Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
A CHRISTIAN MINORITY RESIDENTIAL INSTITUTION
AICTE Approved & NAAC Accredited

A SKILL BASED EVALUATION REPORT

SUBMITTED BY

ANSON SAJU GEORGE (URK2CS7064)

COURSE CODE
20CS2035

COURSE NAME

OBJECT ORIENTED PROGRAMMING

OCTOBER 2023

DIVISION OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING AND TECHNOLOGY

INDUSTRIAL CERTIFICATION

Infosys®
Navigate your next

COURSE COMPLETION CERTIFICATE

The certificate is awarded to

ANSON SAJU GEORGE URK22CS7064

for successfully completing the course

Java Programming Fundamentals

on November 1, 2023

Infosys | Springboard

Congratulations! You make us proud!

Thirumala Arohi
Senior Vice President and Head
Education, Training and Assessment (ETA)
Infosys Limited

Issued on: Wednesday, November 1, 2023
To verify, scan the QR code at https://verify.onwingspan.com

TITLE
Map Application

A REAL TIME APPLICATION REPORT

Submitted by

Anson Saju George URK22CS7064
Elias John Sabu URK22CS2024

DIVISION OF COMPUTER SCIENCE AND ENGINEERING

KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed-to-be-under Sec-3 of the UGC Act, 1956)
Karunya Nagar, Coimbatore - 641 114. INDIA

OCTOBER 2023

ABSTRACT

Our updated Maps Application represents a cutting-edge solution in the realm of travel and navigation. With an emphasis on providing an intuitive and user-friendly travel companion, this application leverages Mapbox's powerful mapping services to offer an all-encompassing solution for efficient journey planning. Users can seamlessly visualize routes on interactive maps, harness real-time location services, and navigate with ease. By simplifying route planning and map interaction, this application offers a streamlined and enjoyable travel experience. As a versatile travel companion, it addresses the complexities of modern journey planning and remains committed to enhancing the user's travel journey, making it accessible to individuals with various technical backgrounds. In a world where convenience and efficiency are paramount, our updated Maps Application stands as a testament to modern travel innovation.

PROBLEM STATEMENT

In today's dynamic and interconnected world, travelers frequently encounter the challenge of efficiently planning their journeys, especially when navigating unfamiliar destinations or seeking to optimize routes for cost-effectiveness and timeliness. While conventional navigation and mapping applications have proven useful, they often fall short of offering a holistic solution. Users are left in search of a comprehensive travel companion that seamlessly combines intuitive map visualization, real-time location services, and user-friendly route planning. This deficiency underscores the need for an all-encompassing travel application that simplifies the complexities of modern travel. Our Maps Application is a response to this pressing problem. It aims to provide a solution that not only streamlines journey planning but also simplifies route visualization, ensuring a seamless and enjoyable travel experience. In a world where efficient travel management is paramount, our application strives to become the ultimate solution, revolutionizing how users plan their journeys and navigate through today's ever-evolving landscape.

METHODOLOGY / ARCHITECTURE

In our project, we've meticulously crafted a Map-Based Android Application, skillfully harnessing Mapbox's mapping services to provide users with a seamless and user-friendly travel companion. The central goal of this application is to empower users in planning their journeys, enabling them to visualize routes on an interactive map and ensuring that each travel experience is both hassle-free and enjoyable.

Our methodology revolves around the core components and underlying logic that constitute the application's functionality:

First and foremost, our Mapbox integration forms the backbone of this digital travel tool, serving as the virtual canvas for maps and offering cutting-edge navigation capabilities. This allows users to explore and navigate with ease, just like having a personal GPS at their fingertips.

Furthermore, our implementation of location services enables users to pinpoint their current position on the map, providing valuable context and ensuring they're always oriented. It's akin to having a digital compass, aiding users on their journeys.

Our user interface (UI) design plays a pivotal role by simplifying the process of entering journey details and interacting with the map. We've placed a strong emphasis on making the application accessible to users with diverse technical backgrounds, ensuring that it's intuitive and user-friendly.

Lastly, the application's permission handling mechanism takes care of securing the user's consent to access their device's location information. This not only enhances privacy but also guarantees a seamless user experience.

In conclusion, this methodology statement encapsulates our unwavering commitment to providing a smooth and intuitive travel planning experience. We're dedicated to delivering a digital travel companion that makes journey planning accessible and enjoyable to users from all walks of life.

