

***Centre of Distance and Online Education***

*Vidya Nagari, Kalina, Santacruz East – 400098*.

A Practical Journal Submitted in fulfillment of the degree of

**MASTER OF SCIENCE**

**IN**

**COMPUTER SCIENCE**

**YEAR 2024-25**

Part II

Semester 3

Subject code – 90981R

Subject Name - Ubiquitous Computing

BY

**Ms. Akshata T. Nangrey**

**Application ID- 37123**

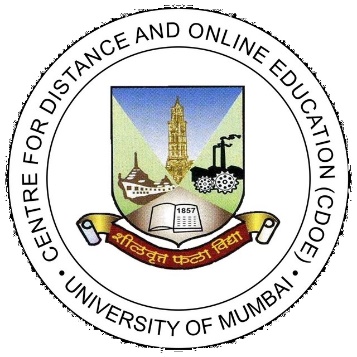
Seat No - 4100209

Centre for Distance and Online Education

(CDOE)

**University of Mumbai**

*Vidya Nagari, Kalina, Santacruz East – 400098*.



**Certificate**

This is to certify that **Miss. Akshata T. Nangrey** student of Masters of Computer Science, Part 2, Semester 3 has completed the specified term work in the subject of **Ubiquitous Computing** in satisfactorily manner within this institute as laid down by University of Mumbai during the Academic year 2024 to 2025.

M.Sc. - CS Coordinator Examiner

Date: Guide

INDEX

|  |  |  |
| --- | --- | --- |
| **Sr No.** | **Practical Name** | **Sign.** |
| 1. | Design and develop location based messaging app |  |
| 2. | Design and develop chat messaging app which is a location-based |  |
| 3. | Design and develop app demonstrating Simple Downstream Messaging |  |
| 4. | Design and develop app demonstrating Send Upstream Messages |  |
| 5. | Demonstrate use of OpenGTS (Open Source GPS Tracking System) |  |
| 6. | Context-Aware system. |  |
| 7. | Develop application demonstrating Human Computer Interaction. |  |
| 8. | Write a Java Card applet |  |

**Practical no. 1**

**Aim :- Design and develop location based messaging app**

**UI Code :-**

*<?*xml version="1.0" encoding="utf-8"*?>*

<RelativeLayoutxmlns:android=["http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) xmlns:tools=["http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent"

android:layout\_height="match\_parent" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context="com.example.prashant.locationmsg.MainActivity">

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge" android:text="Large Text"

android:id="@+id/textView" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true" />

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge" android:text="Large Text"

android:id="@+id/textView3" android:layout\_below="@+id/textView" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true" android:layout\_marginTop="96dp" />

<Button android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Get Longitude and Latitude" android:id="@+id/button" android:layout\_centerVertical="true" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true" />

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge" android:text="Large Text"

android:id="@+id/textView2" android:layout\_alignParentBottom="true" android:layout\_alignRight="@+id/button" android:layout\_alignEnd="@+id/button" />

</RelativeLayout>

**Source code :-**

**import** android.Manifest;

**import** android.content.Context;

**import** android.content.pm.PackageManager;

**import** android.location.Location;

**import** android.location.LocationListener; **import** android.location.LocationManager; **import** android.net.Uri;

**import** android.support.v4.app.ActivityCompat; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle;

**import** android.util.Log; **import** android.view.View; **import** android.widget.Button;

**import** android.widget.TextView;

**import** com.google.android.gms.appindexing.Action;

**import** com.google.android.gms.appindexing.AppIndex;

**import** com.google.android.gms.common.api.GoogleApiClient;**public class** MainActivity**extends**AppCompatActivity**implements**LocationListener { TextView**t1**, **t2**, **t3**;

Button **b1**;

**protected** LocationManager**locationManager**; **protected** LocationListener**locationListener**; **double lat**, **longg**;

**private** GoogleApiClient**client**;

@Override

**protected void** onCreate(Bundle savedInstanceState) { **super**.onCreate(savedInstanceState); setContentView(R.layout.***activity\_main***);

**t1** = (TextView) findViewById(R.id.***textView***); **t2** = (TextView) findViewById(R.id.***textView3***); **t3** = (TextView) findViewById(R.id.***textView2***);

**b1** = (Button) findViewById(R.id.***button***);

**b1**.setOnClickListener(**new** View.OnClickListener() { @Override

**public void** onClick(View view) { **t1**.setText(**"Latitude = "** + **lat**); **t2**.setText(**"Longitude = "** + **longg**);

**if** (**lat**<38 &&**lat**>36 &&**longg**<122 &&**longg**>118) {

**t3**.setText(**"In-Side The Area"**);

} **else** {

**t3**.setText(**"Out-Side The Area"**);

}

}

});

**locationManager**= (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);

**if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***)

!= PackageManager.***PERMISSION\_GRANTED***&&ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {

return;

}

**locationManager**.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***, 0, 0, **this**);

*// locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 0, 0, this);*

*// ATTENTION: This was auto-generated to implement the App Indexing API.*

*// See https://g.co/AppIndexing/AndroidStudio for more information.*

**client** = **new** GoogleApiClient.Builder(**this**).addApi(AppIndex.***API***).build();

}

@Override

**public void** onLocationChanged(Location location) {

*// txtLat = (TextView) findViewById(R.id.textview1);*

**lat**= location.getLatitude(); **longg**= location.getLongitude(); Log.*d*(**""** + **lat**, **""** + **lat**);

Log.*d*(**""** + **longg**, **""** + **longg**);

**if** (**lat**== 38 &&**longg**== 118) {

**t3**.setText(**"You Are at Perfect Place !!!!"**);

} **else** {

**t3**.setText(**"You are not at Perfect Place !!!!"**);

}

*//t1.setText("Latitude:" + location.getLatitude() + ", Longitude:" + location.getLongitude());*

}

@Override

**public void** onProviderDisabled(String provider) { Log.*d*(**"Latitude"**, **"disable"**);

}

@Override

**public void** onProviderEnabled(String provider) { Log.*d*(**"Latitude"**, **"enable"**);

}

@Override

**public void** onStatusChanged(String provider, **int**status, Bundle extras) { Log.*d*(**"Latitude"**, **"status"**);

}

@Override

**public void** onStart() {

**super**.onStart();

*.*

**client**.connect();

Action viewAction = Action.*newAction*( Action.***TYPE\_VIEW***, *// TODO: choose an action type.*

**"Main Page"**,

Uri.*parse*([**"http://host/path"**](http://host/path)),

*// TODO: Make sure this auto-generated app URL is correct.*

Uri.*parse*("android-app://com.example.prashant.locationmsg/http/host/path")

);

AppIndex.***AppIndexApi***.start(**client**, viewAction);

}

@Override

**public void** onStop() {

**super**.onStop();

*// ATTENTION: This was auto-generated to implement the App Indexing API.*

*// See https://g.co/AppIndexing/AndroidStudio for more information.*

Action viewAction = Action.*newAction*( Action.***TYPE\_VIEW***, *// TODO: choose an action type.*

**"Main Page"**, *// TODO: Define a title for the content shown.*

*// TODO: If you have web page content that matches this app activity's*

*content,*

*// make sure this auto-generated web page URL is correct.*

*// Otherwise, set the URL to null.*

Uri.*parse*([**"http://host/path"**](http://host/path)),

*// TODO: Make sure this auto-generated app URL is correct.*

Uri.*parse*("android-app://com.example.prashant.locationmsg/http/host/path")

