安卓系统：作业系统报告书

班级 : 网络91761

姓名 : 何子铭 Hazim bin Shaiful Bahari 学号 : 9999176132

: 谢安娜 Nur Adina Shahira 学号 : 9999176177

Contents

[Introduction 介绍 1](#_Toc8758079)

[Database Structure 数据库结构 2](#_Toc8758080)

[Functionalities 功能说明 3](#_Toc8758081)

# Introduction

HW4Uapp (Homework 4 U application) is an application developed using Android in Eclipse. This application allows teachers to upload homework for their classes and the students can see the homework list. The teacher can also set the homework’s due date and instructions.

The application has 2 access levels: Teacher’s account and Student’s account. In teacher’s account, teachers have the ability to upload new homework, update the due date and class’ status and delete the homework is the record is no longer necessary.

Student’s account has different functionality. Students have restricted abilities to only view homework and has no access to update or delete any homework.

# Database Structure

This project uses one database and three tables. 2 tables store user information and the other one stores homework’s information. These tables are created using DbHelper class. Information written in DbHelper is created automatically when the system runs.

|  |
| --- |
| **public** **void** onCreate(SQLiteDatabase db) {  String sql="create table tcinfo(\_id integer primary key autoincrement,tcName varchar(50), tcUsername varchar(50), tcEmail varchar(50), tcPassword varchar(20))";  db.execSQL(sql);  db.execSQL("insert into tcinfo(tcName, tcUsername, tcEmail, tcPassword) values('adina', 'aadina', 'adina@hw4u.com' ,'aadina12')");  db.execSQL("insert into tcinfo(tcName, tcUsername, tcEmail, tcPassword) values('hazim', 'hahazim', 'hahazim@hw4u.com' ,'hahazim12')");    String sql2="create table stinfo(\_id integer primary key autoincrement,stName varchar(50), stUsername varchar(50), stEmail varchar(50), stPassword varchar(20))";  db.execSQL(sql2);  db.execSQL("insert into stinfo(stName, stUsername, stEmail, stPassword) values('namop', 'nanamop', 'nanamop@hw4u.com' ,'nanamop12')");  db.execSQL("insert into stinfo(stName, stUsername, stEmail, stPassword) values('juhmop', 'juhjuhmop', 'juhjuhmop@hw4u.com' ,'juhjuhmop12')");    String sql3="create table homework(\_id integer primary key autoincrement,hwName varchar(50), hwIns varchar(500), hwClass varchar(50), hwDue varchar(50), hwTeacher varchar(50), hwStatus varchar(50))";  db.execSQL(sql3);  db.execSQL("insert into homework(hwName, hwIns, hwClass, hwDue, hwTeacher, hwStatus) values('Essay', 'List down capital cities of the world in 500 words', 'English' ,'2019-05-12', 'hahazim', 'ENDED')");  db.execSQL("insert into homework(hwName, hwIns, hwClass, hwDue, hwTeacher, hwStatus) values('Integration', 'Intergrate functions in textbook page 60', 'Mathematics' ,'2019-05-15', 'aadina', 'IN SESSION')");  db.execSQL("insert into homework(hwName, hwIns, hwClass, hwDue, hwTeacher, hwStatus) values('Eclipse', 'Install Eclipse', 'Android' ,'2019-05-20', 'hahazim', 'ENDED')");  db.execSQL("insert into homework(hwName, hwIns, hwClass, hwDue, hwTeacher, hwStatus) values('Patent Document', 'Create a patent document using not less than 300 words', 'Innovation' ,'2019-05-25', 'aadina', 'IN SESSION')");  } |

