maps-and-navigation>  
(05/02/2019)

[5] AR Navigation & Nearby GPS, AR Walking Navigation by AR Technology | GPS Navigation | Future Apps:

<https://play.google.com/store/apps/details?id=com.arnavigation.poi.browser&hl=en> (09/02/2019)

[6] Vuforia for Mobile:

[https://docs.unity3d.com/Manual/vuforia\\_requirements.html](https://docs.unity3d.com/Manual/vuforia_requirements.html) (09/02/2019)

[7] ARCore:

<https://developers.google.com/ar/discover/supported-devices> (20/03/2019)

THANK YOU