GTBIT Android Application

DISSERTATION

Submitted in partial fulfillment of the
Requirements for the award of the degree

of

Bachelor of Technology

in

Computer Science & Engineering

By:

Baldeep Singh Handa (09/CSE1/2012)
Divgun Singh Sethi (58/CSE1/2012)
Inderdeep Singh Khanna (11/CSE1/2012)

Under the guidance of :

Mr. Gaurav Sandhu

Department of Computer Science & Engineering
Guru Tegh Bahadur Institute of Technology

Guru Gobind Singh Indraprastha University
Dwarka, New Delhi
Year 2015-2016

DECLARATION

We hereby declare that all the work presented in the dissertation entitled "**GTBIT Android Application**" in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science & Engineering,** Guru Tegh Bahadur Institute of Technology, affiliated to Guru Gobind Singh Indraprastha University Delhi is an authentic record of our own work carried out under the guidance of **Mr. Gaurav Sandhu**

Date:

Baldeep Singh Handa (09/CSE1/2012)

Divgun Singh Sethi (58/CSE1/2012)

Inderdeep Singh Khanna (11/CSE1/2012)

CERTIFICATE

This is to certify that dissertation entitled **"Android Group Chat",** which is submitted by **Mr. Baldeep Singh Handa, Mr. Divgun Singh Sethi** and **Mr. Inderdeep Singh Khanna** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science & Engineering**, Guru Tegh Bahadur Institute of Technology, New Delhi is an authentic record of the candidate's own work carried out by them under our guidance. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Gaurav Sandhu
(Project Guide)

Mrs. Gurpreet Kaur
(Head of Department)
Computer Science & Engineering

Date:

ACKNOWLEDGEMENT

We would like to express our great gratitude towards our supervisor, **Mr. Gaurav Sandhu** who has given us support and suggestions. Without their help we could not have presented this dissertation up to the present standard. We also take this opportunity to give thanks to all others who gave us support for the project or in other aspects of our study at Guru Tegh Bahadur Institute of Technology.

Date:

ABSTRACT

Our purpose is to design and build an android application for “GTBIT”. This Android based application is designed to bring essential Institute information to the user. It uses the client/server architecture to incorporate a fully working group chat functionality. This application consists of eight different pages with different functionalities solving various problems of students. The application is developed to run on the Android operating system, using Android Studio. The programming language is Java.

TABLES AND FIGURES

CONTENTS

Chapter One

INTRODUCTION

INTRODUCTION

This project is aimed at creating a common platform for all the students of any class to retrieve and share information, to chat with one another through groups. This Android based application is designed to bring essential Institute information to the user. This application also aims at sharing image documents in textual format via OCR which will make it easier for the students to share notes during exams. The chat works on Google Cloud Messaging with which the application sends and receives messages to other application. The server requests to GCM is handled by a Python API. The OCR is handled by using the Tesseract project for Optical Character recognition (made by Google).

The system runs on multiple android devices and versions, offers a good and simple GUI interface and connects to a server database.

1.1 Existing system

The existing system involves the member to come to college and view the details. The existing system includes poll messaging where the client and server should be always connected. This results in more battery consumption and increased bandwidth. Also the system does not include cloud thus the client and server are directly connected. So once the message send is not received to the client due to some network problems the data will be lost.

The details are given as mark sheet which is in paper format. The details of symposium are given as announcement or put in notice board and is not known to parents. The parents may not know about the member's daily attendance, it is time consuming. It is difficult to know about others details. It is difficult to know about member activities by parents. The results have to be seen in college website or directly in the college and the parents will not be notified of the result date and time.

1.2 Proposed System

The proposed system is very easy to operate. The main advantages of proposed system are minimizing the network bandwidth and battery consumption. User can find various information including basic information, information about various societies of GTBIT, contact information and information about the latest events and fest. Users can also download their respective syllabus, lecture plan, time table and other notes directly from the app.

It uses existing connections for Google services. This requires the users to sign into their Google account on Android. An application doesn't need to be running to receive data messages. The GCM service handles all aspects of queuing of messages and delivery to the target. Android application running on the target device. Information can be send quickly and easily.

The protection of data from either accidental or unauthorized, intentional modification, destruction is possible. It is adaptability. So it can be reprogrammed to do different tasks. The proposed system is android app which can be installed in all android mobile devices. The user have to register with the college server to get details and attendance details from college. The user can send their query or feedback to the college server. The queries will be answer and all the member activities will be known to parents. This app is time saving and user friendly. An application doesn't need to be running to receive data messages. Information can be send quickly and easily. The protection of data from either accidental or unauthorized, intentional modification, destruction is possible. It is adaptability. So it can be reprogrammed to do different tasks. The member activities are monitored by the parents.

Chapter Two

SYSTEM REQUIREMENTS

REQUIREMENT ANALYSIS (SRS)

2.1 Purpose

Main aim of developing GTBIT mobile Application is to provide an easy way not only to send and retrieve information/messages but also to deliver it to multiple recipients at the same time. It also enable character conversion from an image into machine-encoded text. A SQLAlchemy database is used as a backbone to store the user information.

2.2 Scope

This Application works on smartphones with android OS by using Google Cloud Messaging (GCM) for establishing connection and maintaining database through which users at different locations can connect with each other simultaneously. User can Login into the application using existing Facebook/Google Plus account or Register on the API server. We have a vision to expand the application and incorporate many other features.

2.3 Definition of Terms

- **Emoticon**- An emoticon, etymologically a portmanteau of emotion and icon, is a meta communicative pictorial representation of a facial expression that, in the absence of body language and prosody, serves to draw a receiver's attention to the tenor or temper of a sender's nominal non-verbal communication, changing and improving its interpretation.
- **Google Cloud Messaging** (GCM) - GCM is a service that enables developers to send data from servers to both Android applications or Chrome apps and extensions

- **SQLAlchemy-** It is an open source SQL toolkit and object-relational mapper (ORM) for the Python programming language released under the MIT License.

2.4 Benefits

- To establish communication easily.
- Quickly deliver messages to multiple recipients simultaneously.
- To provide easy and faster access to notices, time tables and schedules.
- To provide user friendly environment.

2.5 Overall Description

2.5.1 Product Perspective

It provides simple database rather than complex ones for high requirements and it provides good and easy graphical user interface to both new as well as experienced user of the android mobile phones.

2.5.2 Software Requirements:

- **Android Studio**- an Integrated Development Environment (IDE) for developing on the Android Platform. Based on JetBrains Intelli J IDEA software, Android Studio is designed specifically for Android development.

- **Android SDK Tools**- Android applications are usually developed in Java programming language using the Android Software Development Kit (SDK). It includes a comprehensive set of development tools.

- **Emulator**- A mobile device Emulator is required for testing and presentation of the application. It comes bundled with Android SDK tools.

- **Tesseract**- Tesseract is probably the most accurate open source OCR engine available. Combined with the Leptonica Image Processing Library it can read a wide variety of image formats and convert them to text. A fork library of tesseract has been used in our application.

- **Emojicon**- Emojicon is an open source android library used to implement emoticons in EditText boxes.

2.5.3 Hardware Requirements:

- **PC**- Personal Computer to with atleast 2 Gb of RAM, 15Gb of storage.
- **Android Device**- A physical android device with atleast 512Mb RAM,Minimum Android Version Ice Cream – Sandwich.

2.6 Functional Requirements

- **Push Notifications:** Our application implements push notification mechanism provided by GCM, where server automatically sends messages to the device whenever available so that it need not poll the server continuously for messages which may drain battery power.

- **Login API:** As GCM server identifies a particular device by its registration ID, our custom designed Login API will help user for one time registration with the GCM server thus enabling the device to send and receive messages to and from the server. We have also provided the facility to login through ones Facebook or Google Plus account.

- **Image & Location Sharing:** Our application provides the facility to share images from gallery with all the members of the group. Also, one can share his/her location

with just a single click in chatroom.

- **Optical Character Recognition(OCR):** One of the most important and useful feature of our application is **OCR.** OCR implemented using Tesseract Android Tools will allow our application to extract text from the image being captured by the device camera and forward the extracted text.

- **Emoticons:** Only text messaging would have been a boring choice, thus to your rescue and to make our application more lively we have implemented emoticons using a custom android library dedicated solely to '**Emojis**'.

- **Speech to Text:** User can provide speech input to our application instead of typing the text and see his voice been converted to text automatically. User can also make his device speak out a message simply by long pressing the text and leave rest to the **Text to Speech (TTS)** feature of our application.

- **#TAGS:** Another cool feature of our application is using **Hashtags** for easier search. Send messages with various hashtags like - #meaning <word> returs meaning of the following word.

2.7 Non-functional Requirements

- The overall system should be fast and error free.
- It should have built in error checking and correction facilities.
- The system should be able to handle large amount of data comfortably.

2.8 Design Constraints

- Application runs on minimum SDK 15 IceCreamSandwich android smartphone.
- The application is developed on Android Studio.
- It uses SQLAlchemy as database management system.

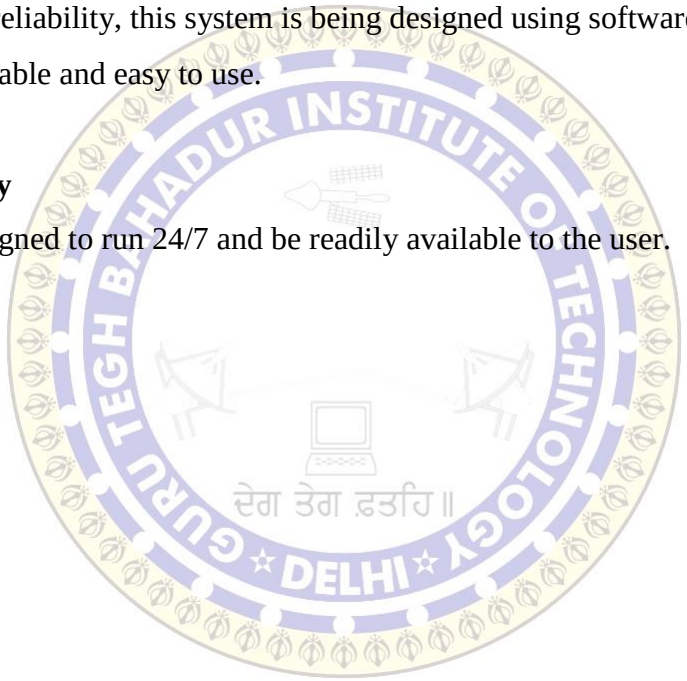
2.9 Attributes

2.9.1 Reliability

In order to ensure reliability, this system is being designed using software that is established to be stable and easy to use.

2.9.2 Availability

This system is designed to run 24/7 and be readily available to the user.



