CS-223
SOFTWARE CODE REVIEW DOCUMENT

for

Project 8
Classroom Visualisation App-2

Prepared by:
Group-2
Harshit Agrawal - 160101030
Harshit Gupta - 160101031
Harshit Sharma - 160101032

April 22, 2018

Contents

1 Introduction

This document contains the complete software review of our app Classroom Visualisation App . Code review for a model is carried out after the module is successfully compiled and the all the syntax errors have been eliminated. Code reviews are extremely cost-effective strategies for reduction in coding errors and to produce high quality code. Normally, two types of reviews are carried out on the code of a module. These two types of code review techniques are code inspection and code walk through. The code testing team in isolation tests different units and modules of the system. Different members of the code testing team have submitted their reports. Although, we are only performing the Code Inspection part wherein each member goes through code to discover some common types of errors caused due to oversight and improper programming. The main objectives of the walk through was to discover the algorithmic and logical errors in the code. The members noted down their findings, which were discussed in a walk through meeting where the coder of the module were also present.

1.1 Goals

The main reasons for performing code review is to :

1. Finding bugs, since bug finding in code review are easier to find and fix, than later in testing.
2. Adherence to coding conventions
3. Improving code quality and understandability
4. Increasing efficiency, by finding trivial programming errors like data resource wastage or use of uninitialized variables.

2 Code

There are 7 modules in the code

2.1 Activity

Activity.java

```
1   package com.example.harshit.post;
2
3   /*
4   <header>
5   Module Name          : Activity
6   Date of Creation     : 10-04-2018
7   Author               : Group 2
8
9   Modification History : 11-04-2018 :- added the activity of display classroom
10
11  Synopsis             : This is the main activity of the app
12
13  Global Variables     : Button showData         button to show students record
14                         Button addData          button to add a student
15                         Button selectSeat       button to select a seat
16                         Button layoutGenerator  button to generate classroom layout
17                         Button displayClassroom button to display classroom
18                         TextView networkStatus  Text view to show the internet connectivity
19
20  Functions            : void addData            function to add a student
21                         void showData           function to show students record
22                         void selectSeat         function to select a seat
23                         void layoutGenerator    function to generate classroom layout
24                         void displayClassroom   function to display classroom
25  </header>
26  */
27
28  import android.content.Context;
29  import android.content.Intent;
30  import android.net.ConnectivityManager;
31  import android.net.NetworkInfo;
32  import android.support.v7.app.AppCompatActivity;
33  import android.os.Bundle;
34  import android.view.View;
35  import android.widget.Button;
36  import android.widget.TextView;
37
```

Activity.java - Google Chrome

```
35  import android.widget.Button;
36  import android.widget.TextView;
37
38  public class Activity extends AppCompatActivity {
39
40      Button showData, addData,selectSeat,layoutGenerator,displayClassroom; // declaration of display buttons on the main screen .
41      TextView networkStatus;   // declaration of text-view of network status
42
43
44      @Override
45      protected void onCreate(Bundle savedInstanceState) {
46
47          super.onCreate(savedInstanceState);   // Calling the constructor of parent class.
48          setContentView(R.layout.activity_);   // Setting the activity content to the main view.
49
50          // ****
51          // assigning the display button and text resource to the declared variable
52          showData  = findViewById(R.id.showData);
53          addData = findViewById(R.id.addData);
54          selectSeat = findViewById(R.id.selectSeat);
55          layoutGenerator = findViewById(R.id.layoutGenerator);
56          displayClassroom = findViewById(R.id.displayClassroom);
57          networkStatus = findViewById(R.id.Network_status);
58          // ****
59
60
61          // checking the internet connectivity of the device
62          ConnectivityManager connectivityManager = (ConnectivityManager)getSystemService(Context.CONNECTIVITY_SERVICE);
63          NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo(); // fetching network information
64
65          if(networkInfo!=null &&networkInfo.isConnected())
66          {
67              networkStatus.setVisibility(View.INVISIBLE);
68          }
69          else
70          {
71              // disabled buttons if internet is not connected
72              showData.setEnabled(false);
73              addData.setEnabled(false);
74              selectSeat.setEnabled(false);
```

```
        displayClassroom = findViewById(R.id.displayClassroom);
        networkStatus = findViewById(R.id.Network_status);
        // ****


        // checking the internet connectivity of the device
        ConnectivityManager connectivityManager = (ConnectivityManager)getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo(); // fetching network information

        if(networkInfo!=null &&networkInfo.isConnected())
        {
            networkStatus.setVisibility(View.INVISIBLE);
        }
        else
        {
            // disabled buttons if internet is not connected
            showData.setEnabled(false);
            addData.setEnabled(false);
            selectSeat.setEnabled(false);
            displayClassroom.setEnabled(false);
            layoutGenerator.setEnabled(false);

        }
    }

    //****
    // Declaring the on-click screens(activity) of the display buttons
    public void addData(View view)
    {
        startActivity(new Intent(this,addData.class));
    }
    public void showData(View view)
    {
        startActivity(new Intent(this,showData.class));
    }
    public void selectSeat(View view) {startActivity(new Intent(this,selectSeat.class));}
    public void layoutGenerator(View view) {startActivity(new Intent(this,Layout.class));}
    public void displayClassroom(View view) {startActivity(new Intent(this,display.class));}
    //****
}
```