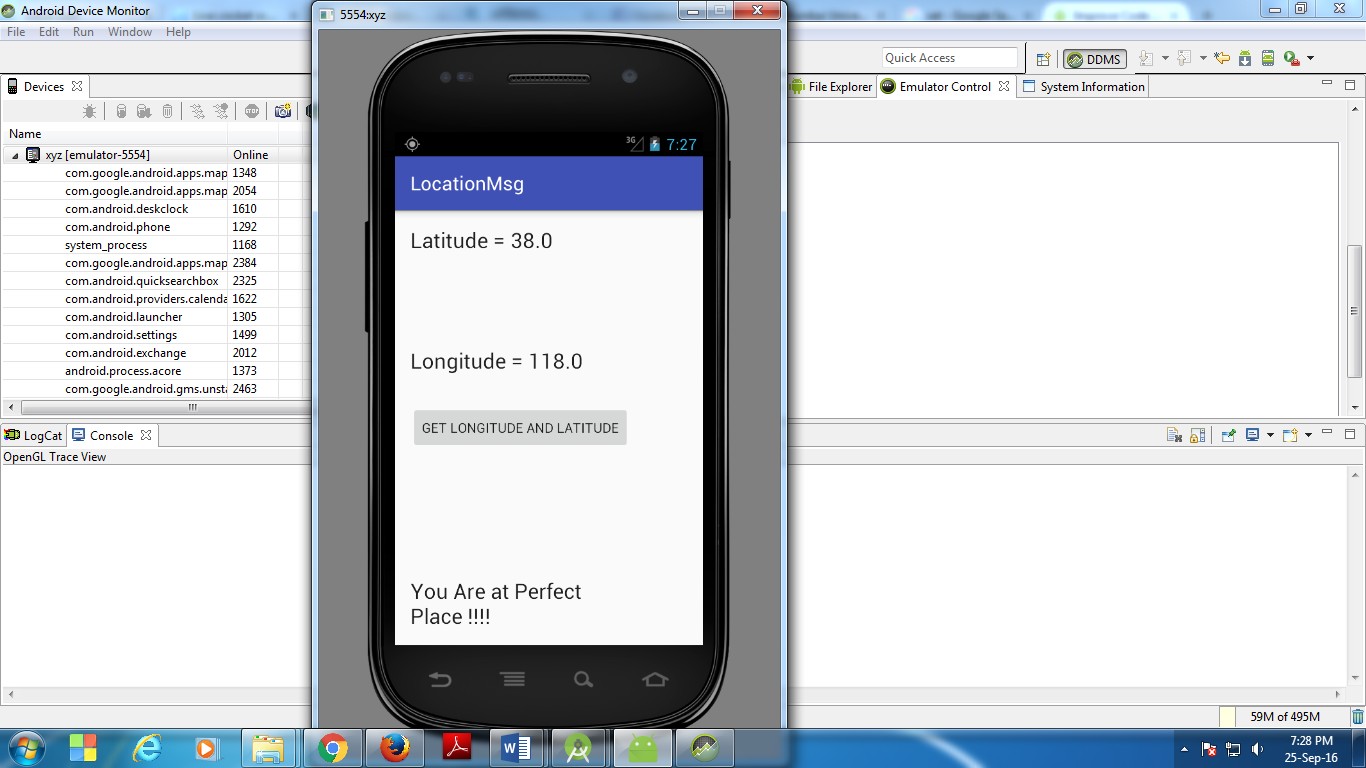
);

AppIndex.***AppIndexApi***.end(**client**, viewAction); client.disconnect();

}

}

**Output:-**



*Permission   
<?*xml version="1.0" encoding="utf-8"*?>*

<manifest xmlns:android=["http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

package="com.example.prashant.locationmsg">

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<uses-permission android:name="android.permission.INTERNET" />

<application android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name" android:supportsRtl="true" android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</**activity**>

*<!-- ATTENTION: This was auto-generated to add Google Play services to your project for*

*App Indexing. See https://g.co/AppIndexing/AndroidStudio for more information. -*

*->*

<meta-data android:name="com.google.android.gms.version"

android:value="@integer/google\_play\_services\_version" />

</**application**>

</**manifest**>

**Practical No .2**

**Aim: - Design and develop chat messaging app which is a location-based**

**UI Code:-**

*<?*xml version="1.0" encoding="utf-8"*?>*

<RelativeLayoutxmlns:android=["http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) xmlns:tools=["http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent"

android:layout\_height="match\_parent" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context="com.example.prashant.gpstrace.MainActivity">

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge" android:text="Large Text"

android:id="@+id/textView" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true" />

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge" android:text="Large Text"

android:id="@+id/textView3" android:layout\_below="@+id/textView" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true" android:layout\_marginTop="96dp" />

<Button android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Get Longitude and Latitude" android:id="@+id/button" android:layout\_centerVertical="true" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true" />

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge" android:text="Large Text"

android:id="@+id/textView2" android:layout\_alignParentBottom="true" android:layout\_alignRight="@+id/button" android:layout\_alignEnd="@+id/button" />

</RelativeLayout>

**Source Code:-**

**package** com.example.prashant.gpstrace;

**import** android.Manifest;

**import** android.content.Context;

**import** android.content.pm.PackageManager;

**import** android.location.Location;

**import** android.location.LocationListener; **import** android.location.LocationManager; **import** android.support.v4.app.ActivityCompat;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle; **import** android.util.Log; **import** android.view.View; **import** android.widget.Button;

**import** android.widget.TextView;

**public class** MainActivity**extends**AppCompatActivity**implements**LocationListener { TextView**t1**, **t2**, **t3**;

Button **b1**;

**protected** LocationManager**locationManager**; **protected** LocationListener**locationListener**; **double** lat, **longg**;

@Override

**protected void** onCreate(Bundle savedInstanceState) { **super**.onCreate(savedInstanceState); setContentView(R.layout.***activity\_main***);

**t1** = (TextView) findViewById(R.id.***textView***); **t2** = (TextView) findViewById(R.id.***textView3***); **t3** = (TextView) findViewById(R.id.***textView2***);

**b1** = (Button) findViewById(R.id.***button***);

**b1**.setOnClickListener(**new** View.OnClickListener() { @Override

**public void** onClick(View view) { **t1**.setText(**"Latitude = "** + lat); **t2**.setText(**"Longitude = "**+ **longg**);

**if**(lat<38 &&lat>36 &&**longg**<122 &&**longg**>118)

{

**t3**.setText(**"In-Side The Area"**);

}

else

{

**t3**.setText(**"Out-Side The Area"**);

}

}

});

**locationManager**= (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);

**if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***)

!= PackageManager.***PERMISSION\_GRANTED***&&ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {

return;

}

**locationManager**.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***, 0, 0, **this**);

*// locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 0, 0, this);*

}

@Override

**public void** onLocationChanged(Location location) {

*// txtLat = (TextView) findViewById(R.id.textview1);*

lat=location.getLatitude(); **longg**= location.getLongitude(); Log.*d*(**""**+lat,**""**+lat);

Log.*d*(**""**+**longg**,**""**+**longg**);

**if**(lat<38 &&lat>36 &&**longg**<122 &&**longg**>118)

{

**t3**.setText(**"In-Side The Area"**);

}

else

{

**t3**.setText(**"Out-Side The Area"**);

}

*//t1.setText("Latitude:" + location.getLatitude() + ", Longitude:" + location.getLongitude());*