IMPLEMENTATION – CODING AND OUTPUT SCREENSHOT

```
package com.example.mapbox;

    import static com.mapbox.maps.plugin.gestures.GesturesUtils.getGestures;
    import
static.com.mapbox.maps.plugin.locationcomponent.LocationComponentUtils.getLocationCo
mponent;

    import android.Manifest;
    import android.content.pm.PackageManager;
    import android.os.Bundle;
    import android.view.View;
    import android.widget.Toast;

    import androidx.activity.result.ActivityResultCallback;
    import androidx.activity.result.ActivityResultLauncher;
    import androidx.activity.result.contract.ActivityResultContracts;
    import androidx.annotation.NonNull;
    import androidx.appcompat.app.AppCompatActivity;
    import androidx.appcompat.content.res.AppCompatResources;
    import androidx.core.app.ActivityCompat;

    import com.google.android.material.floatingactionbutton.FloatingActionButton;
    import com.mapbox.android.gestures.MoveGestureDetector;
    import com.mapbox.geojson.Point;
    import com.mapbox.maps.CameraOptions;
    import com.mapbox.maps.MapView;
    import com.mapbox.maps.Style;
    import com.mapbox.maps.plugin.LocationPuck2D;
    import com.mapbox.maps.plugin.gestures.OnMoveListener;
```

```
import com.mapbox.maps.plugin.locationcomponent.LocationComponentPlugin;
import com.mapbox.maps.plugin.locationcomponent.OnIndicatorBearingChangedListener;
import com.mapbox.maps.plugin.locationcomponent.OnIndicatorPositionChangedListener;


public class MainActivity extends AppCompatActivity {
    private MapView mapView;
    FloatingActionButton floatingActionButton;


    private final ActivityResultLauncher<String> activityResultLauncher = registerForActivityResult(new ActivityResultContracts.RequestPermission(), new ActivityResultCallback<Boolean>() {
        @Override
        public void onActivityResult(Boolean result) {
            if (result) {
                Toast.makeText(MainActivity.this, "Permission granted!", Toast.LENGTH_SHORT).show();
            }
        }
    });


    private final OnIndicatorBearingChangedListener onIndicatorBearingChangedListener = new OnIndicatorBearingChangedListener() {
        @Override
        public void onIndicatorBearingChanged(double v) {
            mapView.getMapboxMap().setCamera(new CameraOptions.Builder().bearing(v).build());
        }
    };


    private final OnIndicatorPositionChangedListener onIndicatorPositionChangedListener = new OnIndicatorPositionChangedListener() {
```

```
        @Override
        public void onIndicatorPositionChanged(@NonNull Point point) {
            mapView.getMapboxMap().setCamera(new
CameraOptions.Builder().center(point).zoom(20.0).build());

getGestures(mapView).setFocalPoint(mapView.getMapboxMap().pixelForCoordinate(point))
;
        }
    };

    private final OnMoveListener onMoveListener = new OnMoveListener() {
        @Override
        public void onMoveBegin(@NonNull MoveGestureDetector moveGestureDetector)
{

getLocationComponent(mapView).removeOnIndicatorBearingChangedListener(onIndicatorB
earingChangedListener);

getLocationComponent(mapView).removeOnIndicatorPositionChangedListener(onIndicatorP
ositionChangedListener);
            getGestures(mapView).removeOnMoveListener(onMoveListener);
            floatingActionButton.show();
        }

        @Override
        public boolean onMove(@NonNull MoveGestureDetector moveGestureDetector) {
            return false;
        }

        @Override
        public void onMoveEnd(@NonNull MoveGestureDetector moveGestureDetector) {

        }
```

```
    };

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

        mapView = findViewById(R.id.mapView);
        floatingActionButton = findViewById(R.id.focusLocation);

        if (ActivityCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

activityResultLauncher.launch(Manifest.permission.ACCESS_FINE_LOCATION);
        }

        floatingActionButton.hide();
        mapView.getMapboxMap().loadStyleUri(Style.SATELLITE, new
Style.OnStyleLoaded() {
            @Override
            public void onStyleLoaded(@NonNull Style style) {
                mapView.getMapboxMap().setCamera(new
CameraOptions.Builder().zoom(20.0).build());
                LocationComponentPlugin locationComponentPlugin =
getLocationComponent(mapView);
                locationComponentPlugin.setEnabled(true);
                LocationPuck2D locationPuck2D = new LocationPuck2D();

locationPuck2D.setBearingImage(AppCompatResources.getDrawable(MainActivity.this,
R.drawable.baseline_location_on_24));
                locationComponentPlugin.setLocationPuck(locationPuck2D);
```

```
locationComponentPlugin.addOnIndicatorBearingChangedListener(onIndicatorBearingChan
gedListener);

locationComponentPlugin.addOnIndicatorPositionChangedListener(onIndicatorPositionChan
gedListener);
                getGestures(mapView).addOnMoveListener(onMoveListener);


               floatingActionButton.setOnClickListener(new View.OnClickListener() {
                  @Override
                  public void onClick(View view) {

locationComponentPlugin.addOnIndicatorBearingChangedListener(onIndicatorBearingChan
gedListener);

locationComponentPlugin.addOnIndicatorPositionChangedListener(onIndicatorPositionChan
gedListener);
                     getGestures(mapView).addOnMoveListener(onMoveListener);
                     floatingActionButton.hide();
                  }
               });
            }
         });
      }
   }
   activity_main.xml
   <?xml version="1.0" encoding="utf-8"?>
   <RelativeLayout 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"
      xmlns:mapbox="http://schemas.android.com/apk/res-auto"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
```

```
    tools:context=".MainActivity">

    <com.mapbox.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        mapbox:mapbox_cameraTargetLat="-122.295252"
        mapbox:mapbox_cameraTargetLng="47.477197"
        mapbox:mapbox_cameraZoom="9.0" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/focusLocation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="16sp"
        android:layout_marginTop="16sp"
        android:layout_marginEnd="16sp"
        android:layout_marginBottom="16sp"
        android:src="@drawable/baseline_my_location_24" />

</RelativeLayout>
```

