TheMainActivity.java

```java
package com.example.baltu.myapplication.MainMenu;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.graphics.Point;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.util.Log;
import android.view.Display;
import android.view.View;
import android.view.WindowManager;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
import android.widget.Toast;
import android.widget.ViewSwitcher;
import com.example.baltu.myapplication.Buses_Schedules;
import com.example.baltu.myapplication.Academic_Calendar;
import com.example.baltu.myapplication.DB_Constructor.Local_DB_Helper;
import com.example.baltu.myapplication.DB_Constructor.User_DB_Helper;
import com.example.baltu.myapplication.GPA_Calculator;
import com.example.baltu.myapplication.Information_Activity.InfoListActivity;
import com.example.baltu.myapplication.News.News_Activity;
import com.example.baltu.myapplication.R;
import com.example.baltu.myapplication.Settings.Settings_activity;
import com.example.baltu.myapplication.Map.University_Map;
import com.example.baltu.myapplication.Map.Offline_Map_Activity;
import com.example.baltu.myapplication.TimeTables.All_TimeTables_Activity;
import com.example.tomek.notepad.MainActivity;
import com.scottyab.aescrypt.AESCrypt;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.security.GeneralSecurityException;
/*Unfortunately The code was rushed and implemented by only one student* so the focus was on t
he functionality and no focus on the documentation*/
public class TheMainActivity extends Activity {
    Point sizes = new Point();
    SQLiteDatabase userdatabase;
    SQLiteDatabase localdatabase;
    Integer Tries = 3;
    static String DB_PATH = "/data/data/com.example.baltu.myapplication/databases/";
    public static final String DB_FILENAME = "LocalDb.db";
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mainmenue2);
        findViewById(R.id.img3).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                Intent intents2 = new Intent(TheMainActivity.this, Academic_Calendar.class);
                startActivity(intents2);
            }
        });
        findViewById(R.id.img5).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            Intent calenderintents = new Intent(TheMainActivity.this, Buses_Schedules.class);
            startActivity(calenderintents);
          }
        });
        findViewById(R.id.img6).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            if (isNetworkAvailable()) {
              Intent NewsIntent = new Intent(TheMainActivity.this, News_Activity.class); //News
              startActivity(NewsIntent);
            } else Toast.makeText(TheMainActivity.this, "Not Connected To the Internet", Toast.LENGTH_SHORT).show();
          }
        });
        findViewById(R.id.img7).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            Intent NotesIntent = new Intent(TheMainActivity.this, MainActivity.class); //Notes
            startActivity(NotesIntent);
          }
        });
        findViewById(R.id.img8).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            Intent InformationIntent = new Intent(TheMainActivity.this, InfoListActivity.class);
            startActivity(InformationIntent);
          }
        });
        findViewById(R.id.img11).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            Intent SettingsIntent = new Intent(TheMainActivity.this, Settings_activity.class);
            startActivity(SettingsIntent);
          }
        });
        findViewById(R.id.img2).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            Intent TimetablesIntent = new Intent(TheMainActivity.this, All_TimeTables_Activity.class);
            startActivity(TimetablesIntent);
          }
        });
        findViewById(R.id.img9).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            Intent gpaIntent = new Intent(TheMainActivity.this, GPA_Calculator.class);
            startActivity(gpaIntent);
              }
            });
            WindowManager wm = (WindowManager) this.getSystemService(Context.WINDOW_SERVICE);
            Display display = wm.getDefaultDisplay();
            display.getSize(sizes);
            Log.d("MainMenue", "onCreate");
            final SharedPreferences prefs = getSharedPreferences(getString(R.string.preference_file_key), MODE_PRIVATE);
            boolean key = prefs.getBoolean("pass_on", false);
            final String password_string = prefs.getString("Password_String", "");
            final ViewSwitcher vs = (ViewSwitcher) findViewById(R.id.Switcher);
            if (!key) {
                    Log.d("Mainmen", "no pass");
                    vs.showNext();
                    if (checkDataBase()) {
                        SQLiteOpenHelper openhelper = new User_DB_Helper(this);
                        userdatabase = openhelper.getWritableDatabase();
                        userdatabase.close();
                        SQLiteOpenHelper localdbhelper = new Local_DB_Helper(this);
                        localdatabase = localdbhelper.getWritableDatabase();
                        localdatabase.close();
                        Toast.makeText(this, "not coppied", Toast.LENGTH_SHORT).show();
                    } else {
                        SQLiteOpenHelper openhelper = new User_DB_Helper(this);
                        userdatabase = openhelper.getWritableDatabase();
                        userdatabase.close();
                        SQLiteOpenHelper localdbhelper = new Local_DB_Helper(this);
                        localdatabase = localdbhelper.getWritableDatabase();
                        localdatabase.close();
                        try {
                            copyDataBase();
                            Toast.makeText(this, "coppied", Toast.LENGTH_SHORT).show();
                        } catch (IOException e) {
                            e.printStackTrace();
                            Toast.makeText(this, "Error", Toast.LENGTH_SHORT).show();
                        }
                    } //Toast.makeText(this, "Created DB", Toast.LENGTH_SHORT).show();
                } else {
                    findViewById(R.id.unlock_bt).setOnClickListener(new View.OnClickListener() {@
                        Override public void onClick(View v) {
                            EditText pass = (EditText) findViewById(R.id.password_box);
                            if (!pass.getText().toString().equals("")) {
                                try {
                                    if (AESCrypt.encrypt("CanTheWorldUnderstand?", "TuNiSbYnIgHt" + pass.getText().toString()).equals(password_string)) {
                                        SharedPreferences.Editor PrefE = prefs.edit();
                                        PrefE.putBoolean("Logged", true);
                                        PrefE.commit();
                                        Log.i("logged", "true");
                                        View view = TheMainActivity.this.getCurrentFocus();
                                        if (view != null) {
                                            InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
                                            imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
                                        }
                                        pass.setText("");
                                        vs.showNext();
                                    } else {
                                        View view = TheMainActivity.this.getCurrentFocus();
                                        if (view != null) {
                                            InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
                                            imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
                                        }
                                        Tries--;
                                        if (Tries == 0) finish();
                                        else {
                                            Toast.makeText(TheMainActivity.this, "    Wrong Password" + "\nYou Have " + Integer.toString(Tries) + " More Tries", Toast.LENGTH_SHORT).show();
```

```java
                                    pass.setText("");
                                }
                            }
                        } catch (GeneralSecurityException e) {
                            e.printStackTrace();
                        }
                    } else Toast.makeText(TheMainActivity.this, "Enter Password", Toast.LENGTH_SHORT).show();
                }
            });
        }
    }@
    Override protected void onStart() {
        super.onStart();
        EditText pass = (EditText) findViewById(R.id.password_box);
        pass.setText("");
    }
    public void clickonMap(View view) {
        if (isNetworkAvailable()) {
            Intent intf = new Intent(this, University_Map.class);
            intf.putExtra("type", 0);
            startActivity(intf);
        } else {
            Toast.makeText(this, "connect to the internet for an interactive map", Toast.LENGTH_LONG).show();
            Intent offlineMap = new Intent(TheMainActivity.this, Offline_Map_Activity.class);
            startActivity(offlineMap);
        }
    }
    private boolean isNetworkAvailable() {
        ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo activeNetworkInfo = connectivityManager.getActiveNetworkInfo();
        return activeNetworkInfo != null && activeNetworkInfo.isConnected();
    }
    private void copyDataBase() throws IOException {
        try {
            InputStream mInputStream = this.getAssets().open("firstdb.db");
            String outFileName = DB_PATH + DB_FILENAME;
            OutputStream mOutputStream = new FileOutputStream(outFileName);
            byte[] buffer = new byte[1024];
            int length;
            while ((length = mInputStream.read(buffer)) > 0) {
                mOutputStream.write(buffer, 0, length);
            }
            mOutputStream.flush();
            mOutputStream.close();
            mInputStream.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    private boolean checkDataBase() {
        try {
            final String mPath = DB_PATH + DB_FILENAME;
            final File file = new File(mPath);
            if (file.exists()) return true;
            else return false;
        } catch (SQLiteException e) {
            e.printStackTrace();
```

```
                    return false;
                }
            }
        }
```

All_TimeTables_Activity.java

```
package com.example.baltu.myapplication.TimeTables;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.TabHost;
import com.example.baltu.myapplication.DataTypes.Exam_class;
import com.example.baltu.myapplication.DataTypes.Tasks_Data;
import com.example.baltu.myapplication.DataTypes.TimeTable_Classes;
import com.example.baltu.myapplication.Data_Provider.UserDB_Provider;
import com.example.baltu.myapplication.Data_Provider.Adapters.CoursesItemAdapter;
import com.example.baltu.myapplication.Data_Provider.Adapters.ExamsItemsAdapter;
import com.example.baltu.myapplication.R;
import com.example.baltu.myapplication.Data_Provider.Adapters.TasksItemsAdapter;
import java.util.List;
public class All_TimeTables_Activity extends Activity {
    UserDB_Provider mDatasource;
    List < TimeTable_Classes > thisdaycourses;
    List < Tasks_Data > allTasks;
    List < Exam_class > allExams;@
    Override protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.timetables);
        mDatasource = new UserDB_Provider(this);
        mDatasource.open();
        final Spinner tabletypes = (Spinner) findViewById(R.id.tablestype);
        ArrayAdapter < CharSequence > items = ArrayAdapter.createFromResource(this, R.array.ti
metables_types, android.R.layout.simple_spinner_item);
        items.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        tabletypes.setAdapter(items);
        final TabHost th = (TabHost) findViewById(R.id.tabhost);
        th.setup();
        final TabHost.TabSpec spec1 = th.newTabSpec("Mon").setContent(R.id.tab1).setIndicator(
"Mon");
        th.addTab(spec1);
        TabHost.TabSpec spec2 = th.newTabSpec("Tue").setContent(R.id.tab2).setIndicator("Tue")
;
        th.addTab(spec2);
        TabHost.TabSpec spec3 = th.newTabSpec("Wed").setContent(R.id.tab3).setIndicator("Wed")
;
        th.addTab(spec3);
        TabHost.TabSpec spec4 = th.newTabSpec("Thu").setContent(R.id.tab4).setIndicator("Thu")
;
        th.addTab(spec4);
        TabHost.TabSpec spec5 = th.newTabSpec("Fri").setContent(R.id.tab5).setIndicator("Fri")
;
        th.addTab(spec5);
        TabHost.TabSpec spec6 = th.newTabSpec("Sat").setContent(R.id.tab6).setIndicator("Sat")
;
        th.addTab(spec6);
        final TabHost uvo = (TabHost) findViewById(R.id.upcomingvsold2);
        uvo.setup();
        final TabHost.TabSpec spec11 = uvo.newTabSpec("Upcoming").setContent(R.id.tab11).setIndicator("Upcoming");
        uvo.addTab(spec11);
        TabHost.TabSpec spec21 = uvo.newTabSpec("Old").setContent(R.id.tab21).setIndicator("Old");
        uvo.addTab(spec21);
        findViewById(R.id.addnewbt).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                if (tabletypes.getSelectedItemPosition() == 0) {
                    Intent calenderintents = new Intent(All_TimeTables_Activity.this, Courses_Viewer_Editor_Activity.class);
                    calenderintents.putExtra("day", th.getCurrentTab());
                    calenderintents.putExtra("name", "");
                    startActivity(calenderintents);
                } else if (tabletypes.getSelectedItemPosition() == 1) {
                    Intent newtask = new Intent(All_TimeTables_Activity.this, Tasks_Viewer_Editor_Activity.class);
                    startActivity(newtask);
                } else if (tabletypes.getSelectedItemPosition() == 2) {
                    Intent newexam = new Intent(All_TimeTables_Activity.this, Exams_Viewer_Editor_Activity.class);
                    startActivity(newexam);
                }
            }
        });
        thisdaycourses = mDatasource.getcourses(th.getCurrentTab());
        courseslistsetup(thisdaycourses);
        th.setOnTabChangedListener(new TabHost.OnTabChangeListener() {@
            Override public void onTabChanged(String tabId) {
                thisdaycourses = mDatasource.getcourses(th.getCurrentTab());
                courseslistsetup(thisdaycourses);
            }
        });
        uvo.setOnTabChangedListener(new TabHost.OnTabChangeListener() {@
            Override public void onTabChanged(String tabId) {
                if (tabletypes.getSelectedItemPosition() == 1) {
                    allTasks = mDatasource.GetTasks(uvo.getCurrentTab());
                    Taskslistsetup(allTasks);
                } else if (tabletypes.getSelectedItemPosition() == 2) {
                    allExams = mDatasource.GetExams(uvo.getCurrentTab());
                    Examslistsetup(allExams);
                }
            }
        });
        tabletypes.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {@
            Override public void onItemSelected(AdapterView <? > parent, View view, int position, long id) {
                View hs = findViewById(R.id.hscrollc);
                View upvol = findViewById(R.id.upcomingvsold);
                if (position != 0) {
                    hs.setVisibility(View.GONE);
                    upvol.setVisibility(View.VISIBLE);
                } else {
                    hs.setVisibility(View.VISIBLE);
                    thisdaycourses = mDatasource.getcourses(th.getCurrentTab());
                          courseslistsetup(thisdaycourses);
                          upvol.setVisibility(View.GONE);
                      }
                      if (position == 1) {
```

```java
              allTasks = mDatasource.GetTasks(uvo.getCurrentTab());
              Taskslistsetup(allTasks);
            } else if (position == 2) {
              allExams = mDatasource.GetExams(uvo.getCurrentTab());
              Examslistsetup(allExams);
            }
          }@
          Override public void onNothingSelected(AdapterView <? > parent) {
            /*long tm=new Date().getTime();                Random rand = new Random(tm);                 int   r = rand.nextInt(256);               int   g = rand.nextInt(256);               int   b = rand.nextInt(256);               th.getCurrentTabView().setBackgroundColor(Color.argb(100,r,g,b));                th.getCurrentView().setBackgroundColor(Color.argb(50,r,g,b));*/
          }
        });
      }@
      Override protected void onPause() {
        super.onPause();
        mDatasource.close();
      }@
      Override protected void onResume() {
        super.onResume();
        TabHost th = (TabHost) findViewById(R.id.tabhost);
        final TabHost uvo = (TabHost) findViewById(R.id.upcomingvsold2);
        mDatasource.open();
        Spinner tabletypes = (Spinner) findViewById(R.id.tablestype);
        if (tabletypes.getSelectedItemPosition() == 0) {
          thisdaycourses = mDatasource.getcourses(th.getCurrentTab());
          courseslistsetup(thisdaycourses);
        } else if (tabletypes.getSelectedItemPosition() == 1) {
          allTasks = mDatasource.GetTasks(uvo.getCurrentTab());
          Taskslistsetup(allTasks);
        } else {
          allExams = mDatasource.GetExams(uvo.getCurrentTab());
          Examslistsetup(allExams);
        }
      }
      private void courseslistsetup(List < TimeTable_Classes > x) {
        View v = findViewById(R.id.crecycler);
        RecyclerView crn = (RecyclerView) v;
        CoursesItemAdapter cadpter = new CoursesItemAdapter(this, x);
        crn.setAdapter(cadpter);
      }
      private void Taskslistsetup(List < Tasks_Data > x) {
        View v = findViewById(R.id.crecycler); //assert (v!=null);
        RecyclerView crn = (RecyclerView) v;
        TasksItemsAdapter cadpter = new TasksItemsAdapter(this, x);
        crn.setAdapter(cadpter);
      }
      private void Examslistsetup(List < Exam_class > x) {
        View v = findViewById(R.id.crecycler); //assert (v!=null);
        RecyclerView crn = (RecyclerView) v;
        ExamsItemsAdapter cadpter = new ExamsItemsAdapter(this, x);
        crn.setAdapter(cadpter);
      }
    }
```