}

@Override

**public void** onProviderDisabled(String provider) { Log.*d*(**"Latitude"**,**"disable"**);

}

@Override

**public void** onProviderEnabled(String provider) { Log.*d*(**"Latitude"**,**"enable"**);

}

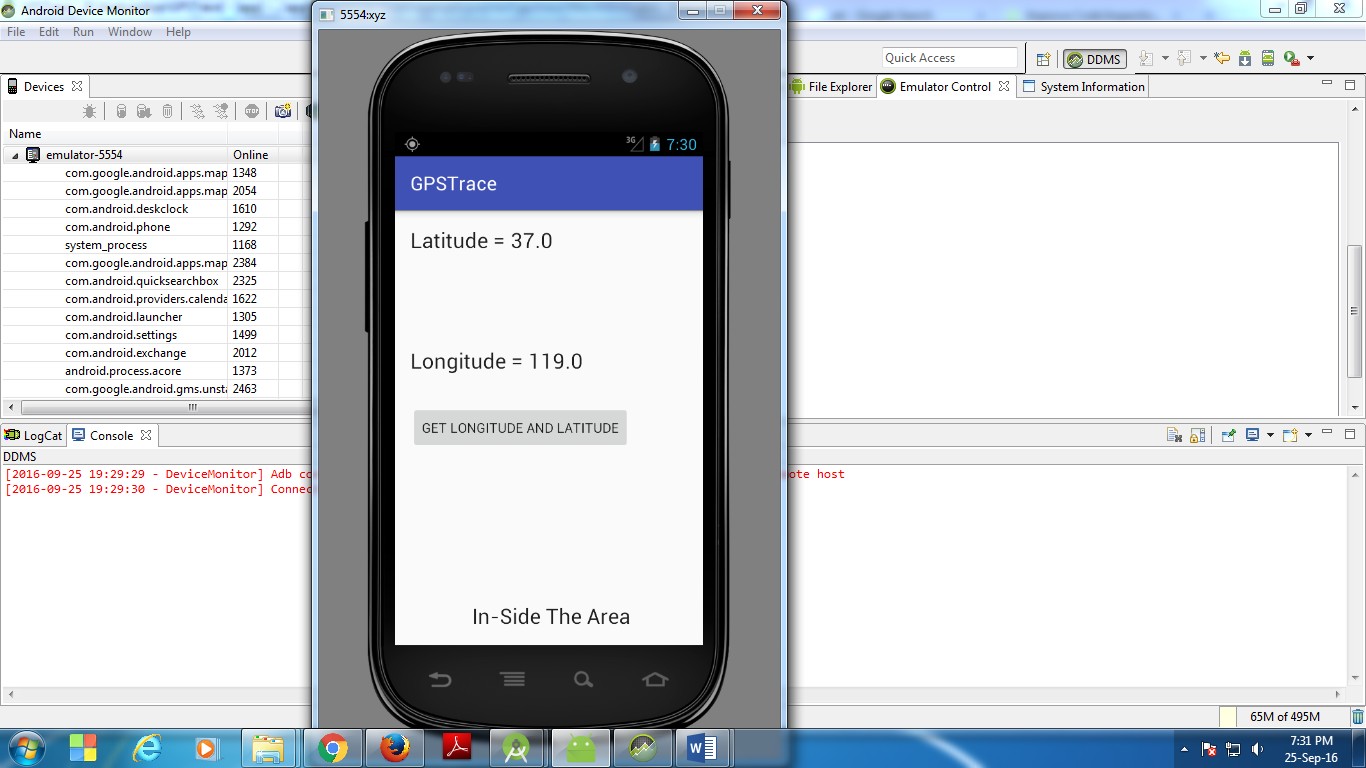
@Override

**public void** onStatusChanged(String provider, **int**status, Bundle extras) { Log.*d*(**"Latitude"**,**"status"**);

}

}

**Output:-**



**Permission :-**

*<?*xml version="1.0" encoding="utf-8"*?>*

<manifest xmlns:android=["http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) package="com.example.prashant.gpstrace">

<application android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name" android:supportsRtl="true" android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</**activity**>

</**application**>

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

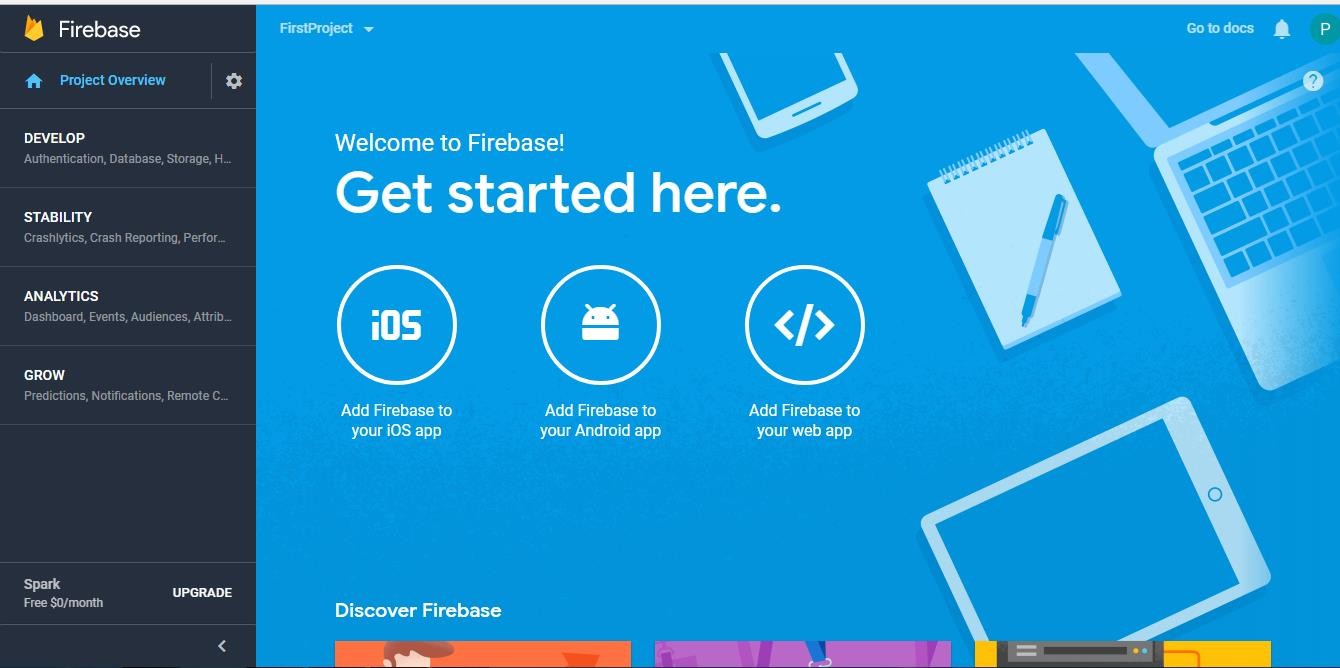
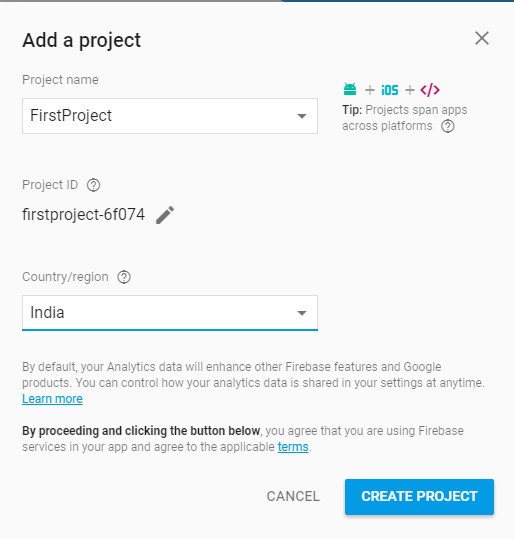
<uses-permission android:name="android.permission.INTERNET" />