Code : Initialize database, tables and records

# Functionalities

* Homepage

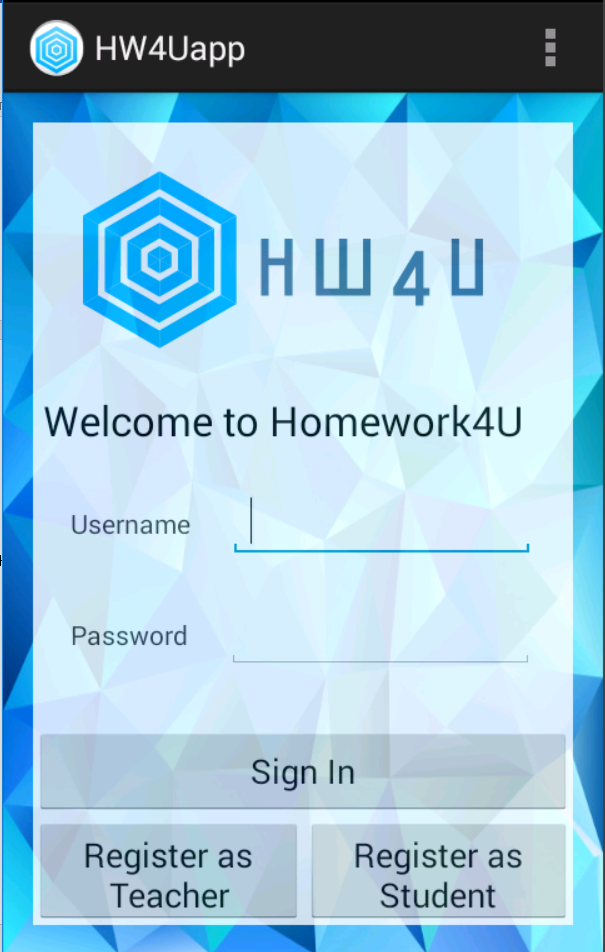


Figure : HW4Uapp homepage

On the homepage, registered users can login while new users can register for new accounts. The layout is coded in **activity\_main.xml** and logging in and registering processes are done in **DbService.java**. This page is coded in **MainActivity.java**.

* Login Function

On HW4Uapp’s main page, it uses only 1 login form but it checks 2 different tables. This function enables teachers and students to log in on the same page.

|  |
| --- |
| **private** **void** initview() {  // **TODO** Auto-generated method stub    btn\_login.setOnClickListener(**new** OnClickListener(){  @Override  **public** **void** onClick(View arg0) {  // **TODO** Auto-generated method stub  String etname=et\_username.getText().toString();  String etpass=et\_password.getText().toString();  *teachername* = et\_username.getText().toString();    **if** (TextUtils.*isEmpty*(etname)|| TextUtils.*isEmpty*(etpass))  {  Toast.*makeText*(MainActivity.**this**,"This field cannot be empty",Toast.*LENGTH\_SHORT*).show();  //Toast.makeText(getApplicationContext(),"This field cannot be empty",Toast.LENGTH\_SHORT).show();  }  **else**  {  DbService dService=**new** DbService(MainActivity.**this**);  **boolean** flag=dService.loginTc(etname,etpass);  **if**(flag)  {  Intent intent=**new** Intent(MainActivity.**this**,HomeTeacherActivity.**class**);  intent.putExtra("teachername", etname);  startActivity(intent);  et\_username.setText("");  et\_password.setText("");  }    **else**  {  DbService dService2=**new** DbService(MainActivity.**this**);  **boolean** flag2=dService2.loginStu(etname,etpass);  **if**(flag2)  {  Intent intent=**new** Intent(MainActivity.**this**,HomeStudentActivity.**class**);  startActivity(intent);  et\_username.setText("");  et\_password.setText("");  }  **else**  {  et\_username.setText("");  et\_password.setText("");  Toast.*makeText*(getApplicationContext(),"Login Error",Toast.*LENGTH\_SHORT*).show();  }  }  }    }}); |

Code : Code on MainActivity.java

On **MainActivity.java**, the login button has an onClick listener that calls the method **loginTc** and **loginStu** from class **DbService.java**. Using **if…else** conditional statement, when the program fails to find matching data from **loginTc**’s method, it will find matching data from **loginStu** method.

|  |
| --- |
| **public** **boolean** loginTc(String username,String password) //Teacher login  {  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String sqlTc="select \* from tcinfo where tcUsername=? and tcPassword=?";  Cursor cursorTc=sdb.rawQuery(sqlTc, **new** String[]{username,password});  **if**(cursorTc.moveToFirst()==**true**)  {  cursorTc.close();  **return** **true**;  }    **else**  {  **return** **false**;  }  } |

Code : LoginTc method in DbService class

This method takes value from EditText controls in **MainActivity.java** and uses cursor to check if the input matches any data in table **tcinfo**.

|  |
| --- |
| **public** **boolean** loginStu(String username,String password) //Student Login  {  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String sqlStu="select \* from stinfo where stUsername=? and stPassword=?";  Cursor cursorStu=sdb.rawQuery(sqlStu, **new** String[]{username,password});  **if**(cursorStu.moveToFirst()==**true**)  {  cursorStu.close();  **return** **true**;  }  **else**  {  **return** **false**;  }  } |

Code : loginStu method in DbService class

This method takes value from EditText controls in **MainActivity.java** and uses cursor to check if the input matches any data in table **stinfo**.