Courses_Viewer_Editor_Activity.java

```java
package com.example.baltu.myapplication.TimeTables;
import android.app.Activity;
import android.app.Dialog;
import android.app.DialogFragment;
import android.app.TimePickerDialog;
import android.content.DialogInterface;
import android.database.sqlite.SQLiteException;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AlertDialog;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ViewSwitcher;
import com.example.baltu.myapplication.DataTypes.TimeTable_Classes;
import com.example.baltu.myapplication.Data_Provider.UserDB_Provider;
import com.example.baltu.myapplication.R;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
/** * Created by Baltu on 2017-05-07. */
public class Courses_Viewer_Editor_Activity extends Activity {
    private UserDB_Provider mDatasource;
    private String Id;
    private TextView vnoc;
    private TextView DofW;
    private TextView vstoc;
    private TextView vetoc;
    private TextView vnoteofc;
    private Spinner tabletypes;
    private TextView noc; //name of course
    private static TextView sth; //start time hours
    private static TextView eth; //end time hours
    private TextView nnc;
    private ViewSwitcher vsc;
    private static DateFormat timeformat = new SimpleDateFormat("HH:mm");@
    Override protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.new_course2);
        vsc = (ViewSwitcher) findViewById(R.id.ViewSwitcherCourses);
        mDatasource = new UserDB_Provider(this);
        mDatasource.open();
        vnoc = (TextView) findViewById(R.id.name); //V name of course
        DofW = (TextView) findViewById(R.id.DayOfTheWeek); //V Day of week
        vstoc = (TextView) findViewById(R.id.starttimeee); //V start time
        vetoc = (TextView) findViewById(R.id.endtimeee); //V end time
        vnoteofc = (TextView) findViewById(R.id.notesnewcourseee); //V Notes
        tabletypes = (Spinner) findViewById(R.id.daysofweek);
        noc = (TextView) findViewById(R.id.editText2); //E name
        sth = (TextView) findViewById(R.id.starttime); //E start time
        eth = (TextView) findViewById(R.id.endtime); //E end time
        nnc = (TextView) findViewById(R.id.notesnewcourse); //E Notes
        ArrayAdapter < CharSequence > items = ArrayAdapter.createFromResource(this, R.array.Da
ys_of_Week, android.R.layout.simple_spinner_item);
        items.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        tabletypes.setAdapter(items);
        if (getIntent().getStringExtra("id") == null) {
            vsc.showNext();
        } else {
            vnoc.setText(getIntent().getStringExtra("name"));
            noc.setText(getIntent().getStringExtra("name"));
            DofW.setText(items.getItem(getIntent().getIntExtra("day", 0)));
            tabletypes.setSelection(getIntent().getIntExtra("day", 0));
            vstoc.setText(getIntent().getStringExtra("start"));
            sth.setText(getIntent().getStringExtra("start"));
            vetoc.setText(getIntent().getStringExtra("end"));
            eth.setText(getIntent().getStringExtra("end"));
            vnoteofc.setText(getIntent().getStringExtra("note"));
            nnc.setText(getIntent().getStringExtra("note"));
        }
        findViewById(R.id.button5).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                vsc.showNext();
            }
        });
        findViewById(R.id.button6).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                new AlertDialog.Builder(Courses_Viewer_Editor_Activity.this).setTitle("Delete ").setMessage("Are you sure you want to delete " + vnoc.getText().toString() + "?").setIcon(android.R.drawable.ic_dialog_alert).setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int whichButton) {
                        if (mDatasource.deletecourse(getIntent().getStringExtra("id"))) {
                            Toast.makeText(Courses_Viewer_Editor_Activity.this, "Deleted", Toast.LENGTH_LONG).show();
                            finish();
                        } else {
                            Toast.makeText(Courses_Viewer_Editor_Activity.this, "Error", Toast.LENGTH_LONG).show();
                            finish();
                        }
                    }
                }).setNegativeButton(android.R.string.no, null).show();
            }
        });
        findViewById(R.id.starttimelayout).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                Bundle g = new Bundle();
                g.putString("time", sth.getText().toString());
                DialogFragment df = new timepicker();
                df.setArguments(g);
                      df.show(getFragmentManager(), "timepicker");
                  }
              });
              findViewById(R.id.endtimelayout2).setOnClickListener(new View.OnClickListener() {@
                  Override public void onClick(View v) {
                      Bundle g = new Bundle();
                      g.putString("time", eth.getText().toString());
                      DialogFragment df = new timepicker();
                      df.setArguments(g);
                      df.show(getFragmentManager(), "timepicker1");
                  }
              });
              findViewById(R.id.savecourse).setOnClickListener(new View.OnClickListener() {@

                  Override public void onClick(View v) {
                      TimeTable_Classes gh;
                      gh = new TimeTable_Classes(getIntent().getStringExtra("id"), noc.getText().toString(), sth.getText().toString(), eth.getText().toString(), tabletypes.getSelectedItemPosition(), nnc.getText().toString());
```

```java
                    if (noc.getText().toString().equals("") || (sth.getText().toString().equals("00:00") && eth.getText().toString().equals("00:00"))) {
                        Toast.makeText(Courses_Viewer_Editor_Activity.this, "Error: Complete your input", Toast.LENGTH_SHORT).show();
                    } else if (getIntent().getStringExtra("id") == null) {
                        try {
                            mDatasource.addnewcourse(gh);
                        } catch (SQLiteException e) {
                            e.printStackTrace();
                        } finally {
                            Toast.makeText(Courses_Viewer_Editor_Activity.this, "Saved", Toast.LENGTH_LONG).show();
                        }
                        finish();
                    } else {
                        try {
                            mDatasource.updatecourse(gh);
                        } catch (SQLiteException e) {
                            e.printStackTrace();
                        } finally {
                            Toast.makeText(Courses_Viewer_Editor_Activity.this, "Updated", Toast.LENGTH_SHORT).show();
                        }
                        finish();
                    }
                }
            });
        }@
        Override protected void onPause() {
            super.onPause();
            mDatasource.close();
        }
        public static class timepicker extends DialogFragment implements TimePickerDialog.OnTimeSetListener {@
            Override public Dialog onCreateDialog(Bundle savedInstanceState) {
                final Calendar c = Calendar.getInstance();
                final int Hours = c.get(Calendar.HOUR_OF_DAY);
                int Minutes = c.get(Calendar.MINUTE);
                if (sth.getText().toString().equals("00:00") && eth.getText().toString().equals("00:00")) {
                    return new TimePickerDialog(getActivity(), this, Hours, Minutes, true);

                } else {
                    DateFormat df = new SimpleDateFormat("HH:mm");
                    try {
                        return new TimePickerDialog(getActivity(), this, df.parse(getArguments().getString("time", "00:00")).getHours(), df.parse(getArguments().getString("time", "00:00")).getMinutes(), true);
                    } catch (ParseException e) {
                        e.printStackTrace();
                        return new TimePickerDialog(getActivity(), this, Hours, Minutes, true);
                    }
                }
            }@
            Override public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
                TextView sth = (TextView) getActivity().findViewById(R.id.starttime);
                TextView eth = (TextView) getActivity().findViewById(R.id.endtime);
                if (getTag() == "timepicker") {
                    sth.setText(String.valueOf(hourOfDay) + ":" + String.valueOf(minute));
```

```java
                        if (eth.getText().toString().equals("00:00")) eth.setText(String.valueOf((hourOfDay + 2) % 24) + ":" + String.valueOf(minute));
                    } else {
                        eth.setText(String.valueOf(hourOfDay) + ":" + String.valueOf(minute));

                        if (sth.getText().toString().equals("00:00")) sth.setText(String.valueOf((hourOfDay + 22) % 24) + ":" + String.valueOf(minute));
                    }
                }
            }
        }
```

Exams_Viewer_Editor_Activity.java

```java
package com.example.baltu.myapplication.TimeTables;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.sqlite.SQLiteException;
import android.os.Bundle;
import android.provider.CalendarContract;
import android.support.annotation.Nullable;
import android.support.v4.app.FragmentActivity;
import android.support.v7.app.AlertDialog;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ViewFlipper;
import com.example.baltu.myapplication.DataTypes.Exam_class;
import com.example.baltu.myapplication.Data_Provider.UserDB_Provider;
import com.example.baltu.myapplication.R;
import com.github.jjobes.slidedatetimepicker.SlideDateTimeListener;
import com.github.jjobes.slidedatetimepicker.SlideDateTimePicker;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
/** * Created by Baltu on 2017-05-09. */
public class Exams_Viewer_Editor_Activity extends FragmentActivity {
    UserDB_Provider msource;@
    Override protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.new_exam);
        final ViewFlipper vf = (ViewFlipper) findViewById(R.id.viewFlipper);
        final String id = getIntent().getStringExtra("id");
        final TextView cname2 = (TextView) findViewById(R.id.newtaskcn2); //view task name
        final TextView cdesc2 = (TextView) findViewById(R.id.newtaskscd2); // new task description
        final TextView Datet2 = (TextView) findViewById(R.id.newtaskdate2); //new task date
        final TextView ntn2 = (TextView) findViewById(R.id.newtasksnote2); //new task note
        final DateFormat all2 = new SimpleDateFormat("dd/MM/yyyy  hh:mm aaa");
        findViewById(R.id.Editebuttun2).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                vf.showNext();
            }
        });
        msource = new UserDB_Provider(this);
        msource.open();
        final Date nowd = new Date();
        final TextView cname = (TextView) findViewById(R.id.newtaskcn); //new task name
        final TextView cdesc = (TextView) findViewById(R.id.newtaskscd); // new task description
        final TextView Timet = (TextView) findViewById(R.id.newtasktime); //new task time
        final TextView Datet = (TextView) findViewById(R.id.newtaskdate); //new task date
        final TextView ntn = (TextView) findViewById(R.id.newtasksnote); //new task note
        final DateFormat tt = new SimpleDateFormat("hh:mm aaa");
        final DateFormat dd = new SimpleDateFormat("dd/MM/yyyy");
        final DateFormat all = new SimpleDateFormat("dd/MM/yyyyhh:mm aaa");
        Timet.setText(tt.format(nowd));
        Datet.setText(dd.format(nowd));
        if (id != null) {
            vf.showNext();
            cname2.setText(getIntent().getStringExtra("name"));
            cname.setText(getIntent().getStringExtra("name"));
            cdesc2.setText(getIntent().getStringExtra("desc"));
            cdesc.setText(getIntent().getStringExtra("desc"));
            Datet2.setText(all2.format(new Date(getIntent().getLongExtra("date", 0))));
            Datet.setText(dd.format(new Date(getIntent().getLongExtra("date", 0))));
            Timet.setText(tt.format(new Date(getIntent().getLongExtra("date", 0))));
            ntn2.setText(getIntent().getStringExtra("notes"));
            ntn.setText(getIntent().getStringExtra("notes"));
        }
        findViewById(R.id.newtasktime).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                SlideDateTimeListener listner = new SlideDateTimeListener() {@
                    Override public void onDateTimeSet(Date date) {
                        Timet.setText(tt.format(date));
                        Datet.setText(dd.format(date));
                    }
                };
                Date nowtime = new Date();
                try {
                    nowtime = all.parse(Datet.getText().toString() + Timet.getText().toString());
                } catch (ParseException e) {
                    e.printStackTrace();
                }
                new SlideDateTimePicker.Builder(getSupportFragmentManager()).setListener(listner).setInitialDate(nowtime).setMaxDate(new Date(nowtime.getTime() + (long) 315360000000.0)).setMinDate(new Date(nowtime.getTime() - (long) 315360000000.0)).setCurrentd_or_t(1).build().show();
            }
        });
        findViewById(R.id.alarm).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_INSERT);
                intent.setType("vnd.android.cursor.item/event");
                intent.putExtra(CalendarContract.Events.TITLE, cname2.getText().toString());
                intent.putExtra(CalendarContract.Events.DESCRIPTION, cdesc2.getText().toString());
                Date taskdate = new Date(); // Setting dates
                try {
                    taskdate = all2.parse(Datet2.getText().toString());
                } catch (ParseException e) {
                    e.printStackTrace();
                }
                intent.putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, taskdate.getTime());

                intent.putExtra(CalendarContract.EXTRA_EVENT_END_TIME, taskdate.getTime());
                startActivity(intent); // make it a full day event // intent.putExtra(CalendarContract.EXTRA_EVENT_ALL_DAY, true); // make it a recurring Event //intent.putExtra(CalendarContract.Events.RRULE, "FREQ=WEEKLY;COUNT=11;WKST=SU;BYDAY=TU,TH"); // Making it private and shown as busy // intent.putExtra(CalendarContract.Events.ACCESS_LEVEL, CalendarContract.Events.AC
CESS_PRIVATE); // intent.putExtra(CalendarContract.Events.AVAILABILITY, CalendarContract.Events.AVAILABILITY_BUSY);
      }
    });
        findViewById(R.id.newtaskdate).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            SlideDateTimeListener listner = new SlideDateTimeListener() {@
              Override public void onDateTimeSet(Date date) {
                Timet.setText(tt.format(date));
                Datet.setText(dd.format(date));
              }
            };
            Date nowtime = new Date();
            try {
              nowtime = all.parse(Datet.getText().toString() + Timet.getText().toString());
            } catch (ParseException e) {
              e.printStackTrace();
            }
            new SlideDateTimePicker.Builder(getSupportFragmentManager()).setListener(listner).setInitialDate(nowtime).setMaxDate(new Date(nowtime.getTime() + (long) 315360000000.0)).setMinDate(new Date(nowtime.getTime() - (long) 315360000000.0)).setCurrentd_or_t(0).build().show();
          }
        });
        findViewById(R.id.newtasksaveb).setOnClickListener(new View.OnClickListener() {@
          Override public void onClick(View v) {
            Exam_class gh = new Exam_class();
            Date isitold = new Date();
            try {
              isitold = all.parse(Datet.getText().toString() + Timet.getText().toString());
            } catch (ParseException e) {
              e.printStackTrace();
            }
            if (!cname.getText().toString().isEmpty() && isitold.getTime() > new Date().getTime()) {
              gh.setCourse(cname.getText().toString());
              gh.setExam_Date(isitold.getTime());
              gh.setDiscription(cdesc.getText().toString());
              gh.setNotes(ntn.getText().toString());
              if (id == null) {
                try {
                  msource.addnewExam(gh);
                } catch (SQLiteException e) {
                  e.printStackTrace();
                }
                Toast.makeText(Exams_Viewer_Editor_Activity.this, "saved", Toast.LENGTH_LONG).show();
                finish();
              } else {
                gh.setID(id);
                try {
                  msource.updateExam(gh);
                } catch (SQLiteException e) {
                  e.printStackTrace();
                }
                Toast.makeText(Exams_Viewer_Editor_Activity.this, "updated", Toast.LENGTH_LONG).show();
```

```java
                                finish();
                            }
                        } else {
                            Toast.makeText(Exams_Viewer_Editor_Activity.this, "Error\ncheck your Inputs", Toast.LENGTH_LONG).show();
                        }
                    }
                });
                findViewById(R.id.deleteb).setOnClickListener(new View.OnClickListener() {@
                    Override public void onClick(View v) {
                        new AlertDialog.Builder(Exams_Viewer_Editor_Activity.this).setTitle("Delete ").setMessage("Are you sure you want to delete " + cname2.getText().toString() + "?").setIcon(android.R.drawable.ic_dialog_alert).setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog, int whichButton) {
                                if (msource.DeleteExam(id) == 1) {
                                    Toast.makeText(Exams_Viewer_Editor_Activity.this, "Deleted", Toast.LENGTH_LONG).show();
                                    finish();
                                } else {
                                    Toast.makeText(Exams_Viewer_Editor_Activity.this, "Error", Toast.LENGTH_LONG).show();
                                    finish();
                                }
                            }
                        }).setNegativeButton(android.R.string.no, null).show();
                    }
                });
            }@
            Override protected void onPause() {
                super.onPause();
                msource.close();
            }
        }
```

Tasks_Viewer_Editor_Activity.java

```java
package com.example.baltu.myapplication.TimeTables;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.sqlite.SQLiteException;
import android.os.Bundle;
import android.provider.CalendarContract;
import android.support.annotation.Nullable;
import android.support.v4.app.FragmentActivity;
import android.support.v7.app.AlertDialog;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ViewFlipper;
import com.example.baltu.myapplication.DataTypes.Tasks_Data;
import com.example.baltu.myapplication.Data_Provider.UserDB_Provider;
import com.example.baltu.myapplication.R;
import com.github.jjobes.slidedatetimepicker.SlideDateTimeListener;
import com.github.jjobes.slidedatetimepicker.SlideDateTimePicker;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
/** * Created by Baltu on 2017-05-09. */
public class Tasks_Viewer_Editor_Activity extends FragmentActivity {
    UserDB_Provider msource;@
    Override protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.newtasks);
        final ViewFlipper vf = (ViewFlipper) findViewById(R.id.viewFlipper);
        final String id = getIntent().getStringExtra("id");
        final TextView cname2 = (TextView) findViewById(R.id.newtaskcn2); //view task name
        final TextView cdesc2 = (TextView) findViewById(R.id.newtaskscd2); // new task description
        final TextView Datet2 = (TextView) findViewById(R.id.newtaskdate2); //new task date
        final TextView ntn2 = (TextView) findViewById(R.id.newtasksnote2); //new task note
        final DateFormat all2 = new SimpleDateFormat("dd/MM/yyyy  hh:mm aaa");
        findViewById(R.id.Editebuttun2).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                vf.showNext();
            }
        });
        findViewById(R.id.alarm).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_INSERT);
                intent.setType("vnd.android.cursor.item/event");
                intent.putExtra(CalendarContract.Events.TITLE, cname2.getText().toString());
                intent.putExtra(CalendarContract.Events.DESCRIPTION, cdesc2.getText().toString());
                Date taskdate = new Date(); // Setting dates
                try {
                    taskdate = all2.parse(Datet2.getText().toString());
                } catch (ParseException e) {
                    e.printStackTrace();
                }
                intent.putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, taskdate.getTime());

                intent.putExtra(CalendarContract.EXTRA_EVENT_END_TIME, taskdate.getTime());
                startActivity(intent); // make it a full day event // intent.putExtra(CalendarContract.EXTRA_EVENT_ALL_DAY, true); // make it a recurring Event //intent.putExtra(CalendarContract.Events.RRULE, "FREQ=WEEKLY;COUNT=11;WKST=SU;BYDAY=TU,TH"); // Making it private and shown as busy // intent.putExtra(CalendarContract.Events.ACCESS_LEVEL, CalendarContract.Events.ACCESS_PRIVATE); // intent.putExtra(CalendarContract.Events.AVAILABILITY, CalendarContract.Events.AVAILABILITY_BUSY);
            }
        });
        msource = new UserDB_Provider(this);
        msource.open();
        final Date nowd = new Date();
        final TextView cname = (TextView) findViewById(R.id.newtaskcn); //new task name
        final TextView cdesc = (TextView) findViewById(R.id.newtaskscd); // new task description
        final TextView Timet = (TextView) findViewById(R.id.newtaskdate); //new task time
        final TextView Datet = (TextView) findViewById(R.id.newtasktime); //new task date
        final TextView ntn = (TextView) findViewById(R.id.newtasksnote); //new task note
        final DateFormat tt = new SimpleDateFormat("hh:mm aaa");
        final DateFormat dd = new SimpleDateFormat("dd/MM/yyyy");
        final DateFormat all = new SimpleDateFormat("dd/MM/yyyyhh:mm aaa");
        Timet.setText(tt.format(nowd));
        Datet.setText(dd.format(nowd));
        if (id != null) {
            vf.showNext();
            cdesc2.setText(getIntent().getStringExtra("desc"));
            cdesc.setText(getIntent().getStringExtra("desc"));
            Datet2.setText(all2.format(new Date(getIntent().getLongExtra("date", 0))));
            Datet.setText(dd.format(new Date(getIntent().getLongExtra("date", 0))));
            Timet.setText(tt.format(new Date(getIntent().getLongExtra("date", 0))));
            ntn2.setText(getIntent().getStringExtra("notes"));
            ntn.setText(getIntent().getStringExtra("notes"));
            cname2.setText(getIntent().getStringExtra("name"));
            cname.setText(getIntent().getStringExtra("name"));
        }
        findViewById(R.id.newtasktime).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                SlideDateTimeListener listner = new SlideDateTimeListener() {@
                    Override public void onDateTimeSet(Date date) {
                        Timet.setText(tt.format(date));
                        Datet.setText(dd.format(date));
                    }
                };
                Date nowtime = new Date();
                try {
                    nowtime = all.parse(Datet.getText().toString() + Timet.getText().toString());
                } catch (ParseException e) {
                    e.printStackTrace();
                }
                new SlideDateTimePicker.Builder(getSupportFragmentManager()).setListener(listner).setInitialDate(nowtime).setMaxDate(new Date(nowtime.getTime() + (long) 315360000000.0)).setMinDate(new Date(nowtime.getTime() - (long) 315360000000.0)).setCurrentd_or_t(0).build().show();
            }
        });
              findViewById(R.id.newtaskdate).setOnClickListener(new View.OnClickListener() {@

                  Override public void onClick(View v) {
                      SlideDateTimeListener listner = new SlideDateTimeListener() {@
                          Override public void onDateTimeSet(Date date) {
                              Timet.setText(tt.format(date));
                              Datet.setText(dd.format(date));
                          }
                      };
                      Date nowtime = new Date();
                      try {
                          nowtime = all.parse(Datet.getText().toString() + Timet.getText().toString());
                      } catch (ParseException e) {
                          e.printStackTrace();
                      }
                      new SlideDateTimePicker.Builder(getSupportFragmentManager()).setListener(listner).setInitialDate(nowtime).setMaxDate(new Date(nowtime.getTime() + (long) 315360000000.0)).setMinDate(new Date(nowtime.getTime() - (long) 315360000000.0)).setCurrentd_or_t(1).build().show();
                  }
              });
              findViewById(R.id.newtasksaveb).setOnClickListener(new View.OnClickListener() {@
                  Override public void onClick(View v) {
                      Tasks_Data gh = new Tasks_Data();
                      Date isitold = new Date();
                      try {
                          isitold = all.parse(Datet.getText().toString() + Timet.getText().toString());
                      } catch (ParseException e) {
                          e.printStackTrace();
                      }
                      if (!cname.getText().toString().isEmpty() && isitold.getTime() > new Date().getTime()) {
                          gh.setDeadline_Date(isitold.getTime());
                          gh.setDiscription(cdesc.getText().toString());
                          gh.setNotes(ntn.getText().toString());
                          gh.setCourse(cname.getText().toString());
                          if (id == null) {
                              try {
                                  msource.addnewTask(gh);
                              } catch (SQLiteException e) {
                                  e.printStackTrace();
                              }
                              Toast.makeText(Tasks_Viewer_Editor_Activity.this, "saved", Toast.LENGTH_LONG).show();
                              finish();
                          } else {
                              gh.setID(id);
                              try {
                                  msource.updateTask(gh);
                              } catch (SQLiteException e) {
                                  e.printStackTrace();
                              }
                              Toast.makeText(Tasks_Viewer_Editor_Activity.this, "updated", Toast.LENGTH_LONG).show();
                              finish();
                          }
                      } else {
                          Toast.makeText(Tasks_Viewer_Editor_Activity.this, "Error\ncheck your Inputs", Toast.LENGTH_LONG).show();
                      }
                  }
              });
              findViewById(R.id.deleteb).setOnClickListener(new View.OnClickListener() {@
                  Override public void onClick(View v) {
                      new AlertDialog.Builder(Tasks_Viewer_Editor_Activity.this).setTitle("Delete ").setMessage("Are you sure you want to delete " + cname2.getText().toString() + "?").setIcon(android.R.drawable.ic_dialog_alert).setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
                          public void onClick(DialogInterface dialog, int whichButton) {
                              if (msource.DeleteTask(id) == 1) {
                                  Toast.makeText(Tasks_Viewer_Editor_Activity.this, "Deleted", Toast.LENGTH_LONG).show();
                                  finish();
                              } else {
                                  Toast.makeText(Tasks_Viewer_Editor_Activity.this, "Error", Toast.LENGTH_LONG).show();
                                  finish();
                              }
                          }
                      }).setNegativeButton(android.R.string.no, null).show();
                  }
              });
          }@
          Override protected void onPause() {
              super.onPause();
              msource.close();
          }
      }
```