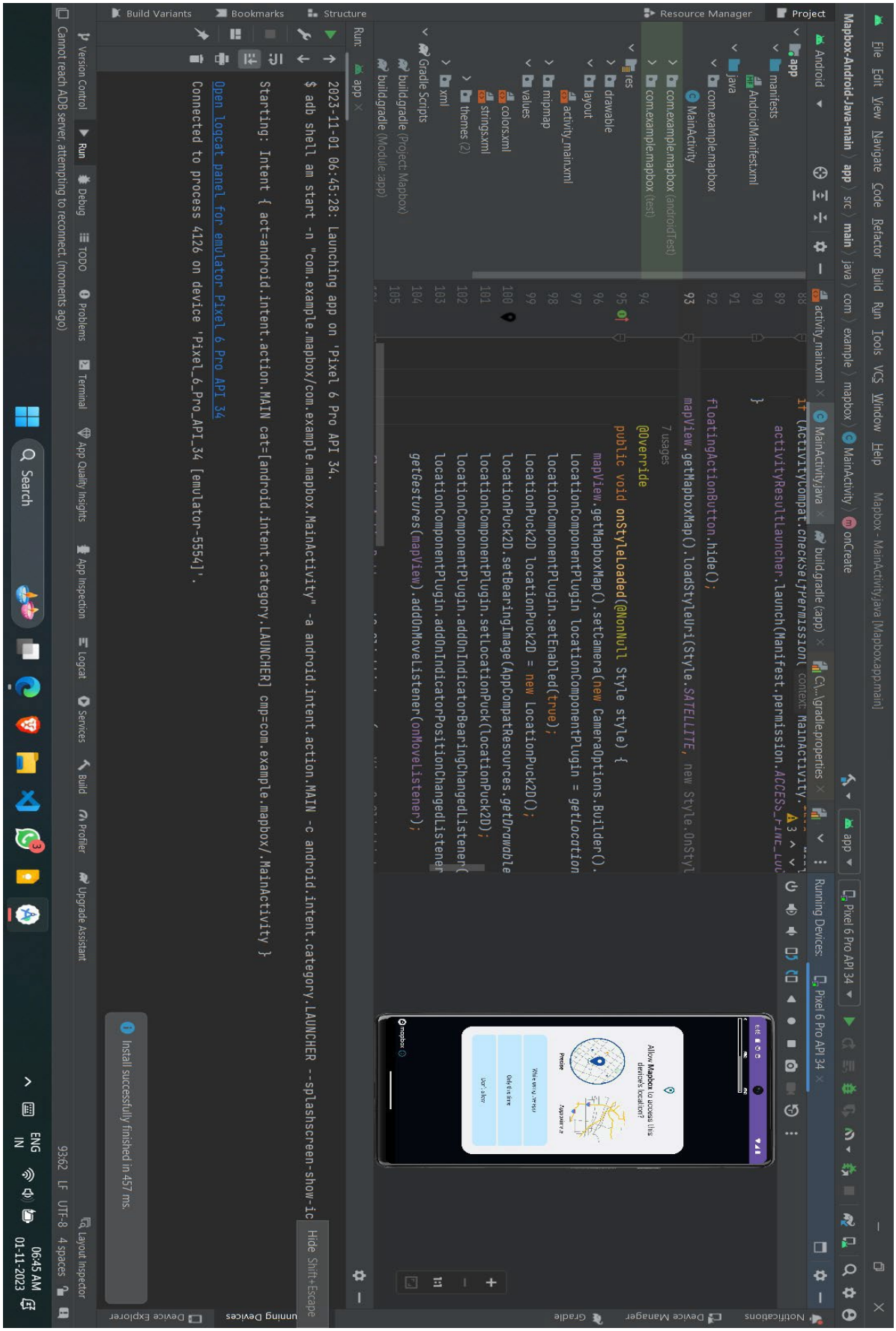
build.graddle

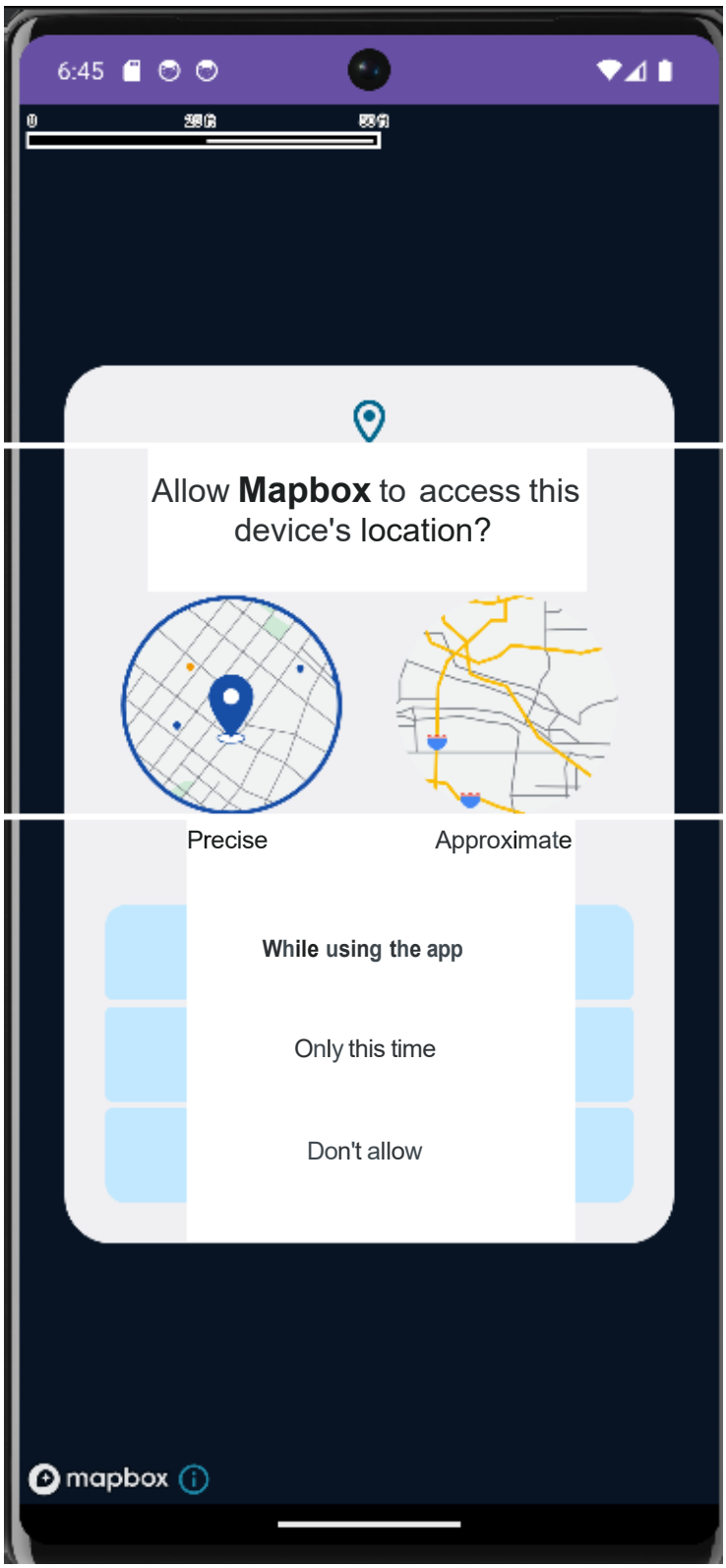
```
plugins {
    id 'com.android.application'
}

android {
    namespace 'com.example.mapbox'
```