* Register Account Function

HW4U’s homepage provides 2 buttons to register new accounts, each specifically for teacher and student. These 2 pages have similar layout but they call different methods to insert new data into respective databases.

|  |  |
| --- | --- |
|  |  |

Figure : Differences in register pages for teacher and student

On **RegisterTeaActivity.java**, the ‘sign up’ button calls for method **registerTc** in **DbService** class. This function also depends on getters and setters for EditText controls in **UserTc.java** class.

|  |
| --- |
| Log.*i*("TAG", teachername + "\_" + teacherusername + "\_" + teacheremail + "\_" + teacherpassword);  DbService tcService = **new** DbService(RegisterTeaActivity.**this**);  UserTc teacher = **new** UserTc();  teacher.setTcName(teachername);  teacher.setTcUsername(teacherusername);  teacher.setTcEmail(teacheremail);  teacher.setTcPassword(teacherpassword);  tcService.registerTc(teacher);  Toast.*makeText*(RegisterTeaActivity.**this**, "Register Success, please re-login to access", Toast.*LENGTH\_LONG*)  .show();  mainpage(); |

Code : Using userTc method in RegisterTeaActivity.java

|  |
| --- |
| **package** com.example.hw4uapp;  **import** java.io.Serializable;  **public** **class** UserTc **implements** Serializable {    **private** **int** \_id;  **private** String tcName;  **private** String tcUsername;  **private** String tcEmail;  **private** String tcPassword;  **public** UserTc() {  **super**();  // **TODO** Auto-generated constructor stub  }    **public** UserTc(String tcName, String tcUsername, String tcEmail, String tcPassword) {  **super**();  **this**.tcName = tcName;  **this**.tcUsername = tcUsername;  **this**.tcEmail = tcEmail;  **this**.tcPassword = tcPassword;  }  **public** **int** get\_id() {  **return** \_id;  }  **public** **void** set\_id(**int** \_id) {  **this**.\_id = \_id;  }  **public** String getTcName() {  **return** tcName;  }  **public** **void** setTcName(String tcName) {  **this**.tcName = tcName;  }  **public** String getTcUsername() {  **return** tcUsername;  }  **public** **void** setTcUsername(String tcUsername) {  **this**.tcUsername = tcUsername;  }  **public** String getTcEmail() {  **return** tcEmail;  }  **public** **void** setTcEmail(String tcEmail) {  **this**.tcEmail = tcEmail;  }  **public** String getTcPassword() {  **return** tcPassword;  }  **public** **void** setTcPassword(String tcPassword) {  **this**.tcPassword = tcPassword;  }  } |

Code : Getters and setters in UserTc.java

By clicking the ‘Sign Up’ button, the information entered in the register page will be inserted into **tcinfo** table using **registerTc** method in **DbService.java** class.

|  |
| --- |
| **public** **boolean** registerTc(UserTc user){ //register teacher  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String sql="insert into tcinfo(tcName,tcUsername,tcEmail,tcPassword) values(?,?,?,?)";  Object obj[]={user.getTcName(),user.getTcUsername(),user.getTcEmail(),user.getTcPassword()};  sdb.execSQL(sql, obj);  **return** **true**;  } |

Code : registerTc method in DbService class

The same code is used for student’s registration.

* Logging in Student’s account

Demonstration to log in student’s account uses below information:

Username: nanamop

Password: nanamop12

When login succeeds, the student will be redirected to a Student’s homepage.

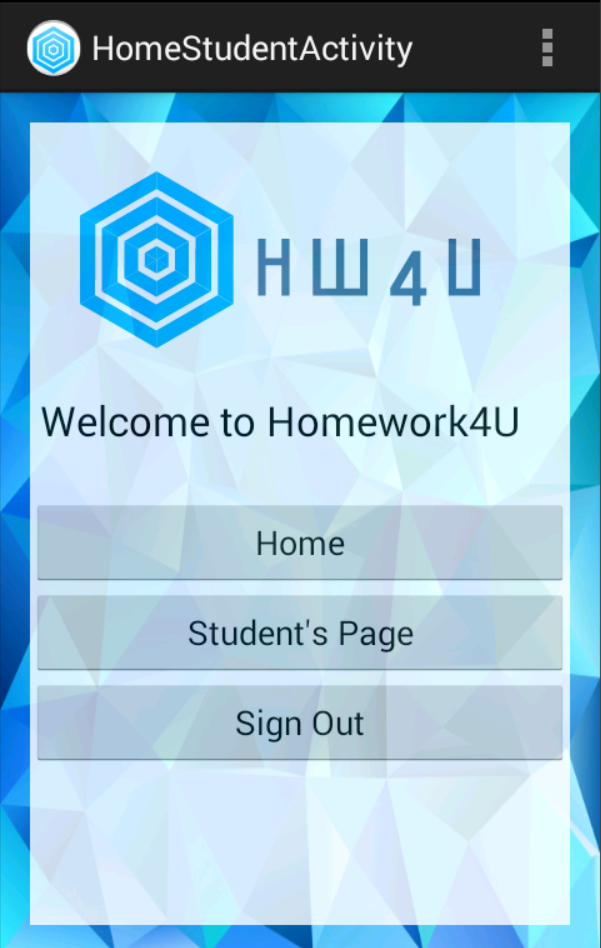


Figure : Student's homepage

The home button will open the homepage again. If the student’s click on the ‘Student’s Page’ button, the student can see the list of homework given by the teacher.