2.2 Add data

addData.java

```
package com.example.harshit.post;

/*
<header>
Module Name          : addData
Date of Creation     : 10-04-2018
Author               : Group 2

Modification History : 11-04-2018 :- added exception for negative student id.

Synopsis             : This module is responsible for adding a student in the record

Global Variables     : EditText ID          Input from the user for student id
                       EditText Name       Input from the user for student name
                       String Student_ID,  Variable to send student id to database
                       String StudentName  Variable to send student name to database

Functions            : void saveInfo       save the student id and the student name to the database
</header>
*/

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
```

```java
import java.net.URLEncoder;

public class addData extends AppCompatActivity {

    EditText ID,Name; // declaration of text input on the add data screen .
    String Student_ID,StudentName;   // declaration of Student ID , Student Name .

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);         // Calling the constructor of parent class.
        setContentView(R.layout.activity_add_data); // Setting the activity content to the add data view.

        //***
        // Get the widgets reference from XML layout
        // assigning the display button and text resource to the declared variable
        ID = findViewById(R.id.ID);
        Name = findViewById(R.id.Name);
        //***

    }


    public void saveInfo(View view){
        // fetching the id and the name from the user input
        Student_ID = ID.getText().toString();
        StudentName = Name.getText().toString();
        // Starting the add request to the online database server
        BackgroundTask backgroundTask = new BackgroundTask();
        backgroundTask.execute(Student_ID,StudentName);
        finish(); // exit the current screen when data is added .
    }

    class BackgroundTask extends AsyncTask<String, Void, String> {

        String ADD_DATA_URL;  // url of the database request handling script
        @Override
        protected void onPreExecute(){  // task to be preformed before execution of the request to the server

            ADD_DATA_URL = "https://22harshit.000webhostapp.com/add_info.php";
```

```java
        @Override
        protected void onPreExecute(){  // task to be preformed before execution of the request to the server

            ADD_DATA_URL = "https://22harshit.000webhostapp.com/add_info.php";
        }


        @Override
        protected String doInBackground(String... args) {
            String id, name;  // declaring the id and the name from the arguments given to the function
            // fetching the id and the name from the arguments given to the function
            id = args[0];
            name = args[1];
            if (Integer.valueOf(id) >= 0){
                try {
                    //creating a URL
                    URL url = new URL(ADD_DATA_URL);
                    //Opening the URL using HttpURLConnection
                    HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
                    // specifying the request method
                    httpURLConnection.setRequestMethod("POST");
                    httpURLConnection.setDoOutput(true);
                    // creating a output stream
                    OutputStream outputStream = httpURLConnection.getOutputStream();
                    //We will use a buffered reader to read the string from service
                    BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));
                    // encoding the data string to be sent
                    String data_string = URLEncoder.encode("ID", "UTF-8") + "=" + URLEncoder.encode(id, "UTF-8") + "&" +
                            URLEncoder.encode("Name", "UTF-8") + "=" + URLEncoder.encode(name, "UTF-8");
                    // writing the data to the buffer
                    bufferedWriter.write(data_string);

                    // closing and flushing the connections
                    bufferedWriter.flush();
                    bufferedWriter.close();
                    outputStream.close();
                    InputStream inputStream = httpURLConnection.getInputStream();
                    inputStream.close();
                    httpURLConnection.disconnect();
                    //finally returning the status string
                    return "One row data inserted";
```

addData.java - Google Chrome

```
                    InputStream inputStream = httpURLConnection.getInputStream();
                    inputStream.close();
                    httpURLConnection.disconnect();
                    //finally returning the status string
                    return "One row data inserted";

                } catch (MalformedURLException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                return null;
            }
            else
            {
                return "Error : negative Student ID";
            }
        }

        @Override
        protected void onProgressUpdate(Void ... values){
            super.onProgressUpdate(values);
        }

        @Override
        protected void onPostExecute(String result)
        {
            // printing the success message
            Toast.makeText(getApplicationContext(),result,Toast.LENGTH_LONG).show();
        }


    }
}
```