Chapter Three

SYSTEM DESIGN

SYSTEM DESIGN

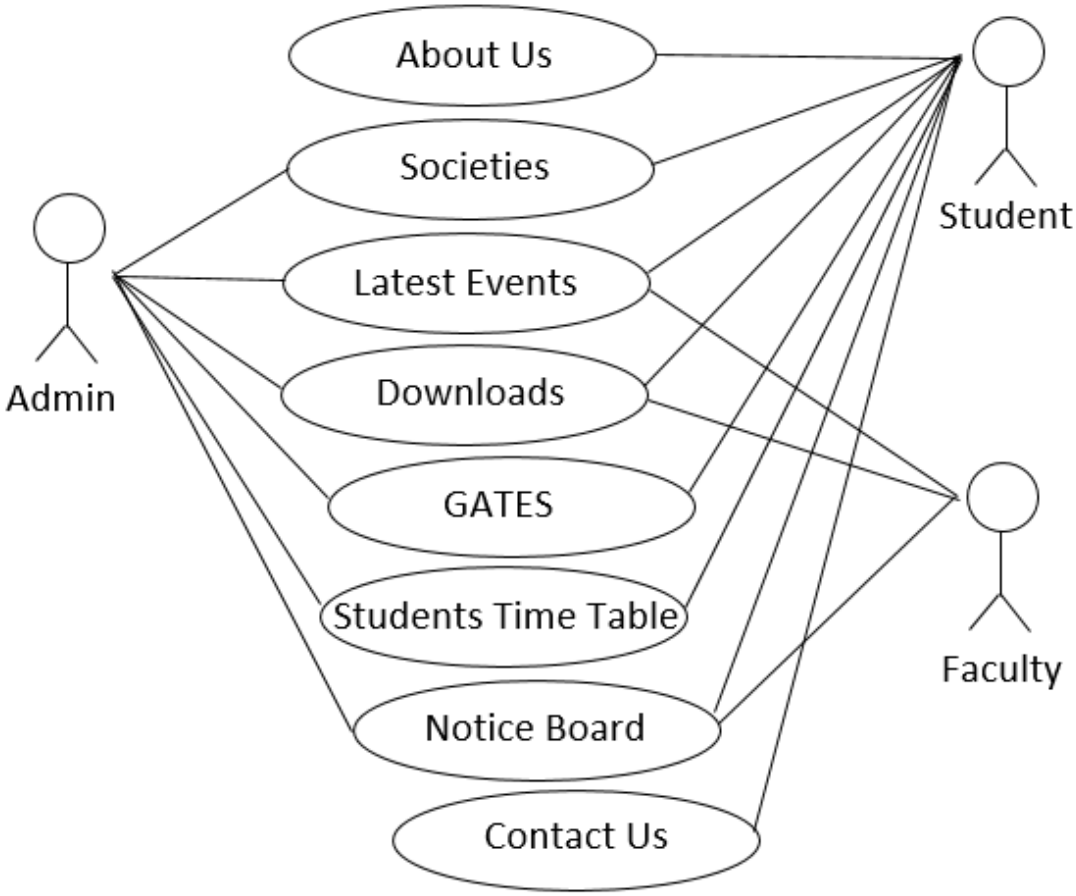


Fig. 3.1- Use Case Diagram

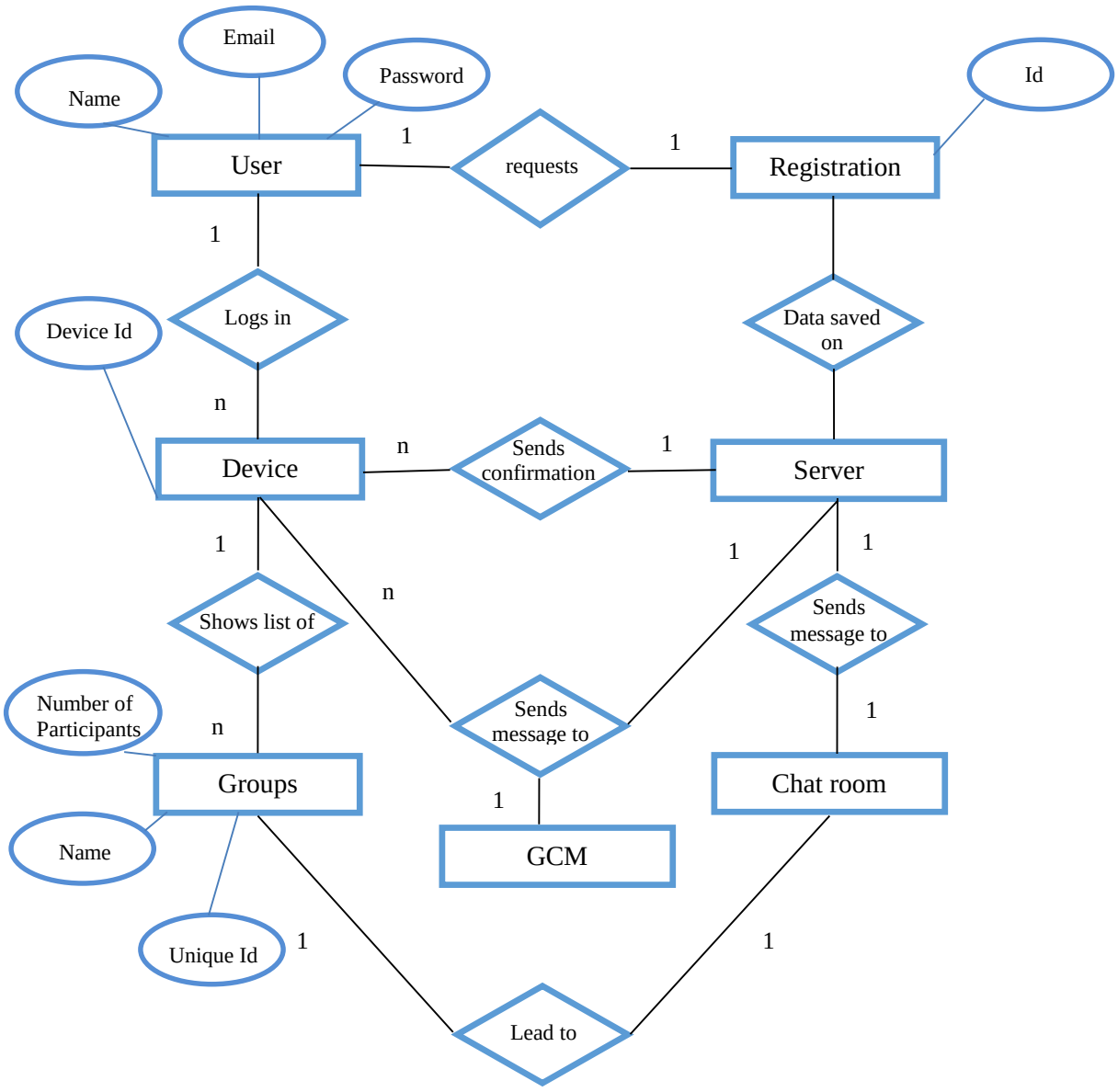
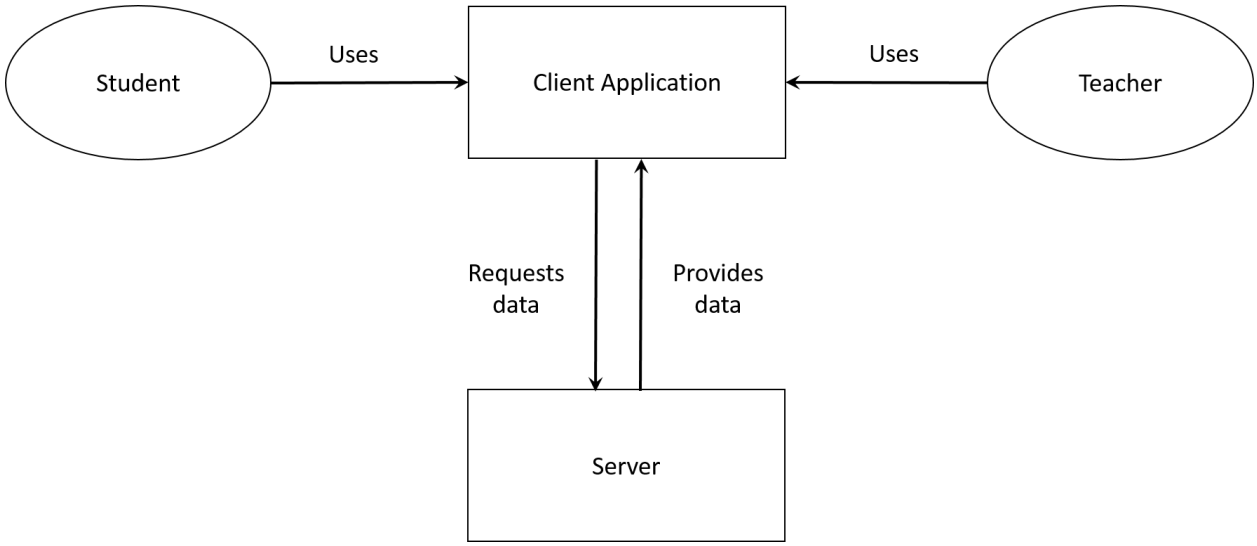