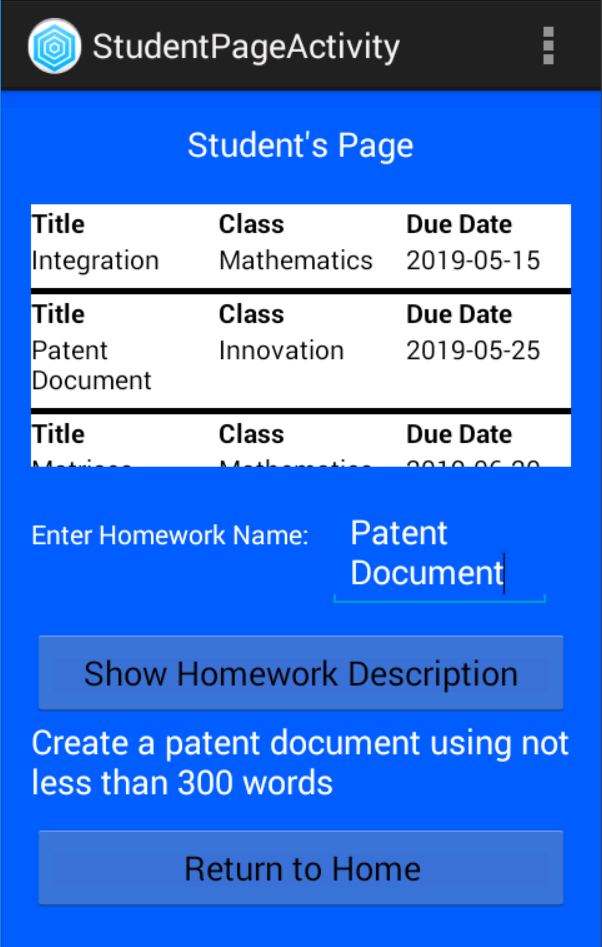


Figure : Student's Page

The student can view the list of homework in **ListView**. When the student inserts the homework name and clicks on the ‘Show Homework Description’ button, a detailed instruction of the specific homework will appear.

|  |
| --- |
| //BUTTON SELECT EVENT  OnClickListener select = **new** OnClickListener(){  @Override  **public** **void** onClick(View v) {  // **TODO** Auto-generated method stub  String hwname = et\_hwname.getText().toString().trim();  select(hwname);  }  };    btn\_Select.setOnClickListener(select);  et\_hwname.setText("");  }      //SELECT FUNCTION  **public** **void** select(String name)  {  TextView showInst=(TextView) **super**.findViewById(R.id.*tvdisplay*);  EditText et\_hwname=(EditText) **super**.findViewById(R.id.*etHwName*);  String hwinst ="";  Cursor hw =uService.querydata(name);    **if** (hw==**null**)  {  android.app.AlertDialog.Builder adInfo=**new** AlertDialog.Builder(**this**);  adInfo.setTitle("Homework Doesn't Exist");  adInfo.setMessage("There is no such homework");  adInfo.setIcon(R.drawable.*logo*);  adInfo.create();  adInfo.show();  et\_hwname.setText("");  showInst.setText("");  }    **else**  {  **for**(hw.moveToFirst();!hw.isAfterLast();hw.moveToNext())  {  String hwname=hw.getString(1);  String hwins=hw.getString(2);  String hwClass=hw.getString(3);  String hwDue=hw.getString(4);  String hwTeacher=hw.getString(5);  String hwStatus=hw.getString(6);  String data=hwins;  hwinst=data;  }    showInst.setText(hwinst);    }    }  @Override  **public** **boolean** onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.*student\_page*, menu);  **return** **true**;  } |