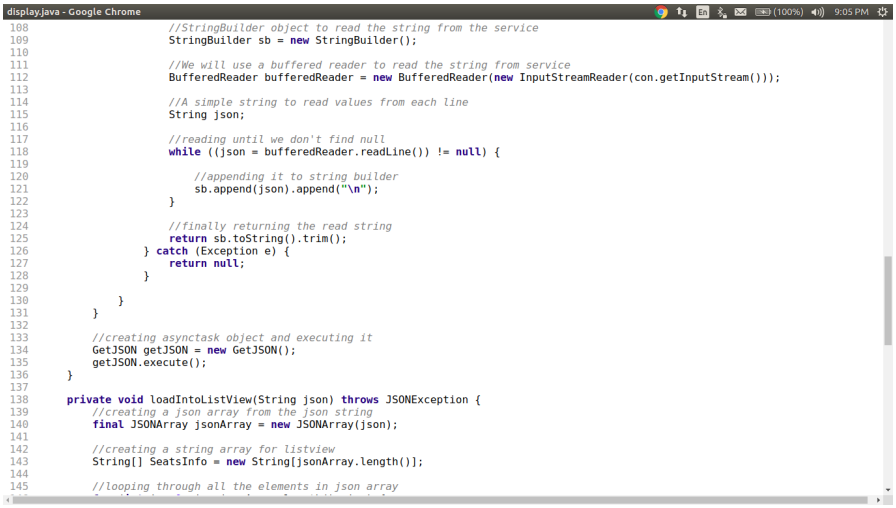
2.3 Display

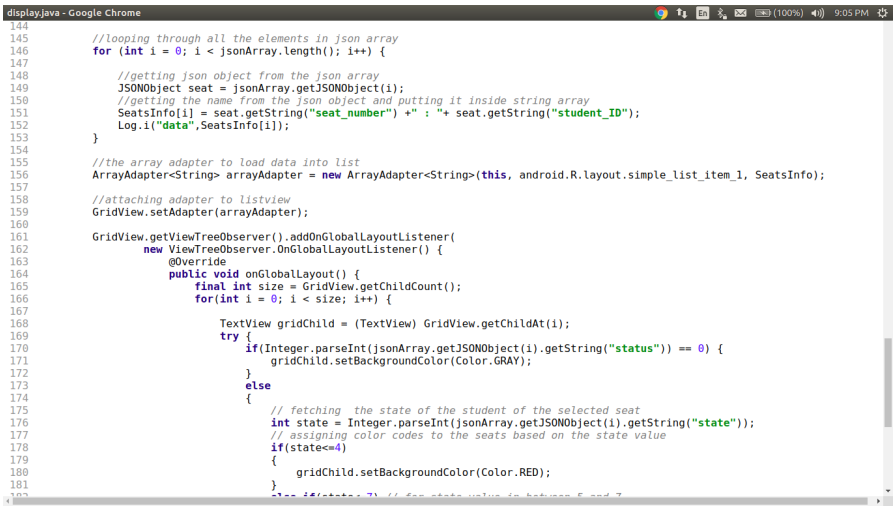
display.java

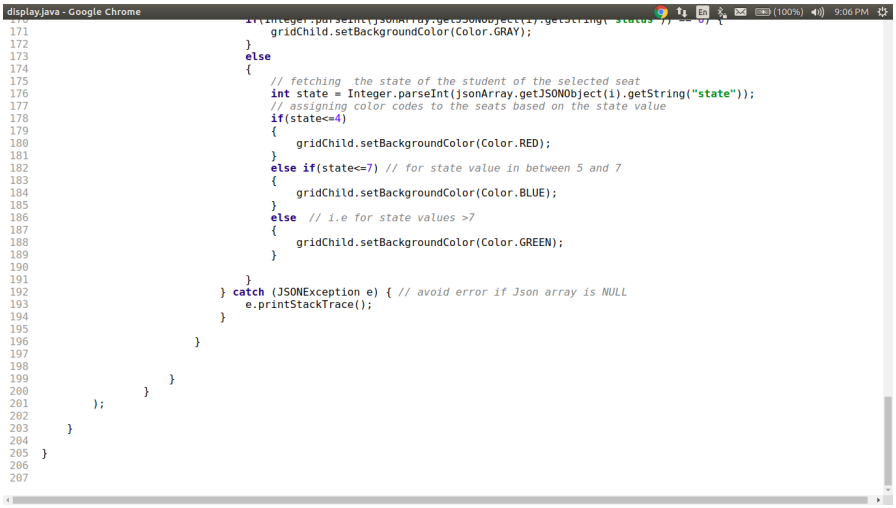
```
package com.example.harshit.post;

/*
<header>
Module Name          : display
Date of Creation     : 10-04-2018
Author               : Group 2

Modification History : 11-04-2018 :- added try catch statements to handle crashing of the app

Synopsis             : This module is responsible for displaying the classroom

Global Variables     : GridView GridView   gridview to represent seats

Functions            : void getJSON          get seat information in json format from database
                       void loadIntoListView load the json format array to list view to display
</header>
*/


import android.graphics.Color;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.ViewTreeObserver;
import android.widget.ArrayAdapter;
import android.widget.GridView;
import android.widget.TextView;
import android.widget.Toast;

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

import java.io.BufferedReader;
```

```
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class display extends AppCompatActivity {

    GridView GridView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); // Calling the constructor of parent class.
        setContentView(R.layout.activity_display); // Setting the activity content to the display view

        // Get the widgets reference from XML layout
        // assigning the display button and text resource to the declared variable
        GridView = (GridView)findViewById(R.id.display);

        getJSON("https://22harshit.000webhostapp.com/display.php");  // Fetching database entries of the seats information  database
    }

    private void getJSON(final String urlWebService) {
       /*
        * As fetching the json string is a network operation
        * And we cannot perform a network operation in main thread
        * so we need an AsyncTask
        * The constrains defined here are
        * Void -> We are not passing anything
        * Void -> Nothing at progress update as well
        * String -> After completion it should return a string and it will be the json string
        * */
        class GetJSON extends AsyncTask<Void, Void, String> {

            //this method will be called before execution
            //you can display a progress bar or something
            //so that user can understand that he should wait
            //as network operation may take some time
            @Override
```

```
            //you can display a progress bar or something
            //so that user can understand that he should wait
            //as network operation may take some time
            @Override
            protected void onPreExecute() {
                super.onPreExecute();
            }

            //this method will be called after execution
            //so here we are displaying a toast with the json string
            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                if(s!=null) {

                    try {
                        loadIntoListView(s);
                    } catch (JSONException e) { // try catch exception for JSON
                        e.printStackTrace();
                    }
                }
                else
                {
                    Toast.makeText(getApplicationContext(), "Internet not working", Toast.LENGTH_LONG).show();
                }
            }

            //in this method we are fetching the json string
            @Override
            protected String doInBackground(Void... voids) {

                try {
                    //creating a URL
                    URL url = new URL(urlWebService);

                    //Opening the URL using HttpURLConnection
                    HttpURLConnection con = (HttpURLConnection) url.openConnection();

                    //StringBuilder object to read the string from the service
                    StringBuilder sb = new StringBuilder();
```


display.java - Google Chrome
(100%) 9:05 PM
//StringBuilder object to read the string from the service
StringBuilder sb = new StringBuilder();
//We will use a buffered reader to read the string from service
BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(con.getInputStream()));
//A simple string to read values from each line
String json;
//reading until we don't find null
while ((json = bufferedReader.readLine()) != null) {
//appending it to string builder
sb.append(json).append("\n");
}
//finally returning the read string
return sb.toString().trim();
} catch (Exception e) {
return null;
}
}
}
//creating asynctask object and executing it
GetJSON getJSON = new GetJSON();
getJSON.execute();
}
private void loadIntoListView(String json) throws JSONException {
//creating a json array from the json string
final JSONArray jsonArray = new JSONArray(json);
//creating a string array for listview
String[] SeatsInfo = new String[jsonArray.length()];
//looping through all the elements in json array


display.java - Google Chrome
(100%) 9:05 PM
//looping through all the elements in json array
for (int i = 0; i < jsonArray.length(); i++) {
//getting json object from the json array
JSONObject seat = jsonArray.getJSONObject(i);
//getting the name from the json object and putting it inside string array
SeatsInfo[i] = seat.getString("seat_number") +" : "+ seat.getString("student_ID");
Log.i("data",SeatsInfo[i]);
}
//the array adapter to load data into list
ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, SeatsInfo);
//attaching adapter to listview
GridView.setAdapter(arrayAdapter);
GridView.getViewTreeObserver().addOnGlobalLayoutListener(
new ViewTreeObserver.OnGlobalLayoutListener() {
@Override
public void onGlobalLayout() {
final int size = GridView.getChildCount();
for(int i = 0; i < size; i++) {
TextView gridChild = (TextView) GridView.getChildAt(i);
try {
if(Integer.parseInt(jsonArray.getJSONObject(i).getString("status")) == 0) {
gridChild.setBackgroundColor(Color.GRAY);
}
else
{
// fetching the state of the student of the selected seat
int state = Integer.parseInt(jsonArray.getJSONObject(i).getString("state"));
// assigning color codes to the seats based on the state value
if(state<=4)
{
gridChild.setBackgroundColor(Color.RED);
}




```
                                    gridChild.setBackgroundColor(Color.GRAY);
                                }
                                else
                                {
                                    // fetching  the state of the student of the selected seat
                                    int state = Integer.parseInt(jsonArray.getJSONObject(i).getString("state"));
                                    // assigning color codes to the seats based on the state value
                                    if(state<=4)
                                    {
                                        gridChild.setBackgroundColor(Color.RED);
                                    }
                                    else if(state<=7) // for state value in between 5 and 7
                                    {
                                        gridChild.setBackgroundColor(Color.BLUE);
                                    }
                                    else  // i.e for state values >7
                                    {
                                        gridChild.setBackgroundColor(Color.GREEN);
                                    }

                                }
                            } catch (JSONException e) { // avoid error if Json array is NULL
                                e.printStackTrace();
                            }

                        }


                    }
                }
            );

    }
}
```