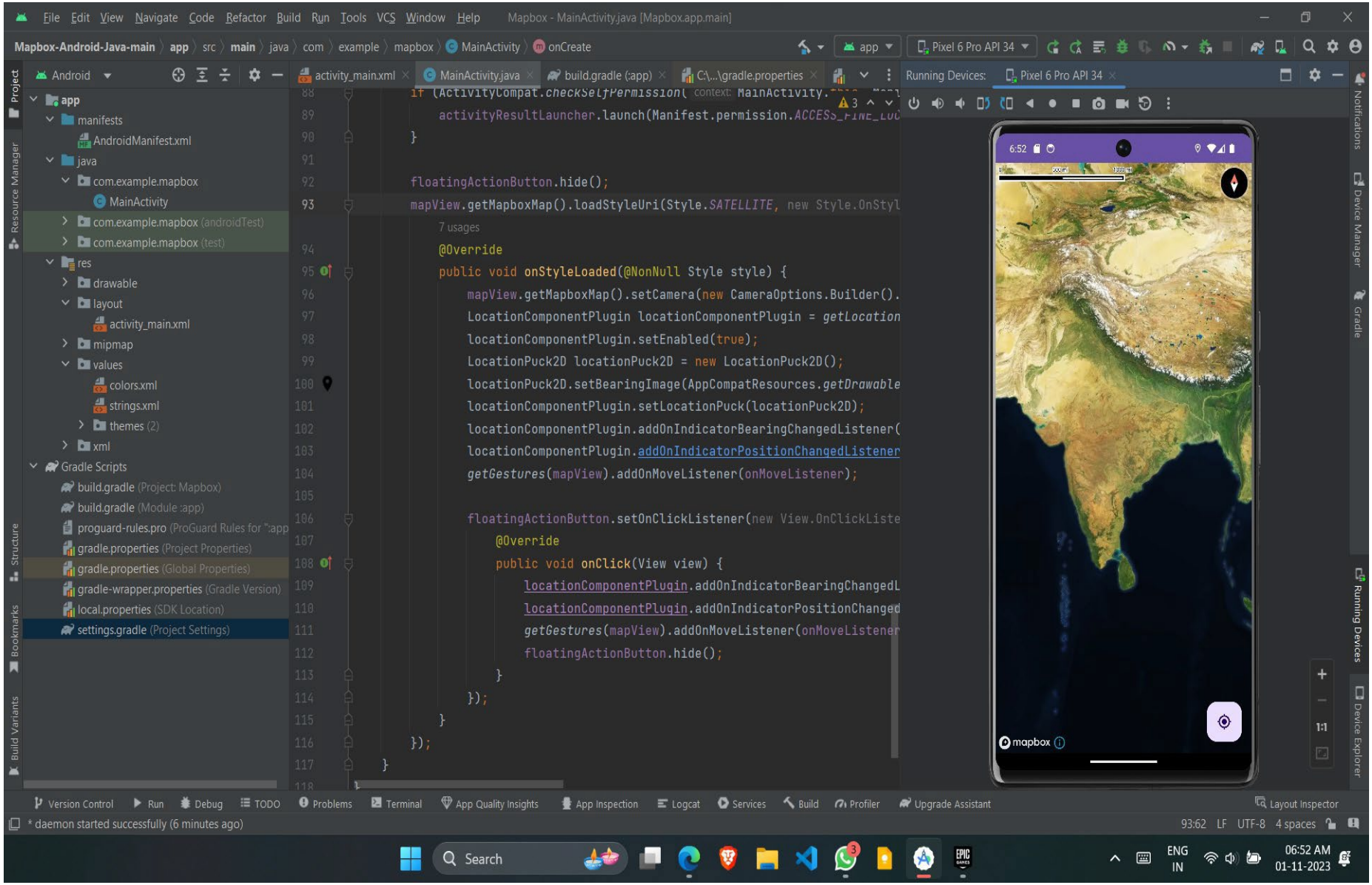
```
    compileSdk 33

    defaultConfig {
        applicationId "com.example.mapbox"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-
rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.9.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
```

```
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    implementation 'com.mapbox.maps:android:10.16.1'
}
```