Fig. 3.2- Entity Relationship Diagram



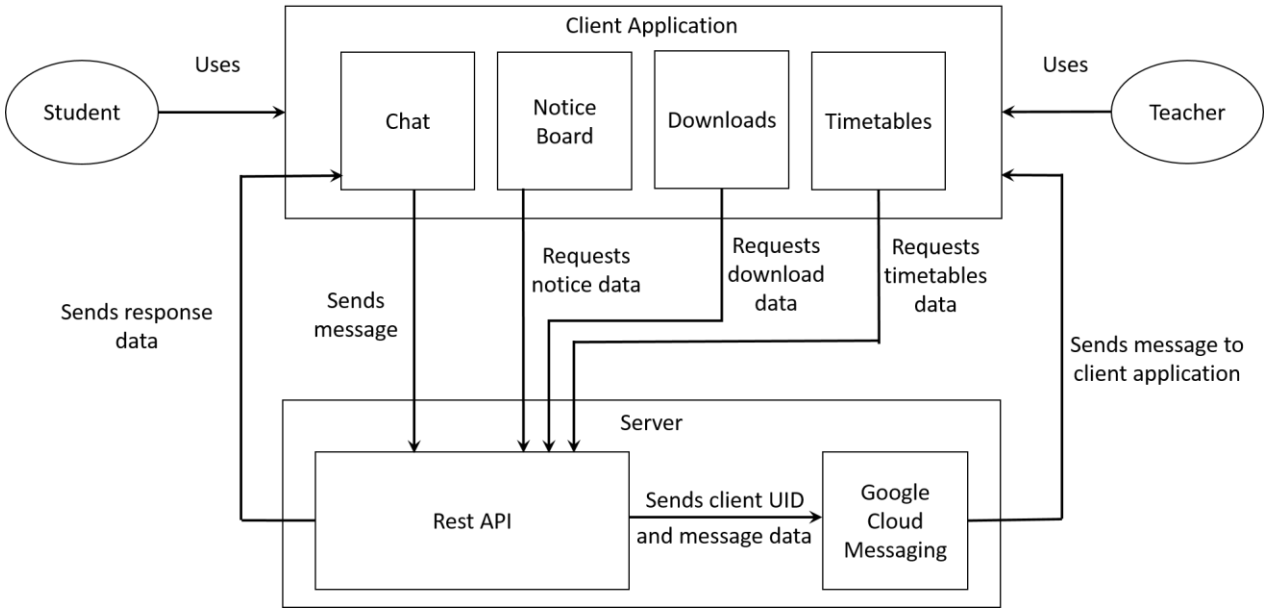


Fig. 3.3- Data Flow Diagram

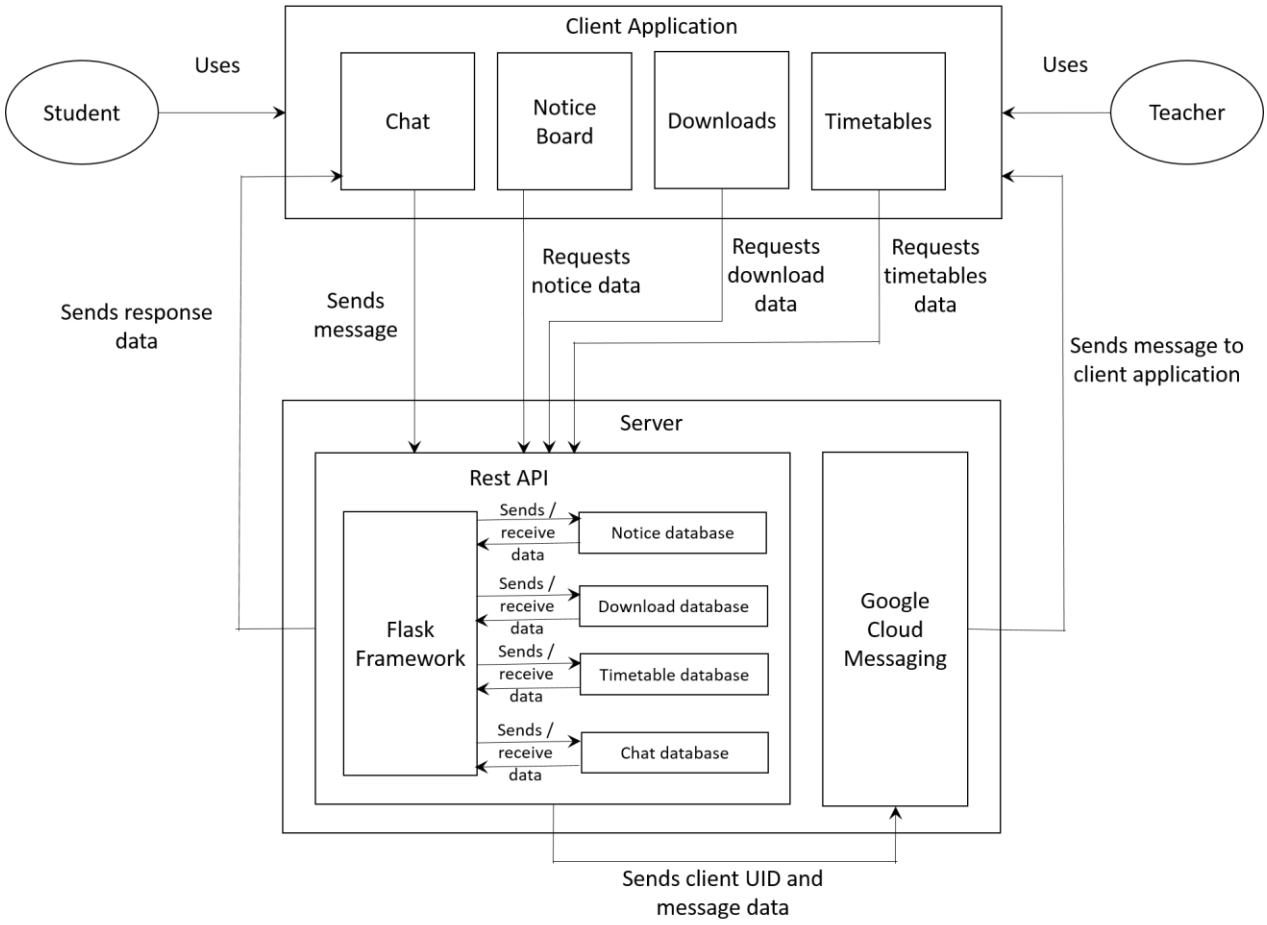


Fig. 3.3- Data Flow Diagram

Chapter Four

BODY OF THESIS

ANDROID

Android provides a rich application framework that allows you to build innovative apps and games for mobile devices in a Java language environment. The documents listed in the left navigation provide details about how to build apps using Android's various APIs.

Android Apps Provide Multiple Entry Points:

Android apps are built as a combination of distinct components that can be invoked individually. For instance, an individual *activity* provides a single screen for a user interface, and a *service* independently performs work in the background.

From one component you can start another component using an *intent*. You can even start a component in a different app, such as an activity in a maps app to show an address. This model provides multiple entry points for a single app and allows any app to behave as a user's "default" for an action that other apps may invoke.

Android Apps Adapt to different devices:

Android provides an adaptive app framework that allows you to provide unique resources for different device configurations. For example, you can create different XML layout files for different screen sizes and the system determines which layout to apply based on the current device's screen size.

You can query the availability of device features at runtime if any app features require specific hardware such as a camera. If necessary, you can also declare features your app requires so app markets such as Google Play Store do not allow installation on devices that do not support that feature.

CLOUD COMPUTING

When you store your photos online instead of on your home computer, or use webmail or a social networking site, you are using a "cloud computing" service. If you are an organization, and you want to use, for example, an online invoicing service instead of updating the in-house one you have been using for many years, that online invoicing service is a "cloud computing" service.

Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. Doing so may give rise to certain privacy implications. For that reason the Office of the Privacy Commissioner of Canada (OPC) has prepared some responses to Frequently Asked Questions (FAQs). We have also developed a Fact Sheet that provides detailed information on cloud computing and the privacy challenges it presents.

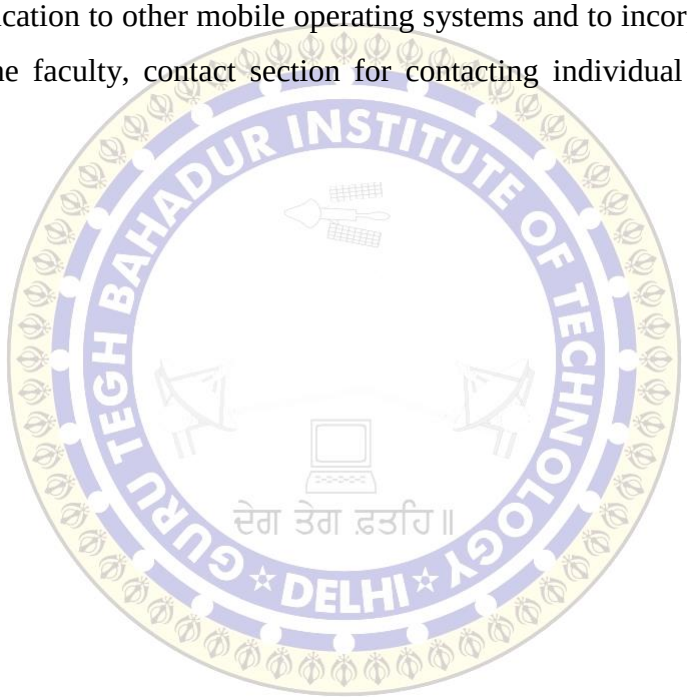
Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications

Chapter Five

FUTURE SCOPE

FUTURE SCOPE

This Application works on smartphones with android OS by using Google Cloud Messaging (GCM) for establishing connection and maintaining database through which users at different locations can connect with each other simultaneously. User can Login into the application using existing Facebook/Google Plus account or Register on the API server. We have a vision to expand the application to other mobile operating systems and to incorporate features like admin login for the faculty, contact section for contacting individual department of the college, etc.



Chapter Six

REFERENCES

REFERENCES

The following links were searched and exploited extensively for the project development and implementation.

1. Professional Android 2 Application Development (Wrox) by Reto Meier- 1st edition (26 February 2010).
2. Hello, Android: Introducing Google's Mobile Development Platform- Pragmatic Bookshelf; 4 edition (4 May 2015)
3. "Google Cloud Messaging for Android — Android Developers." [Online]. Available: http://developer.android.com/google/gcm/index.html
4. "Crowdreply at google play store," https://play.google.com/store/apps/details?id=edu.buffalo.cse.ubicomp.crowdonline
5. "Android Push Notifications using Google Cloud Messaging (GCM), PHP and MySQL"- Ravi Tamada (14 October 2012) http://www.androidhive.info/2012/10/android-push-notifications-using-google-cloud-messaging-gcm-php-and-mysql/
6. "Android Speech To Text Tutorial"- Ravi Tamada (13 July 2014) http://www.androidhive.info/2014/07/android-speech-to-text-tutorial/
7. "Android Chat with Google GCM XMPP"- Joe JavaPapers (17 June 2014) http://javapapers.com/android/android-chat-with-google-gcm-xmpp/
8. https://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract3
9. http://www.codeproject.com/Tips/840623/Android-Character-Recognition
10. https://en.wikipedia.org/wiki/Optical_character_recognition
11. https://github.com/rmtheis/android-ocr
12. http://gaut.am/making-an-ocr-android-app-using-tesseract/
13. https://github.com/rockerhieu/emojicon

Appendix A

SOURCE CODE

SOURCE CODE

// Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.bidibchat" >
   <uses-permission android:name="android.permission.INTERNET"/>
   <uses-permission android:name="android.permission.GET_ACCOUNTS"/>
   <uses-permission android:name="android.permission.USE_CREDENTIALS"/>
   <uses-permission android:name="android.permission.WAKE_LOCK"/>
   <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
   <uses-permission android:name="com.bidibchat.permission.C2D_MESSAGE" />
   <permission android:name="com.bidibchat.permission.C2D_MESSAGE"
      android:protectionLevel="signature" />

   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >

      <activity
         android:name=".SplashActivity"
         android:label="@string/app_name" >
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>

      <activity
         android:name=".GroupsListActivity"
         android:label="Groups" />
      <activity
         android:name=".ChatActivity"
         android:label="@string/title_activity_chat"
         android:launchMode="singleInstance"
         >
      </activity>
      <activity
         android:name=".LoginActivity"
         android:label="LoginActivity"
         />
      <activity android:name="com.facebook.FacebookActivity"
         android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation"
         android:theme="@android:style/Theme.Translucent.NoTitleBar"
```

Appendix

```
        android:label="@string/app_name" />

    <meta-data android:name="com.facebook.sdk.ApplicationId"
android:value="@string/facebook_app_id"/>

    <provider android:authorities="com.facebook.app.FacebookContentProvider871647029617364"
              name="com.facebook.FacebookContentProvider"
              exported="true"/>

    <receiver
      android:name="com.google.android.gms.gcm.GcmReceiver"
      android:exported="true"
      android:permission="com.google.android.c2dm.permission.SEND" >
      <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
        <category android:name="com.bidibchat" />
      </intent-filter>
    </receiver>