Academic_Calendar.java

```
package com.example.baltu.myapplication;
import android.app.Activity;
import android.app.Fragment;
```

```java
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.view.View;
import com.example.baltu.myapplication.DataTypes.Calender_Data;
import com.example.baltu.myapplication.Data_Provider.Adapters.CalendarItemsAdapter;
import com.example.baltu.myapplication.Data_Provider.Local_DB_Main_Provider;
import java.util.List;
public class Academic_Calendar extends Activity {
    Local_DB_Main_Provider mSource;
    List < Calender_Data > allevents;@
    Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mSource = new Local_DB_Main_Provider(this);
        setContentView(R.layout.activity_calender);
        RecyclerView crv = (RecyclerView) findViewById(R.id.calendar_recycler);
        allevents = mSource.GetCalendarItems();
        CalendarItemsAdapter cda = new CalendarItemsAdapter(this, allevents);
        crv.setAdapter(cda);
    }
}
```

University_Map.java

```java
package com.example.baltu.myapplication.Map;
import android.content.Intent;
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.support.v4.widget.DrawerLayout;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import com.example.baltu.myapplication.DataTypes.University_Locations;
import com.example.baltu.myapplication.Data_Provider.Local_DB_Main_Provider;
import com.example.baltu.myapplication.Data_Provider.JsonHelper;
import com.example.baltu.myapplication.R;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.LatLngBounds;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.maps.android.data.Feature;
import com.google.maps.android.data.kml.KmlLayer;
import org.xmlpull.v1.XmlPullParserException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
public class University_Map extends FragmentActivity implements OnMapReadyCallback {
    private Map < String, Places_on_map > all_places = new HashMap < > ();
    private Map < String, University_Locations > all_Locations = new HashMap < > ();
    private static final String File_Name = "mapplaces.json";
    private GoogleMap mMap;
    static final int MY_REQUEST_CODE = 5584;
    private String[] mPlanetTitles;
    private DrawerLayout mDrawerLayout;
    private ListView mDrawerList;
    private FloatingActionButton myfab;
    private Local_DB_Main_Provider mSource;;@
    Override protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_un__map);
            mSource = new Local_DB_Main_Provider(this);
            final List < University_Locations > AllLocations = mSource.GetMapLocations();
            mPlanetTitles = getResources().getStringArray(R.array.Credits);
            mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
            mDrawerList = (ListView) findViewById(R.id.left_drawer);
            myfab = (FloatingActionButton) findViewById(R.id.floatingActionButtonmap);
            mDrawerList.setAdapter(new ArrayAdapter < String > (this, android.R.layout.simple_list_item_1, mPlanetTitles));
            mDrawerList.setOnItemClickListener(new AdapterView.OnItemClickListener() {@
                Override public void onItemClick(AdapterView <? > parent, View view, int position, long id) {
                    University_Locations item = AllLocations.get(position);
                    Intent loc_info = new Intent(University_Map.this, Locations_on_Map_Details.class);
                    loc_info.putExtra("Location", item.getShortName());
                    loc_info.putExtra("Full_Name", item.getLongName());
                    loc_info.putExtra("url", item.getImage_Url());
                    loc_info.putExtra("Info", item.getInformation());
                    loc_info.putExtra("Longt", item.getLongitude());
                    loc_info.putExtra("Lat", item.getLatitude());
                    mDrawerLayout.closeDrawers();
                    startActivityForResult(loc_info, MY_REQUEST_CODE);
                }
            });
            myfab.setOnClickListener(new View.OnClickListener() {@
                Override public void onClick(View v) {
                    mDrawerLayout.openDrawer(mDrawerList);
                }
            });
            List < String > ListString = new ArrayList < > ();
            if (AllLocations != null) {
                for (University_Locations n: AllLocations) {
                    ListString.add(n.getLongName());
                    all_Locations.put(n.getShortName(), n);
                }
                mDrawerList.setAdapter(new ArrayAdapter < String > (this, android.R.layout.simple_list_item_1, ListString));
            }
            List < Places_on_map > myplaces = JsonHelper.Importer(University_Map.this);
            if (myplaces != null) {
                for (Places_on_map pl: myplaces) {
                    all_places.put(pl.getPlace_Name(), pl);
                }
            } // Obtain the SupportMapFragment and get notified when the map is ready to be used.
            SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
            mapFragment.getMapAsync(this);
        }
        /**     * Manipulates the map once available.     * This callback is triggered when the map is ready to be used.     * This is where we can add markers or lines, add listeners or move the camera. In this case,     * we just add a marker near Sydney, Australia.     * If Google Play services is not installed on the device, the user will be prompted to install     * it
```

```
 inside the SupportMapFragment. This method will only be triggered once the user has     * installed Google Play services and returned to the app.     */
        @
    Override public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        int maptype = getIntent().getIntExtra("type", 0);
        try {
            if (maptype == 0) {
                KmlLayer layer = new KmlLayer(mMap, R.raw.campusmap, getApplicationContext());

                layer.addLayerToMap();
                layer.setOnFeatureClickListener(new KmlLayer.OnFeatureClickListener() {@
                    Override public void onFeatureClick(Feature feature) {
                        String namec = null;
                        try {
                                    namec = feature.getProperty("name").toString();
                                } catch (Exception e) {
                                    e.printStackTrace();
                                }
                                if (all_Locations.containsKey(namec)) {
                                    Intent loc_info = new Intent(University_Map.this, Locations_on_Map_Details.class);
                                    loc_info.putExtra("Location", all_Locations.get(namec).getShortName());
                                    loc_info.putExtra("Full_Name", all_Locations.get(namec).getLongName());
                                    loc_info.putExtra("url", all_Locations.get(namec).getImage_Url());
                                    loc_info.putExtra("Info", all_Locations.get(namec).getInformation());
                                    loc_info.putExtra("Longt", all_Locations.get(namec).getLongitude());
                                    loc_info.putExtra("Lat", all_Locations.get(namec).getLatitude());
                                    startActivityForResult(loc_info, MY_REQUEST_CODE);
                                } else if (all_places.containsKey(namec)) {
                                    Intent loc_info = new Intent(University_Map.this, Locations_on_Map_Details.class);
                                    loc_info.putExtra("Location", namec);
                                    loc_info.putExtra("Full_Name", all_places.get(namec).getFull_Name());
                                    loc_info.putExtra("imagename", all_places.get(namec).getImage_Name());
                                    startActivityForResult(loc_info, MY_REQUEST_CODE);
                                }
                            }
                        });
                    } else {
                        int line = getIntent().getIntExtra("line", 0);
                        switch (line) {
                            case 0:
                                KmlLayer layer = new KmlLayer(mMap, R.raw.emubusall, getApplicationContext());
                                layer.addLayerToMap();
                                layer.setOnFeatureClickListener(new KmlLayer.OnFeatureClickListener() {@
                                    Override public void onFeatureClick(Feature feature) {
                                        String namec = feature.getProperty("name").toString();

                                    }
                                });
```

```java
                                break;
                            case 1:
                                KmlLayer layer2 = new KmlLayer(mMap, R.raw.emubus1, getApplicationContext());
                                layer2.addLayerToMap();
                                layer2.setOnFeatureClickListener(new KmlLayer.OnFeatureClickListener() {@
                                    Override public void onFeatureClick(Feature feature) {
                                        String namec = feature.getProperty("name").toString();

                                    }
                                });
                                break;
                            case 2:
                                KmlLayer layer3 = new KmlLayer(mMap, R.raw.emubus2, getApplicationContext());
                                layer3.addLayerToMap();
                                layer3.setOnFeatureClickListener(new KmlLayer.OnFeatureClickListener() {@
                                    Override public void onFeatureClick(Feature feature) {
                                        String namec = feature.getProperty("name").toString();

                                    }
                                });
                                break;
                            case 3:
                                KmlLayer layer4 = new KmlLayer(mMap, R.raw.emubus3, getApplicationContext());
                                layer4.addLayerToMap();
                                layer4.setOnFeatureClickListener(new KmlLayer.OnFeatureClickListener() {@
                                    Override public void onFeatureClick(Feature feature) {
                                        String namec = feature.getProperty("name").toString();

                                    }
                                });
                                break;
                            case 4:
                                KmlLayer layer5 = new KmlLayer(mMap, R.raw.emubus4, getApplicationContext());
                                layer5.addLayerToMap();
                                layer5.setOnFeatureClickListener(new KmlLayer.OnFeatureClickListener() {@
                                    Override public void onFeatureClick(Feature feature) {
                                        String namec = feature.getProperty("name").toString();

                                    }
                                });
                                break;
                            case 5:
                                KmlLayer layer6 = new KmlLayer(mMap, R.raw.emubus5, getApplicationContext());
                                layer6.addLayerToMap();
                                break;
                        }
                    }
                } catch (XmlPullParserException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                LatLng emu = new LatLng(35.14328, 33.9102722);
                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(emu, 16));
                mMap.setMinZoomPreference(14);
                LatLng emua = new LatLng(35.0749442, 33.8914845); //SW
                LatLng emub = new LatLng(35.2309717, 33.9522488); //NE
                LatLngBounds emuc = new LatLngBounds(emua, emub);
                mMap.setLatLngBoundsForCameraTarget(emuc);
            }@
            Override protected void onActivityResult(int requestCode, int resultCode, Intent da
ta) {
                super.onActivityResult(requestCode, resultCode, data);
                if (requestCode == resultCode) {
                    MarkerOptions g = new MarkerOptions();
                    g.position(new LatLng(data.getDoubleExtra("Lati", 33.9085024), data.getDoub
leExtra("longt", 35.1460493)));
                    g.title(data.getStringExtra("Title"));
                    mMap.addMarker(g);
                    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(g.getPosition(), 16));
                }
            }
        }
```

Offline_Map_Activity.java

```
package com.example.baltu.myapplication.Map;
import android.app.Activity;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import com.example.baltu.myapplication.R;
import com.github.chrisbanes.photoview.PhotoView;
/** * Created by Baltu on 2017-04-05. */
public class Offline_Map_Activity extends Activity {@
    Override protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.offlinemap);
        PhotoView gh = (PhotoView) findViewById(R.id.zoomablemap);
        gh.setMaximumScale(6);
        gh.setMediumScale((float) 3);
    }
}
Places_on_map.java package com.example.baltu.myapplication.Map;
/** * Created by Baltu on 2017-04-03. */
public class Places_on_map {
    private String Place_Name;
    private String Full_Name;
    private String Image_Name;
    public String getPlace_Name() {
        return Place_Name;
    }
    public String getFull_Name() {
        return Full_Name;
    }
    public Places_on_map() {}
    public void setPlace_Name(String place_Name) {
        Place_Name = place_Name;
    }
    public void setFull_Name(String full_Name) {
        Full_Name = full_Name;
    }
    public String getImage_Name() {
        return Image_Name;
```

```java
    }
    public void setImage_Name(String image_Name) {
        Image_Name = image_Name;
    }
    public Places_on_map(String place_Name, String full_Name, String image_Name) {
        Place_Name = place_Name;
        Full_Name = full_Name;
        Image_Name = image_Name;
    }
}
```

Locations_on_Map_Details.java

```java
package com.example.baltu.myapplication.Map;
import android.app.Activity;
import android.app.Instrumentation;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.Drawable;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.util.Log;
import android.view.Display;
import android.view.View;
import android.webkit.WebView;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.example.baltu.myapplication.R;
import java.io.IOException;
import java.io.InputStream;
/** * Created by Baltu on 2017-04-09. */
public class Locations_on_Map_Details extends Activity {
    String url;
    private String name;
    private Double longt;
    private Double Lati;
    private ImageView mv;@
    Override protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map_location);
        Intent k = getIntent();
        name = k.getStringExtra("Full_Name");
        url = k.getStringExtra("url");
        if (url != null) {
            AsyncTask sk = new DownloadImageTask(((ImageView) findViewById(R.id.Location_Image
)));
            sk.execute(new String[] {
                url
            });
        } else {
            String ImageS = getIntent().getStringExtra("imagename");
            if (ImageS != null) try {
                InputStream imagei = getAssets().open("images/" + ImageS);
                Drawable imagef = Drawable.createFromStream(imagei, null);
                ((ImageView) findViewById(R.id.Location_Image)).setImageDrawable(imagef);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }((TextView) findViewById(R.id.Place_full_name)).setText(name);
        ((TextView) findViewById(R.id.Place_Info)).setText(getIntent().getStringExtra("Info"))
;
        longt = k.getDoubleExtra("Longt", 33.9085024);
        Lati = k.getDoubleExtra("Lat", 35.1460493);
        findViewById(R.id.Cloase_Location).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                finish();
            }
        });
        findViewById(R.id.Location_Location).setOnClickListener(new View.OnClickListener() {@

            Override public void onClick(View v) {
                Intent data = new Intent();
                data.putExtra("longt", longt);
                data.putExtra("Lati", Lati);
                data.putExtra("Title", name);
                setResult(5584, data);
                finish();
            }
        });
    }
    private class DownloadImageTask extends AsyncTask < String, Void, Bitmap > {
        ImageView mMyImageView;
        public DownloadImageTask(ImageView MyImageView) {
            this.mMyImageView = MyImageView;
        }
        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) {
            mMyImageView.setImageBitmap(result);
        }
    }
}
```