</**manifest**>

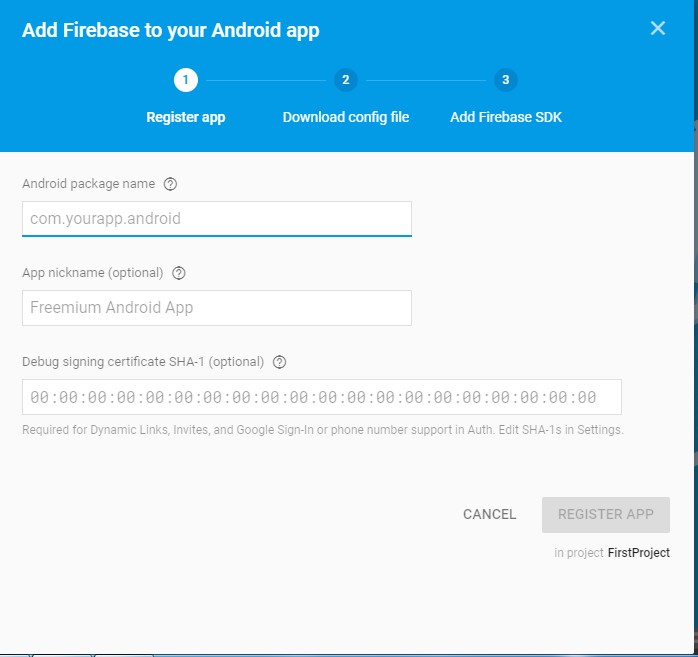
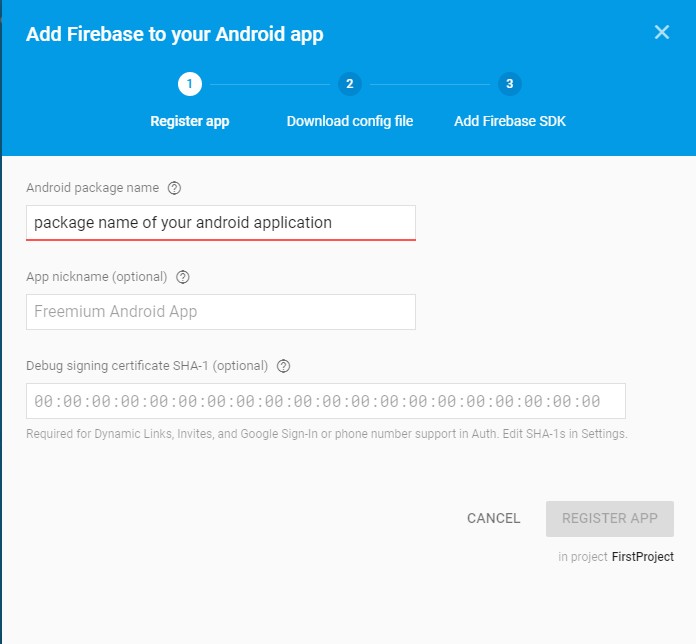
**Practical No. 3**

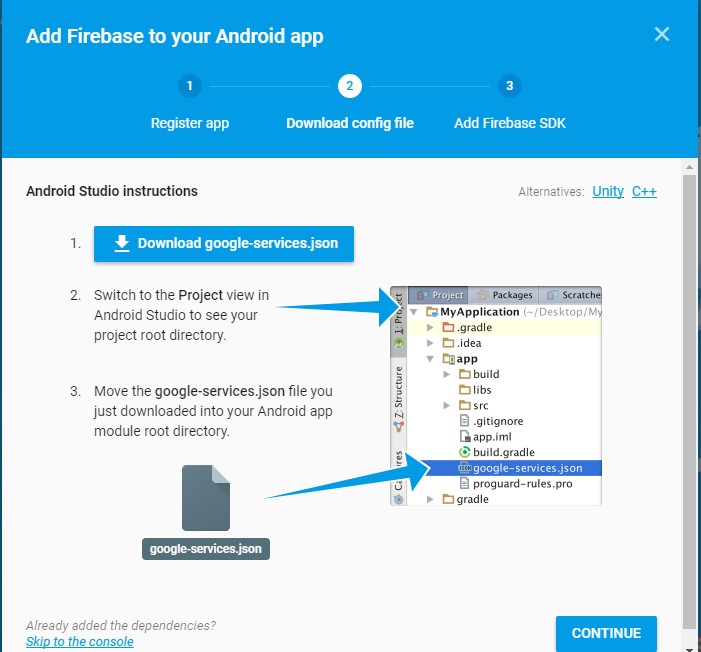
# Aim: - Design and develop app demonstrating Simple Downstream Messaging.