    <service
      android:name="com.bidibchat.Services.MyGcmListenerService"
      android:exported="false" >
      <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
      </intent-filter>
    </service>

    <service
      android:name="com.bidibchat.Services.GCMInstanceIDListenerService"
      android:exported="false">
      <intent-filter>
        <action android:name="com.google.android.gms.iid.InstanceID"/>
      </intent-filter>
    </service>

    <service android:name="com.bidibchat.Services.InstanceIDService" android:exported="false">
      <intent-filter>
        <action android:name="com.google.android.gms.iid.InstanceID"/>
      </intent-filter>
    </service>

  </application>

</manifest>



// Activities

// Splash Activity
```

```
package com.bidibchat;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.CountDownTimer;
```

Appendix

```
enceManager;
ctionBarActivity;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;


public class SplashActivity extends AppCompatActivity {

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

      Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
      setSupportActionBar(toolbar);

      new CountDownTimer(3000, 1000) {

         public void onTick(long millisUntilFinished) {

         }

         public void onFinish() {

            SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
            Boolean isLoggedIn = prefs.getBoolean("isLoggedIn", false);
            if(isLoggedIn){

               Intent launchLoginActivity = new Intent(getApplicationContext(),GroupsListActivity.class);
               startActivity(launchLoginActivity);
               //destroy splash screen
               finish();


            }else {

               Intent launchLoginActivity = new Intent(getApplicationContext(), LoginActivity.class);
               startActivity(launchLoginActivity);
               //destroy splash screen
               finish();
            }
         }
```

```
        }.start();


    }
```

Appendix

```
// Login Activity

package com.bidibchat;

    import android.content.Context;
    import android.content.Intent;
    import android.content.IntentSender;
    import android.content.SharedPreferences;
    import android.graphics.Bitmap;
    import android.graphics.BitmapFactory;
    import android.os.AsyncTask;
    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.EditText;
    import android.widget.ImageView;
    import android.widget.Toast;

    import com.android.volley.Request;
    import com.android.volley.RequestQueue;
    import com.android.volley.Response;
    import com.android.volley.VolleyError;
    import com.android.volley.VolleyLog;
    import com.android.volley.toolbox.JsonObjectRequest;
    import com.android.volley.toolbox.Volley;
    import com.bidibchat.Services.InstanceIDService;
    import com.bidibchat.models.UserModel;
    import com.facebook.AccessToken;
    import com.facebook.CallbackManager;
    import com.facebook.FacebookCallback;
    import com.facebook.FacebookException;
    import com.facebook.FacebookSdk;
    import com.facebook.GraphRequest;
```

```
import com.facebook.GraphResponse;
import com.facebook.login.LoginManager;
import com.facebook.login.LoginResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
```

Appendix

```
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks;
import com.google.android.gms.common.api.GoogleApiClient.OnConnectionFailedListener;
import com.google.android.gms.plus.Plus;
import com.google.android.gms.plus.model.people.Person;
import com.google.gson.Gson;

import org.json.JSONObject;

import java.io.InputStream;

public class LoginActivity extends AppCompatActivity implements View.OnClickListener,
    ConnectionCallbacks, OnConnectionFailedListener {


  private static final int RC_SIGN_IN = 0;
  // Logcat tag
  private static final String TAG = "MainActivity";

  //= Volley.newRequestQueue(this);
  // Profile pic image size in pixels
  private static final int PROFILE_PIC_SIZE = 400;

  // Google client to interact with Google API
  private GoogleApiClient mGoogleApiClient;

  /**
   * A flag indicating that a PendingIntent is in progress and prevents us
   * from starting further intents.
   */
  private boolean mIntentInProgress;

  private boolean mSignInClicked;

  private ConnectionResult mConnectionResult;

  private SignInButton btnSignIn;

  CallbackManager callbackManager;
  Button login;

  EditText emailField,passwordfield,nameField;
  String group="Food";

  @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    FacebookSdk.sdkInitialize(getApplicationContext());
    setContentView(R.layout.activity_login);

    callbackManager = CallbackManager.Factory.create();
Appendix                    dViewById(R.id.facebook_login_button);

    emailField = (EditText) findViewById(R.id.email_field);
    passwordfield = (EditText) findViewById(R.id.password_field);
    nameField = (EditText) findViewById(R.id.name_field);


    btnSignIn = (SignInButton) findViewById(R.id.btn_sign_in);

    // Button click listeners
    btnSignIn.setOnClickListener(this);

    if(AccessToken.getCurrentAccessToken() != null){
        RequestData();
        onLoginRegisterClick(null);

    }

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

            LoginManager.getInstance().registerCallback(callbackManager, new
FacebookCallback<LoginResult>() {
                @Override
                public void onSuccess(LoginResult loginResult) {

                    if (AccessToken.getCurrentAccessToken() != null) {
                        RequestData();
                    }
                }

                @Override
                public void onCancel() {

                }

                @Override
                public void onError(FacebookException exception) {
                }
            });


    }
});
```

```
        mGoogleApiClient = new GoogleApiClient.Builder(this)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this).addApi(Plus.API)
                .addScope(Plus.SCOPE_PLUS_LOGIN).build();
    }
```

Appendix

```
    public void RequestData(){
        GraphRequest request = GraphRequest.newMeRequest(AccessToken.getCurrentAccessToken(), new
    GraphRequest.GraphJSONObjectCallback() {
            @Override
            public void onCompleted(JSONObject object,GraphResponse response) {

                JSONObject json = response.getJSONObject();
    //              try {
                    if(json != null){
                        //TODO - add data to shared prefrences
                    }

    //              } catch (JSONException e) {
    //                  e.printStackTrace();
    //              }
                onLoginRegisterClick(null);
            }

        });
        Bundle parameters = new Bundle();
        parameters.putString("fields", "id,name,link,email,picture");
        request.setParameters(parameters);
        request.executeAsync();
    }

    @Override
    protected void onResume() {
        super.onResume();
        mGoogleApiClient.connect();
    }

    @Override
    protected void onPause() {
        super.onPause();
        if (mGoogleApiClient.isConnected()) {
            mGoogleApiClient.disconnect();
        }
    }

    /**
     * Button on click listener
     * */
    @Override
    public void onClick(View v) {
```

```
    switch (v.getId()) {
        case R.id.btn_sign_in:
            // Signin button clicked
            signInWithGplus();
            break;
    }
```

Appendix

```
@Override
public void onConnectionFailed(ConnectionResult result) {
    if (!result.hasResolution()) {
        GooglePlayServicesUtil.getErrorDialog(result.getErrorCode(), this,
                0).show();
        return;
    }

    if (!mIntentInProgress) {
        // Store the ConnectionResult for later usage
        mConnectionResult = result;

        if (mSignInClicked) {
            // The user has already clicked 'sign-in' so we attempt to
            // resolve all
            // errors until the user is signed in, or they cancel.
            resolveSignInError();
        }
    }

}

@Override
protected void onActivityResult(int requestCode, int responseCode,
                        Intent intent) {

    if (requestCode == RC_SIGN_IN) {
        if (responseCode != RESULT_OK) {
            mSignInClicked = false;
        }

        mIntentInProgress = false;

        if (!mGoogleApiClient.isConnecting()) {
            mGoogleApiClient.connect();
        }
    }
    callbackManager.onActivityResult(requestCode, responseCode, intent);
}

@Override
public void onConnected(Bundle arg0) {
    mSignInClicked = false;
```

```
    //Toast.makeText(this, "User is connected!", Toast.LENGTH_LONG).show();
    // Get user's information
    getProfileInformation();
    onLoginRegisterClick(null);
```

Appendix

```
@Override
public void onConnectionSuspended(int arg0) {
    mGoogleApiClient.connect();
}

/**
 * Updating the UI, showing/hiding buttons and profile layout
 * */

/**
 * Sign-in into google
 * */
private void signInWithGplus() {
    if (!mGoogleApiClient.isConnecting()) {
        mSignInClicked = true;
        resolveSignInError();
    }
}

/**
 * Method to resolve any signin errors
 * */
private void resolveSignInError() {
    if (mConnectionResult.hasResolution()) {
        try {
            mIntentInProgress = true;
            mConnectionResult.startResolutionForResult(this, RC_SIGN_IN);
        } catch (IntentSender.SendIntentException e) {
            mIntentInProgress = false;
            mGoogleApiClient.connect();
        }
    }
}

/**
 * Fetching user's information name, email, profile pic
 * */
private void getProfileInformation() {
    try {
        if (Plus.PeopleApi.getCurrentPerson(mGoogleApiClient) != null) {
            Person currentPerson = Plus.PeopleApi
                    .getCurrentPerson(mGoogleApiClient);
            String personName = currentPerson.getDisplayName();
            String personPhotoUrl = currentPerson.getImage().getUrl();
```

```
                String personGooglePlusProfile = currentPerson.getUrl();
                String email = Plus.AccountApi.getAccountName(mGoogleApiClient);

                Log.e(TAG, "Name: " + personName + ", plusProfile: "
                        + personGooglePlusProfile + ", email: " + email
                        + ", Image: " + personPhotoUrl);
```

Appendix

```
                //TODO send info to shared preferences

                // by default the profile url gives 50x50 px image only
                // we can replace the value with whatever dimension we want by
                // replacing sz=X

                //we can use this profile image
//                personPhotoUrl = personPhotoUrl.substring(0,
//                        personPhotoUrl.length() - 2)
//                        + PROFILE_PIC_SIZE;


            } else {
                Toast.makeText(getApplicationContext(),
                        "Person information is null", Toast.LENGTH_LONG).show();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * Background Async task to load user profile picture from url
     * */
    private class LoadProfileImage extends AsyncTask<String, Void, Bitmap> {
        ImageView bmImage;

        public LoadProfileImage(ImageView bmImage) {
            this.bmImage = bmImage;
        }

        protected Bitmap doInBackground(String... urls) {
            String urldisplay = urls[0];
            Bitmap mIcon11 = null;
            try {
                InputStream in = new java.net.URL(urldisplay).openStream();
                mIcon11 = BitmapFactory.decodeStream(in);
            } catch (Exception e) {
                Log.e("Error", e.getMessage());
                e.printStackTrace();
            }
            return mIcon11;
        }
```

```
        protected void onPostExecute(Bitmap result) {
            bmImage.setImageBitmap(result);
        }
    }

    /**
Appendix
     * */
    private void signOutFromGplus() {
        if (mGoogleApiClient.isConnected()) {
            Plus.AccountApi.clearDefaultAccount(mGoogleApiClient);
            mGoogleApiClient.disconnect();
            mGoogleApiClient.connect();
        }
    }

    public void onLoginRegisterClick(View view){

        Intent intent = new Intent(this, InstanceIDService.class);
        intent.putExtra("email",emailField.getText().toString());
        intent.putExtra("password",passwordfield.getText().toString());
        intent.putExtra("name",nameField.getText().toString());
        intent.putExtra("groupname",group);
        startService(intent);