Buses_Schedules.java

```
package com.example.baltu.myapplication;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.RecyclerView;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.TabHost;
import android.widget.ViewAnimator;
import com.example.baltu.myapplication.Data_Provider.Adapters.BusItemsAdapter;
import com.example.baltu.myapplication.DataTypes.Bus_Data;
import com.example.baltu.myapplication.Data_Provider.Adapters.BusItemsAdapterweekend;
import com.example.baltu.myapplication.Data_Provider.Local_DB_Main_Provider;
```

```java
import com.github.jjobes.slidedatetimepicker.TimeFragment;
import java.util.List;
import java.util.Timer;
import java.util.TimerTask;
public class Buses_Schedules extends AppCompatActivity {
    Local_DB_Main_Provider mSource;
    List < Bus_Data > allbuses;
    List < Bus_Data > allbuses2;@
    Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bus_sche);
        mSource = new Local_DB_Main_Provider(this);
        final Spinner lineNumber = (Spinner) findViewById(R.id.buses_lines);
        ArrayAdapter < CharSequence > items = ArrayAdapter.createFromResource(this, R.array.buses_lines, android.R.layout.simple_spinner_item);
        items.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        lineNumber.setAdapter(items);
        final TabHost th = (TabHost) findViewById(R.id.To_From_Tab);
        th.setup();
        final TabHost.TabSpec spec1 = th.newTabSpec("To").setContent(R.id.tab1).setIndicator("To");
        th.addTab(spec1);
        TabHost.TabSpec spec2 = th.newTabSpec("From").setContent(R.id.tab2).setIndicator("From");
        th.addTab(spec2);
        final ViewAnimator ani = (ViewAnimator) findViewById(R.id.viewanimator);
        final ViewAnimator anit = (ViewAnimator) findViewById(R.id.busestitle);
        View leftb = findViewById(R.id.leftArrow);
        View rightb = findViewById(R.id.rightArrow);
        allbuses = mSource.GetAllBus(th.getCurrentTab(), 1);
        allbuses2 = mSource.GetAllBus(th.getCurrentTab(), 0);
        Buseslistsetup(allbuses, allbuses2);
        lineNumber.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {@
            Override public void onItemSelected(AdapterView <? > parent, View view, int position, long id) {
                if (position == 0) {
                    allbuses = mSource.GetAllBus((th.getCurrentTab() + 1) % 2, 1);
                    allbuses2 = mSource.GetAllBus((th.getCurrentTab() + 1) % 2, 0);
                    Buseslistsetup(allbuses, allbuses2);
                } else {
                    allbuses = mSource.Get1lineBus(position, (th.getCurrentTab() + 1) % 2, 1);

                    allbuses2 = mSource.Get1lineBus(position, (th.getCurrentTab() + 1) % 2, 0);
                    Buseslistsetup(allbuses, allbuses2);
                }
            }@
            Override public void onNothingSelected(AdapterView <? > parent) {}
        });
        th.setOnTabChangedListener(new TabHost.OnTabChangeListener() {@
            Override public void onTabChanged(String tabId) {
                if (lineNumber.getSelectedItemPosition() == 0) {
                    allbuses = mSource.GetAllBus((th.getCurrentTab() + 1) % 2, 1);
                    allbuses2 = mSource.GetAllBus((th.getCurrentTab() + 1) % 2, 0);
                    Buseslistsetup(allbuses, allbuses2);
                } else {
                    allbuses = mSource.Get1lineBus(lineNumber.getSelectedItemPosition(), (th.getCurrentTab() + 1) % 2, 1);
                    allbuses2 = mSource.Get1lineBus(lineNumber.getSelectedItemPosition(), (th.getCurrentTab() + 1) % 2, 0);
                    Buseslistsetup(allbuses, allbuses2);
                }
            }
        });
        leftb.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                ani.showPrevious();
                anit.showPrevious();
            }
        });
        rightb.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                ani.showNext();
                anit.showNext();
            }
        });
    }
    private void Buseslistsetup(List < Bus_Data > x, List < Bus_Data > y) {
        View v = findViewById(R.id.buscyc1);
        View v2 = findViewById(R.id.buscyc2); //assert (v!=null);
        RecyclerView crn = (RecyclerView) v;
        RecyclerView crn2 = (RecyclerView) v2;
        BusItemsAdapter cadpter = new BusItemsAdapter(x);
        BusItemsAdapterweekend cadpter2 = new BusItemsAdapterweekend(y);
        crn.setAdapter(cadpter);
        crn2.setAdapter(cadpter2);
    }
}
```

News_Activity.java

```java
package com.example.baltu.myapplication.News;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.RecyclerView;
import android.widget.TabHost;
import com.example.baltu.myapplication.DataTypes.News_Class;
import com.example.baltu.myapplication.R;
import org.xmlpull.v1.XmlPullParserException;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.List;
public class News_Activity extends AppCompatActivity {
    String urlString = "http://ww1.emu.edu.tr/rss/en/21";
    String EventsurlString = "http://ww1.emu.edu.tr/rss/en/33";
    String AnnurlString = "http://ww1.emu.edu.tr/rss/en/32";
    List < News_Class > news;
    List < News_Class > Events;
    List < News_Class > Announcments;
    InputStream stream;
    InputStream stream2;
    InputStream stream3;@
    Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_news);
        TabHost NTH = (TabHost) findViewById(R.id.news_tab);
        NTH.setup();
        final TabHost.TabSpec spec1 = NTH.newTabSpec("News").setContent(R.id.tab1).setIndicato
r("News");
        NTH.addTab(spec1);
        TabHost.TabSpec spec2 = NTH.newTabSpec("Events").setContent(R.id.tab2).setIndicator("Events");
        NTH.addTab(spec2);
        TabHost.TabSpec spec3 = NTH.newTabSpec("Announ").setContent(R.id.tab3).setIndicator("Announ.");
        NTH.addTab(spec3);
        RSSFEED d = new RSSFEED();
        d.execute(new String[] {
            urlString, EventsurlString, AnnurlString
        });
        if (news != null) {
            RecyclerView nr = (RecyclerView) findViewById(R.id.news_recycler);
            NewsItemsAdapter NRA = new NewsItemsAdapter(this, news);
            nr.setAdapter(NRA);
        }
    }
    private class RSSFEED extends AsyncTask < String, Void, String > {@
        Override protected String doInBackground(String...urls) {
            try {
                stream = downloadUrl(urls[0]);
                news = NewsRssParser.parse(stream);
                stream2 = downloadUrl(urls[1]);
                Events = NewsRssParser.parse(stream2);
                stream3 = downloadUrl(urls[2]);
                Announcments = NewsRssParser.parse(stream3);
            } catch (XmlPullParserException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            if (stream != null) {
                try {
                    stream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            if (stream2 != null) {
                try {
                    stream2.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            if (stream3 != null) {
                try {
                    stream3.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            return "y";
        }@
        Override protected void onPostExecute(String s) {
            super.onPostExecute(s);
            if (news != null) {
                RecyclerView nr = (RecyclerView) findViewById(R.id.news_recycler);
                NewsItemsAdapter NRA = new NewsItemsAdapter(News_Activity.this, news);
                nr.setAdapter(NRA);
                RecyclerView nr2 = (RecyclerView) findViewById(R.id.Events_recycler);
                NewsItemsAdapter NRA2 = new NewsItemsAdapter(News_Activity.this, Events);
```

```
                    nr2.setAdapter(NRA2);
                    RecyclerView nr3 = (RecyclerView) findViewById(R.id.Anno_recycler);
                    NewsItemsAdapter NRA3 = new NewsItemsAdapter(News_Activity.this, Announcments)
;
                    nr3.setAdapter(NRA3);
                }
            }
        }
    private InputStream downloadUrl(String urlString) throws IOException {
        URL url = new URL(urlString);
               HttpURLConnection conn = (HttpURLConnection) url.openConnection();
               conn.setReadTimeout(10000 /* milliseconds */ );
               conn.setConnectTimeout(15000 /* milliseconds */ );
               conn.setRequestMethod("GET");
               conn.setDoInput(true); // Starts the query
               conn.connect();
               return conn.getInputStream();
           }
       }
```

NewsRssParser.java

```
package com.example.baltu.myapplication.News;
import android.util.Xml;
import com.example.baltu.myapplication.DataTypes.News_Class;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import java.io.IOException;
import java.io.InputStream;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
/** * Created by Baltu on 2017-05-19. */
public class NewsRssParser {
    private static final String ns = null;
    private static final String ns2 = null;
    public static List parse(InputStream in ) throws XmlPullParserException, IOException {
        try {
            XmlPullParser parser = Xml.newPullParser();
            parser.setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES, false);
            parser.setInput( in , null);
            parser.nextTag();
            return readRSS(parser);
        } finally { in .close();
        }
    }
    private static List readRSS(XmlPullParser parser) throws XmlPullParserException, IOExcepti
on {
        List entries = new ArrayList();
        parser.require(XmlPullParser.START_TAG, ns, "rss");
        while (parser.next() != XmlPullParser.END_TAG) {
            if (parser.getEventType() != XmlPullParser.START_TAG) {
                continue;
            }
            String name = parser.getName();
            if (name.equals("item")) {
                entries.add(readItem(parser));
            } else if (name.equals("channel")) {
                continue;
```

```java
            } else {
                skip(parser);
            }
        }
        return entries;
    }
    private static News_Class readItem(XmlPullParser parser) throws XmlPullParserException, IOException {
        parser.require(XmlPullParser.START_TAG, ns, "item");
        String Title = null;
        String Link = null;
        String Description = null;
        Date Date = null;
        while (parser.next() != XmlPullParser.END_TAG) {
            if (parser.getEventType() != XmlPullParser.START_TAG) {
                continue;
            }
            String name = parser.getName();
            if (name.equals("title")) {
                Title = readTitle(parser);
            } else if (name.equals("link")) {
                Link = readLink(parser);
            } else if (name.equals("description")) {
                Description = readDes(parser);
            } else if (name.equals("pubDate")) {
                Date = readDate(parser);
            } else {
                skip(parser);
            }
        }
        return new News_Class(Title, Link, Description, Date);
    }
    private static String readTitle(XmlPullParser parser) throws IOException, XmlPullParserException {
        parser.require(XmlPullParser.START_TAG, ns, "title");
        String title = readText(parser);
        parser.require(XmlPullParser.END_TAG, ns, "title");
        return title;
    }
    private static String readLink(XmlPullParser parser) throws IOException, XmlPullParserException {
        parser.require(XmlPullParser.START_TAG, ns, "link");
        String title = readText(parser);
        parser.require(XmlPullParser.END_TAG, ns, "link");
        return title;
    }
    private static String readDes(XmlPullParser parser) throws IOException, XmlPullParserException {
        parser.require(XmlPullParser.START_TAG, ns, "description");
        String title = readText(parser);
        parser.require(XmlPullParser.END_TAG, ns, "description");
        return title;
    }
    private static Date readDate(XmlPullParser parser) throws IOException, XmlPullParserException {
        parser.require(XmlPullParser.START_TAG, ns, "pubDate");
        String title = readText(parser);
        parser.require(XmlPullParser.END_TAG, ns, "pubDate");
        DateFormat df = new SimpleDateFormat("E, dd MMM yyyy hh:mm");
        Date dt = new Date();
        try {
            dt = df.parse(title);
        } catch (ParseException e) {
            e.printStackTrace();
        }
               return dt;
           }
           private static String readText(XmlPullParser parser) throws IOException, XmlPullParserException {
               String result = "";
               if (parser.next() == XmlPullParser.TEXT) {
                   result = parser.getText();
                   parser.nextTag();
               }
               return result;
           }
           private static void skip(XmlPullParser parser) throws XmlPullParserException, IOException {
               if (parser.getEventType() != XmlPullParser.START_TAG) {
                   throw new IllegalStateException();
               }
               int depth = 1;
               while (depth != 0) {
                   switch (parser.next()) {
                       case XmlPullParser.END_TAG:
                           depth--;
                           break;
                       case XmlPullParser.START_TAG:
                           depth++;
                           break;
                   }
               }
           }
       }
       NewsItemsAdapter.java package com.example.baltu.myapplication.News;
       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.TextView;
       import com.example.baltu.myapplication.DataTypes.News_Class;
       import com.example.baltu.myapplication.MyWebViwer;
       import com.example.baltu.myapplication.R;
       import java.text.DateFormat;
       import java.text.SimpleDateFormat;
       import java.util.Date;
       import java.util.List;
       public class NewsItemsAdapter extends RecyclerView.Adapter < NewsItemsAdapter.ViewHolder > {
           private List < News_Class > mItems;
           private Context mContext;
           int f;
           public NewsItemsAdapter(Context context, List < News_Class > items) {
               this.mContext = context;
               this.mItems = items;
           }@
           Override public NewsItemsAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
               LayoutInflater inflater = LayoutInflater.from(mContext);
               View itemView = inflater.inflate(R.layout.news_layout, parent, false);
```

```java
                ViewHolder viewHolder = new ViewHolder(itemView);
                return viewHolder;
            }@
            Override public void onBindViewHolder(final NewsItemsAdapter.ViewHolder holder, int
    position) {
                final News_Class item = mItems.get(position);
                holder.Title.setText(item.getTitle());
                holder.TDescription.setText(item.getDescription());
                Date d = item.getDate();
                DateFormat dtd = new SimpleDateFormat("EEE dd MMM yyyy hh:mm");
                holder.DATE.setText(dtd.format(d));
                holder.mView.setOnClickListener(new View.OnClickListener() {@
                    Override public void onClick(View v) {
                        Intent newsintent = new Intent(holder.mView.getContext(), MyWebViwer.cl
ass);
                        newsintent.putExtra("URL", item.getLink());
                        holder.mView.getContext().startActivity(newsintent);
                    }
                });
            }@
            Override public int getItemCount() {
                return mItems.size();
            }
            public static class ViewHolder extends RecyclerView.ViewHolder {
                public TextView Title;
                public TextView TDescription;
                public TextView DATE;
                public View mView;
                public ViewHolder(View itemView) {
                    super(itemView);
                    Title = (TextView) itemView.findViewById(R.id.News_Title);
                    TDescription = (TextView) itemView.findViewById(R.id.News_Desc);
                    DATE = (TextView) itemView.findViewById(R.id.News_Date);
                    mView = itemView;
                }
            }
        }
```

InfoListActivity.java

```java
package com.example.baltu.myapplication.Information_Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import com.example.baltu.myapplication.DataTypes.InfoContent;
import com.example.baltu.myapplication.Information_Activity.InfoDetailActivity;
import com.example.baltu.myapplication.Information_Activity.InfoDetailFragment;
import com.example.baltu.myapplication.R;
import java.util.List;
/** * An activity representing a list of Items. This activity * has different presentations fo
r handset and tablet-
size devices. On * handsets, the activity presents a list of items, which when touched, * lead
 to a {@link InfoDetailActivity} representing * item details. On tablets, the activity present
s the list of items and * item details side-by-side using two vertical panes. */
```

```java
public class InfoListActivity extends AppCompatActivity {
    /**     * Whether or not the activity is in two-pane mode, i.e. running on a tablet     * device.     */
    private boolean mTwoPane;@
    Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_info_list);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        toolbar.setTitle(getTitle());
        View recyclerView = findViewById(R.id.info_list);
        assert recyclerView != null;
        setupRecyclerView((RecyclerView) recyclerView);
        if (findViewById(R.id.info_detail_container) != null) { // The detail container view will be present only in the // large-screen layouts (res/values-w900dp). // If this view is present, then the // activity should be in two-pane mode.
            mTwoPane = true;
        }
    }
    private void setupRecyclerView(@NonNull RecyclerView recyclerView) {
        recyclerView.setAdapter(new SimpleItemRecyclerViewAdapter(InfoContent.ITEMS));
    }
    public class SimpleItemRecyclerViewAdapter extends RecyclerView.Adapter < SimpleItemRecyclerViewAdapter.ViewHolder > {
        private final List < InfoContent.InfoItem > mValues;
        public SimpleItemRecyclerViewAdapter(List < InfoContent.InfoItem > items) {
            mValues = items;
        }@
        Override public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
            View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.info_list_content, parent, false);
            return new ViewHolder(view);
        }@
        Override public void onBindViewHolder(final ViewHolder holder, int position) {
            holder.mItem = mValues.get(position); //holder.mIdView.setText(mValues.get(position).id);
            holder.mContentView.setText(mValues.get(position).content);
            holder.mView.setOnClickListener(new View.OnClickListener() {@
                Override public void onClick(View v) {
                    if (mTwoPane) {
                        Bundle arguments = new Bundle();
                        arguments.putString(InfoDetailFragment.ARG_ITEM_ID, holder.mItem.id);

                        InfoDetailFragment fragment = new InfoDetailFragment();
                        fragment.setArguments(arguments);
                        getSupportFragmentManager().beginTransaction().replace(R.id.info_detail_container, fragment).commit();
                    } else {
                        Context context = v.getContext();
                        Intent intent = new Intent(context, InfoDetailActivity.class);
                        intent.putExtra(InfoDetailFragment.ARG_ITEM_ID, holder.mItem.id);
                        context.startActivity(intent);
                    }
                }
            });
        }@
        Override public int getItemCount() {
            return mValues.size();
        }
        public class ViewHolder extends RecyclerView.ViewHolder {
            public final View mView; //public final TextView mIdView;
            public final TextView mContentView;
            public InfoContent.InfoItem mItem;
            public ViewHolder(View view) {
                super(view);
                mView = view; //mIdView = (TextView) view.findViewById(R.id.id);
                mContentView = (TextView) view.findViewById(R.id.content);
            }@
            Override public String toString() {
                return super.toString() + " '" + mContentView.getText() + "'";
            }
        }
    }
}
```

InfoDetailActivity.java

```
package com.example.baltu.myapplication.Information_Activity;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.app.ActionBar;
import android.view.MenuItem;
import com.example.baltu.myapplication.R;
/** * An activity representing a single Info detail screen. This * activity is only used narro
w width devices. On tablet-size devices, * item details are presented side-by-
side with a list of items * in a {@link InfoListActivity}. */
public class InfoDetailActivity extends AppCompatActivity {@
    Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_info_detail);
        Toolbar toolbar = (Toolbar) findViewById(R.id.detail_toolbar);
        setSupportActionBar(toolbar);
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View view) {
                Snackbar.make(view, "Replace with your own detail action", Snackbar.LENGTH_LON
G).setAction("Action", null).show();
            }
        }); // Show the Up button in the action bar.
        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            actionBar.setDisplayHomeAsUpEnabled(true);
        } // savedInstanceState is non-
null when there is fragment state // saved from previous configurations of this activity // (e
.g. when rotating the screen from portrait to landscape). // In this case, the fragment will a
utomatically be re-
added // to its container so we don't need to manually add it. // For more information, see th
e Fragments API guide at: // // http://developer.android.com/guide/components/fragments.html /
/
        if (savedInstanceState == null) { // Create the detail fragment and add it to the acti
vity // using a fragment transaction.
            Bundle arguments = new Bundle();
            arguments.putString(InfoDetailFragment.ARG_ITEM_ID, getIntent().getStringExtra(Inf
oDetailFragment.ARG_ITEM_ID));
            InfoDetailFragment fragment = new InfoDetailFragment();
            fragment.setArguments(arguments);
            getSupportFragmentManager().beginTransaction().add(R.id.info_detail_container, fra
gment).commit();
```

```
        }
    }@
    Override public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == android.R.id.home) { // This ID represents the Home or Up button. In the case of this // activity, the Up button is shown. For // more details, see the Navigation pattern on Android Design: // // http://developer.android.com/design/patterns/navigation.html#up-vs-back //
            navigateUpTo(new Intent(this, InfoListActivity.class));
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

InfoDetailFragment.java

```
package com.example.baltu.myapplication.Information_Activity;
import android.app.Activity;
import android.support.design.widget.CollapsingToolbarLayout;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import com.example.baltu.myapplication.DataTypes.InfoContent;
import com.example.baltu.myapplication.R;
/** * A fragment representing a single Info detail screen. * This fragment is either contained in a {@link InfoListActivity} * in two-pane mode (on tablets) or a {@link InfoDetailActivity} * on handsets. */
public class InfoDetailFragment extends Fragment {
    /**     * The fragment argument representing the item ID that this fragment     * represents.     */
    public static final String ARG_ITEM_ID = "item_id";
    /**     * The dummy content this fragment is presenting.     */
    private InfoContent.InfoItem mItem;
    /**     * Mandatory empty constructor for the fragment manager to instantiate the     * fragment (e.g. upon screen orientation changes).     */
    public InfoDetailFragment() {}@
    Override public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments().containsKey(ARG_ITEM_ID)) { // Load the dummy content specified by the fragment // arguments. In a real-world scenario, use a Loader // to load content from a content provider.
            mItem = InfoContent.ITEM_MAP.get(getArguments().getString(ARG_ITEM_ID));
            Activity activity = this.getActivity();
            CollapsingToolbarLayout appBarLayout = (CollapsingToolbarLayout) activity.findViewById(R.id.toolbar_layout);
            if (appBarLayout != null) {
                appBarLayout.setTitle(mItem.content);
            }
        }
    }@
    Override public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.info_detail, container, false); // Show the dummy content as text in a TextView.
        if (mItem != null) {
            ((TextView) rootView.findViewById(R.id.info_detail)).setText(mItem.details);
        }
        return rootView;
    }
}
```

GPA_Calculator.java