2.4 Seat Status

seatStatus.java

```
package com.example.harshit.post;

/*
<header>
Module Name          : seatStatus
Date of Creation     : 10-04-2018
Author               : Group 2

Modification History : 11-04-2018 :- added colour to the last seat selected

Synopsis             : This module is responsible for selecting unoccupied seats by
                       checking their status.

Global Variables     :     TextView ID          declaration of text view  on the seatStatus screen .
                           GridView GridView    declaration of Seat Grid View on the seatStatus screen .
                           Button ConfirmSeat   declaration of Confirm button on the seatStatus screen .
                           String SeatNumber    variable to send seat no. to the database
                           String StudentID     variable to send student id. to the database
                           String SelectedSeat  variable to send seat no. of the seat selected to the database
                           int previousSelected Position  represents the last seat selected

Functions            : void saveInfo             function that saves the information in the database
                       void onItemSelected       function thatfetches the seat no. of the selected seat
                       void getJSON              get seat information in json format from database
                       void loadIntoListView     load the json format array to list view to display
</header>
*/

import android.graphics.Color;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.GridView;
```

```
import android.widget.GridView;
import android.widget.TextView;
import android.widget.Toast;

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

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Random;

public class seatStatus extends AppCompatActivity implements AdapterView.OnItemSelectedListener {

    TextView ID; // declaration of text view  on the seatStatus screen .
    GridView GridView; // declaration of Seat Grid View on the seatStatus screen .
    Button ConfirmSeat; // declaration of Confirm button on the seatStatus screen .
    String SeatNumber;
    String StudentID="";
    String SelectedSeat;
    int previousSelectedPosition=-1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); // Calling the constructor of parent class.
        setContentView(R.layout.activity_seat_status);  // Setting the activity content to the add data view.

        // Get the widgets reference from XML layout
        // assigning the display button and text resource to the declared variable
        GridView = (GridView) findViewById(R.id.gridLayout);
        ID = (TextView)findViewById(R.id.Sudent_ID);
```

```
        ID = (TextView)findViewById(R.id.Sudent_ID);
        ConfirmSeat = (Button)findViewById(R.id.confirm_seat);

        // extracting the information passed on by previous activity
        Bundle bundle = getIntent().getExtras();
        ID.setText(bundle.getString("ID"));
        StudentID = bundle.getString("ID");

        // Fetching database entries of the seats available
        getJSON("https://22harshit.000webhostapp.com/out_seats.php");

    }

    public void saveInfo(View view){
        // fetching the id and the seat number from the user input
        StudentID = ID.getText().toString();
        SeatNumber = SelectedSeat;
        BackgroundTask backgroundTask = new BackgroundTask();
        // Starting the add request to the online database server
        backgroundTask.execute(StudentID,SeatNumber);
        finish(); // exit the current screen when data is added .

    }


    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        // On selecting a spinner item
        SelectedSeat = parent.getItemAtPosition(position).toString();

        // Showing selected spinner item
        Toast.makeText(parent.getContext(), "Selected: " + SelectedSeat, Toast.LENGTH_LONG).show();

    }
    public void onNothingSelected(AdapterView<?> arg0) {
    }


    private void getJSON(final String urlWebService) {
       /*
       * As fetching the json string is a network operation
```

```
112
113     private void getJSON(final String urlWebService) {
114        /*
115        * As fetching the json string is a network operation
116        * And we cannot perform a network operation in main thread
117        * so we need an AsyncTask
118        * The constrains defined here are
119        * Void -> We are not passing anything
120        * Void -> Nothing at progress update as well
121        * String -> After completion it should return a string and it will be the json string
122        * */
123         class GetJSON extends AsyncTask<Void, Void, String> {
124
125             //this method will be called before execution
126             //you can display a progress bar or something
127             //so that user can understand that he should wait
128             //as network operation may take some time
129             @Override
130             protected void onPreExecute() {
131                 super.onPreExecute();
132             }
133
134             //this method will be called after execution
135             //so here we are displaying a toast with the json string
136             @Override
137             protected void onPostExecute(String s) {
138                 super.onPostExecute(s);
139
140                 try {
141                     loadIntoListView(s);
142                 } catch (JSONException e) {
143                     e.printStackTrace();
144                 }
145
146             }
147
148             //in this method we are fetching the json string
149             @Override
150             protected String doInBackground(Void... voids) {
151
152
```

```
146             //in this method we are fetching the json string
149             @Override
150             protected String doInBackground(Void... voids) {
151
152
153                 try {
154                     //creating a URL
155                     URL url = new URL(urlWebService);
156
157                     //Opening the URL using HttpURLConnection
158                     HttpURLConnection con = (HttpURLConnection) url.openConnection();
159
160                     //StringBuilder object to read the string from the service
161                     StringBuilder sb = new StringBuilder();
162
163                     //We will use a buffered reader to read the string from service
164                     BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(con.getInputStream()));
165
166                     //A simple string to read values from each line
167                     String json;
168
169                     //reading until we don't find null
170                     while ((json = bufferedReader.readLine()) != null) {
171
172                         //appending it to string builder
173                         sb.append(json).append("\n");
174                     }
175
176                     //finally returning the read string
177                     return sb.toString().trim();
178                 } catch (Exception e) {
179                     return null;
180                 }
181
182             }
183         }
184
185         //creating asynctask object and executing it
186         GetJSON getJSON = new GetJSON();
187         getJSON.execute();
188     }
```

```
        getJSON.execute();
    }

    private void loadIntoListView(String json) throws JSONException {
        //creating a json array from the json string
        JSONArray jsonArray = new JSONArray(json);

        //creating a string array for listview
        String[] seatStatus = new String[jsonArray.length()];

        //looping through all the elements in json array
        for (int i = 0; i < jsonArray.length(); i++) {

            //getting json object from the json array
            JSONObject obj = jsonArray.getJSONObject(i);
            //getting the name from the json object and putting it inside string array
            seatStatus[i] = obj.getString("seat_number") ;

        }

        //the array adapter to load data into list
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, seatStatus);

        //attaching adapter to list view
        GridView.setAdapter(arrayAdapter);
        GridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {


            @Override

            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                // Get the selected item text
                String selectedItem = parent.getItemAtPosition(position).toString();

                // Get the current selected view as a TextView
                TextView tv = (TextView) view;

                // Set the current selected item background color
                tv.setBackgroundColor(Color.parseColor("#FF9AD082"));

                // Set the current selected item text color
```

```
                // Set the current selected item text color
                tv.setTextColor(Color.BLUE);

                // Get the last selected View from GridView
                TextView previousSelectedView = (TextView) GridView.getChildAt(previousSelectedPosition);
                SelectedSeat=Integer.toString(position+1);
                // If there is a previous selected view exists
                if (previousSelectedPosition != -1)
                {
                    // Set the last selected View to deselect
                    previousSelectedView.setSelected(false);

                    // Set the last selected View background color as deselected item
                    previousSelectedView.setBackgroundColor(Color.parseColor("#FFFFFF"));

                    // Set the last selected View text color as deselected item
                    previousSelectedView.setTextColor(Color.DKGRAY);
                }

                // Set the current selected view position as previousSelectedPosition
                previousSelectedPosition = position;

            }
        });

    }


    class BackgroundTask extends AsyncTask<String, Void, String> {

        String SELECT_SEAT_URL;
        @Override
        protected void onPreExecute(){

            SELECT_SEAT_URL = "http://22harshit.000webhostapp.com/new_seat.php";
        }


        @Override
        protected String doInBackground(String... args) {
```

```
        String SELECT_SEAT_URL;
        @Override
        protected void onPreExecute(){

            SELECT_SEAT_URL = "http://22harshit.000webhostapp.com/new_seat.php";
        }


        @Override
        protected String doInBackground(String... args) {
            String id,seat_number;
            Random rnd = new Random();
            String state = Integer.toString(rnd.nextInt(10) + 1);
            id = args[0];
            seat_number = args[1];
            try {
                //creating a URL
                URL url  = new URL(SELECT_SEAT_URL);
                //Opening the URL using HttpURLConnection
                HttpURLConnection httpURLConnection  = (HttpURLConnection) url.openConnection();
                // specifying the request method
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);
                // creating a output stream
                OutputStream outputStream = httpURLConnection.getOutputStream();
                //We will use a buffered reader to read the string from service
                BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream,"UTF-8"));
                // encoding the data string to be sent
                String data_string = URLEncoder.encode("ID","UTF-8")+"="+URLEncoder.encode(id,"UTF-8")+"&"+
                        URLEncoder.encode("seat_number","UTF-8")+"="+URLEncoder.encode(seat_number,"UTF-8")+"&"+
                        URLEncoder.encode("state","UTF-8")+"="+URLEncoder.encode(state,"UTF-8");
                // writing the data to the buffer
                bufferedWriter.write(data_string);

                // closing and flushing the connections
                bufferedWriter.flush();
                bufferedWriter.close();
                bufferedWriter.close();
                outputStream.close();

                // reading the response from server
```

```
                bufferedWriter.close();
                outputStream.close();

                // reading the response from server
                InputStream inputStream = httpURLConnection.getInputStream();
                BufferedReader r = new BufferedReader(new InputStreamReader(inputStream));
                StringBuilder statusMessage = new StringBuilder();
                String line;
                while ((line = r.readLine()) != null) {
                    statusMessage.append(line).append('\n');
                }
                inputStream.close();
                httpURLConnection.disconnect();

                //finally returning the status string
                return statusMessage.toString();

            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onProgressUpdate(Void ... values){
            super.onProgressUpdate(values);
        }

        @Override
        protected void onPostExecute(String result)
        {
            // printing the status message
            Toast.makeText(getApplicationContext(),result,Toast.LENGTH_LONG).show();
        }


    }
```