    }

}



// Chat Activity

package com.bidibchat;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.widget.EditText;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
```

```
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import com.bidibchat.Services.MyGcmListenerService;
import com.bidibchat.adapters.ChatListAdapter;
import com.bidibchat.models.MessageModel;
```

Appendix

```
import com.bidibchat.models.UserModel;
import com.google.gson.Gson;

import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;


public class ChatActivity extends AppCompatActivity {

    static RecyclerView chatList;
    MessageModel model;
    static List<MessageModel> modelList;
    static ChatListAdapter adapter;

    EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_chat);

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        chatList = (RecyclerView) findViewById(R.id.chat_recycler_view);
        editText = (EditText) findViewById(R.id.chatTextField);
        editText.setSelected(false);

        modelList = new ArrayList<MessageModel>();
        //TODO : add data fetched from server


        adapter = new ChatListAdapter(this,modelList);

        chatList.setLayoutManager(new LinearLayoutManager(this));
        chatList.setAdapter(adapter);

        MyGcmListenerService.OnNewMessageListener onNewMessageListener = new
MyGcmListenerService.OnNewMessageListener(){

            @Override
            public void onNewMessageListenerReceived(MessageModel model) {
```

```
            //do something

            Log.d("Message Recieved from",model.getSenderName());
            refreshMessageListOnNotification(model.getSenderName(), model.getContent());
            //clear
//            MyGcmListenerService.clearOnNewLocationListener(this);
```

Appendix

```
        };
        MyGcmListenerService.setOnNewLocationListener(onNewMessageListener);

    }

    public void performSend(View v){

        model = new MessageModel();
        //TODO : Get sender name from shared preferences

        SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        String email = prefs.getString("email",null);
        String name = prefs.getString("name",null);

        model.setSenderName(name);
        model.setSendersEmail(email);
        model.setContent(editText.getText().toString());
        model.setSenderType(MessageModel.SENDER);
        modelList.add(model);

        adapter.notifyDataSetChanged();
        chatList.smoothScrollToPosition(modelList.size() - 1);
        sendMessageToServer(model);
        editText.setText("");


    }

    public void refreshMessageListOnNotification(final String header,final String content){

        this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                MessageModel model = new MessageModel();
                //TODO : Get sender name from shared preferences
                model.setSenderName(header);
                model.setContent(content);
                if (modelList == null) {

                    modelList = new ArrayList<MessageModel>();

                }
                modelList.add(model);
```

```
                adapter.notifyDataSetChanged();
            }
        });
        chatList.smoothScrollToPosition(modelList.size() - 1);

    }
```

Appendix

```
    private void sendMessageToServer(MessageModel model) {
        RequestQueue queue = Volley.newRequestQueue(getApplicationContext());

        final String URL = "https://bidibchat.herokuapp.com/sendall";
        // Post params to be sent to the server
        Gson gson = new Gson();
        String json = gson.toJson(model);

        JsonObjectRequest req = new JsonObjectRequest(Request.Method.POST,URL,json ,
                new Response.Listener<JSONObject>() {
                    @Override
                    public void onResponse(JSONObject response) {

                    }
                }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                VolleyLog.e("Error: ", error.getMessage());
            }
        });

        queue.add(req);
    }

}


// Group List Activity

package com.bidibchat;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.widget.Toast;
```

```
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.JsonObjectRequest;
```



```
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.bidibchat.adapters.GroupsListAdapter;
import com.bidibchat.models.GroupsListModel;
import com.bidibchat.models.UserModel;
import com.google.gson.Gson;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Iterator;


public class GroupsListActivity extends AppCompatActivity {

    Toolbar toolbar;
    RecyclerView groupListView;
    ArrayList<GroupsListModel> dataList;
    GroupsListAdapter adapter;


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

        toolbar = (Toolbar) findViewById(R.id.groups_toolbar);
        setSupportActionBar(toolbar);

        groupListView = (RecyclerView) findViewById(R.id.groups_list_recycler_view);

        dataList = new ArrayList<>();

        getListFromServer();
        adapter = new GroupsListAdapter(this,dataList);
        groupListView.setLayoutManager(new LinearLayoutManager(this));
        groupListView.setAdapter(adapter);

    }

    public void onAddGroupButtonClicked(View v){

        GroupsListModel temp = new GroupsListModel();
        temp.setGroupImageName("");
```

```
        temp.setNumberOfGroupMembers("1");
        temp.setGroupName("New Group Name");
        dataList.add(temp);
        adapter.notifyDataSetChanged();
        groupListView.smoothScrollToPosition(dataList.size() - 1);
```

Appendix

```
    public void getListFromServer() {

        RequestQueue queue = Volley.newRequestQueue(getApplicationContext());

        final String URL = "https://bidibchat.herokuapp.com/grouplist";

        StringRequest req = new StringRequest(Request.Method.GET, URL,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        try {
                            JSONObject obj = new JSONObject(response);
                            Iterator<String> iter = obj.keys();
                            while (iter.hasNext()) {

                                String key = iter.next();
                                GroupsListModel groups = new GroupsListModel();
                                groups.setGroupName(key);
                                int numOfMembers = (int) obj.get(key);
                                groups.setNumberOfGroupMembers(Integer.toString(numOfMembers));
                                dataList.add(groups);

                            }
                            adapter.notifyDataSetChanged();
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }

                    }
                }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {

                }
        });
        queue.add(req);

    }
}


// Services
```

// GCM Listener Service

```
package com.bidibchat.Services;

import android.app.NotificationManager;
```



```
import android.content.Context;
import android.content.Intent;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.NotificationCompat;
import android.util.Log;

import com.bidibchat.ChatActivity;
import com.bidibchat.LoginActivity;
import com.bidibchat.R;
import com.bidibchat.models.MessageModel;
import com.google.android.gms.gcm.GcmListenerService;

import java.util.ArrayList;

/**
 * Created by Inderdeep on 09-11-2015.
 */


public class MyGcmListenerService extends GcmListenerService {

    public static ArrayList<OnNewMessageListener> arrOnNewMessageListener =
            new ArrayList<OnNewMessageListener>();

    private static final String TAG = "MyGcmListenerService";

    public interface OnNewMessageListener {
        public abstract void onNewMessageListenerReceived(MessageModel model);
    }


    // Allows the user to set an Listener and react to the event
    public static void setOnNewLocationListener(
            OnNewMessageListener listener) {
        arrOnNewMessageListener.add(listener);
    }

    public static void clearOnNewLocationListener(
            OnNewMessageListener listener) {
        arrOnNewMessageListener.remove(listener);
    }
```

```
    // This function is called after the new point received
    private static void OnNewMessageRecieved(MessageModel model) {
        // Check if the Listener was set, otherwise we'll get an Exception when
        // we try to call it
        if (arrOnNewMessageListener != null) {
Appendix                                    nt, when we have any listener
            for (int i = arrOnNewMessageListener.size() - 1; i >= 0; i--) {
                arrOnNewMessageListener.get(i).onNewMessageListenerReceived(
                        model);
            }
        }
    }

    @Override
    public void onMessageReceived(String from, Bundle data) {
        String header = data.getString("header");
        String message = data.getString("content");
        Log.d(TAG, "Header: " + header);
        Log.d(TAG, "From: " + from);
        Log.d(TAG, "Message: " + message);

        MessageModel model = new MessageModel();
        model.setSenderType(MessageModel.RECIEVER);
        model.setContent(message);
        model.setSenderName(header);
        OnNewMessageRecieved(model);
        sendNotification(header,message);

    }

    /**
     * Create and show a simple notification containing the received GCM message.
     *
     * @param message GCM message received.
     */
    private void sendNotification(String header,String message) {
        Intent intent = new Intent(this, LoginActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0 /* Request code */, intent,
                PendingIntent.FLAG_ONE_SHOT);

        Uri defaultSoundUri= RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        NotificationCompat.Builder notificationBuilder = new
  NotificationCompat.Builder(getApplicationContext())
                .setSmallIcon(R.mipmap.ic_launcher)
                .setContentTitle(header)
                .setContentText(message)
                .setSound(defaultSoundUri)
                .setContentIntent(pendingIntent);
```

```
        NotificationManager notificationManager =
                (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        notificationManager.notify(0, notificationBuilder.build());
    }
}
```

Appendix

```
// Instance ID Service

package com.bidibchat.Services;

import android.app.IntentService;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import android.util.Log;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import com.bidibchat.GroupsListActivity;
import com.bidibchat.LoginActivity;
import com.bidibchat.R;
import com.bidibchat.models.UserModel;
import com.google.android.gms.gcm.GcmPubSub;
import com.google.android.gms.gcm.GoogleCloudMessaging;
import com.google.android.gms.iid.InstanceID;
import com.google.gson.Gson;

import org.json.JSONObject;

import java.io.IOException;

/**
 * Created by Inderdeep on 09-11-2015.
 */
public class InstanceIDService extends IntentService {

    private static final String TAG = "InstanceIDService";
    private static final String[] TOPICS = {"global"};

    private static String email,password,name,groupname;

    public InstanceIDService() {
```

```
    super(TAG);
}

@Override
protected void onHandleIntent(Intent intent) {
    SharedPreferences sharedPreferences = PreferenceManager.getDefaultSharedPreferences(this);
```

Appendix

```
    email = intent.getExtras().getString("email");
    password = intent.getExtras().getString("password");
    name = intent.getExtras().getString("name");
    groupname = intent.getExtras().getString("groupname");

    try {

        InstanceID instanceID = InstanceID.getInstance(this);
        String token = instanceID.getToken(getString(R.string.gcm_defaultSenderId),
                GoogleCloudMessaging.INSTANCE_ID_SCOPE, null);

        Log.i(TAG, "GCM Registration Token: " + token);

        // TODO: Implement this method to send any registration to your app's servers.
        sendRegistrationToServer(token);

        // Subscribe to topic channels
        subscribeTopics(token);

        // You should store a boolean that indicates whether the generated token has been
        // sent to your server. If the boolean is false, send the token to your server,
        // otherwise your server should have already received the token.
        sharedPreferences.edit().putBoolean("Token Sent", true).apply();

    } catch (Exception e) {
        Log.d(TAG, "Failed to complete token refresh", e);
        // If an exception happens while fetching the new token or updating our registration data
        // on a third-party server, this ensures that we'll attempt the update at a later time.
        sharedPreferences.edit().putBoolean("Token Sent", false).apply();
    }
}

/**
 * Persist registration to third-party servers.
 *
 * Modify this method to associate the user's GCM registration token with any server-side account
 * maintained by your application.
 *
 * @param token The new token.
 */
private void sendRegistrationToServer(String token) {
    // Add custom implementation, as needed.

    register(token);
```

```
    }


    public void register(String RegId) {

Appendix                    olley.newRequestQueue(getApplicationContext());

        final String URL = "https://bidibchat.herokuapp.com/register";
// Post params to be sent to the server
        Gson gson = new Gson();
        UserModel model = new UserModel();
        model.setGcm_regid(RegId);
        model.setUser_email(email);
        model.setUser_group(groupname);
        model.setUser_name(name);
        model.setUser_password(password);
        String json = gson.toJson(model);