```
package com.example.baltu.myapplication;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.List;
public class GPA_Calculator extends AppCompatActivity {
    private Spinner s11;
    TextView result;
    private Spinner s12;
    private Spinner s13;
    private Spinner s14;
    private Spinner s15;
    private Spinner s16;
    private Spinner s17;
    private Spinner s21;
    private Spinner s22;
    private Spinner s23;
    private Spinner s24;
    private Spinner s25;
    private Spinner s26;
    private Spinner s27;
    List < grades > allgrades = new ArrayList < > ();
    TextView semsterc;
    TextView semstergpa;
    TextView totalc;
    TextView cgpatext;
    TextView newCGPA;@
    Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gpa__calculator);
        s11 = (Spinner) findViewById(R.id.spinner111);
        s12 = (Spinner) findViewById(R.id.spinner12);
        s13 = (Spinner) findViewById(R.id.spinner13);
        s14 = (Spinner) findViewById(R.id.spinner14);
        s15 = (Spinner) findViewById(R.id.spinner15);
        s16 = (Spinner) findViewById(R.id.spinner16);
        s17 = (Spinner) findViewById(R.id.spinner17);
        s21 = (Spinner) findViewById(R.id.spinner21);
        s22 = (Spinner) findViewById(R.id.spinner22);
        s23 = (Spinner) findViewById(R.id.spinner23);
        s24 = (Spinner) findViewById(R.id.spinner24);
        s25 = (Spinner) findViewById(R.id.spinner25);
        s26 = (Spinner) findViewById(R.id.spinner26);
        s27 = (Spinner) findViewById(R.id.spinner27);
        result = (TextView) findViewById(R.id.GPAResult);
        semsterc = (TextView) findViewById(R.id.semesterCredits);
        semstergpa = (TextView) findViewById(R.id.yourGPA);
        totalc = (TextView) findViewById(R.id.TotalOldCredits);
        cgpatext = (TextView) findViewById(R.id.yourCGPA);
        newCGPA = (TextView) findViewById(R.id.new_CGPA);
        setall1spinners();
```

```java
        setall2spinners();
        View caculate = findViewById(R.id.calculat_bt);
        caculate.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                allgrades.clear();
                int allmycredits = 0;
                double allmygrades = 0;
                double gpa = 0;
                if (s11.getSelectedItemPosition() != 0 && s21.getSelectedItemPosition() != 0) addgrade(s11.getSelectedItemPosition(), s21.getSelectedItemPosition());
                if (s12.getSelectedItemPosition() != 0 && s22.getSelectedItemPosition() != 0) addgrade(s12.getSelectedItemPosition(), s22.getSelectedItemPosition());
                if (s13.getSelectedItemPosition() != 0 && s23.getSelectedItemPosition() != 0) addgrade(s13.getSelectedItemPosition(), s23.getSelectedItemPosition());
                if (s14.getSelectedItemPosition() != 0 && s24.getSelectedItemPosition() != 0) addgrade(s14.getSelectedItemPosition(), s24.getSelectedItemPosition());
                if (s15.getSelectedItemPosition() != 0 && s25.getSelectedItemPosition() != 0) addgrade(s15.getSelectedItemPosition(), s25.getSelectedItemPosition());
                if (s16.getSelectedItemPosition() != 0 && s26.getSelectedItemPosition() != 0) addgrade(s16.getSelectedItemPosition(), s26.getSelectedItemPosition());
                if (s17.getSelectedItemPosition() != 0 && s27.getSelectedItemPosition() != 0) addgrade(s17.getSelectedItemPosition(), s27.getSelectedItemPosition());
                if (allgrades.size() != 0) {
                    for (grades item: allgrades) {
                        allmycredits += item.getCredit();
                        allmygrades += ((item.getGrade() / 4.0) * (double) item.getCredit());

                    }
                    gpa = (allmygrades / (double) allmycredits) * 4.0;
                    DecimalFormat df = new DecimalFormat("0.00");
                    result.setText(df.format(gpa));
                    semstergpa.setText(df.format(gpa));
                    semsterc.setText(Integer.toString(allmycredits));
                } else Toast.makeText(GPA_Calculator.this, "Error", Toast.LENGTH_SHORT).show();
            }
        });
        View caculatecgpa = findViewById(R.id.CalculateCGPA);
        caculatecgpa.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                if (!semstergpa.getText().toString().equals("") && !semsterc.getText().toString().equals("") && !totalc.getText().toString().equals("") && !cgpatext.getText().toString().equals("")) {
                    int sc = Integer.parseInt(semsterc.getText().toString());
                    double sgpa = Double.parseDouble(semstergpa.getText().toString());
                    int tc = Integer.parseInt(totalc.getText().toString());
                    double cgpa = Double.parseDouble(cgpatext.getText().toString());
                    if (sgpa > 4 || cgpa > 4) {
                        Toast.makeText(GPA_Calculator.this, "Gpa and CGPA should be less than 4", Toast.LENGTH_LONG).show();
                        return;
                    }
                    if (sc == 0) {
                        Toast.makeText(GPA_Calculator.this, "semester credits can't be 0", Toast.LENGTH_LONG).show();
                                return;
                            }
                            if (sc > 30) {
                                Toast.makeText(GPA_Calculator.this, "semester credits are too large", Toast.LENGTH_LONG).show();
                                return;
                    }
                    double ncgpa = ((((sgpa / 4.0) * (double) sc) + ((cgpa / 4.0) * (double) tc)) / (sc + tc)) * 4;
                    DecimalFormat df = new DecimalFormat("0.00");
                    newCGPA.setText(df.format(ncgpa));
                } else {
                    Toast.makeText(GPA_Calculator.this, "Enter All Required Details", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
    private void addgrade(int x, int y) {
        double mygrade = 0;
        switch (y) {
            case 1:
                mygrade = 4;
                break;
            case 2:
                mygrade = 3.7;
                break;
            case 3:
                mygrade = 3.3;
                break;
            case 4:
                mygrade = 3;
                break;
            case 5:
                mygrade = 2.7;
                break;
            case 6:
                mygrade = 2.3;
                break;
            case 7:
                mygrade = 2;
                break;
            case 8:
                mygrade = 1.7;
                break;
            case 9:
                mygrade = 1.3;
                break;
            case 10:
                mygrade = 1;
                break;
            case 11:
                mygrade = 0.7;
                break;
            case 12:
                mygrade = 0;
                break;
        }
        allgrades.add(new grades(x, mygrade));
    }
    private void setall1spinners() {
        ArrayAdapter < CharSequence > items = ArrayAdapter.createFromResource(this, R.array.Credits, android.R.layout.simple_spinner_item);
        items.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        s11.setAdapter(items);
        s12.setAdapter(items);
        s13.setAdapter(items);
```

```java
                s14.setAdapter(items);
                s15.setAdapter(items);
                s16.setAdapter(items);
                s17.setAdapter(items);
            }
            private void setall2spinners() {
                ArrayAdapter < CharSequence > items = ArrayAdapter.createFromResource(this, R.array.Grades, android.R.layout.simple_spinner_item);
                items.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
                s21.setAdapter(items);
                s22.setAdapter(items);
                s23.setAdapter(items);
                s24.setAdapter(items);
                s25.setAdapter(items);
                s26.setAdapter(items);
                s27.setAdapter(items);
            }
            private class grades {
                private int credit;
                private double grade;
                public grades(int credit, double grade) {
                    this.credit = credit;
                    this.grade = grade;
                }
                public int getCredit() {
                    return credit;
                }
                public void setCredit(int credit) {
                    this.credit = credit;
                }
                public double getGrade() {
                    return grade;
                }
                public void setGrade(double grade) {
                    this.grade = grade;
                }
            }
        }
```

Settings_activity.java

```java
package com.example.baltu.myapplication.Settings;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
import com.example.baltu.myapplication.R;
import com.example.baltu.myapplication.Updater.Main_Updator2;
public class Settings_activity extends Activity {@
    Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.settings);
        findViewById(R.id.button4).setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                if (isNetworkAvailable()) {
                    new Main_Updator2(Settings_activity.this);
                } else Toast.makeText(Settings_activity.this, "Connect To the Internet", Toast.LENGTH_SHORT).show();
            }
        });
    }
    public void pass(View view) {
        Intent calenderintents2 = new Intent(Settings_activity.this, Password_Settings.class);

        startActivity(calenderintents2);
    }
    private boolean isNetworkAvailable() {
        ConnectivityManager connectivityManager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo activeNetworkInfo = connectivityManager.getActiveNetworkInfo();
        return activeNetworkInfo != null && activeNetworkInfo.isConnected();
    }
}
```

Password_Settings.java

```
package com.example.baltu.myapplication.Settings;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.v7.widget.SwitchCompat;
import android.text.InputType;
import android.util.Log;
import android.widget.CompoundButton;
import android.widget.EditText;
import com.example.baltu.myapplication.R;
import com.scottyab.aescrypt.AESCrypt;
public class Password_Settings extends Activity {
    private String m_Text = "";@
    Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_password__settings);
        SharedPreferences prefs = getSharedPreferences(getString(R.string.preference_file_key), MODE_PRIVATE);
        boolean key = prefs.getBoolean("pass_on", false);
        final SwitchCompat witch = (SwitchCompat) findViewById(R.id.switchpass);
        witch.setChecked(key);
        witch.setOnCheckedChangeListener(new SwitchCompat.OnCheckedChangeListener() {@
            Override public void onCheckedChanged(CompoundButton switchCompat, boolean isChecked) {
                SharedPreferences prefs = getSharedPreferences(getString(R.string.preference_file_key), MODE_PRIVATE);
                boolean key = prefs.getBoolean("pass_on", false);
                final SharedPreferences.Editor prefsE = prefs.edit();
                if (!isChecked) {
                    prefsE.putBoolean("pass_on", false);
                    prefsE.commit();
                    prefsE.putString("Password_String", "");
                    Log.d("Pass_On", "null");
                } else {
                    AlertDialog.Builder builder = new AlertDialog.Builder(Password_Settings.this);
                    builder.setTitle("Enter New Password"); // Set up the input
                    final EditText input = new EditText(Password_Settings.this); // Specify the type of input expected; this, for example, sets the input as a password, and will mask the text
                    input.setInputType(InputType.TYPE_CLASS_NUMBER | InputType.TYPE_NUMBER_VARIATION_PASSWORD);
                    builder.setView(input); // Set up the buttons
                    builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {@
                        Override public void onClick(DialogInterface dialog, int which) {
                            m_Text = input.getText().toString();
                            if (m_Text.isEmpty()) {
                                witch.setChecked(false);
                                return;
                            } else {
                                try {
                                    prefsE.putBoolean("pass_on", true);
                                    prefsE.putString("Password_String", AESCrypt.encrypt("CanT
heWorldUnderstand?", "TuNiSbYnIgHt" + m_Text));
                                    prefsE.commit();
                                } catch (Exception e) {
                                    e.printStackTrace();
                                }
                            }
                        }
                    });
                    builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{@
                        Override public void onClick(DialogInterface dialog, int which) {
                            dialog.cancel();
                            witch.setChecked(false);
                        }
                    });
                    builder.show(); //while(m_Text == "***"){}
                }
            }
        });
    }
}
```

Bus_Data.java

```
package com.example.baltu.myapplication.DataTypes;
import android.content.ContentValues;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.Buss_Table;
import com.example.baltu.myapplication.DB_Constructor.User_DB.TasksTable;
import java.util.UUID;
/** * Created by Baltu on 2017-04-24. */
public class Bus_Data {@
    com.google.gson.annotations.SerializedName("id") private String ID;
    private Integer Line_Number;
    private Integer Weekday;
    private Integer To_from;@
    com.google.gson.annotations.SerializedName("time") private String Time;
    public Bus_Data(String ID, Integer line_Number, Integer weekday, Integer to_from, String t
ime) {
        if (ID == null) ID = UUID.randomUUID().toString();
        this.ID = ID;
        Line_Number = line_Number;
        Weekday = weekday;
        To_from = to_from;
        Time = time;
    }
    public Bus_Data() {
        if (ID == null) ID = UUID.randomUUID().toString();
    }
    public void setTime(String time) {
        Time = time;
    }
```

```java
    public String getID() {
        return ID;
    }
    public void setID(String ID) {
        this.ID = ID;
    }
    public Integer getLine_Number() {
        return Line_Number;
    }
    public void setLine_Number(Integer line_Number) {
        Line_Number = line_Number;
    }
    public Integer getWeekday() {
        return Weekday;
    }
    public void setWeekday(Integer weekday) {
        Weekday = weekday;
    }
    public Integer getTo_from() {
        return To_from;
    }
    public void setTo_from(Integer to_from) {
        To_from = to_from;
    }
    public String getTime() {
        return Time;
    }
    public ContentValues toValues() {
        ContentValues values = new ContentValues(5);
        values.put(Buss_Table.COLUMN_ID, ID);
        values.put(Buss_Table.COLUMN_LINE_NUMBER, Line_Number);
        values.put(Buss_Table.COLUMN_WEEKDAY, Weekday);
        values.put(Buss_Table.COLUMN_TO_FROM, To_from);
        values.put(Buss_Table.COLUMN_TIME, Time);
        return values;
    }
}
```

Calender_Data.java

```java
package com.example.baltu.myapplication.DataTypes;
import android.content.ContentValues;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.Calender_Table;
import java.util.Calendar;
import java.util.Date;
/** * Created by Baltu on 2017-04-24. */
public class Calender_Data {
    private String ID;
    private String Event_Name;
    private Long StartDate;
    private Long EndDate;
    public Calender_Data() {}
    public Calender_Data(String ID, String event_Name, Long startDate, Long endDate) {
        this.ID = ID;
        Event_Name = event_Name;
        StartDate = startDate;
        EndDate = endDate;
    }
    public String getID() {
        return ID;
    }
    public void setID(String ID) {
        this.ID = ID;
    }
    public String getEvent_Name() {
        return Event_Name;
    }
    public void setEvent_Name(String event_Name) {
        Event_Name = event_Name;
    }
    public Long getStartDate() {
        return StartDate;
    }
    public void setStartDate(Long startDate) {
        StartDate = startDate;
    }
    public Long getEndDate() {
        return EndDate;
    }
    public void setEndDate(Long endDate) {
        EndDate = endDate;
    }
    public ContentValues toValues() {
        ContentValues values = new ContentValues(4);
        values.put(Calender_Table.COLUMN_ID, ID);
        values.put(Calender_Table.COLUMN_Event_Name, Event_Name);
        values.put(Calender_Table.COLUMN_SDATE, StartDate);
        values.put(Calender_Table.COLUMN_EDATE, EndDate);
        return values;
    }
}
```

Exam_class.java

```java
package com.example.baltu.myapplication.DataTypes;
import android.content.ContentValues;
import com.example.baltu.myapplication.DB_Constructor.User_DB.ExamsTable;
import java.util.UUID;
/** * Created by Baltu on 2017-04-24. */
public class Exam_class {
    private String ID;
    private String Course;
    private String discription;
    private long Exam_Date;
    private String Notes;
    public String getID() {
        return ID;
    }
    public void setID(String ID) {
        this.ID = ID;
    }
    public String getCourse() {
        return Course;
    }
    public void setCourse(String course) {
        Course = course;
    }
    public String getDiscription() {
        return discription;
    }
    public void setDiscription(String discription) {
        this.discription = discription;
    }
    public long getExam_Date() {
        return Exam_Date;
    }
    public void setExam_Date(long deadline_Date) {
        Exam_Date = deadline_Date;
    }
    public String getNotes() {
        return Notes;
    }
    public void setNotes(String notes) {
        Notes = notes;
    }
    public Exam_class(String ID, String course, String discription, long deadline_Date, String notes) {
        if (ID == null) {
            ID = UUID.randomUUID().toString();
        }
        this.ID = ID;
        Course = course;
        this.discription = discription;
        Exam_Date = deadline_Date;
        Notes = notes;
    }
    public Exam_class() {
        ID = UUID.randomUUID().toString();
    }
    public ContentValues toValues() {
        ContentValues values = new ContentValues(5);
        values.put(ExamsTable.COLUMN_ID, ID);
        values.put(ExamsTable.COLUMN_CourseNAME, Course);
        values.put(ExamsTable.COLUMN_Discription, discription);
        values.put(ExamsTable.COLUMN_EXAMDATE, Exam_Date);
        values.put(ExamsTable.COLUMN_NOTES, Notes);
        return values;
    }
}
```

InfoContent.java

```java
package com.example.baltu.myapplication.DataTypes;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
/** * Helper class for providing sample content for user interfaces created by * Android template wizards. * <p> * TODO: Replace all uses of this class before publishing your app. */
public class InfoContent {
    /**     * An array of sample (dummy) items.     */
    public static final List < InfoItem > ITEMS = new ArrayList < InfoItem > ();
    /**     * A map of sample (dummy) items, by ID.     */
    public static final Map < String, InfoItem > ITEM_MAP = new HashMap < String, InfoItem > ();
    private static final int COUNT = 15;
    static { // Add some sample items.
        for (int i = 1; i <= COUNT; i++) {
            addItem(createDummyItem(i));
        }
    }
    private static void addItem(InfoItem item) {
        ITEMS.add(item);
        ITEM_MAP.put(item.id, item);
    }
    private static InfoItem createDummyItem(int position) {
        return new InfoItem(String.valueOf(position), "Info For Students No" + position, makeD
etails(position));
    }
    private static String makeDetails(int position) {
            StringBuilder builder = new StringBuilder();
            builder.append("Details about Item: ").append(position);
            builder.append("\nTimeTables or Phone Number here");
            return builder.toString();
        }
        /**     * A dummy item representing a piece of content.     */
    public static class InfoItem {
        public final String id;
        public final String content;
        public final String details;
        public InfoItem(String id, String content, String details) {
            this.id = id;
            this.content = content;
            this.details = details;
        }@
        Override public String toString() {
            return content;
        }
    }
}
```

Information.java