Code : Coding StudentPageActivity.java

**initlist()** functions allows the listview to display data from database.

|  |
| --- |
| //INITLIST - Implement data initialization, fetch user data and display them in a list.  **public** **void** initlist(){    ListView lvHomework = (ListView) **this**.findViewById(R.id.*listView1*);    //Get the cursor  Cursor hw = uService.queryall();  **if**(hw.isFirst())  {  Log.*i*("TAG","ok");  SimpleCursorAdapter adapter=**new** SimpleCursorAdapter(**this**,R.layout.*item*, hw,**new** String[]{"hwName", "hwClass", "hwDue","hwTeacher", "hwStatus"}, **new** **int**[]{R.id.*itemhwName*, R.id.*itemhwClass*, R.id.*itemhwDue*,R.id.*itemhwTc*, R.id.*itemhwSt*},0);  //Create a SimpleCursorAdapter adapter to bind data to item display controls    lvHomework.setAdapter(adapter);  }  } |

Code : initlist() method

**initview()** method shows the instruction in TextView when the ‘Show Homework Description’ button is clicked.

|  |
| --- |
| //INITVIEW  **public** **void** initview()  {  TextView showInst=(TextView) **super**.findViewById(R.id.*tvdisplay*);  String data1="";  String data2="";  Cursor hw =uService.queryall();  **if**(hw.moveToFirst()==**true**)  {  String hwid=hw.getString(0);  String hwname=hw.getString(1);  String hwins=hw.getString(2);  String hwclass=hw.getString(3);  String hwdue=hw.getString(4);  String hwteacher=hw.getString(5);  String hwstatus=hw.getString(6);  String data=hwid+" "+hwname+" "+hwins+" "+hwclass+" "+hwdue+" "+hwteacher+" "+hwstatus;    **while**(hw.moveToNext())  {  hwid=hw.getString(0);  hwname=hw.getString(1);  hwins=hw.getString(2);  hwclass=hw.getString(3);  hwdue=hw.getString(4);  hwteacher=hw.getString(5);  hwstatus=hw.getString(6);    data2 =data2+"\n"+ hwid +" "+ hwname+" "+hwins+" "+hwclass+" "+hwdue+" "+hwteacher+" "+hwstatus;  Log.*i*("TAG",data2);  //Toast.makeText(getApplicationContext(), "User Information:"+data2,Toast.LENGTH\_SHORT).show();  }  data1=data+"\r"+data2;  showInst.setText(data1);  }  } |

Code : initview() method

Suitable toast also appears if the spelling of homework’s name is incorrect or if the field is empty.

**queryall** method from **DbService** class is also used to query all information from database before adapted into ListView.

|  |
| --- |
| //UNTUK INITVIEW NGAN INITLIST  **public** Cursor queryall(){  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String[] columns = **new** String[]{"\_id","hwName", "hwIns", "hwClass","hwDue", "hwTeacher", "hwStatus"};  Cursor cursor=sdb.query("homework", columns,**null**, **null**, **null**,**null**, **null**, **null**);  **if**(!cursor.moveToNext()){  cursor.close();  **return** **null**;  }  **return** cursor;  } |

**querydata** method from **DbService** class is used to query only selected data’s information from database to display in TextView.

|  |
| --- |
| //GUNA UNTUK SELECT  **public** Cursor querydata(String uname){  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String[] columns = **new** String[]{"\_id","hwName", "hwIns","hwClass","hwDue", "hwTeacher", "hwStatus"};  Cursor cursor=sdb.query("homework", columns, "hwName='"+uname+"'", **null**, **null**,**null**, **null**, **null**);  **if**(!cursor.moveToNext()){  cursor.close();  **return** **null**;  }  **return** cursor;  } |

* Logging in teacher’s account

Demonstration to log in teacher’s account uses below information:

Username: hahazim

Password: hahazim12

When login succeeds, the student will be redirected to a teacher’s homepage.