# 

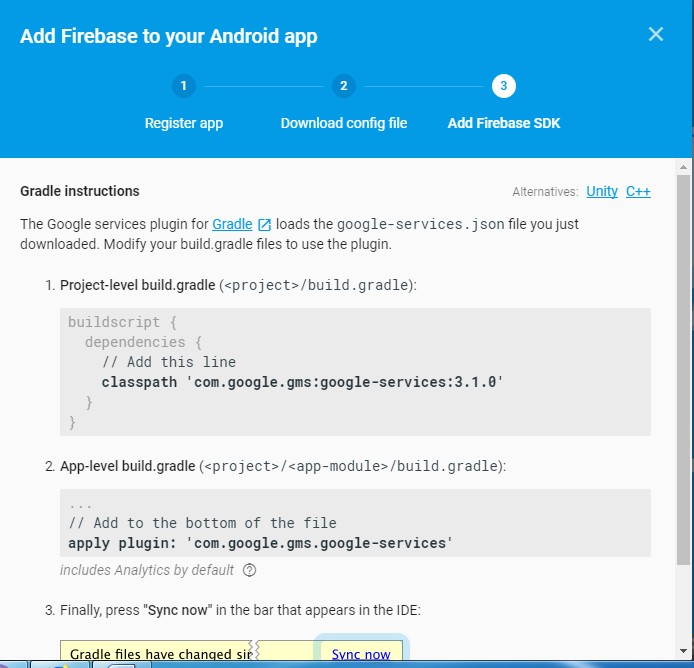


# 1) Create project in firebase console and follow their steps:

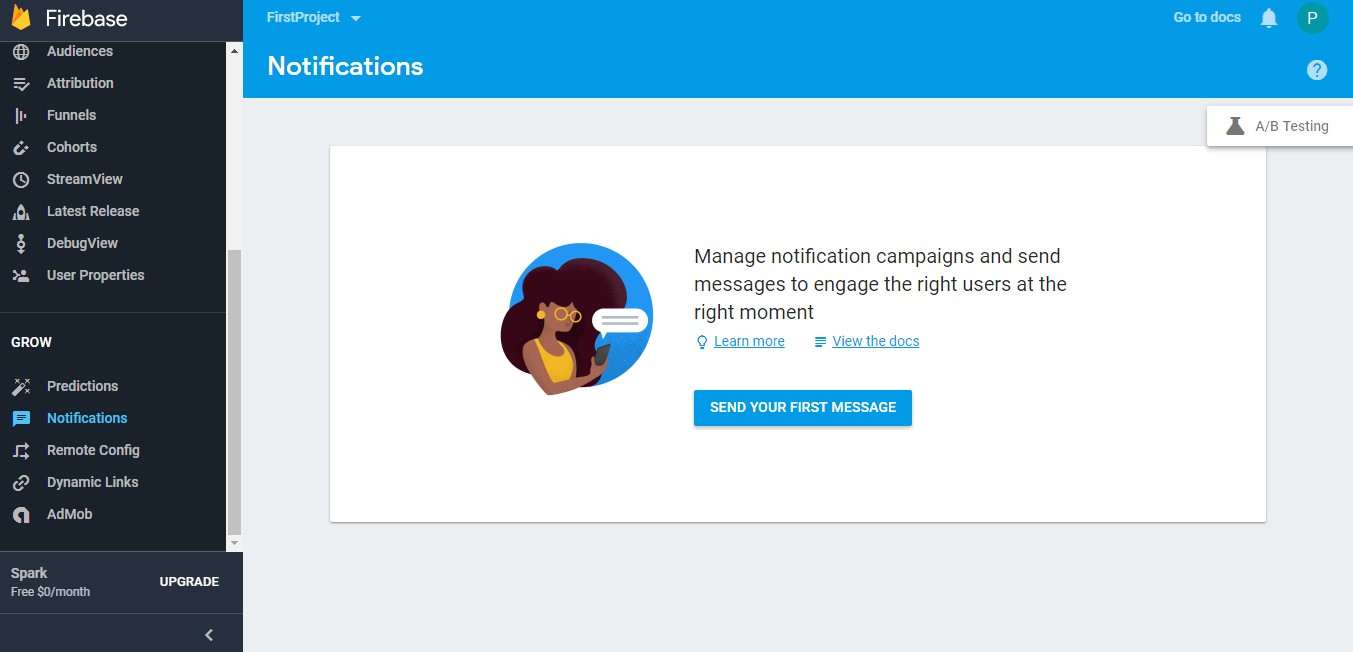




**Download google-services.json and copy it to app folder and follow the steps**



**In Grow tab select notification for sending messages**



**MainActivity.java**

package com.example.prashant.firebasedemo;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

import android.app.Notification;

import android.app.NotificationManager; import android.content.Context;

import android.os.Build; import android.os.Bundle;

import android.support.v7.app.AppCompatActivity; import android.util.Log;

import android.view.View; import android.widget.Button; import android.widget.Toast;

import com.google.firebase.iid.FirebaseInstanceId; import com.google.firebase.messaging.FirebaseMessaging;

public class MainActivity extends AppCompatActivity {

private static final String TAG = "MainActivity"; @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

if (getIntent().getExtras() != null) {

for (String key : getIntent().getExtras().keySet()) { Object value = getIntent().getExtras().get(key); Log.d(TAG, "Key: " + key + " Value: " + value);

}

}

// [END handle\_data\_extras]

Button subscribeButton = (Button)findViewById(R.id.button3); subscribeButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v) {

// [START subscribe\_topics]

FirebaseMessaging.getInstance().subscribeToTopic("news");

// [END subscribe\_topics]

// Log and toast

String msg = getString(R.string.msg\_subscribed); Log.d(TAG, msg); Toast.makeText(MainActivity.this, msg,

Toast.LENGTH\_SHORT).show();

}

});

Button logTokenButton = (Button)findViewById(R.id.button4); logTokenButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// Get token String token =

FirebaseInstanceId.getInstance().getToken();

// Log and toast Log.d(TAG, "Done");

Toast.makeText(MainActivity.this, "Done",

Toast.LENGTH\_SHORT).show();

}

});

}

}

# MyFirebaseInstanceIDService.java

package com.example.prashant.firebasedemo;

/\*\*

\* Created by PRASHANT on 13-Nov-17.

\*/

import android.util.Log;

import com.google.firebase.iid.FirebaseInstanceId;

import com.google.firebase.iid.FirebaseInstanceIdService;

public class MyFirebaseInstanceIDService extends FirebaseInstanceIdService {

private static final String TAG = "MyFirebaseIIDService";

/\*\*

* Called if InstanceID token is updated. This may occur if the security of
* the previous token had been compromised. Note that this is called when the InstanceID token
* is initially generated so this is where you would retrieve the token.

\*/

// [START refresh\_token] @Override

public void onTokenRefresh() {

// Get updated InstanceID token. String refreshedToken =

FirebaseInstanceId.getInstance().getToken();

Log.d(TAG, "Refreshed token: " + refreshedToken);

// If you want to send messages to this application instance

or

// manage this apps subscriptions on the server side, send the

// Instance ID token to your app server. sendRegistrationToServer(refreshedToken);

}

// [END refresh\_token]

/\*\*

* Persist token to third-party servers.

\*

* Modify this method to associate the user's FCM InstanceID token with any server-side account
* maintained by your application.

\*

* @param token The new token.

\*/

private void sendRegistrationToServer(String token) {

// TODO: Implement this method to send token to your app

server.

}

}

# MyFirebaseMessagingService.java

package com.example.prashant.firebasedemo;

/\*\*

\* Created by PRASHANT on 13-Nov-17.

\*/

import android.app.NotificationManager; import android.app.PendingIntent; import android.content.Context;

import android.content.Intent;

import android.media.RingtoneManager; import android.net.Uri;

import android.support.v4.app.NotificationCompat; import android.util.Log;

import com.google.firebase.\*;

import com.google.firebase.messaging.FirebaseMessagingService; import com.google.firebase.messaging.RemoteMessage;

import com.firebase.jobdispatcher.Constraint;

import com.firebase.jobdispatcher.FirebaseJobDispatcher; import com.firebase.jobdispatcher.GooglePlayDriver; import com.firebase.jobdispatcher.Job;

public class MyFirebaseMessagingService extends FirebaseMessagingService {

private static final String TAG = "MyFirebaseMsgService";

/\*\*

* Called when message is received.

\*

* @param remoteMessage Object representing the message received from Firebase Cloud Messaging.

\*/

// [START receive\_message] @Override

public void onMessageReceived(RemoteMessage remoteMessage) {

// [START\_EXCLUDE]

// There are two types of messages data messages and notification messages. Data messages are handled

// here in onMessageReceived whether the app is in the foreground or background. Data messages are the type

// traditionally used with GCM. Notification messages are only received here in onMessageReceived when the app

// is in the foreground. When the app is in the background an automatically generated notification is displayed.

// When the user taps on the notification they are returned to the app. Messages containing both notification

// and data payloads are treated as notification messages. The Firebase console always sends notification

// messages. For more see: https://firebase.google.com/docs/cloud-messaging/concept-options

// [END\_EXCLUDE]

// TODO(developer): Handle FCM messages here.