        JsonObjectRequest req = new JsonObjectRequest(Request.Method.POST,URL,json ,
                new Response.Listener<JSONObject>() {
                    @Override
                    public void onResponse(JSONObject response) {

                        Log.d("Response",response.toString());

                        SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
                        SharedPreferences.Editor prefEditor = prefs.edit();
                        prefEditor.putBoolean("isLoggedIn",true);
                        prefEditor.putString("email",email);
                        prefEditor.putString("name",name);
                        prefEditor.apply();

                        Intent openGroupsActivity = new Intent(getApplicationContext(), GroupsListActivity.class);
                        openGroupsActivity.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                        startActivity(openGroupsActivity);

                    }
                }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                VolleyLog.e("Error: ", error.getMessage());
                Toast.makeText(getApplicationContext(),"Some Error
Occured!",Toast.LENGTH_SHORT).show();
            }
        });

        queue.add(req);

    }
```

```
    /**
     * Subscribe to any GCM topics of interest, as defined by the TOPICS constant.
     *
     * @param token GCM token
     * @throws IOException if unable to reach the GCM PubSub service
     */
    private void subscribeTopics(String token) throws IOException {
        GcmPubSub pubSub = GcmPubSub.getInstance(this);
        for (String topic : TOPICS) {
            pubSub.subscribe(token, "/topics/" + topic, null);
        }
    }

}


// GCM Instance ID Listener Service

package com.bidibchat.Services;

import android.content.Intent;

import com.google.android.gms.iid.InstanceIDListenerService;

/**
 * Created by Inderdeep on 09-11-2015.
 */
public class GCMInstanceIDListenerService extends InstanceIDListenerService {

    private static final String TAG = "MyInstanceIDLS";

    /**
     * Called if InstanceID token is updated. This may occur if the security of
     * the previous token had been compromised. This call is initiated by the
     * InstanceID provider.
     */
    // [START refresh_token]
    @Override
    public void onTokenRefresh() {
        // Fetch updated Instance ID token and notify our app's server of any changes (if applicable).
        Intent intent = new Intent(this, InstanceIDService.class);
        startService(intent);
    }
    // [END refresh_token]
}
```

```
// Models

// User Model
```

Appendix ;

```
import android.content.Intent;

import com.google.android.gms.iid.InstanceIDListenerService;

/**
 * Created by Inderdeep on 09-11-2015.
 */
public class GCMInstanceIDListenerService extends InstanceIDListenerService {

    private static final String TAG = "MyInstanceIDLS";

    /**
     * Called if InstanceID token is updated. This may occur if the security of
     * the previous token had been compromised. This call is initiated by the
     * InstanceID provider.
     */
    // [START refresh_token]
    @Override
    public void onTokenRefresh() {
        // Fetch updated Instance ID token and notify our app's server of any changes (if applicable).
        Intent intent = new Intent(this, InstanceIDService.class);
        startService(intent);
    }
    // [END refresh_token]
}


// Message Model

package com.bidibchat.models;

import java.text.SimpleDateFormat;

/**
 * Created by tekvy on 17/9/15.
 */
public class MessageModel {


    public final static int RECIEVER = 0;
    public final static int SENDER = 1;

    private String content;
```

```
private SimpleDateFormat timeStamp;
private String messageImageUrl;
private String header;

public int getSenderType() {
    return senderType;
```

Appendix

```
public void setSenderType(int senderType) {
    this.senderType = senderType;
}

private int senderType;
private String sendersEmail;

public String getSendersEmail() {
    return sendersEmail;
}

public void setSendersEmail(String sendersEmail) {
    this.sendersEmail = sendersEmail;
}



public String getSenderName() {
    return header;
}

public void setSenderName(String senderName) {
    this.header = senderName;
}



public String getMessageImageUrl() {
    return messageImageUrl;
}

public void setMessageImageUrl(String messageImageUrl) {
    this.messageImageUrl = messageImageUrl;
}

public SimpleDateFormat getTimeStamp() {
    return timeStamp;
}

public void setTimeStamp(SimpleDateFormat timeStamp) {
    this.timeStamp = timeStamp;
}
```

```
    public String getContent() {
        return content;
    }

    public void setContent(String text) {
        this.content = text;
```

Appendix

```
}


// Group List Model

package com.bidibchat.models;

/**
 * Created by Inderdeep on 17-09-2015.
 */
public class GroupsListModel {

    private String groupName;
    private String numberOfGroupMembers;
    private String groupImageName;

    public String getGroupName() {
        return groupName;
    }

    public void setGroupName(String _groupName) {
        this.groupName = _groupName;
    }

    public String geNumberOfGroupMembers() {
        return numberOfGroupMembers;
    }

    public void setNumberOfGroupMembers(String _numberOfGroupMembers) {
        this.numberOfGroupMembers = _numberOfGroupMembers;
    }

    public String getGroupImageName() {
        return groupImageName;
    }

    public void setGroupImageName(String _groupImageName) {
        this.groupImageName = _groupImageName;
    }

}
```

```
// Adapters

// Group List Adapter

package com.bidibchat.adapters;
```

Appendix

```
import android.content.Context;
import android.content.Intent;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;


import com.bidibchat.ChatActivity;
import com.bidibchat.R;
import com.bidibchat.models.GroupsListModel;

import java.util.ArrayList;

/**
 * Created by Inderdeep on 17-09-2015.
 */
public class GroupsListAdapter extends RecyclerView.Adapter<GroupsListAdapter.ViewHolder>{

    Context context;
    ArrayList<GroupsListModel> modelList;

    public GroupsListAdapter(Context context, ArrayList<GroupsListModel> modelList){

        this.context = context;
        this.modelList = modelList;

    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemLayoutView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.group_list_item, null);
        final ViewHolder viewHolder = new ViewHolder(itemLayoutView);

        viewHolder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent openChatActivity = new Intent(context, ChatActivity.class);
                context.startActivity(openChatActivity);
            }
        });
```

```
        return viewHolder;
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
```

Appendix

```
        holder.groupImage.setImageResource(R.drawable.profile_mask);
        holder.groupNameLabel.setText(modelList.get(position).getGroupName());
        holder.groupMembersLabel.setText(modelList.get(position).geNumberOfGroupMembers());

    }

    @Override
    public int getItemCount() {
        return modelList.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder{


        public ImageView groupImage;
        public TextView groupNameLabel;
        public TextView groupMembersLabel;

        private int pos = -1;

        public int getPos() {
            return pos;
        }

        public void setPos(int pos) {
            this.pos = pos;
        }

        public ViewHolder(View itemView) {
            super(itemView);

            groupImage = (ImageView) itemView.findViewById(R.id.group_image);
            groupNameLabel = (TextView) itemView.findViewById(R.id.group_name_label);
            groupMembersLabel = (TextView) itemView.findViewById(R.id.number_of_group_members_label);

        }
    }
}
```

// Chat List Adapter

```
package com.bidibchat.adapters;
```

```
import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
```

Appendix

```
import com.bidibchat.R;
import com.bidibchat.models.GroupsListModel;
import com.bidibchat.models.MessageModel;

import java.util.ArrayList;
import java.util.List;

public class ChatListAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {

    Context context;
    List<MessageModel> modelList;


    public ChatListAdapter(Context context, List<MessageModel> modelList) {
        this.context = context;
        this.modelList = modelList;
    }


    @Override
    public int getItemCount() {
        return modelList.size();
    }

    @Override
    public int getItemViewType(int position) {
        // Just as an example, return 0 or 1 depending on position
        return modelList.get(position).getSenderType();
    }

    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
//        View itemLayoutView = LayoutInflater.from(parent.getContext())
//                .inflate(R.layout.receive_message, null);
//        final ViewHolderSend viewHolder = new ViewHolderSend(itemLayoutView);
//
//        viewHolder.itemView.setOnClickListener(new View.OnClickListener() {
//            @Override
//            public void onClick(View v) {
//                Toast.makeText(context, "Message Name : " + modelList.get(viewHolder.getPos()).getText(),
Toast.LENGTH_SHORT).show();
//            }
```

```
//      });
//
//      return viewHolder;

      switch (viewType) {
        case 1:
Appendix                    = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.send_message,parent, false);

            final ViewHolderSend viewHolder = new ViewHolderSend(itemLayoutView);
            viewHolder.itemView.setOnClickListener(new View.OnClickListener() {
              @Override
              public void onClick(View v) {

                Toast.makeText(context, Integer.toString(viewHolder.getPos()),
Toast.LENGTH_SHORT).show();

              }
            });
            return viewHolder;

          case 0:

            View headerView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.receive_message,parent, false);

            final ViewHolderReceive headerViewHolder = new ViewHolderReceive(headerView);
            headerViewHolder.itemView.setOnClickListener(new View.OnClickListener() {
              @Override
              public void onClick(View v) {
                Toast.makeText(context, Integer.toString(headerViewHolder.getPos()),
Toast.LENGTH_SHORT).show();
              }
            });
            return headerViewHolder;

          default:
            return null;

      }


    }

    @Override
    public void onBindViewHolder(RecyclerView.ViewHolder holder, int position) {

      if(holder.getClass().equals(ViewHolderReceive.class)){

        ((ViewHolderReceive)holder).setPos(position);
        ((ViewHolderReceive)holder).messageImage.setImageResource(R.drawable.profile_mask);
```

```
        ((ViewHolderReceive)holder).senderNameLabel.setText(modelList.get(position).getSenderName());
        ((ViewHolderReceive)holder).textLabel.setText(modelList.get(position).getContent());

    }
    if(holder.getClass().equals(ViewHolderSend.class)){
```

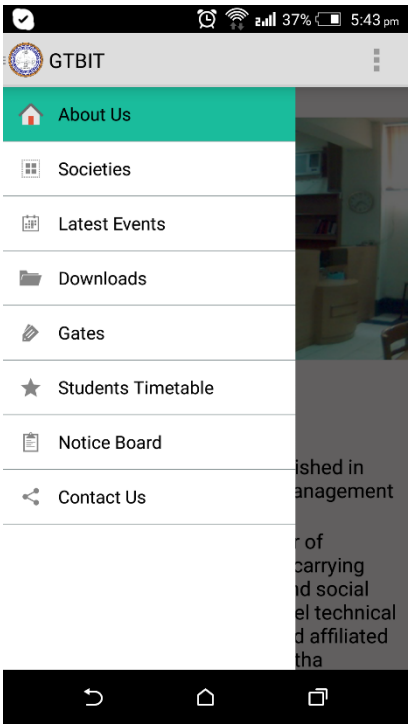
Appendix

```
        der).setPos(position);
        ((ViewHolderSend)holder).textLabel.setText(modelList.get(position).getContent());
    }