```
package com.example.baltu.myapplication.DataTypes;
/** * Created by Baltu on 2017-05-20. */
public class Information {
    String ID;
    String Title;
    String Content;
    String Image;
    public Information(String ID, String title, String content, String image) {
        this.ID = ID;
        Title = title;
        Content = content;
        Image = image;
    }
    public String getID() {
        return ID;
    }
    public void setID(String ID) {
        this.ID = ID;
    }
    public String getTitle() {
        return Title;
    }
    public void setTitle(String title) {
        Title = title;
    }
    public String getContent() {
        return Content;
    }
    public void setContent(String content) {
        Content = content;
    }
    public String getImage() {
        return Image;
```

```java
    }
    public void setImage(String image) {
        Image = image;
    }
}
```

News_Class.java

```java
package com.example.baltu.myapplication.DataTypes;
import java.util.Date;
/** * Created by Baltu on 2017-04-24. */
public class News_Class {
    private String Title;
    private String Link;
    private String Description;
    private Date mDate;
    public News_Class(String title, String link, String description, java.util.Date date) {
        Title = title;
        Link = link;
        Description = description;
        mDate = date;
    }
    public String getTitle() {
        return Title;
    }
    public void setTitle(String title) {
        Title = title;
    }
    public String getLink() {
        return Link;
    }
    public void setLink(String link) {
        Link = link;
    }
    public String getDescription() {
        return Description;
    }
    public void setDescription(String description) {
        Description = description;
    }
    public java.util.Date getDate() {
        return mDate;
    }
    public void setDate(java.util.Date date) {
        mDate = date;
    }
}
```

Tasks_Data.java

```java
package com.example.baltu.myapplication.DataTypes;
import android.content.ContentValues;
import com.example.baltu.myapplication.DB_Constructor.User_DB.TasksTable;
import java.util.UUID;
/** * Created by Baltu on 2017-04-24. */
public class Tasks_Data {
    private String ID;
    private String Course;
    private String discription;
    private long Deadline_Date;
    private String Notes;
    public String getID() {
        return ID;
    }
    public void setID(String ID) {
        this.ID = ID;
    }
    public String getCourse() {
        return Course;
    }
    public void setCourse(String course) {
        Course = course;
    }
    public String getDiscription() {
        return discription;
    }
    public void setDiscription(String discription) {
        this.discription = discription;
    }
    public long getDeadline_Date() {
        return Deadline_Date;
    }
    public void setDeadline_Date(long deadline_Date) {
        Deadline_Date = deadline_Date;
    }
    public String getNotes() {
        return Notes;
    }
    public void setNotes(String notes) {
        Notes = notes;
    }
    public Tasks_Data(String ID, String course, String discription, long deadline_Date, String
notes) {
        if (ID == null) {
            ID = UUID.randomUUID().toString();
        }
        this.ID = ID;
        Course = course;
        this.discription = discription;
        Deadline_Date = deadline_Date;
        Notes = notes;
    }
    public Tasks_Data() {
        ID = UUID.randomUUID().toString();
    }
    public ContentValues toValues() {
        ContentValues values = new ContentValues(5);
        values.put(TasksTable.COLUMN_ID, ID);
        values.put(TasksTable.COLUMN_CourseNAME, Course);
        values.put(TasksTable.COLUMN_Discription, discription);
        values.put(TasksTable.COLUMN_ENDDATE, Deadline_Date);
        values.put(TasksTable.COLUMN_NOTES, Notes);
        return values;
    }
}
```

TimeTable_Classes.java

```java
package com.example.baltu.myapplication.DataTypes;
import android.content.ContentValues;
import com.example.baltu.myapplication.DB_Constructor.User_DB.Courses_Table;
import java.util.UUID;
/** * Created by Baltu on 2017-04-04. */
public class TimeTable_Classes {
    private String ID;
    private String Course;
    private String starting_time;
    private String finishing_time;
    private int Day_of_week;
    private String Notes;
    public TimeTable_Classes() {
        if (ID == null) {
            ID = UUID.randomUUID().toString();
        }
    }
    public TimeTable_Classes(String ID, String course, String starting_time, String finishing_time, int day_of_week, String notes) {
        if (ID == null) {
            ID = UUID.randomUUID().toString();
        }
        this.ID = ID;
        Course = course;
        this.starting_time = starting_time;
        this.finishing_time = finishing_time;
        Day_of_week = day_of_week;
        Notes = notes;
    }
    public String getID() {
        return ID;
    }
    public String getCourse() {
        return Course;
    }
    public String getStarting_time() {
        return starting_time;
    }
    public String getFinishing_time() {
        return finishing_time;
    }
    public int getDay_of_week() {
        return Day_of_week;
    }
    public String getNotes() {
        return Notes;
    }
    public void setID(String ID) {
        this.ID = ID;
    }
    public void setCourse(String course) {
        Course = course;
    }
    public void setStarting_time(String starting_time) {
        this.starting_time = starting_time;
    }
    public void setFinishing_time(String finishing_time) {
        this.finishing_time = finishing_time;
    }
    public void setDay_of_week(int day_of_week) {
        Day_of_week = day_of_week;
    }
    public void setNotes(String notes) {
        Notes = notes;
    }
    public ContentValues toValues() {
        ContentValues values = new ContentValues(6);
        values.put(Courses_Table.COLUMN_ID, ID);
        values.put(Courses_Table.COLUMN_DAY, Day_of_week);
        values.put(Courses_Table.COLUMN_NAME, Course);
        values.put(Courses_Table.COLUMN_NOTES, Notes);
        values.put(Courses_Table.COLUMN_STARTTIME, starting_time);
        values.put(Courses_Table.COLUMN_ENDTIME, finishing_time);
        return values;
    }
}
```

University_Locations.java

```java
package com.example.baltu.myapplication.DataTypes;
import android.content.ContentValues;
import android.content.pm.LabeledIntent;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.MapLocations_Table;
/** * Created by Baltu on 2017-05-23. */
public class University_Locations {
    private String ID;
    private String ShortName;
    private String LongName;
    private String Information;
    private Double Latitude;
    private Double Longitude;@
    com.google.gson.annotations.SerializedName("ImageUrl") private String Image_Url;
    public String getID() {
        return ID;
    }
    public void setID(String ID) {
        this.ID = ID;
    }
    public String getShortName() {
        return ShortName;
    }
    public void setShortName(String shortName) {
        ShortName = shortName;
    }
    public String getLongName() {
        return LongName;
    }
    public void setLongName(String longName) {
        LongName = longName;
    }
    public String getInformation() {
        return Information;
    }
    public void setInformation(String information) {
        Information = information;
    }
    public Double getLatitude() {
        return Latitude;
    }
    public void setLatitude(Double latitude) {
        Latitude = latitude;
    }
    public Double getLongitude() {
        return Longitude;
    }
    public void setLongitude(Double longitude) {
        Longitude = longitude;
    }
    public String getImage_Url() {
        return Image_Url;
    }
    public void setImage_Url(String image_Url) {
        Image_Url = image_Url;
    }
    public University_Locations(String ID, String shortName, String longName, String informati
on, Double latitude, Double longitude, String image_Url) {
        this.ID = ID;
        ShortName = shortName;
        LongName = longName;
        Information = information;
        Latitude = latitude;
        Longitude = longitude;
        Image_Url = image_Url;
    }
    public University_Locations() {}
    public ContentValues ToValues() {
        ContentValues Values = new ContentValues(7);
        Values.put(MapLocations_Table.COLUMN_ID, ID);
        Values.put(MapLocations_Table.COLUMN_SHORTNAME, ShortName);
        Values.put(MapLocations_Table.COLUMN_LONGNAME, LongName);
        Values.put(MapLocations_Table.COLUMN_INFO, Information);
        Values.put(MapLocations_Table.COLUMN_LATITUDE, Latitude);
        Values.put(MapLocations_Table.COLUMN_LONGITUDE, Longitude);
        Values.put(MapLocations_Table.COLUMN_IMAGE_URL, Image_Url);
        return Values;
    }
}
```

Local_DB_Helper.java

```java
package com.example.baltu.myapplication.DB_Constructor;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.widget.Toast;
import com.example.baltu.myapplication.DB_Constructor.*;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.Buss_Table;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.Calender_Table;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.Information_Table;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.MapLocations_Table;
import com.example.baltu.myapplication.DB_Constructor.User_DB.Courses_Table;
import com.example.baltu.myapplication.DataTypes.Bus_Data;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Calendar;
import java.util.List;
/** * Created by Baltu on 2017-05-06. */
public class Local_DB_Helper extends SQLiteOpenHelper {
    static String DB_PATH = "/data/data/com.example.baltu.myapplication/databases/";
    public static final String DB_FILENAME = "LocalDb.db";
    public static final int DB_Version = 1;
    Context mcontext;
    public Local_DB_Helper(Context context) {
        super(context, DB_FILENAME, null, DB_Version);
        mcontext = context;
    }@
    Override public void onCreate(SQLiteDatabase db) {
        db.execSQL(Buss_Table.SQL_CREATE);
        db.execSQL(Calender_Table.SQL_CREATE);
        db.execSQL(Information_Table.SQL_CREATE);
        db.execSQL(MapLocations_Table.SQL_CREATE);
    }@
    Override public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {}@
    Override public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        super.onDowngrade(db, oldVersion, newVersion);
    }
}
```

User_DB_Helper.java

```
package com.example.baltu.myapplication.DB_Constructor;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import com.example.baltu.myapplication.DB_Constructor.User_DB.Courses_Table;
import com.example.baltu.myapplication.DB_Constructor.User_DB.ExamsTable;
import com.example.baltu.myapplication.DB_Constructor.User_DB.Settings;
import com.example.baltu.myapplication.DB_Constructor.User_DB.TasksTable;
/** * Created by Baltu on 2017-05-06. */
public class User_DB_Helper extends SQLiteOpenHelper {
    public static final String DB_FILENAME = "UserDb.db";
    public static final int DB_Version = 1;
    public User_DB_Helper(Context context) {
        super(context, DB_FILENAME, null, DB_Version);
    }@
    Override public void onCreate(SQLiteDatabase db) {
        db.execSQL(Courses_Table.SQL_CREATE);
        db.execSQL(TasksTable.SQL_CREATE);
        db.execSQL(ExamsTable.SQL_CREATE);
    }@
    Override public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {}
}
```

Buss_Table.java

```
package com.example.baltu.myapplication.DB_Constructor.Local_DB;
/** * Created by Baltu on 2017-05-11. */
public class Buss_Table {
    public static final String TABLE_NAME = "buss";
    public static final String COLUMN_ID = "id";
    public static final String COLUMN_LINE_NUMBER = "Line_Number";
    public static final String COLUMN_WEEKDAY = "Weekday";
    public static final String COLUMN_TO_FROM = "To_from";
    public static final String COLUMN_TIME = "time";
    public static final String[] allColoumns = {
        COLUMN_ID, COLUMN_LINE_NUMBER, COLUMN_WEEKDAY, COLUMN_TO_FROM, COLUMN_TIME
    };
    public static final String SQL_CREATE = "CREATE TABLE " + TABLE_NAME + "(" + COLUMN_ID + "
 TEXT PRIMARY KEY," + COLUMN_LINE_NUMBER + " INTEGER," + COLUMN_WEEKDAY + " INTEGER," + COLUMN
_TO_FROM + " INTEGER," + COLUMN_TIME + " TEXT" + ");";
    public static final String SQL_DELETE = "DROP TABLE " + TABLE_NAME;
}
```

Calender_Table.java

```
package com.example.baltu.myapplication.DB_Constructor.Local_DB;
/** * Created by Baltu on 2017-05-11. */
public class Calender_Table {
    public static final String TABLE_NAME = "calender";
    public static final String COLUMN_ID = "ID";
    public static final String COLUMN_Event_Name = "event_name";
    public static final String COLUMN_SDATE = "start_date";
    public static final String COLUMN_EDATE = "end_date";
    public static final String[] allColoumns = {
```

```java
        COLUMN_ID, COLUMN_Event_Name, COLUMN_SDATE, COLUMN_EDATE
    };
    public static final String SQL_CREATE = "CREATE TABLE " + TABLE_NAME + "(" + COLUMN_ID + " TEXT PRIMARY KEY," + COLUMN_Event_Name + " TEXT," + COLUMN_SDATE + " LONG," + COLUMN_EDATE + " LONG" + ");";
    public static final String SQL_DELETE = "DROP TABLE " + TABLE_NAME;
}
```

Information_Table.java

```java
package com.example.baltu.myapplication.DB_Constructor.Local_DB;
/** * Created by Baltu on 2017-05-11. */
public class Information_Table {
    public static final String TABLE_NAME = "info";
    public static final String COLUMN_ID = "ID";
    public static final String COLUMN_TITLE = "Title";
    public static final String COLUMN_CONTENT = "Content";
    public static final String COLUMN_Image = "Image";
    public static final String[] allColoumns = {
        COLUMN_ID, COLUMN_TITLE, COLUMN_CONTENT, COLUMN_Image
    };
    public static final String SQL_CREATE = "CREATE TABLE " + TABLE_NAME + "(" + COLUMN_ID + " TEXT PRIMARY KEY," + COLUMN_TITLE + " TEXT," + COLUMN_CONTENT + " TEXT," + COLUMN_Image + " TEXT" + ");";
    public static final String SQL_DELETE = "DROP TABLE " + TABLE_NAME;
}
```

MapLocations_Table.java

```java
package com.example.baltu.myapplication.DB_Constructor.Local_DB;
/** * Created by Baltu on 2017-05-11. */
public class MapLocations_Table {
    public static final String TABLE_NAME = "MapLocations";
    public static final String COLUMN_ID = "ID";
    public static final String COLUMN_SHORTNAME = "ShortName";
    public static final String COLUMN_LONGNAME = "LongName";
    public static final String COLUMN_INFO = "Information";
    public static final String COLUMN_LATITUDE = "Latitude";
    public static final String COLUMN_LONGITUDE = "Longitude";
    public static final String COLUMN_IMAGE_URL = "ImageUrl";
    public static final String[] allColoumns = {
        COLUMN_ID, COLUMN_SHORTNAME, COLUMN_LONGNAME, COLUMN_INFO, COLUMN_LATITUDE, COLUMN_LONGITUDE, COLUMN_IMAGE_URL
    };
    public static final String SQL_CREATE = "CREATE TABLE " + TABLE_NAME + "(" + COLUMN_ID + " TEXT PRIMARY KEY," + COLUMN_SHORTNAME + " TEXT," + COLUMN_LONGNAME + " TEXT," + COLUMN_INFO + " TEXT," + COLUMN_LATITUDE + " REAL," + COLUMN_LONGITUDE + " REAL," + COLUMN_IMAGE_URL + " TEXT" + ");";
    public static final String SQL_DELETE = "DROP TABLE " + TABLE_NAME;
}
```

Courses_Table.java

```java
package com.example.baltu.myapplication.DB_Constructor.User_DB;
public class Courses_Table {
    public static final String TABLE_ITEMS = "lectures";
    public static final String COLUMN_ID = "itemId";
    public static final String COLUMN_NAME = "itemName";
    public static final String COLUMN_NOTES = "notes";
    public static final String COLUMN_STARTTIME = "starttime";
    public static final String COLUMN_ENDTIME = "wndtime";
    public static final String COLUMN_DAY = "day";
    public static final String[] allColoumns = {
        COLUMN_ID, COLUMN_NAME, COLUMN_NOTES, COLUMN_STARTTIME, COLUMN_ENDTIME, COLUMN_DAY
    };
    public static final String SQL_CREATE = "CREATE TABLE " + TABLE_ITEMS + "(" + COLUMN_ID + " TEXT PRIMARY KEY," + COLUMN_NAME + " TEXT," + COLUMN_NOTES + " TEXT," + COLUMN_STARTTIME + " TEXT," + COLUMN_ENDTIME + " TEXT," + COLUMN_DAY + " TEXT" + ");";
    public static final String SQL_DELETE = "DROP TABLE " + TABLE_ITEMS;
}
```

ExamsTable.java

```
package com.example.baltu.myapplication.DB_Constructor.User_DB;
public class ExamsTable {
    public static final String TABLE_NAME = "exams";
    public static final String COLUMN_ID = "id";
    public static final String COLUMN_CourseNAME = "course";
    public static final String COLUMN_Discription = "discription";
    public static final String COLUMN_EXAMDATE = "exam_date";
    public static final String COLUMN_NOTES = "notes";
    public static final String[] allColoumns = {
        COLUMN_ID, COLUMN_CourseNAME, COLUMN_Discription, COLUMN_NOTES, COLUMN_EXAMDATE, COLUMN_NOTES
    };
    public static final String SQL_CREATE = "CREATE TABLE " + TABLE_NAME + "(" + COLUMN_ID + " TEXT PRIMARY KEY," + COLUMN_CourseNAME + " TEXT," + COLUMN_Discription + " TEXT," + COLUMN_EXAMDATE + " INTEGER ," + COLUMN_NOTES + " TEXT" + ");";
    public static final String SQL_DELETE = "DROP TABLE " + TABLE_NAME;
}
```

Settings.java

```
package com.example.baltu.myapplication.DB_Constructor.User_DB;
public class Settings {
    public static final String TABLE_NAME = "settings";
    public static final String COLUMN_ID = "ID";
    public static final String COLUMN_OPTIONNAME = "option";
    public static final String COLUMN_Discription = "discription";
    public static final String COLUMN_CONTENT = "content";
    public static final String[] allColoumns = {
        COLUMN_ID, COLUMN_OPTIONNAME, COLUMN_Discription, COLUMN_CONTENT
    };
    public static final String SQL_CREATE = "CREATE TABLE " + TABLE_NAME + "(" + COLUMN_ID + " TEXT PRIMARY KEY," + COLUMN_OPTIONNAME + " TEXT," + COLUMN_Discription + " TEXT," + COLUMN_CONTENT + " TEXT " + ");";
    public static final String SQL_DELETE = "DROP TABLE " + TABLE_NAME;
}
```

TasksTable.java