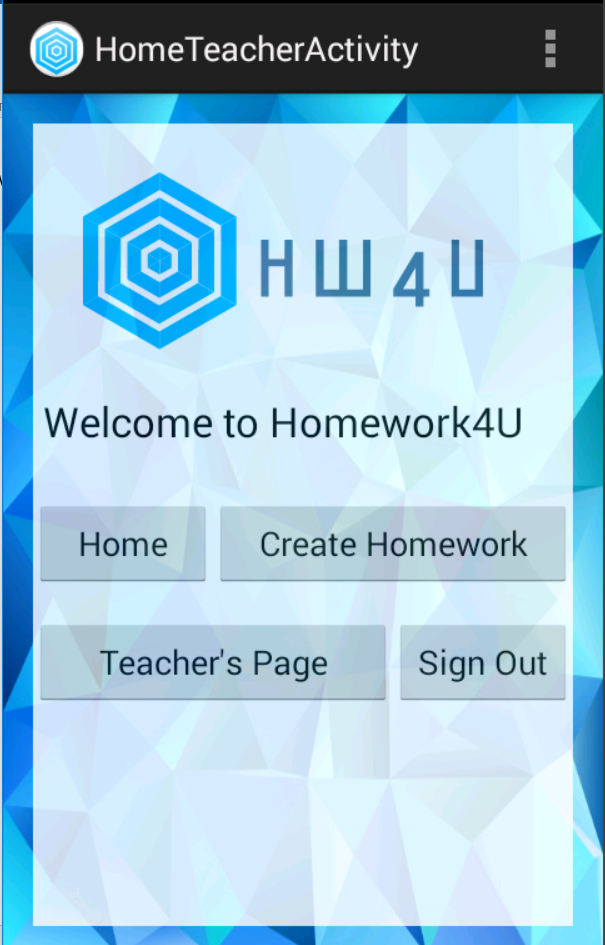


Figure : Teacher's homepage

* Teacher creating homework

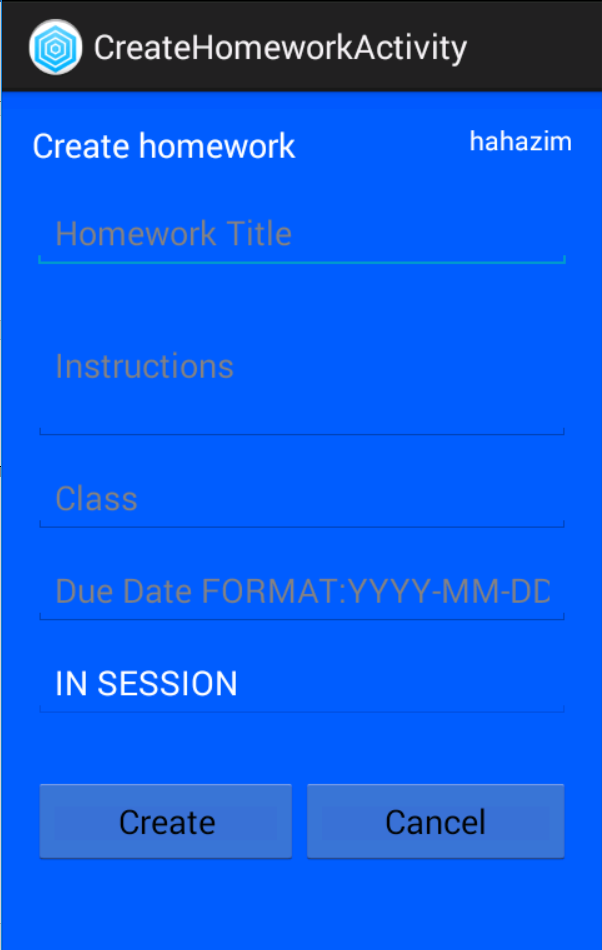


Figure : Teacher's creating homework page

Teacher’s creating homework function is the same as register function.

On **CreateHomeworkActivity.java**, the ‘Create button calls for method **createHomework** in **DbService** class. This function also depends on getters and setters for EditText controls in **UserHw.java** class.

|  |
| --- |
| Log.*i*("TAG", homeworkname + "\_" + homeworkinstruction + "\_" + homeworkclass + "\_" + homeworkduedate + "\_" + classstatus + "\_" + gettcname);  DbService hwService = **new** DbService(CreateHomeworkActivity.**this**);  UserHw homework = **new** UserHw();  homework.setHwName(homeworkname);  homework.setHwIns(homeworkinstruction);  homework.setHwClass(homeworkclass);  homework.setHwDue(homeworkduedate);  homework.setHwStatus(classstatus);  homework.setHwTeacher(gettcname);  hwService.createHomework(homework);  Toast.*makeText*(CreateHomeworkActivity.**this**, "Homework Created!", Toast.*LENGTH\_LONG*).show();  teacherpage(); |