}

public class ViewHolderSend extends RecyclerView.ViewHolder{

    public TextView textLabel;

    private int pos = -1;

    public int getPos() {
        return pos;
    }

    public void setPos(int pos) {
        this.pos = pos;
    }

    public ViewHolderSend(View itemView) {
        super(itemView);

        textLabel = (TextView) itemView.findViewById(R.id.send_message_text);
    }
}

public class ViewHolderReceive extends RecyclerView.ViewHolder{

    public ImageView messageImage;
    public TextView senderNameLabel;
    public TextView textLabel;

    private int pos = -1;

    public int getPos() {
        return pos;
    }

    public void setPos(int pos) {
        this.pos = pos;
    }

    public ViewHolderReceive(View itemView) {
        super(itemView);
```

```
            messageImage = (ImageView) itemView.findViewById(R.id.sender_image);
            senderNameLabel = (TextView) itemView.findViewById(R.id.sender_name);
            textLabel = (TextView) itemView.findViewById(R.id.receive_message_text);

        }
    }
}
```

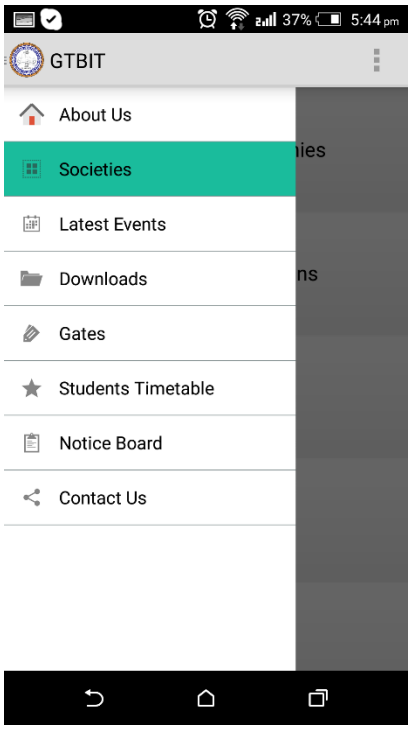
Appendix B

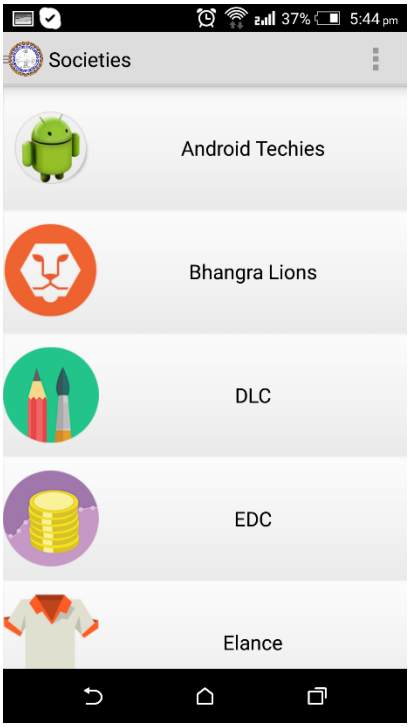
SCREENSHOTS


37% 5:43 pm
GTBIT
About Us
Societies
Latest Events
Downloads
Gates
Students Timetable
Notice Board
Contact Us

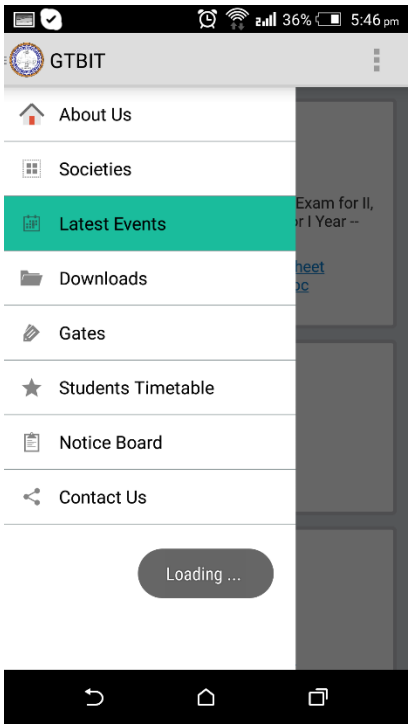

37% 5:44 pm
About Us
Introduction
Guru Tegh Bahadur Institute of Technology (GTBIT) was established in 1999 by Delhi Sikh Gurdwara Management Committee
which is running a large number of educational institutes, besides carrying out large number of religious and social activities. GTBIT is a degree level technical institute, approved by AICTE and affiliated to Guru Gobind Singh Indraprastha

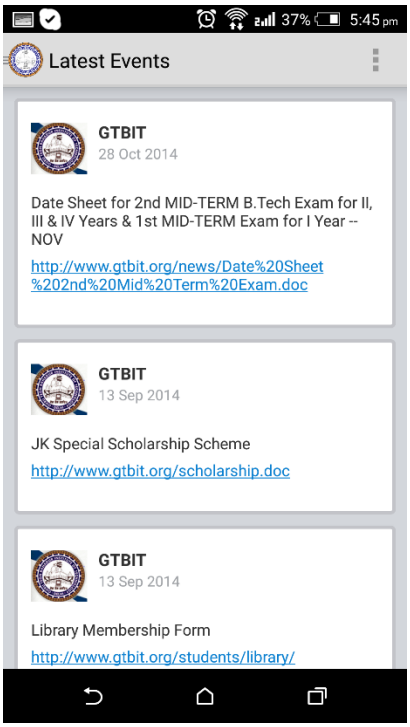

37% 5:44 pm
GTBIT
Vision Statement:
(GTBIT) aims to be among the premier institutes of engineering and technology in the country, recognised for excellence in teaching
To help our students in developing their


GTBIT
About Us
Societies
Latest Events
Downloads
Gates
Students Timetable
Notice Board
Contact Us


Societies
Android Techies
Bhangra Lions
DLC
EDC
Elance

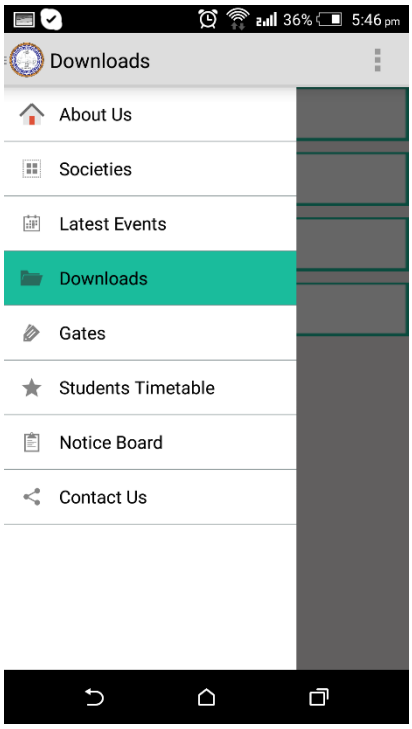

GTBIT
Android Techies
The need of knowing 'Android', being engineering students, was felt and recognised by the core members of the society, sprouting the idea of building a society to support this thought. The idea was welcomed by the authorities of the college and was marked as an influential field. Under the constant support of the Director General of the college, Sardar Gurmeet Singh Soin, coupled with the adequate guidance of Mr. Gaurav Sandhu, the current teacher coordinator of Android Techies, has led to the rise of the society in enormous proportions. The success of getting a cent per cent enrolment of students of GTBIT with Android has been immense. Android Techies

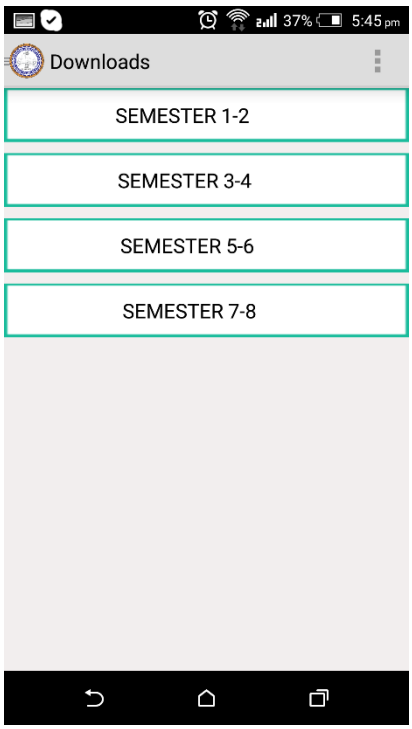

36% 5:46 pm
GTBIT
About Us
Societies
Latest Events
Downloads
Gates
Students Timetable
Notice Board
Contact Us
Loading ...

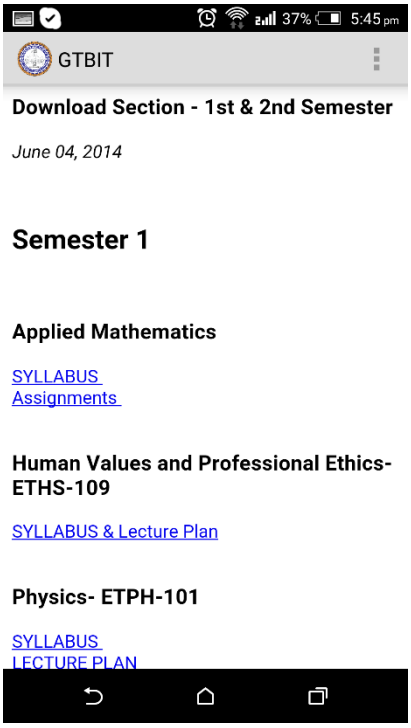

37% 5:45 pm
Latest Events
GTBIT
28 Oct 2014
Date Sheet for 2nd MID-TERM B.Tech Exam for II, III & IV Years & 1st MID-TERM Exam for I Year -- NOV
http://www.gtbit.org/news/Date%20Sheet%202nd%20Mid%20Term%20Exam.doc
GTBIT
13 Sep 2014
JK Special Scholarship Scheme
http://www.gtbit.org/scholarship.doc