2.5 Select Seat

selectSeat.java


```
1   package com.example.harshit.post;
2
3   /*
4   <header>
5   Module Name          : selectSeat
6   Date of Creation     : 10-04-2018
7   Author               : Group 2
8
9   Modification History : 11-04-2018 :- added the exception for existing id allocated seat
10
11  Synopsis             : This module is responsible for checking the student id and proceed for seat selection
12
13  Global Variables     : EditText ID         declaration of text input on the select seat screen .
14                         String StudentID    declaration of Student ID .
15                         Button selectSeat   button to go for seat selection
16
17  Functions            : void checkStudent   check if the student is enrolled in the course or not
18                         void seatStatus     fetches seat status and proceed for seat selection
19  </header>
20  */
21
22  import android.content.Intent;
23  import android.os.AsyncTask;
24  import android.support.v7.app.AppCompatActivity;
25  import android.os.Bundle;
26  import android.util.Log;
27  import android.view.View;
28  import android.widget.Button;
29  import android.widget.EditText;
30  import android.widget.Toast;
31
32  import java.io.BufferedReader;
33  import java.io.BufferedWriter;
34  import java.io.IOException;
35  import java.io.InputStream;
36  import java.io.InputStreamReader;
37  import java.io.OutputStream;
```

```
37  import java.io.OutputStream;
38  import java.io.OutputStreamWriter;
39  import java.net.HttpURLConnection;
40  import java.net.MalformedURLException;
41  import java.net.URL;
42  import java.net.URLEncoder;
43
44  import static java.lang.Boolean.FALSE;
45  import static java.lang.Boolean.TRUE;
46
47  public class selectSeat extends AppCompatActivity {
48
49      EditText ID; // declaration of text input on the select seat screen .
50      String StudentID; // declaration of Student ID .
51      Button selectSeat; // declaration of select seat button on the select seat screen . .
52
53      @Override
54      protected void onCreate(Bundle savedInstanceState) {
55          super.onCreate(savedInstanceState);// Calling the constructor of parent class.
56          setContentView(R.layout.activity_select_seat);// Setting the activity content to theselect seat view
57
58          // Get the widgets reference from XML layout
59          // assigning the display button and text resource to the declared variable
60          ID = (EditText) findViewById(R.id.StudentID);
61          selectSeat = (Button) findViewById(R.id.selectSeat);
62      }
63
64      public void checkStudent(View view) {
65          // fetching the id from the user input
66          StudentID = ID.getText().toString();
67          BackgroundTask backgroundTask = new BackgroundTask();
68          backgroundTask.execute(StudentID);
69      }
70
71      class BackgroundTask extends AsyncTask<String, Void, String> {
72
73          String CHECK_INFO_URL;
74
75          @Override
76          protected void onPreExecute() {
```

```
        protected void onPreExecute() {

            CHECK_INFO_URL = "https://22harshit.000webhostapp.com/check.php";
        }


        @Override
        protected String doInBackground(String... args) {
            String id;
            id = args[0];
            try {
                //creating a URL
                URL url = new URL(CHECK_INFO_URL);
                //Opening the URL using HttpURLConnection
                HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
                // specifying the request method
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);
                // creating a output stream
                OutputStream outputStream = httpURLConnection.getOutputStream();
                //We will use a buffered reader to read the string from service
                BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));
                // encoding the data string to be sent
                String data_string = URLEncoder.encode("ID", "UTF-8") + "=" + URLEncoder.encode(id, "UTF-8");
                // writing the data to the buffer
                bufferedWriter.write(data_string);

                // closing and flushing the connections
                bufferedWriter.flush();
                bufferedWriter.close();
                bufferedWriter.close();
                outputStream.close();

                // reading the response from server
                InputStream inputStream = httpURLConnection.getInputStream();
                BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
                StringBuilder statusMessage = new StringBuilder();
                String line;
                while ((line = bufferedReader.readLine()) != null) {
                    statusMessage.append(line).append('\n');
```

```
                StringBuilder statusMessage = new StringBuilder();
                String line;
                while ((line = bufferedReader.readLine()) != null) {
                    statusMessage.append(line).append('\n');
                }
                inputStream.close();
                httpURLConnection.disconnect();
                //finally returning the status string
                return statusMessage.toString();


            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onProgressUpdate(Void... values) {
            super.onProgressUpdate(values);
        }

        @Override
        protected void onPostExecute(String result) {

            super.onPostExecute(result);

            // trimming of the data to convert to integer
            if(result!=null) {
                result = result.replaceAll("\\s", "");
                int status = Integer.parseInt(result.replaceAll("[\\D]", ""));
                // checking whether the student id is in the database or not
                if (status == 1) {
                    selectSeat.setEnabled(TRUE);
                } else {
                    // printing the status message
                    Toast.makeText(getApplicationContext(), "Enter a valid Student ID", Toast.LENGTH_LONG).show();
                    selectSeat.setEnabled(FALSE);
```

```
            super.onProgressUpdate(values);
        }

        @Override
        protected void onPostExecute(String result) {

            super.onPostExecute(result);

            // trimming of the data to convert to integer
            if(result!=null) {
                result = result.replaceAll("\\s", "");
                int status = Integer.parseInt(result.replaceAll("[\\D]", ""));
                // checking whether the student id is in the database or not
                if (status == 1) {
                    selectSeat.setEnabled(TRUE);
                } else {
                    // printing the status message
                    Toast.makeText(getApplicationContext(), "Enter a valid Student ID", Toast.LENGTH_LONG).show();
                    selectSeat.setEnabled(FALSE);
                }
            }
            else
                Toast.makeText(getApplicationContext(), "Internet not working", Toast.LENGTH_LONG).show();
        }


    }

    public void seatStatus(View view){ // function displaying available seats for the students
        Bundle bundle = new Bundle(); // creating a information bundle to pass data to the next view
        bundle.putString("ID",ID.getText().toString()); // adding information to the bundle
        Intent intent = new Intent(this,seatStatus.class); // creating a new intent
        intent.putExtras(bundle); // attaching the extra information with the intent
        startActivity(intent); // starting a new intent ( screen )
    }

}
```