// Not getting messages here? See why this may be: https://goo.gl/39bRNJ

Log.d(TAG, "From: " + remoteMessage.getFrom());

// Check if message contains a data payload. if (remoteMessage.getData().size() > 0) {

Log.d(TAG, "Message data payload: " + remoteMessage.getData());

if (/\* Check if data needs to be processed by long running job \*/ true) {

// For long-running tasks (10 seconds or more) use Firebase Job Dispatcher.

scheduleJob();

} else {

// Handle message within 10 seconds handleNow();

}

}

// Check if message contains a notification payload. if (remoteMessage.getNotification() != null) {

Log.d(TAG, "Message Notification Body: " + remoteMessage.getNotification().getBody());

}

// Also if you intend on generating your own notifications as a result of a received FCM

// message, here is where that should be initiated. See sendNotification method below.

}

// [END receive\_message]

/\*\*

* Schedule a job using FirebaseJobDispatcher.

\*/

private void scheduleJob() {

// [START dispatch\_job] FirebaseJobDispatcher dispatcher = new

FirebaseJobDispatcher(new GooglePlayDriver(this));

Job myJob = dispatcher.newJobBuilder()

.setService(MyJobService.class)

.setTag("my-job-tag")

.build(); dispatcher.schedule(myJob);

// [END dispatch\_job]

}

/\*\*

* Handle time allotted to BroadcastReceivers.

\*/

private void handleNow() {

Log.d(TAG, "Short lived task is done.");

}

/\*\*

* Create and show a simple notification containing the received FCM message.

\*

* @param messageBody FCM message body received.

\*/

private void sendNotification(String messageBody) { Intent intent = new Intent(this, MainActivity.class); intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

PendingIntent pendingIntent = PendingIntent.getActivity(this,

0 /\* Request code \*/, intent,

PendingIntent.FLAG\_ONE\_SHOT);

// String channelId = getString(R.string.default\_notification\_channel\_id);

Uri defaultSoundUri= RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);

NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this)

.setSmallIcon(R.mipmap.ic\_launcher)

.setContentTitle("FCM Message")

.setContentText(messageBody)

.setAutoCancel(true)

.setSound(defaultSoundUri)

.setContentIntent(pendingIntent);

NotificationManager notificationManager = (NotificationManager)

getSystemService(Context.NOTIFICATION\_SERVICE);

notificationManager.notify(0 /\* ID of notification \*/, notificationBuilder.build());

}

}

# MyJobService.java

package com.example.prashant.firebasedemo;

/\*\*

\* Created by PRASHANT on 13-Nov-17.

\*/

import android.util.Log;

import com.firebase.jobdispatcher.JobParameters; import com.firebase.jobdispatcher.JobService;

public class MyJobService extends JobService {

private static final String TAG = "MyJobService"; @Override

public boolean onStartJob(JobParameters jobParameters) { Log.d(TAG, "Performing long running task in scheduled job");

// TODO(developer): add long running task here. return false;

}

@Override

public boolean onStopJob(JobParameters jobParameters) { return false;

}

}

**Practical No. 4**

**Aim: - Design and develop app demonstrating Send Upstream Messages.**

**Code :-**

Init.php

<?php

$host = "localhost";

$db\_user = "root";

$db\_password = "";

$db\_name = "fcm\_db";

$con = mysqli\_connect($host, $db\_user, $db\_password, $db\_name);

if($con)

echo " Connection Successful ";

else

?>

echo "Connection error" ;

Fcm\_insert.php

<?php

require "init.php";

$fcm\_token = $\_POST["fcm\_token"];

$sql = "insert into fcm\_info values('".$fcm\_token."');"; mysqli\_query($con, $sql);

mysqli\_close($con);

?>

MainActivity.java

package com.example.prashant.gcmmanager;

import android.content.Context;

import android.content.SharedPreferences;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

import android.view.View; import android.widget.Button;

import com.android.volley.AuthFailureError; import com.android.volley.Request;

import com.android.volley.Response; import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import java.util.HashMap; import java.util.Map;

public class MainActivity extends AppCompatActivity { Button button;

String app\_server\_url = "http://10.0.2.2/fcmtest/fcm\_insert.php";

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

button = (Button)findViewById(R.id.button2); button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) { SharedPreferences sharedPreferences =

getApplicationContext().getSharedPreferences(getString(R.string.FCM\_PR EF), Context.MODE\_PRIVATE);

final String token = sharedPreferences.getString(getString(R.string.FCM\_TOKEN),"");

StringRequest stringRequest = new StringRequest(Request.Method.POST, app\_server\_url, new Response.Listener<String>() {

@Override

public void onResponse(String response) {

}

}, new Response.ErrorListener() { @Override

public void onErrorResponse(VolleyError error) {

}

})

{

AuthFailureError {

@Override

protected Map<String, String> getParams() throws

Map<String,String> params = new

HashMap<String, String>();

params.put("fcm\_token",token); return params;

}

};

MySingleton.getmInstance(MainActivity.this).addtoreque(stringRequest);

}

});

}

}

FCMInstanceIDService.java

package com.example.prashant.gcmmanager; import android.content.Context;

import android.content.SharedPreferences;

import com.google.firebase.iid.FirebaseInstanceId;

import com.google.firebase.iid.FirebaseInstanceIdService;

/\*\*

\* Created by PRASHANT on 15-Nov-17.

\*/

public class FCMInstanceIDService extends FirebaseInstanceIdService { @Override

public void onTokenRefresh() {

String recent\_token = FirebaseInstanceId.getInstance().getToken();

SharedPreferences sharedPreferences = getApplicationContext().getSharedPreferences(getString(R.string.FCM\_PR EF), Context.MODE\_PRIVATE);

SharedPreferences.Editor editor= sharedPreferences.edit(); editor.putString(getString(R.string.FCM\_TOKEN),recent\_token); editor.commit();

}

}

FCMMessagingService.java

package com.example.prashant.gcmmanager;

import android.app.NotificationManager; import android.app.PendingIntent; import android.content.Context;

import android.content.Intent;

import android.support.v7.app.NotificationCompat;

import com.google.firebase.messaging.FirebaseMessagingService; import com.google.firebase.messaging.RemoteMessage;

/\*\*

\* Created by PRASHANT on 15-Nov-17.

\*/

public class FCMMessagingService extends FirebaseMessagingService { @Override

public void onMessageReceived(RemoteMessage remoteMessage) {

String title = remoteMessage.getNotification().getTitle(); String message = remoteMessage.getNotification().getBody();

Intent intent = new Intent(this,MainActivity.class); intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK); PendingIntent pendingIntent =

PendingIntent.getActivity(this,0,intent,PendingIntent.FLAG\_ONE\_SHOT); NotificationCompat.Builder notificationBuilder = new

NotificationCompat.Builder(this); notificationBuilder.setContentTitle(title); notificationBuilder.setContentText(message); notificationBuilder.setSmallIcon(R.mipmap.ic\_launcher); notificationBuilder.setAutoCancel(true); notificationBuilder.setContentIntent(pendingIntent); NotificationManager notificationManager =

(NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE); notificationManager.notify(0, notificationBuilder.build());

super.onMessageReceived(remoteMessage);

}

}

MySingleton.java package com.example.prashant.gcmmanager; import android.content.Context;

import com.android.volley.Request; import com.android.volley.RequestQueue;

import com.android.volley.toolbox.Volley;

/\*\*

\* Created by PRASHANT on 15-Nov-17.