```
package com.example.baltu.myapplication.DB_Constructor.User_DB;
public class TasksTable {
    public static final String TABLE_NAME = "tasks";
    public static final String COLUMN_ID = "ID";
    public static final String COLUMN_CourseNAME = "course";
    public static final String COLUMN_Discription = "discription";
    public static final String COLUMN_ENDDATE = "deadline_date";
    public static final String COLUMN_NOTES = "notes";
    public static final String[] allColoumns = {
        COLUMN_ID, COLUMN_CourseNAME, COLUMN_Discription, COLUMN_NOTES, COLUMN_ENDDATE, COLUMN_NOTES
    };
    public static final String SQL_CREATE = "CREATE TABLE " + TABLE_NAME + "(" + COLUMN_ID + " TEXT PRIMARY KEY," + COLUMN_CourseNAME + " TEXT," + COLUMN_Discription + " TEXT," + COLUMN_ENDDATE + " INTEGER ," + COLUMN_NOTES + " TEXT" + ");";
    public static final String SQL_DELETE = "DROP TABLE " + TABLE_NAME;
}
```

UserDB_Provider.java

```java
package com.example.baltu.myapplication.Data_Provider;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.DatabaseUtils;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import com.example.baltu.myapplication.DB_Constructor.User_DB.Courses_Table;
import com.example.baltu.myapplication.DB_Constructor.User_DB.ExamsTable;
import com.example.baltu.myapplication.DB_Constructor.User_DB.Settings;
import com.example.baltu.myapplication.DB_Constructor.User_DB.TasksTable;
import com.example.baltu.myapplication.DB_Constructor.User_DB_Helper;
import com.example.baltu.myapplication.DataTypes.Exam_class;
import com.example.baltu.myapplication.DataTypes.Tasks_Data;
import com.example.baltu.myapplication.DataTypes.TimeTable_Classes;
import com.scottyab.aescrypt.AESCrypt;
import java.security.GeneralSecurityException;
import java.util.ArrayList;
import java.util.List;
public class UserDB_Provider {
    private Context mContext;
    private final static String passid = "15968987559";
    private SQLiteDatabase UserDb;
    SQLiteOpenHelper mUserDBhelper;
    public UserDB_Provider(Context context) {
        this.mContext = context;
        mUserDBhelper = new User_DB_Helper(mContext);
        UserDb = mUserDBhelper.getWritableDatabase();
    }
    public void open() {
        UserDb = mUserDBhelper.getWritableDatabase();
    }
    public void close() {
            mUserDBhelper.close();
        } //----------------------------------------------------------------- //--------------
---------------------------------------------------
    public TimeTable_Classes addnewcourse(TimeTable_Classes ttc) {
        UserDb.insert(Courses_Table.TABLE_ITEMS, null, ttc.toValues());
        return ttc;
    }
    public long numberofcourses(TimeTable_Classes ttc) {
        return DatabaseUtils.queryNumEntries(UserDb, Courses_Table.TABLE_ITEMS, "itemId=?", ne
w String[] {
            ttc.getID()
        });
    }
    public TimeTable_Classes updatecourse(TimeTable_Classes ttc) {
        UserDb.update(Courses_Table.TABLE_ITEMS, ttc.toValues(), "itemId=?", new String[] {
            ttc.getID()
        });
        return ttc;
    }
    public boolean deletecourse(String id) {
        UserDb.delete(Courses_Table.TABLE_ITEMS, "itemId=?", new String[] {
            id
        });
        return true;
    }
    public List < TimeTable_Classes > getcourses(int day) {
            List < TimeTable_Classes > nelist = new ArrayList < > ();
            String[] arg = {
                Integer.toString(day)
            };
            Cursor cur = UserDb.query(Courses_Table.TABLE_ITEMS, Courses_Table.allColoumns, "day=?", arg, null, null, "starttime");
            while (cur.moveToNext()) {
                TimeTable_Classes ttc = new TimeTable_Classes();
                ttc.setID(cur.getString(cur.getColumnIndex(Courses_Table.COLUMN_ID)));
                ttc.setCourse(cur.getString(cur.getColumnIndex(Courses_Table.COLUMN_NAME)));
                ttc.setDay_of_week(cur.getInt(cur.getColumnIndex(Courses_Table.COLUMN_DAY)));

                ttc.setStarting_time(cur.getString(cur.getColumnIndex(Courses_Table.COLUMN_STARTTIME)));
                ttc.setFinishing_time(cur.getString(cur.getColumnIndex(Courses_Table.COLUMN_ENDTIME)));
                ttc.setNotes(cur.getString(cur.getColumnIndex(Courses_Table.COLUMN_NOTES)));
                nelist.add(ttc);
            }
            return nelist;
        } //=====================================================================================
    public Tasks_Data addnewTask(Tasks_Data ttc) {
        UserDb.insert(TasksTable.TABLE_NAME, null, ttc.toValues());
        return ttc;
    }
    public Tasks_Data updateTask(Tasks_Data ttc) {
        UserDb.update(TasksTable.TABLE_NAME, ttc.toValues(), "ID=?", new String[] {
            ttc.getID()
        });
        return ttc;
    }
    public int DeleteTask(String id) {
        String[] arg = {
            id
        };
        return UserDb.delete(TasksTable.TABLE_NAME, "ID=?", arg);
    }
    public List < Tasks_Data > GetTasks(int x) {
            List < Tasks_Data > nelist = new ArrayList < > ();
            String[] arg = {
                Long.toString((new java.util.Date()).getTime())
            };
            Cursor cur;
            if (x == 0) {
                cur = UserDb.query(TasksTable.TABLE_NAME, TasksTable.allColoumns, "deadline_date>?", arg, null, null, TasksTable.COLUMN_ENDDATE);
            } else {
                cur = UserDb.query(TasksTable.TABLE_NAME, TasksTable.allColoumns, "deadline_date<=?", arg, null, null, TasksTable.COLUMN_ENDDATE);
            }
            while (cur.moveToNext()) {
                Tasks_Data ttc = new Tasks_Data();
                ttc.setID(cur.getString(cur.getColumnIndex(TasksTable.COLUMN_ID)));
                ttc.setCourse(cur.getString(cur.getColumnIndex(TasksTable.COLUMN_CourseNAME)));
                ttc.setDeadline_Date(cur.getLong(cur.getColumnIndex(TasksTable.COLUMN_ENDDATE)));
                ttc.setDiscription(cur.getString(cur.getColumnIndex(TasksTable.COLUMN_Discription)));
                ttc.setNotes(cur.getString(cur.getColumnIndex(TasksTable.COLUMN_NOTES)));
                nelist.add(ttc);
            }
            return nelist;
        } //====================================================================================================
    public Exam_class addnewExam(Exam_class ttc) {
        UserDb.insert(ExamsTable.TABLE_NAME, null, ttc.toValues());
        return ttc;
    }
    public Exam_class updateExam(Exam_class ttc) {
        UserDb.update(ExamsTable.TABLE_NAME, ttc.toValues(), "id=?", new String[] {
            ttc.getID()
        });
        return ttc;
    }
    public int DeleteExam(String id) {
        String[] arg = {
            id
        };
        return UserDb.delete(ExamsTable.TABLE_NAME, "id=?", arg);
    }
    public List < Exam_class > GetExams(int x) {
            List < Exam_class > nelist = new ArrayList < > ();
            String[] arg = {
                Long.toString((new java.util.Date()).getTime())
            };
            Cursor cur;
            if (x == 0) {
                cur = UserDb.query(ExamsTable.TABLE_NAME, ExamsTable.allColoumns, "exam_date>?", arg, null, null, ExamsTable.COLUMN_EXAMDATE);
            } else {
                cur = UserDb.query(ExamsTable.TABLE_NAME, ExamsTable.allColoumns, "exam_date<=?", arg, null, null, ExamsTable.COLUMN_EXAMDATE);
            }
            while (cur.moveToNext()) {
                Exam_class ttc = new Exam_class();
                ttc.setID(cur.getString(cur.getColumnIndex(ExamsTable.COLUMN_ID)));
                ttc.setCourse(cur.getString(cur.getColumnIndex(ExamsTable.COLUMN_CourseNAME)));
                ttc.setExam_Date(cur.getLong(cur.getColumnIndex(ExamsTable.COLUMN_EXAMDATE)));
                ttc.setDiscription(cur.getString(cur.getColumnIndex(ExamsTable.COLUMN_Discription)));
                ttc.setNotes(cur.getString(cur.getColumnIndex(ExamsTable.COLUMN_NOTES)));
                nelist.add(ttc);
            }
            return nelist;
        } //=========================================================================================================== //Settings
    public int isPasswordOn() {
        Cursor cur;
        String[] arg = {
            passid
        };
        cur = UserDb.query(Settings.TABLE_NAME, Settings.allColoumns, Settings.COLUMN_ID + "=? ", arg, null, null, null);
        if (cur.getCount() == 1) {
```

```java
                if (cur.getString(cur.getColumnIndex(Settings.COLUMN_Discription)).equals("on")) {
                    return 1;
                } else if (cur.getString(cur.getColumnIndex(Settings.COLUMN_Discription)).equals("off")) {
                    return 0;
                } else return 2;
            } else return 2;
        }
        public String GetPasswordon() {
            Cursor cur;
            String[] arg = {
                passid
            };
            cur = UserDb.query(Settings.TABLE_NAME, Settings.allColoumns, Settings.COLUMN_ID + "=? ", arg, null, null, null);
            return cur.getString(cur.getColumnIndex(Settings.COLUMN_CONTENT)) + passid;
        }
        public boolean SetPassword(String mypass) throws GeneralSecurityException {
            ContentValues values = new ContentValues(4);
            values.put(Settings.COLUMN_ID, passid);
            values.put(Settings.COLUMN_Discription, "on");
            values.put(Settings.COLUMN_OPTIONNAME, "Password");
            values.put(Settings.COLUMN_CONTENT, AESCrypt.encrypt(passid, mypass));
            String[] arg = {
                passid
            };
            try {
                UserDb.delete(Settings.TABLE_NAME, Settings.COLUMN_ID + "=?", arg);
                UserDb.insert(Settings.TABLE_NAME, null, values);
            } catch (SQLiteException e) {
                e.printStackTrace();
                return false;
            } finally {
                return true;
            }
        }
        public boolean SetPasswordoff() throws GeneralSecurityException {
            ContentValues values = new ContentValues(4);
            values.put(Settings.COLUMN_ID, passid);
            values.put(Settings.COLUMN_Discription, "off");
            values.put(Settings.COLUMN_OPTIONNAME, "Password");
            values.put(Settings.COLUMN_CONTENT, "");
            String[] arg = {
                passid
            };
            try {
                UserDb.delete(Settings.TABLE_NAME, Settings.COLUMN_ID + "=?", arg);
                UserDb.insert(Settings.TABLE_NAME, null, values);
            } catch (SQLiteException e) {
                e.printStackTrace();
                return false;
            } finally {
                return true;
            }
        }
    }
```

Local_DB_Main_Provider.java

```java
package com.example.baltu.myapplication.Data_Provider;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.Buss_Table;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.Calender_Table;
import com.example.baltu.myapplication.DB_Constructor.Local_DB.MapLocations_Table;
import com.example.baltu.myapplication.DB_Constructor.Local_DB_Helper;
import com.example.baltu.myapplication.DataTypes.Bus_Data;
import com.example.baltu.myapplication.DataTypes.Calender_Data;
import com.example.baltu.myapplication.DataTypes.University_Locations;
import java.util.ArrayList;
import java.util.List;
public class Local_DB_Main_Provider {
    private Context mContext;
    private SQLiteDatabase LocalDB;
    SQLiteOpenHelper mLocalDBhelper;
    public Local_DB_Main_Provider(Context context) {
        this.mContext = context;
        mLocalDBhelper = new Local_DB_Helper(mContext);
        LocalDB = mLocalDBhelper.getWritableDatabase();
    }
    public void open() {
        LocalDB = mLocalDBhelper.getWritableDatabase();
    }
    public void close() {
            mLocalDBhelper.close();
        } //------------------------------------------------------------------ //--------------
---------------------------------------------------
    public List < Bus_Data > GetAllBus(int x, int y) {
        List < Bus_Data > nelist = new ArrayList < > ();
        String[] arg = {
            Long.toString((new java.util.Date()).getTime())
        };
        Cursor cur;
        cur = LocalDB.query(Buss_Table.TABLE_NAME, Buss_Table.allColoumns, " To_from=? AND  We
ekday=? ", new String[] {
            String.valueOf(x), Integer.toString(y)
        }, null, null, Buss_Table.COLUMN_TIME);
        while (cur.moveToNext()) {
            Bus_Data ttc = new Bus_Data();
            ttc.setID(cur.getString(cur.getColumnIndex(Buss_Table.COLUMN_ID)));
            ttc.setLine_Number(cur.getInt(cur.getColumnIndex(Buss_Table.COLUMN_LINE_NUMBER)));

            ttc.setTime(cur.getString(cur.getColumnIndex(Buss_Table.COLUMN_TIME)));
            ttc.setTo_from(cur.getInt(cur.getColumnIndex(Buss_Table.COLUMN_TO_FROM)));
            ttc.setWeekday(cur.getInt(cur.getColumnIndex(Buss_Table.COLUMN_WEEKDAY)));
            nelist.add(ttc);
        }
        return nelist;
    }
    public List < Bus_Data > Get1lineBus(int x, int y, int z) {
        List < Bus_Data > nelist = new ArrayList < > ();
        String[] arg = {
            Long.toString((new java.util.Date()).getTime())
        };
        Cursor cur;
        cur = LocalDB.query(Buss_Table.TABLE_NAME, Buss_Table.allColoumns, "Line_Number=? AND
To_from=? AND  Weekday=? ", new String[] {
            String.valueOf(x), Integer.toString(y), Integer.toString(z)
        }, null, null, Buss_Table.COLUMN_TIME);
        while (cur.moveToNext()) {
            Bus_Data ttc = new Bus_Data();
            ttc.setID(cur.getString(cur.getColumnIndex(Buss_Table.COLUMN_ID)));
            ttc.setLine_Number(cur.getInt(cur.getColumnIndex(Buss_Table.COLUMN_LINE_NUMBER)));
            ttc.setTime(cur.getString(cur.getColumnIndex(Buss_Table.COLUMN_TIME)));
            ttc.setTo_from(cur.getInt(cur.getColumnIndex(Buss_Table.COLUMN_TO_FROM)));
            ttc.setWeekday(cur.getInt(cur.getColumnIndex(Buss_Table.COLUMN_WEEKDAY)));
            nelist.add(ttc);
        }
        return nelist;
    }
    public List < Calender_Data > GetCalendarItems() {
        List < Calender_Data > nelist = new ArrayList < > ();
        Cursor cur;
        cur = LocalDB.query(Calender_Table.TABLE_NAME, Calender_Table.allColoumns, null, null, null, null, Calender_Table.COLUMN_SDATE);
        while (cur.moveToNext()) {
            Calender_Data ttc = new Calender_Data();
            ttc.setID(cur.getString(cur.getColumnIndex(Calender_Table.COLUMN_ID)));
            ttc.setEvent_Name(cur.getString(cur.getColumnIndex(Calender_Table.COLUMN_Event_Name)));
            ttc.setStartDate(cur.getLong(cur.getColumnIndex(Calender_Table.COLUMN_SDATE)));
            ttc.setEndDate(cur.getLong(cur.getColumnIndex(Calender_Table.COLUMN_EDATE)));
            nelist.add(ttc);
        }
        return nelist;
    }
    public List < University_Locations > GetMapLocations() {
        List < University_Locations > Locationslist = new ArrayList < > ();
        Cursor cur;
        cur = LocalDB.query(MapLocations_Table.TABLE_NAME, MapLocations_Table.allColoumns, null, null, null, null, MapLocations_Table.COLUMN_LONGNAME);
        while (cur.moveToNext()) {
            University_Locations location = new University_Locations();
            location.setID(cur.getString(cur.getColumnIndex(MapLocations_Table.COLUMN_ID)));
            location.setShortName(cur.getString(cur.getColumnIndex(MapLocations_Table.COLUMN_SHORTNAME)));
            location.setLongName(cur.getString(cur.getColumnIndex(MapLocations_Table.COLUMN_LONGNAME)));
            location.setInformation(cur.getString(cur.getColumnIndex(MapLocations_Table.COLUMN_INFO)));
            location.setLatitude(cur.getDouble(cur.getColumnIndex(MapLocations_Table.COLUMN_LATITUDE)));
            location.setLongitude(cur.getDouble(cur.getColumnIndex(MapLocations_Table.COLUMN_LONGITUDE)));
            location.setImage_Url(cur.getString(cur.getColumnIndex(MapLocations_Table.COLUMN_IMAGE_URL)));
            Locationslist.add(location);
        }
        return Locationslist;
    }
}
```

BusItemsAdapter.java

```java
package com.example.baltu.myapplication.Data_Provider.Adapters;
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.TextView;
import com.example.baltu.myapplication.DataTypes.Bus_Data;
import com.example.baltu.myapplication.R;
import com.example.baltu.myapplication.Map.University_Map;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
public class BusItemsAdapter extends RecyclerView.Adapter < BusItemsAdapter.ViewHolder > { // The items to display in your RecyclerView
    private List < Bus_Data > items; // Provide a suitable constructor (depends on the kind of dataset)
    public BusItemsAdapter(List < Bus_Data > items) {
            this.items = items;
        } // Return the size of your dataset (invoked by the layout manager)
        @
    Override public int getItemCount() {
        return this.items.size();
    }@
    Override public int getItemViewType(int position) {
        return items.get(position).getLine_Number();
    }@
    Override public BusItemsAdapter.ViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());
        final ViewHolder viewHolder;
        View v1;
        switch (viewType) {
            case 1:
                v1 = inflater.inflate(R.layout.bus1_layout, viewGroup, false);
                viewHolder = new ViewHolder(v1);
                break;
            case 2:
                v1 = inflater.inflate(R.layout.bus2_layout, viewGroup, false);
                viewHolder = new ViewHolder(v1);
                break;
            case 3:
                v1 = inflater.inflate(R.layout.bus3_layout, viewGroup, false);
                viewHolder = new ViewHolder(v1);
                break;
            case 4:
                v1 = inflater.inflate(R.layout.bus4_layout, viewGroup, false);
                viewHolder = new ViewHolder(v1);
                break;
            case 5:
                v1 = inflater.inflate(R.layout.bus5_layout, viewGroup, false);
                viewHolder = new ViewHolder(v1);
                break;
            default:
                v1 = inflater.inflate(R.layout.bus1_layout, viewGroup, false);
                viewHolder = new ViewHolder(v1);
                break;
        }
        return viewHolder;
    }@
    Override public void onBindViewHolder(final BusItemsAdapter.ViewHolder viewHolder, final int position) {
        final Bus_Data item = items.get(position);
        DateFormat df = new SimpleDateFormat("hh:mm aaa");
        DateFormat tlf = new SimpleDateFormat("HH:mm");
        Date now = new Date();
        String day_of_week = (new SimpleDateFormat("E")).format(now);
        String time = tlf.format(now);
        Date bustime = new Date();
        try {
            bustime = tlf.parse(item.getTime());
            now = tlf.parse(time);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        viewHolder.Timet.setText(df.format(bustime));
        Date left = new Date(0);
        if (day_of_week.equals("Sun") || day_of_week.equals("Mon") || day_of_week.equals("Tue") || day_of_week.equals("Wed") || day_of_week.equals("Thu")) {
            if (bustime.after(now)) {
                left = new Date(0, 0, 0, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
            } else {
                left = new Date(0, 0, 0, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
            }
        } else if (day_of_week.equals("Fri")) {
            if (bustime.after(now)) {
                left = new Date(0, 0, 0, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
            } else {
                left = new Date(0, 0, 2, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
            }
        } else if (day_of_week.equals("Sat")) {
            if (bustime.after(now)) {
                left = new Date(0, 0, 2, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
            } else {
                left = new Date(0, 0, 1, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
            }
        }
        viewHolder.Timeleft.setText(Integer.toString(left.getDay()) + "d:" + Integer.toString(left.getHours()) + "h:" + Integer.toString(left.getMinutes()) + "m");
        viewHolder.mView.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                Intent intf = new Intent(viewHolder.mView.getContext(), University_Map.class);

                intf.putExtra("type", 1);
                      intf.putExtra("line", items.get(position).getLine_Number());
                      viewHolder.mView.getContext().startActivity(intf);
                  }
              });
          }
          public static class ViewHolder extends RecyclerView.ViewHolder {
              public TextView Timet;
              public TextView Timeleft;
              public View mView;
              public Long Timer;
              public ViewHolder(View itemView) {
                  super(itemView);
                  Timet = (TextView) itemView.findViewById(R.id.bustime);
                  Timeleft = (TextView) itemView.findViewById(R.id.timelefttime);
                  mView = itemView;
              }
          }
```