2.6 Show Data

showData.java

```
package com.example.harshit.post;

/*
<header>
Module Name          : showData
Date of Creation     : 10-04-2018
Author               : Group 2

Modification History : 11-04-2018 :- added try catch statements for error handling

Synopsis             : This module is responsible for showing student record.

Global Variables     : ListView listView        - View to show the existing students enrolled in the course


Functions            : void getJSON              get student information in json format from database
                       void loadIntoListView     load the json format array to list view to display student record
</header>
*/

import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

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

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
```

```
import java.net.URL;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class showData extends AppCompatActivity {

    ListView listView ; // declaration of list view on the show data screen .

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        setContentView(R.layout.activity_show_data);    // Setting the activity content to the add data view.
        super.onCreate(savedInstanceState);             // Calling the constructor of parent class
        listView = findViewById(R.id.student_list);     // assigning the list view resource to the declared variable
        getJSON("https://22harshit.000webhostapp.com/out.php"); // Fetching database entries of the students enrolled
    }

    private void getJSON(final String urlWebService) {
       /*
        * As fetching the json string is a network operation
        * And we cannot perform a network operation in main thread
        * so we need an AsyncTask
        * The constrains defined here are
        * Void -> We are not passing anything
        * Void -> Nothing at progress update as well
        * String -> After completion it should return a string and it will be the json string
        * */
        class GetJSON extends AsyncTask<Void, Void, String> {

            //this method will be called before execution
            //you can display a progress bar or something
            //so that user can understand that he should wait
            //as network operation may take some time
            @Override
            protected void onPreExecute() {
                super.onPreExecute();
            }

            //this method will be called after execution
```

```
                super.onPreExecute();
            }

            //this method will be called after execution
            //so here we are displaying a toast with the json string
            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                try {
                    if(s!=null)
                    {
                        loadIntoListView(s);
                    }
                    else
                        Toast.makeText(getApplicationContext(), "Internet connection is not working ", Toast.LENGTH_LONG).show();

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

            //in this method we are fetching the json string
            @Override
            protected String doInBackground(Void... voids) {



                try {
                    //creating a URL
                    URL url = new URL(urlWebService);

                    //Opening the URL using HttpURLConnection
                    HttpURLConnection con = (HttpURLConnection) url.openConnection();

                    //StringBuilder object to read the string from the service
                    StringBuilder sb = new StringBuilder();

                    //We will use a buffered reader to read the string from service
                    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(con.getInputStream()));

                    //A simple string to read values from each line
```

```
                    //A simple string to read values from each line
                    String json;

                    //reading until we don't find null
                    while ((json = bufferedReader.readLine()) != null) {

                        //appending it to string builder
                        sb.append(json).append("\n");
                    }

                    //finally returning the read string
                    return sb.toString().trim();
                } catch (Exception e) {
                    return null;
                }

            }
        }

        //creating asynctask object and executing it
        GetJSON getJSON = new GetJSON();
        getJSON.execute();
    }

    private void loadIntoListView(String json) throws JSONException {
        //creating a json array from the json string
        JSONArray jsonArray = new JSONArray(json);

        //creating a string array for listview
        String[] studentdataList = new String[jsonArray.length()];

        //looping through all the elements in json array
        for (int i = 0; i < jsonArray.length(); i++) {

            //getting json object from the json array
            JSONObject student = jsonArray.getJSONObject(i);
            //getting the name from the json object and putting it inside string array
            studentdataList[i] = student.getString("ID") + " : " + student.getString("Name");
            Log.i("data",studentdataList[i]);
        }
```

```
                    return null;
                }

            }
        }

        //creating asynctask object and executing it
        GetJSON getJSON = new GetJSON();
        getJSON.execute();
    }

    private void loadIntoListView(String json) throws JSONException {
        //creating a json array from the json string
        JSONArray jsonArray = new JSONArray(json);

        //creating a string array for listview
        String[] studentdataList = new String[jsonArray.length()];

        //looping through all the elements in json array
        for (int i = 0; i < jsonArray.length(); i++) {

            //getting json object from the json array
            JSONObject student = jsonArray.getJSONObject(i);
            //getting the name from the json object and putting it inside string array
            studentdataList[i] = student.getString("ID") + " : " + student.getString("Name");
            Log.i("data",studentdataList[i]);
        }

        //the array adapter to load data into list
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, studentdataList);

        //attaching adapter to listview
        Log.i("data",arrayAdapter.toString());
        listView.setAdapter(arrayAdapter);
    }

}
```

2.7 Layout

Layout.java

```
1   package com.example.harshit.post;
2
3   /*
4   <header>
5   Module Name          : Layout
6   Date of Creation     : 10-04-2018
7   Author               : Group 2
8
9   Modification History : 11-04-2018 :- added exception for negative no. of seats
10
11  Synopsis             : This module is responsible for generating classroom layout
12
13  Global Variables     : GridView GridView;       gridview to represent seats
14                         private int previousSelectedPosition = -1;    represents previousSelectedPosition
15                         EditText EditText;         User input for no. of seats
16                         Button GenerateLayout      button to generate classroom layout
17
18  Functions            : void onClick    clicking the button will generate the layout based on no. of seats
19  </header>
20  */
21
22  import android.graphics.Color;
23  import android.os.AsyncTask;