\*/

public class MySingleton {

private static MySingleton mInstance; private static Context mctx;

private RequestQueue requestQueue; private MySingleton(Context context)

{

mctx = context;

requestQueue =getRequestQueue();

}

private RequestQueue getRequestQueue()

{

if(requestQueue ==null)

{

requestQueue = Volley.newRequestQueue(mctx.getApplicationContext());

}

return requestQueue;

}

public static synchronized MySingleton getmInstance(Context context)

{

if(mInstance == null)

{

mInstance = new MySingleton(context);

}

return mInstance;

}

public void addtoreque(Request<?> request)

{

getRequestQueue().add(request);

}

}

**Practical No. 5**

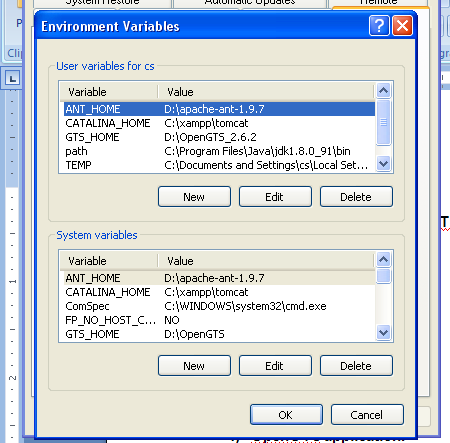
**Aim: - Demonstrate use of OpenGTS (Open Source GPS Tracking System). Settings:-**

**Required Software:-**

1. **JDK 1.6**
2. **XAMPP Server**
3. **Mysql-java connector**
4. **OpenGTS application.**

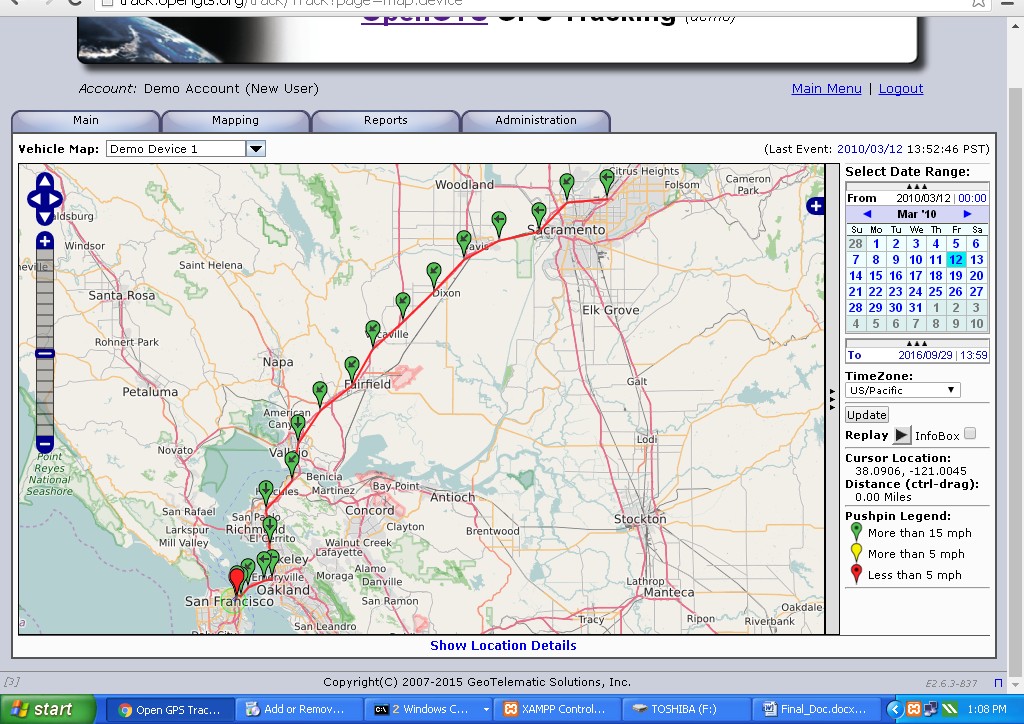
[**http://www.opengts.org/**](http://www.opengts.org/)

**Set Environment Variables:-**



1. **Open command Prompt and go to D:\OpenGTS\_2.6.2**
2. **Type command ant all**
3. **Type command ant track**
4. **Type command ant track.deploy**
5. **Type command initdb.bat –rootUser=root –pass=123456789**
6. **Type Command admin.bat Account –account:admin –pass:123456 –create**
7. **Type url 127.0.0.0:8080/track/Track and login with admin and 123456**



****

**Practical No. 6**

**Aim: - Context-Aware system.**

Context-awareness is a key concept in ubiquitous computing. The Java Context-Awareness Framework (JCAF) is a Java-based context-awareness infrastructure and programming API for creating context-aware applications

<https://sourceforge.net/projects/jcaf/>

**Practical No. 7**

**Aim: - Develop application demonstrating Human Computer Interactions**

**Code :-**

import java.awt.\*; importjava.awt.event.\*;

public class CloseableSimpleWarning extends Frame implements WindowListener

{

static private final intframe\_height = 150; static private final intframe\_width = 250;

publicCloseableSimpleWarning()

{

//setBackground(Color.red);

//setForeground(Color.black); setTitle("Warning");

setSize(frame\_width, frame\_height); addWindowListener(this);

}

public void windowClosing (WindowEvent e)

{

System.exit(0);

}

public void windowClosed (WindowEvent e)

{

System.exit(0);

}

public void windowIconified (WindowEvent e)

{

System.exit(0);

}

public void windowDeiconified (WindowEvent e)

{

System.exit(0);

}

public void windowOpened (WindowEvent e)

{

System.exit(0);

}

public void windowActivated (WindowEvent e)

{

System.exit(0);

}

public void windowDeactivated (WindowEvent e)

{

System.exit(0);

}

public static void main(String [] args)

{

CloseableSimpleWarning f = new CloseableSimpleWarning();

f.show();