Code : Create homework function

|  |
| --- |
| **package** com.example.hw4uapp;  **import** java.io.Serializable;  **public** **class** UserHw **implements** Serializable{    **private** **int** \_id;  **private** String hwName;  **private** String hwIns;  **private** String hwClass;  **private** String hwDue;  **private** String hwStatus;  **private** String hwTeacher;    **public** UserHw() {  **super**();  // **TODO** Auto-generated constructor stub  }    **public** UserHw(String hwName, String hwIns, String hwClass, String hwDue) {  **super**();  **this**.hwName = hwName;  **this**.hwIns = hwIns;  **this**.hwClass = hwClass;  **this**.hwDue = hwDue;  **this**.hwStatus = hwStatus;  **this**.hwTeacher = hwTeacher;  }  **public** **int** get\_id() {  **return** \_id;  }  **public** **void** set\_id(**int** \_id) {  **this**.\_id = \_id;  }  **public** String getHwName() {  **return** hwName;  }  **public** **void** setHwName(String hwName) {  **this**.hwName = hwName;  }  **public** String getHwIns() {  **return** hwIns;  }  **public** **void** setHwIns(String hwIns) {  **this**.hwIns = hwIns;  }  **public** String getHwClass() {  **return** hwClass;  }  **public** **void** setHwClass(String hwClass) {  **this**.hwClass = hwClass;  }  **public** String getHwDue() {  **return** hwDue;  }  **public** **void** setHwDue(String hwDue) {  **this**.hwDue = hwDue;  }  **public** String getHwStatus() {  **return** hwStatus;  }  **public** **void** setHwStatus(String hwStatus) {  **this**.hwStatus = hwStatus;  }  **public** String getHwTeacher() {  **return** hwTeacher;  }  **public** **void** setHwTeacher(String hwTeacher) {  **this**.hwTeacher = hwTeacher;  }  } |

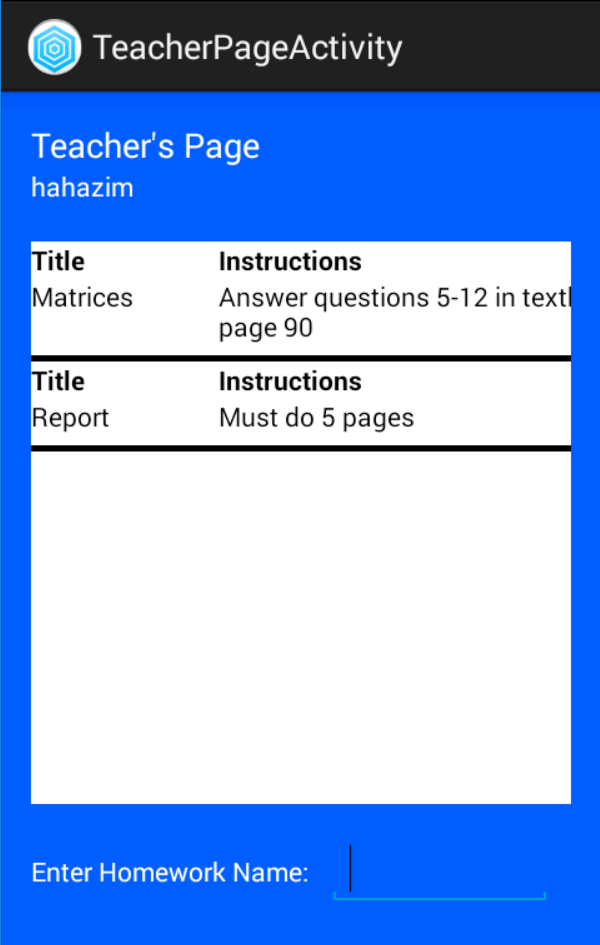
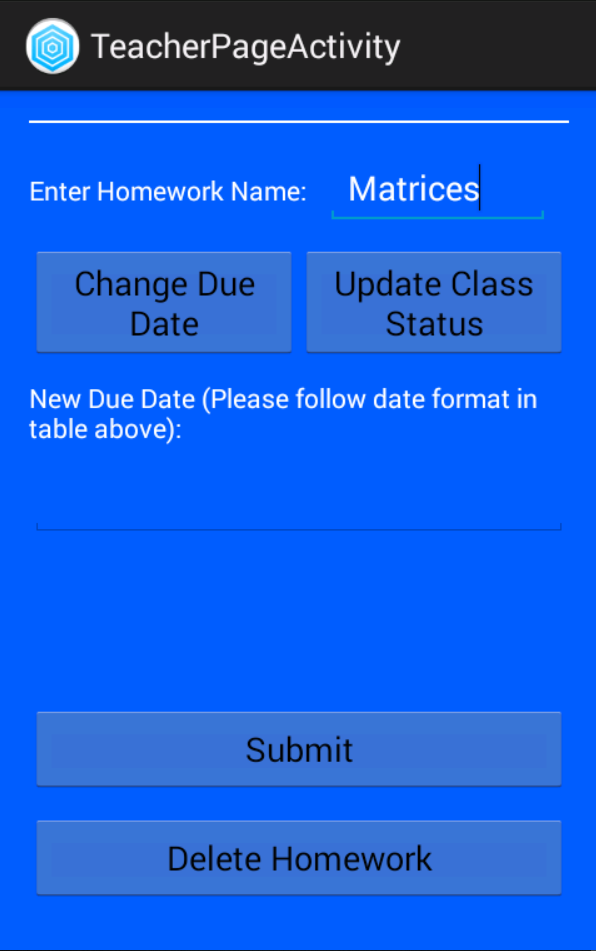
Code : UserHw getters and setters