```
    }
```

BusItemsAdapterweekend.java

```
package com.example.baltu.myapplication.Data_Provider.Adapters;
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.TextView;
import com.example.baltu.myapplication.DataTypes.Bus_Data;
import com.example.baltu.myapplication.R;
import com.example.baltu.myapplication.Map.University_Map;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
public class BusItemsAdapterweekend extends RecyclerView.Adapter < BusItemsAdapterweekend.ViewHolder > { // The items to display in your RecyclerView
  private List < Bus_Data > items; // Provide a suitable constructor (depends on the kind of dataset)
  public BusItemsAdapterweekend(List < Bus_Data > items) {
      this.items = items;
    } // Return the size of your dataset (invoked by the layout manager)
    @
  Override public int getItemCount() {
    return this.items.size();
  }@
  Override public int getItemViewType(int position) {
    return items.get(position).getLine_Number();
  }@
  Override public BusItemsAdapterweekend.ViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType) {
    LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());
    final ViewHolder viewHolder;
    View v1;
    switch (viewType) {
      case 1:
        v1 = inflater.inflate(R.layout.bus1_layout, viewGroup, false);
        viewHolder = new ViewHolder(v1);
        break;
      case 2:
        v1 = inflater.inflate(R.layout.bus2_layout, viewGroup, false);
        viewHolder = new ViewHolder(v1);
        break;
      case 3:
        v1 = inflater.inflate(R.layout.bus3_layout, viewGroup, false);
        viewHolder = new ViewHolder(v1);
        break;
      case 4:
        v1 = inflater.inflate(R.layout.bus4_layout, viewGroup, false);
        viewHolder = new ViewHolder(v1);
        break;
      case 5:
        v1 = inflater.inflate(R.layout.bus5_layout, viewGroup, false);
        viewHolder = new ViewHolder(v1);
        break;
      default:
        v1 = inflater.inflate(R.layout.bus1_layout, viewGroup, false);
        viewHolder = new ViewHolder(v1);
```

```java
                break;
        }
        return viewHolder;
    }@
    Override public void onBindViewHolder(final BusItemsAdapterweekend.ViewHolder viewHolder, final int position) {
        final Bus_Data item = items.get(position);
        DateFormat df = new SimpleDateFormat("hh:mm aaa");
        DateFormat tlf = new SimpleDateFormat("HH:mm");
        Date now = new Date();
        DateFormat day_of_week_fomat = new SimpleDateFormat("E");
        String day_of_week = day_of_week_fomat.format(now);
        String time = tlf.format(now);
        Date bustime = new Date();
        try {
            bustime = tlf.parse(item.getTime());
            now = tlf.parse(time);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        viewHolder.Timet.setText(df.format(bustime));
        Date left = new Date(0);
        if (day_of_week.equals("Sat")) {
            if (bustime.after(now)) {
                left = new Date(0, 0, 0, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
            } else {
                left = new Date(0, 0, 0, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
            }
        } else if (day_of_week.equals("Fri")) {
            if (bustime.after(now)) {
                left = new Date(0, 0, 1, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
            } else {
                left = new Date(0, 0, 0, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
            }
        } else if (day_of_week.equals("Sun")) {
            if (bustime.after(now)) {
                left = new Date(0, 0, 6, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
            } else {
                left = new Date(0, 0, 5, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
            }
        } else if (day_of_week.equals("Mon")) {
            if (bustime.after(now)) {
                left = new Date(0, 0, 5, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
            } else {
                left = new Date(0, 0, 4, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
                    }
                } else if (day_of_week.equals("Tue")) {
                    if (bustime.after(now)) {
                        left = new Date(0, 0, 4, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
                    } else {
                        left = new Date(0, 0, 3, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
                    }
                } else if (day_of_week.equals("Wed")) {
                    if (bustime.after(now)) {
                        left = new Date(0, 0, 3, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
                    } else {
                        left = new Date(0, 0, 2, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
                    }
                } else if (day_of_week.equals("Thu")) {
                    if (bustime.after(now)) {
                        left = new Date(0, 0, 2, bustime.getHours() - now.getHours(), bustime.getMinutes() - now.getMinutes(), bustime.getSeconds() - now.getSeconds());
                    } else {
                        left = new Date(0, 0, 1, 23 - (now.getHours() - bustime.getHours()), 60 - (now.getMinutes() - bustime.getMinutes()), 59 - (now.getSeconds() - bustime.getSeconds()));
                    }
                }
                viewHolder.Timeleft.setText(Integer.toString(left.getDay()) + "d:" + Integer.toString(left.getHours()) + "h:" + Integer.toString(left.getMinutes()) + "m");
                viewHolder.mView.setOnClickListener(new View.OnClickListener() {@
                    Override public void onClick(View v) {
                        Intent intf = new Intent(viewHolder.mView.getContext(), University_Map.class);
                        intf.putExtra("type", 1);
                        intf.putExtra("line", items.get(position).getLine_Number());
                        viewHolder.mView.getContext().startActivity(intf);
                    }
                });
            }
            public static class ViewHolder extends RecyclerView.ViewHolder {
                public TextView Timet;
                public TextView Timeleft;
                public View mView;
                public Long Timer;
                public ViewHolder(View itemView) {
                    super(itemView);
                    Timet = (TextView) itemView.findViewById(R.id.bustime);
                    Timeleft = (TextView) itemView.findViewById(R.id.timelefttime);
                    mView = itemView;
                }
            }
        }
```

CalendarItemsAdapter.java

```java
package com.example.baltu.myapplication.Data_Provider.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.TextView;
import com.example.baltu.myapplication.DataTypes.Calender_Data;
import com.example.baltu.myapplication.R;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
public class CalendarItemsAdapter extends RecyclerView.Adapter < CalendarItemsAdapter.ViewHolder > {
    private List < Calender_Data > mItems;
    private Context mContext;
    public CalendarItemsAdapter(Context context, List < Calender_Data > items) {
        this.mContext = context;
        this.mItems = items;
    }@
    Override public CalendarItemsAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(mContext);
        View itemView = inflater.inflate(R.layout.calendar_item_layout, parent, false);
        ViewHolder viewHolder = new ViewHolder(itemView);
        return viewHolder;
    }@
    Override public void onBindViewHolder(final CalendarItemsAdapter.ViewHolder holder, int position) {
        final Calender_Data item = mItems.get(position);
        DateFormat df = new SimpleDateFormat("dd / MM / yyyy");
        if (item.getEvent_Name().contains("$")) { //Months will contain $
            holder.TITLE.setText(item.getEvent_Name().substring(2));
            holder.TITLE.setTextSize(24);
            holder.EDateView.setVisibility(View.GONE);
            holder.SDate_View.setVisibility(View.GONE);
            holder.TITLE.setTextColor(mContext.getResources().getColor(android.R.color.white));
            holder.mView.setBackgroundColor(mContext.getResources().getColor(android.R.color.holo_blue_dark));
        } else {
            if (item.getEvent_Name().contains("##")) { //Years will contain ##
                holder.TITLE.setText(item.getEvent_Name().substring(2));
                holder.TITLE.setTextSize(24);
                holder.TITLE.setTextColor(mContext.getResources().getColor(android.R.color.white));
                holder.EDateView.setVisibility(View.GONE);
                holder.SDate_View.setVisibility(View.GONE);
                holder.mView.setBackgroundColor(mContext.getResources().getColor(android.R.color.holo_green_dark));
            } else {
                holder.TITLE.setText(item.getEvent_Name());
                if (item.getStartDate().equals(item.getEndDate())) {
                    holder.SDateText.setText(df.format(new Date(item.getStartDate())));
                    holder.EDateView.setVisibility(View.GONE);
                } else {
                    holder.SDateText.setText(df.format(new Date(item.getStartDate())));
                    holder.SDate.setText("Start Date:");
                    holder.EDateText.setText(df.format(new Date(item.getEndDate())));
                }
            }
        }
    }@
    Override public int getItemCount() {
        return mItems.size();
    }
    public static class ViewHolder extends RecyclerView.ViewHolder {
        public TextView TITLE;
        public View SDate_View;
        public TextView SDate;
        public TextView SDateText;
        public TextView EDateText;
        public View EDateView;
        public View mView;
        public ViewHolder(View itemView) {
            super(itemView);
            TITLE = (TextView) itemView.findViewById(R.id.calendar_title);
            SDateText = (TextView) itemView.findViewById(R.id.calendar_date_text);
            SDate = (TextView) itemView.findViewById(R.id.calendar_date);
            EDateText = (TextView) itemView.findViewById(R.id.calendar_Enddate_text);
            EDateView = itemView.findViewById(R.id.calendar_end_view);
            SDate_View = itemView.findViewById(R.id.Calendar_StartDate_View);
            mView = itemView;
        }
    }
}
```

CoursesItemAdapter.java

```
package com.example.baltu.myapplication.Data_Provider.Adapters;
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.TextView;
import com.example.baltu.myapplication.DataTypes.TimeTable_Classes;
import com.example.baltu.myapplication.R;
import com.example.baltu.myapplication.TimeTables.Courses_Viewer_Editor_Activity;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.List;
public class CoursesItemAdapter extends RecyclerView.Adapter < CoursesItemAdapter.ViewHolder >
 {
    private static DateFormat timeformat = new SimpleDateFormat("HH:mm");
    private List < TimeTable_Classes > mItems;
    private Context mContext;
    public CoursesItemAdapter(Context context, List < TimeTable_Classes > items) {
        this.mContext = context;
        this.mItems = items;
    }@
    Override public CoursesItemAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int vie
wType) {
        LayoutInflater inflater = LayoutInflater.from(mContext);
        View itemView = inflater.inflate(R.layout.lecture_layout, parent, false);
        ViewHolder viewHolder = new ViewHolder(itemView);
        return viewHolder;
    }@
    Override public void onBindViewHolder(final CoursesItemAdapter.ViewHolder holder, int posi
tion) {
        final TimeTable_Classes item = mItems.get(position);
        holder.cName.setText(item.getCourse());
        holder.starttime.setText(item.getStarting_time());
        holder.endtime.setText(item.getFinishing_time());
        holder.courseid.setText(item.getID());
        holder.mView.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                Intent ce = new Intent(mContext, Courses_Viewer_Editor_Activity.class);
                ce.putExtra("name", item.getCourse());
                ce.putExtra("day", item.getDay_of_week());
                ce.putExtra("start", item.getStarting_time());
                ce.putExtra("end", item.getFinishing_time());
```

```java
                    ce.putExtra("note", item.getNotes());
                    ce.putExtra("id", item.getID());
                    mContext.startActivity(ce);
                }
            });
        }@
        Override public int getItemCount() {
            return mItems.size();
        }
        public static class ViewHolder extends RecyclerView.ViewHolder {
            public TextView cName;
            public TextView starttime;
            public TextView endtime;
            public TextView courseid;
            public View mView;
            public ViewHolder(View itemView) {
                super(itemView);
                cName = (TextView) itemView.findViewById(R.id.Task_course_name);
                starttime = (TextView) itemView.findViewById(R.id.Startt);
                endtime = (TextView) itemView.findViewById(R.id.Endt);
                courseid = (TextView) itemView.findViewById(R.id.course_id);
                mView = itemView;
            }
        }
    }
```

ExamsItemsAdapter.java

```java
package com.example.baltu.myapplication.Data_Provider.Adapters;
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.TextView;
import com.example.baltu.myapplication.DataTypes.Exam_class;
import com.example.baltu.myapplication.R;
import com.example.baltu.myapplication.TimeTables.Exams_Viewer_Editor_Activity;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
public class ExamsItemsAdapter extends RecyclerView.Adapter < ExamsItemsAdapter.ViewHolder > {

    private List < Exam_class > mItems;
    private Context mContext;
    int f;
    public ExamsItemsAdapter(Context context, List < Exam_class > items) {
        this.mContext = context;
        this.mItems = items;
    }@
    Override public ExamsItemsAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int view
Type) {
        LayoutInflater inflater = LayoutInflater.from(mContext);
        View itemView = inflater.inflate(R.layout.exam_layout, parent, false);
        ViewHolder viewHolder = new ViewHolder(itemView);
        return viewHolder;
    }@
    Override public void onBindViewHolder(final ExamsItemsAdapter.ViewHolder holder, int posit
ion) {
        final Exam_class item = mItems.get(position);
        holder.TName.setText(item.getCourse());
        holder.TDescription.setText(item.getDiscription());
        Date d = new Date(item.getExam_Date()); //String ddate=d.toString();
        DateFormat dtd = new SimpleDateFormat("EEE dd MMM yyyy");
        DateFormat dtt = new SimpleDateFormat("hh:mm aaa");
        holder.DeadLineT.setText(dtt.format(d));
        holder.DeadLineD.setText(dtd.format(d));
        holder.Taskid.setText(item.getID());
        holder.mView.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                Intent In = new Intent(mContext, Exams_Viewer_Editor_Activity.class);
                In.putExtra("id", item.getID());
                In.putExtra("name", item.getCourse());
                In.putExtra("desc", item.getDiscription());
                In.putExtra("date", item.getExam_Date());
                In.putExtra("notes", item.getNotes());
                mContext.startActivity(In);
            }
        });
    }@
    Override public int getItemCount() {
        return mItems.size();
    }
    public static class ViewHolder extends RecyclerView.ViewHolder {
        public TextView TName;
        public TextView TDescription;
        public TextView DeadLineT;
        public TextView DeadLineD;
        public TextView Taskid;
        public View mView;
        public ViewHolder(View itemView) {
            super(itemView);
            TName = (TextView) itemView.findViewById(R.id.Task_course_name);
            TDescription = (TextView) itemView.findViewById(R.id.Task_Description);
            DeadLineT = (TextView) itemView.findViewById(R.id.DeadlineTime);
            DeadLineD = (TextView) itemView.findViewById(R.id.DeadlineDate);
            Taskid = (TextView) itemView.findViewById(R.id.TaskID);
            mView = itemView;
        }
    }
}
```

TasksItemsAdapter.java

```java
package com.example.baltu.myapplication.Data_Provider.Adapters;
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.TextView;
import com.example.baltu.myapplication.DataTypes.Tasks_Data;
import com.example.baltu.myapplication.R;
import com.example.baltu.myapplication.TimeTables.Tasks_Viewer_Editor_Activity;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
```

```
public class TasksItemsAdapter extends RecyclerView.Adapter < TasksItemsAdapter.ViewHolder > {
    private List < Tasks_Data > mItems;
    private Context mContext;
    public TasksItemsAdapter(Context context, List < Tasks_Data > items) {
        this.mContext = context;
        this.mItems = items;
    }@
    Override public TasksItemsAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(mContext);
        View itemView = inflater.inflate(R.layout.task_layout, parent, false);
        ViewHolder viewHolder = new ViewHolder(itemView);
        return viewHolder;
    }@
    Override public void onBindViewHolder(final TasksItemsAdapter.ViewHolder holder, int position) {
        final Tasks_Data item = mItems.get(position);
        holder.TName.setText(item.getCourse());
        holder.TDescription.setText(item.getDiscription());
        Date d = new Date(item.getDeadline_Date()); //String ddate=d.toString();
        DateFormat dtd = new SimpleDateFormat("EEE dd MMM yyyy");
        DateFormat dtt = new SimpleDateFormat("hh:mm aaa");
        holder.DeadLineT.setText(dtt.format(d));
        holder.DeadLineD.setText(dtd.format(d));
        holder.Taskid.setText(item.getID());
        holder.mView.setOnClickListener(new View.OnClickListener() {@
            Override public void onClick(View v) {
                Intent In = new Intent(mContext, Tasks_Viewer_Editor_Activity.class);
                In.putExtra("id", item.getID());
                In.putExtra("name", item.getCourse());
                In.putExtra("desc", item.getDiscription());
                In.putExtra("date", item.getDeadline_Date());
                In.putExtra("notes", item.getNotes());
                mContext.startActivity(In);
            }
        });
    }@
    Override public int getItemCount() {
        return mItems.size();
    }
    public static class ViewHolder extends RecyclerView.ViewHolder {
        public TextView TName;
        public TextView TDescription;
        public TextView DeadLineT;
        public TextView DeadLineD;
        public TextView Taskid;
        public View mView;
        public ViewHolder(View itemView) {
            super(itemView);
            TName = (TextView) itemView.findViewById(R.id.Task_course_name);
            TDescription = (TextView) itemView.findViewById(R.id.Task_Description);
            DeadLineT = (TextView) itemView.findViewById(R.id.DeadlineTime);
            DeadLineD = (TextView) itemView.findViewById(R.id.DeadlineDate);
            Taskid = (TextView) itemView.findViewById(R.id.TaskID);
            mView = itemView;
        }
    }
}
```