24  import android.support.v7.app.AppCompatActivity;
25  import android.os.Bundle;
26  import android.util.Log;
27  import android.view.View;
28  import android.widget.AdapterView;
29  import android.widget.ArrayAdapter;
30  import android.widget.Button;
31  import android.widget.EditText;
32  import android.widget.GridView;
33  import android.widget.TextView;
34  import android.widget.Toast;
35
36  import java.io.BufferedWriter;
37  import java.io.IOException;
```

```
37  import java.io.IOException;
38  import java.io.InputStream;
39  import java.io.OutputStream;
40  import java.io.OutputStreamWriter;
41  import java.net.HttpURLConnection;
42  import java.net.MalformedURLException;
43  import java.net.URL;
44  import java.net.URLEncoder;
45
46  public class Layout extends AppCompatActivity {
47
48      GridView GridView;
49      private int previousSelectedPosition = -1;
50      EditText EditText;
51      Button GenerateLayout;
52      @Override
53      protected void onCreate(Bundle savedInstanceState) {
54          super.onCreate(savedInstanceState); // Calling the constructor of parent class.
55          setContentView(R.layout.activity_layout); // Setting the activity content to the classroom layout view.
56          // Get the widgets reference from XML layout
57          GridView = findViewById(R.id.gridview);
58          EditText = findViewById(R.id.seat_count);
59          GenerateLayout = findViewById(R.id.generate);
60
61      }
62
63      public void onClick(View view){
64
65          String seats = EditText.getText().toString();  // extracting the string from the user input
66          // Initializing a new String Array
67          if(seats.matches(""))
68          {
69              Toast.makeText(getApplicationContext(),"Number of seats entered is empty",Toast.LENGTH_LONG).show();
70              return ;
71          }
72          int number_of_seats = Integer.parseInt(seats); // converting string to integer
73          if(number_of_seats<0)
74          {
75              Toast.makeText(getApplicationContext(),"Number of seats entered is negative",Toast.LENGTH_LONG).show();
76              return ;
```

```
        if(number_of_seats<0)
        {
            Toast.makeText(getApplicationContext(),"Number of seats entered is negative",Toast.LENGTH_LONG).show();
            return ;
        }
        String[] seatInfo = new String[number_of_seats];

        for (int i=0;i<number_of_seats;i++)
        {
            seatInfo[i]=Integer.toString(i+1);
        }

        // Populate a List from Array elements
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, seatInfo);

        GridView.setAdapter(adapter);
        GridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {


            @Override

            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                // Get the selected item text
                String selectedItem = parent.getItemAtPosition(position).toString();

                // Get the current selected view as a TextView
                TextView tv = (TextView) view;

                // Set the current selected item background color
                tv.setBackgroundColor(Color.parseColor("#FF9AD082"));

                // Set the current selected item text color
                tv.setTextColor(Color.BLUE);

                // Get the last selected View from GridView
                TextView previousSelectedView = (TextView) GridView.getChildAt(previousSelectedPosition);

                // If there is a previous selected view exists
                if (previousSelectedPosition != -1)
                {
                    // Set the last selected View to deselect
```

```
                // If there is a previous selected view exists
                if (previousSelectedPosition != -1)
                {
                    // Set the last selected View to deselect
                    previousSelectedView.setSelected(false);

                    // Set the last selected View background color as deselected item
                    previousSelectedView.setBackgroundColor(Color.parseColor("#FFFFFF"));

                    // Set the last selected View text color as deselected item
                    previousSelectedView.setTextColor(Color.DKGRAY);
                }

                // Set the current selected view position as previousSelectedPosition
                previousSelectedPosition = position;
            }
        });

        BackgroundTask backgroundTask = new BackgroundTask();
        backgroundTask.execute(seats);
    }

    class BackgroundTask extends AsyncTask<String, Void, String> {

        String ADD_SEATS_URL;
        @Override
        protected void onPreExecute(){

            ADD_SEATS_URL = "https://22harshit.000webhostapp.com/add_seat.php";
        }


        @Override
        protected String doInBackground(String... args) {
            String seats;
            seats = args[0];

            try {

                //creating a URL
```

```
        @Override
        protected String doInBackground(String... args) {
            String seats;
            seats = args[0];

            try {

                //creating a URL
                URL url  = new URL(ADD_SEATS_URL);
                //Opening the URL using HttpURLConnection
                HttpURLConnection httpURLConnection  = (HttpURLConnection) url.openConnection();
                // specifying the request method
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);
                // creating a output stream
                OutputStream outputStream = httpURLConnection.getOutputStream();
                //We will use a buffered reader to read the string from service
                BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream,"UTF-8"));
                // encoding the data string to be sent
                String data_string = URLEncoder.encode("seats","UTF-8")+"="+URLEncoder.encode(seats,"UTF-8");
                // writing the data to the buffer
                bufferedWriter.write(data_string);

                // closing and flushing the connections
                bufferedWriter.flush();
                bufferedWriter.close();
                bufferedWriter.close();
                outputStream.close();
                InputStream inputStream = httpURLConnection.getInputStream();
                inputStream.close();
                httpURLConnection.disconnect();

                //finally returning the status string
                return "Layout generated";

            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
```

```
                // writing the data to the buffer
                bufferedWriter.write(data_string);

                // closing and flushing the connections
                bufferedWriter.flush();
                bufferedWriter.close();
                bufferedWriter.close();
                outputStream.close();
                InputStream inputStream = httpURLConnection.getInputStream();
                inputStream.close();
                httpURLConnection.disconnect();

                //finally returning the status string
                return "Layout generated";

            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onProgressUpdate(Void ... values){
            super.onProgressUpdate(values);
        }

        @Override
        protected void onPostExecute(String result)
        {
            // printing the success message
            Toast.makeText(getApplicationContext(),result,Toast.LENGTH_LONG).show();
        }


    }
}
```