OUTPUT


Mapbox-Android-Java-main
MainActivity.java
activity_mainxml
Pixel 6 Pro API 34
Running Devices:
Allow Mapbox to access this device's location?
While using the app
Only this time
Don't allow
Install successfully finished in 457 ms.
2023-11-01 06:45:28: Launching app on 'Pixel 6 Pro API 34.
$ adb shell am start -n "com.example.mapbox/com.example.mapbox.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER --splashscreen-show-ic
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.example.mapbox/.MainActivity }
Open logcat panel for emulator Pixel 6 Pro API 34
Connected to process 4126 on device 'Pixel_6_Pro_API_34 [emulator-5554]'.
Cannot reach ADB server, attempting to reconnect. (moments ago)


6:45
Allow Mapbox to access this device's location?
Precise
Approximate
While using the app
Only this time
Don't allow
mapbox


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Mapbox - MainActivity.java [Mapbox.app.main]
Mapbox-Android-Java-main app src main java com example mapbox MainActivity onCreate
app
Pixel 6 Pro API 34
Android
activity_main.xml
MainActivity.java
build.gradle (:app)
C:\...\gradle.properties
Running Devices:
Pixel 6 Pro API 34
app
manifests
AndroidManifest.xml
java
com.example.mapbox
MainActivity
com.example.mapbox (androidTest)
com.example.mapbox (test)
res
drawable
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
themes (2)
xml
Gradle Scripts
build.gradle (Project: Mapbox)
build.gradle (Module :app)
proguard-rules.pro (ProGuard Rules for ":app
gradle.properties (Project Properties)
gradle.properties (Global Properties)
gradle-wrapper.properties (Gradle Version)
local.properties (SDK Location)
settings.gradle (Project Settings)
if (ActivityCompat.checkSelfPermission(context: MainActivity.
activityResultLauncher.launch(Manifest.permission.ACCESS_FINE_LOC
}
floatingActionButton.hide();
mapView.getMapboxMap().loadStyleUri(Style.SATELLITE, new Style.OnStyl
7 usages
@Override
public void onStyleLoaded(@NonNull Style style) {
mapView.getMapboxMap().setCamera(new CameraOptions.Builder().
LocationComponentPlugin locationComponentPlugin = getLocation
locationComponentPlugin.setEnabled(true);
LocationPuck2D locationPuck2D = new LocationPuck2D();
locationPuck2D.setBearingImage(AppCompatResources.getDrawable
locationComponentPlugin.setLocationPuck(locationPuck2D);
locationComponentPlugin.addOnIndicatorBearingChangedListener(
locationComponentPlugin.addOnIndicatorPositionChangedListener
getGestures(mapView).addOnMoveListener(onMoveListener);
floatingActionButton.setOnClickListener(new View.OnClickListe
@Override
public void onClick(View view) {
locationComponentPlugin.addOnIndicatorBearingChangedL
locationComponentPlugin.addOnIndicatorPositionChanged
getGestures(mapView).addOnMoveListener(onMoveListener
floatingActionButton.hide();
}
});
}
});
}
6:52
mapbox
Version Control
Run
Debug
TODO
Problems
Terminal
App Quality Insights
App Inspection
Logcat
Services
Build
Profiler
Upgrade Assistant
Layout Inspector
* daemon started successfully (6 minutes ago)
93:62 LF UTF-8 4 spaces
Search
ENG
IN
06:52 AM
01-11-2023

CONCLUSION

Our maps application represents a comprehensive solution to the challenges faced by modern travelers. By integrating Mapbox's powerful API, we provide accurate and efficient mapping services, while our gas price calculation feature empowers users to make informed, budget-conscious decisions about their journeys. The user-friendly interface ensures that planning routes is a seamless and enjoyable experience. In a world where time and resources are of the essence, our application simplifies route planning and financial management. Whether for daily commutes or cross-country adventures, we offer a valuable tool that enhances the overall travel experience. Embracing technology and usability, our solution is tailored to meet the evolving needs of today's travelers.

EVALUATION SHEET

Reg.No : URK22CS7064

Name: Anson Saju George

Course code: 20CS2035

Course Name: Object Oriented Programming

S.No	Rubrics	Maximum Marks	Marks Obtained
1	Industrial Certification	10	
2	Real – Time Application Design	30	
Total		40	

Rubrics	Excellent	Good	Average	Below Average
Classes and Inheritance				
Concept Used				
GUI				
Database				
Innovation				
Presentation and Viva				
Report				

Signature of the Faculty-in-charge