GTBIT
13 Sep 2014
Library Membership Form
http://www.gtbit.org/students/library/

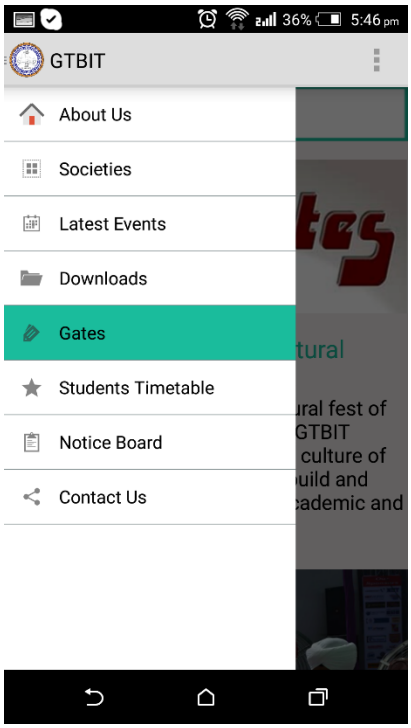

37% 5:45 pm
Gates
GTBIT's Annual Techno-cultural Event for creative Souls
GATES is the annual techno-cultural fest of GTBIT. Its an opportunity for the GTBIT community to showcase the rich culture of GTBITians; a venture to initiate, build and maintain acquaintances in the academic and industrial arena.
GATES is organized in the month of March and has been achieved new landmarks every


37% 5:45 pm
Gates
EVENTS
gates
GTBIT's Annual Techno-cultural Event for creative Souls
GATES is the annual techno-cultural fest of GTBIT. Its an opportunity for the GTBIT community to showcase the rich culture of GTBITians; a venture to initiate, build and maintain acquaintances in the academic and industrial arena.


36% 5:46 pm
Downloads
About Us
Societies
Latest Events
Downloads
Gates
Students Timetable
Notice Board
Contact Us

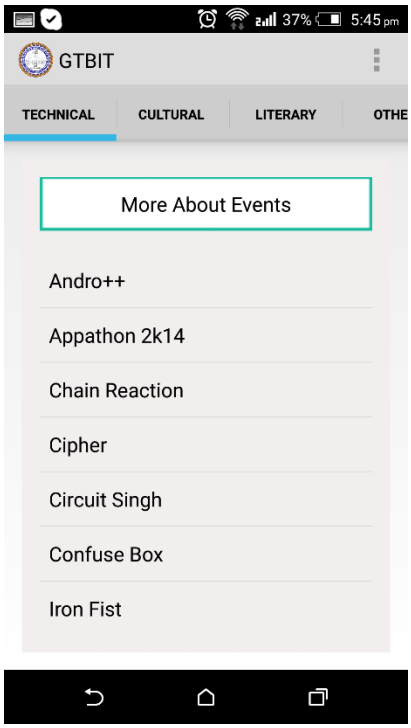

37% 5:45 pm
Downloads
SEMESTER 1-2
SEMESTER 3-4
SEMESTER 5-6
SEMESTER 7-8

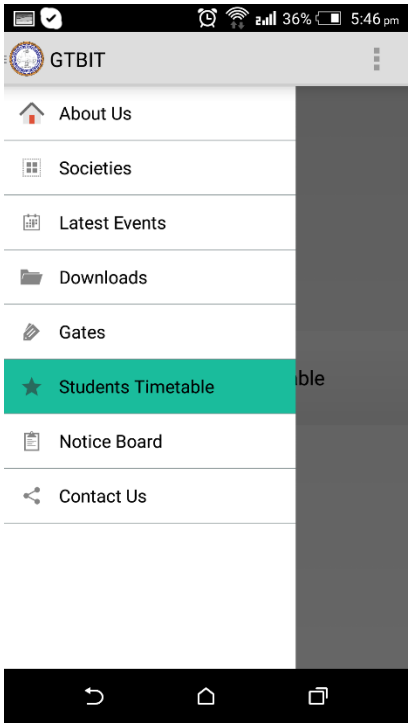

37% 5:45 pm
GTBIT
Download Section - 1st & 2nd Semester
June 04, 2014
Semester 1
Applied Mathematics
SYLLABUS
Assignments
Human Values and Professional Ethics- ETHS-109
SYLLABUS & Lecture Plan
Physics- ETPH-101
SYLLABUS
LECTURE PLAN

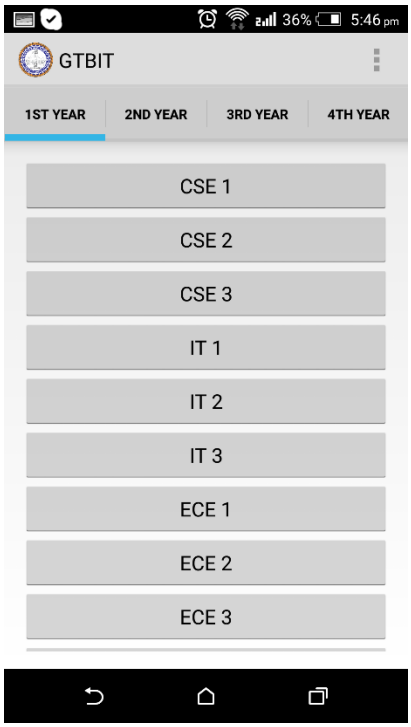

GTBIT
About Us
Societies
Latest Events
Downloads
Gates
Students Timetable
Notice Board
Contact Us

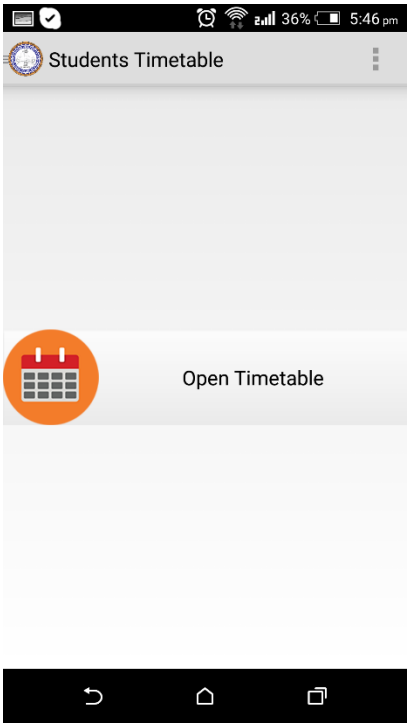

Gates
EVENTS
GTBIT's Annual Techno-cultural Event for creative Souls
GATES is the annual techno-cultural fest of GTBIT. Its an opportunity for the GTBIT community to showcase the rich culture of GTBITians; a venture to initiate, build and maintain acquaintances in the academic and industrial arena.

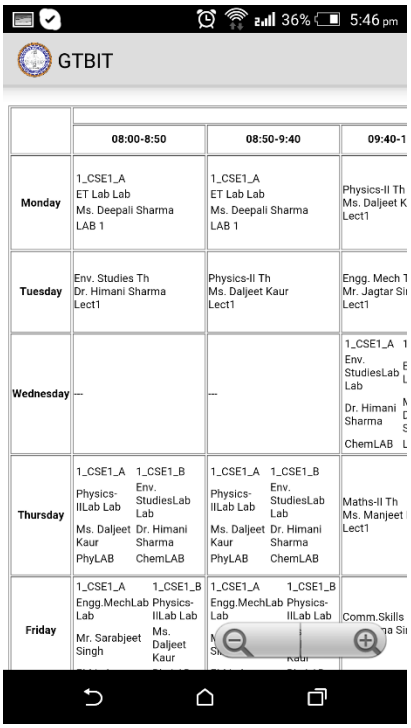

GTBIT
GIGABYTE WARS
THIS TIME CPU WILL CHECK YOUR SPEED AND PERFORMANCE
CASH PRIZES AND CERTIFICATES
ROUND 1 :HARDWARE TEST(MCQ)
ROUND 2 :DECODE THE SYSTEM CONFIGURATION(CIPHER)
AND RACE FOR ASSEMBLING.
FREE REGISTRATION
MANJOT SINGH(EM) :9899231114
JASKRIT KOHLI(EC):9711408060
VISHAL GOEL(EC):9650717103

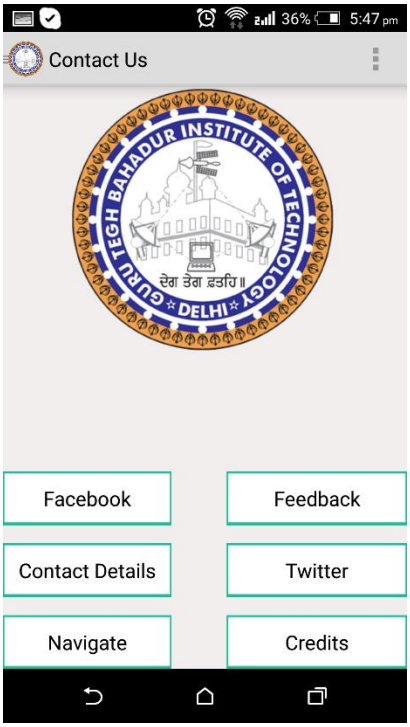

GTBIT
TECHNICAL
CULTURAL
LITERARY
More About Events
Andro++
Appathon 2k14
Chain Reaction
Cipher
Circuit Singh
Confuse Box
Iron Fist

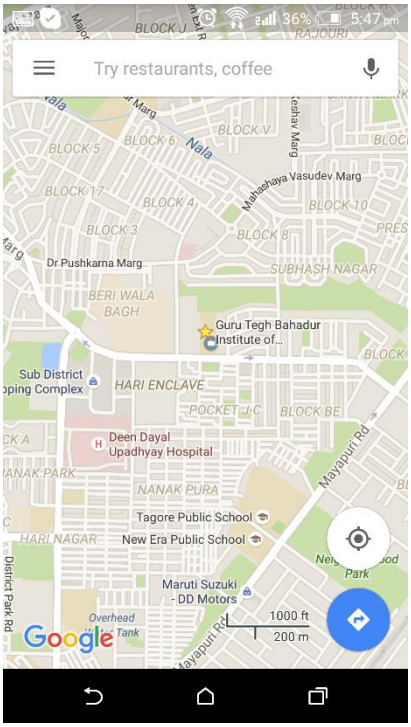

GTBIT
About Us
Societies
Latest Events
Downloads
Gates
Students Timetable
Notice Board
Contact Us

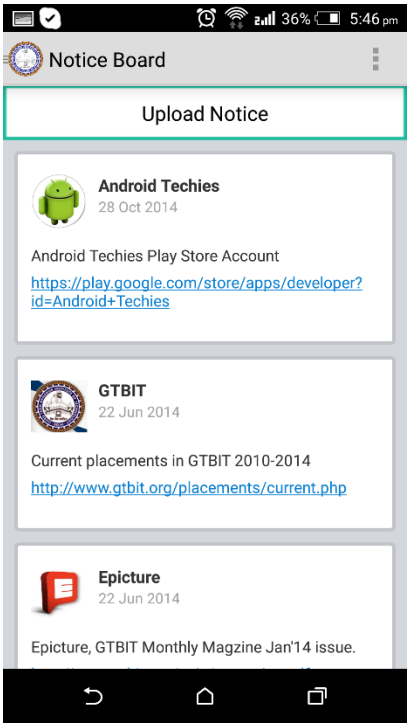

GTBIT
1ST YEAR
2ND YEAR
3RD YEAR
4TH YEAR
CSE 1
CSE 2
CSE 3
IT 1
IT 2
IT 3
ECE 1
ECE 2
ECE 3


Students Timetable
Open Timetable


GTBIT
08:00-8:50
08:50-9:40
Monday
Tuesday
Wednesday
Thursday
Friday


36% 5:47 pm
Contact Us
Facebook
Feedback
Contact Details
Twitter
Navigate
Credits


36% 5:47 pm
Try restaurants, coffee
Guru Tegh Bahadur Institute of...
Deen Dayal Upadhyay Hospital
Tagore Public School
New Era Public School
Maruti Suzuki - DD Motors
Google
1000 ft
200 m


36% 5:46 pm
Notice Board
Upload Notice
Android Techies
28 Oct 2014
Android Techies Play Store Account
https://play.google.com/store/apps/developer?id=Android+Techies
GTBIT
22 Jun 2014
Current placements in GTBIT 2010-2014
http://www.gtbit.org/placements/current.php
Epicture
22 Jun 2014
Epicture, GTBIT Monthly Magzine Jan'14 issue.