3 Code Testing Team

3.1 Team profile

The code testing team comprises of the following members, all of whom are Undergraduates currently pursuing Bachelor of Technology at Indian Institute of Technology Guwahati, India in the Department of Computer Science and Engineering. All of the members are currently in the sophomore year.

1. Mitansh jain
2. Sujoy Ghosh
3. Daman Tekchandani

All members of the team are proficient in Java and have past experience in developing android applications.

4 Code Inspection Report

4.1 Code Report by Mitansh Jain

- There is use of too many global variables.
- In onClick function of Layout Activity, there is no handling of error for null value of variable number_of_seats.
- Indentation is followed properly while writing code.
- No proper bracket coding convention is used
- There are sufficient try catch statements to avoid error occurrence while running the app
- No Jump (go to) statements were used in the modules.
- All the array references used in the code were in the bound of the array.
- Proper care has been taken when there is no internet connection.
- Grid View has been chosen, it may cause problem when number of seats is high.
- The headers of each module had all the details that are required from a good header, like - Name of the module, Date on which the module was created, Author's name, Modification history, Synopsis of the module, Different functions supported, along with their input/output parameters.
- No uninitialized variables were found in any module.
- All the loops will terminate according to their terminating conditions respectively after some finite iterations.

4.2 Code Report by Sujoy Ghosh

- No index of any array is out of bound.
- JUMP (go to) statements were not found in the code.
- No proper bracket coding convention is used
- In function GridView.setOnItemClickListener() of activity Seat Status, no try catch statements are used for error handling. The app may crash for some Null value of seat Selection.
- All the loops used are terminating according to their terminating conditions after performing some finite iterations.
- Proper care has been taken in Indentation while writing code There are no uninitialized variables found in the module
- Grid View has been used for displaying seats, which may create problem when number of seats are high.
- In onClick function of Layout Activity,no care has been taken for handling NULL value of variable number_of_seats.
- The headers of each module have the basic details, like - Name of the module, Date on which the module was created, Author's name, Modification history, Synopsis of the module, Different functions supported, along with their input/output Parameters.
- Many global variables have been found in various modules.
- Proper care has been taken for no internet connectivity.

4.3 Code Report by Daman Tekchandani

- There were none uninitialized variables found in any module.
- All the loops terminated according to their condition. No Non-Terminating Loops were found.
- All the array references used in the code were in the bound of the array.
- No care has been taken to clear the database.
- There is no use of Jump, goto statements in any of the modules.
- While writing code, proper indentation is followed.
- Sufficient use of try catch statements to avoid error occurrence while running the app.
- No care has been taken to clear the database.
- Grid View has been used for displaying seats, which may create problem when number of seats are high.
- No proper bracket coding convention is used
- For internet connection, proper care has been taken.
- Use of too many global variables is found in various modules

5 Conclusion

The members of the code review team submitted the reports during their final meeting with the development team. From these submitted reports, we get to know about a few logical errors that were encountered during the execution of the code inspection and were listed down.

The key errors found from the code review are as follows :-

- There is a use of too many global variables which is not a good coding practice
- The problem of using GridView . For large layouts , the View may crash or behave in an unexpected manner
- There are many functions in various modules in which Exception handling is not taken care of which may lead to crashing of the application .
- No proper bracket coding convention is used .
- No care has been taken if variable (Number of seats) is having null value .