}

}

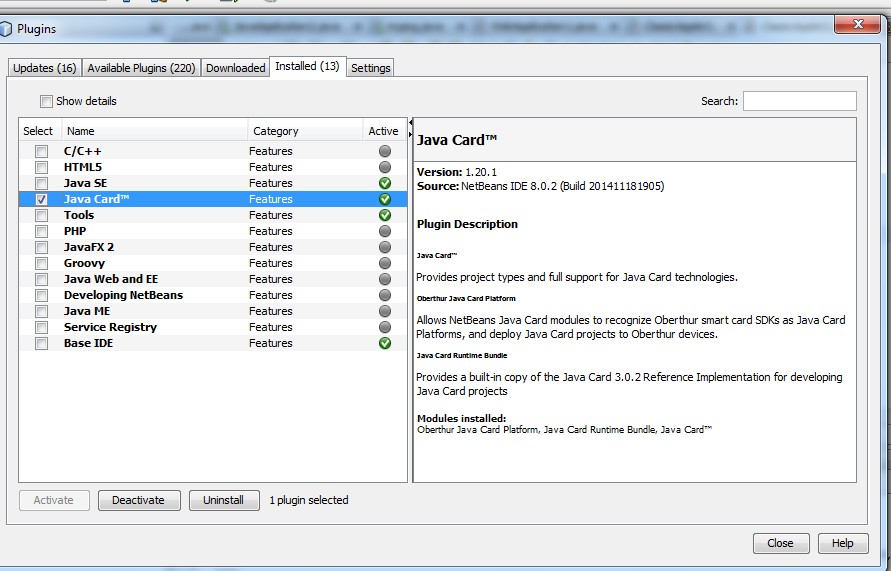
**Practical No. 8**

**Aim: - Write a Java Card applet**

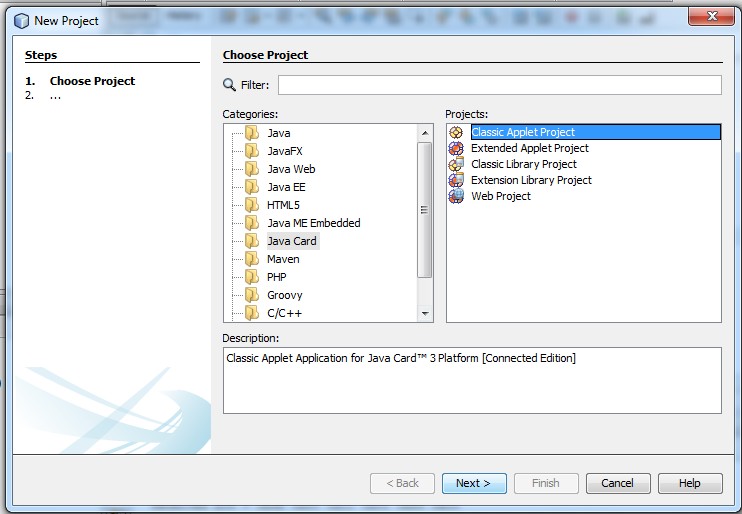
Download Java Card Sdk from

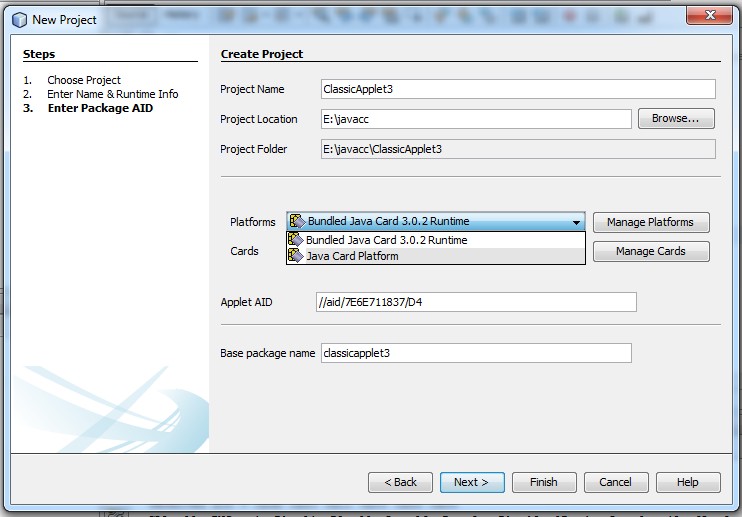
[http://www.oracle.com/technetwork/java/embedded/javacard/downloads/javacard-sdk-](http://www.oracle.com/technetwork/java/embedded/javacard/downloads/javacard-sdk-2043229.html) [2043229.html](http://www.oracle.com/technetwork/java/embedded/javacard/downloads/javacard-sdk-2043229.html)

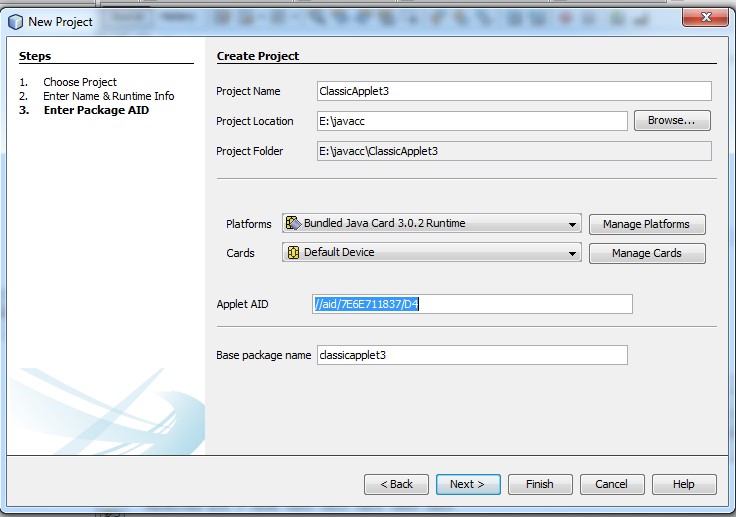
Install in Netbeans as plugin :- Tools -> Plugin

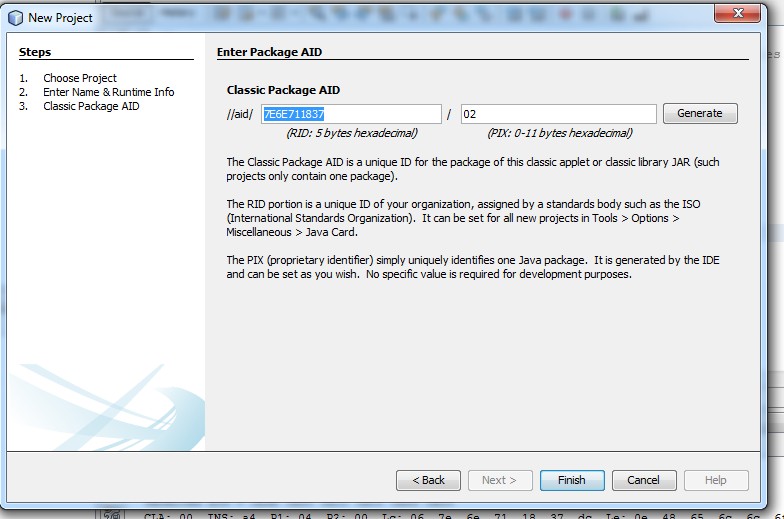


Create an application









Code:-

/\*

* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.

\*/

package classicapplet2;

importjavacard.framework.\*;

/\*\*

\*

* @author Prashant

\*/

public class ClassicApplet2 extends Applet {

/\*\*

* + Installs this applet.

\*

* + @parambArray
  + the array containing installation parameters
  + @parambOffset
  + the starting offset in bArray
  + @parambLength
  + the length in bytes of the parameter data in bArray

\*/

private byte[] received;

private static final short MAX\_LENGTH = 256;

private static final byte[] helloFidesmo = {(byte)'H',(byte)'e',(byte)'l',(byte)'l',(byte)'o',(byte)'

',(byte)'F',(byte)'i',(byte)'d',(byte)'e',(byte)'s',(byte)'m',(byte)'o',(byte)'!'};

public static void install(byte[] bArray, short bOffset, byte bLength) {

new ClassicApplet2();

}

/\*\*

* + Only this class's install method should create the applet object.

\*/

protected ClassicApplet2() {

received = new byte[MAX\_LENGTH]; register();

}

/\*\*

* + Processes an incoming APDU.

\*

* + @see APDU
  + @paramapdu
  + the incoming APDU

\*/

public void process(APDU apdu) {

//Insert your code here byte buffer[] = apdu.getBuffer();

short length = (short) helloFidesmo.length;

Util.arrayCopyNonAtomic(helloFidesmo, (short)0, buffer, (short)0, (short)length); apdu.setOutgoingAndSend((short)0, length);

}

}