|  |
| --- |
| /\* Teachers' Functions for Create Homework \*/    **public** **boolean** createHomework(UserHw newhw)  {  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String sql="insert into homework(hwName,hwIns,hwClass,hwDue,hwTeacher,hwStatus) values(?,?,?,?,?,?)";  Object obj[]={newhw.getHwName(),newhw.getHwIns(),newhw.getHwClass(),newhw.getHwDue(), newhw.getHwTeacher(), newhw.getHwStatus()};  sdb.execSQL(sql, obj);  **return** **true**;  }    /\* Teachers' Functions for Create Homework End \*/ |

Code : CreateHomework method

* Teacher’s updating and deleting homework

From the teacher’s homepage, teachers can view and commit changes to homework’s information by accessing **TeacherPage.java**.

Teachers can change due date of a homework by inserting homework’s name in the EditText and clicking submit. The ‘Submit’ button will call 2 different methods, according to the data filled in the EditText control using **if…else** conditional statement.

|  |
| --- |
| OnClickListener SubmitUpdate = **new** OnClickListener(){  @Override  **public** **void** onClick(View v)  {  String homeworkname = et\_hwname.getText().toString();  String newdate = et\_newDate.getText().toString();  String status = et\_UpStatus.getText().toString();  String gettcname=MainActivity.*teachername*.toString();    **if**(TextUtils.*isEmpty*(status))  {  initviewupdatedate(homeworkname, newdate);  initlist(gettcname);  et\_UpStatus.setVisibility(View.*INVISIBLE*);  tv\_UpdateStatus.setVisibility(View.*INVISIBLE*);    et\_newDate.setVisibility(View.*INVISIBLE*);  tv\_ChangeDue.setVisibility(View.*INVISIBLE*);    et\_hwname.setText("");  et\_newDate.setText("");  et\_UpStatus.setText("");    }  **if**(TextUtils.*isEmpty*(newdate))  {  initviewupdatestatus(homeworkname, status);  initlist(gettcname);  et\_UpStatus.setVisibility(View.*INVISIBLE*);  tv\_UpdateStatus.setVisibility(View.*INVISIBLE*);    et\_newDate.setVisibility(View.*INVISIBLE*);  tv\_ChangeDue.setVisibility(View.*INVISIBLE*);    et\_hwname.setText("");  et\_newDate.setText("");  et\_UpStatus.setText("");  }  }  }; |

Code : SubmitUpdate button onClick Listener

To update the due date, the button calls **initviewupdatedate()** function.

|  |
| --- |
| //INITVIEWUPDATEDATE - Teacher update due date  **public** **void** initviewupdatedate(String hwname,String newduedate)  {  uService.updateDate(hwname, newduedate);  } |

Code : initviewupdatedate() function

The **Initviewupdatedate()** function calls **updateDate** method from **DbService.java** class.

|  |
| --- |
| /\* Teachers' Functions for Update Date \*/    **public** **void** updateDate(String hwname,String newduedate)  {  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String sql="update homework set hwDue='"+ newduedate +"' where hwName='"+ hwname +"'";  sdb.execSQL(sql);  }    /\* Teachers' Functions for Update Date End \*/ |

Code : updateDate method

To update the class status, the button calls **initviewupdatestatus() function**.

|  |
| --- |
| //INITVIEWUPDATESTATUS - Teacher update status  **public** **void** initviewupdatestatus(String hwname,String newstatus)  {  uService.updateStatus(hwname, newstatus);  } |

Code : initviewupdatestatus() function

The **initviewupdatestatus ()** function calls **updateStatus** method from **DbService.java** class.

|  |
| --- |
| /\* Teachers' Functions for Update Status \*/    **public** **void** updateStatus(String hwname,String newstatus)  {  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String sql="update homework set hwStatus='"+ newstatus +"' where hwName='"+ hwname +"'";  sdb.execSQL(sql);  }    /\* Teachers' Functions for Update Status End \*/ |

Code : updateStatus method

* Teacher deleting homework

When the teacher clicks on ‘Delete Homework’ button, it will call the **initviewdeletehw()**.

|  |
| --- |
| //INITVIEWDELETEHW - Teacher delete homework  **public** **void** initviewdeletehw(**final** String hwname, **final** String teachername)  {  Cursor hw =uService.querydata(hwname);  **if** (hw==**null**)  {  Toast.*makeText*(getApplicationContext(),"No such data",Toast.*LENGTH\_SHORT*).show();  }    **else**  {  android.app.AlertDialog.Builder dialog = **new** AlertDialog.Builder(**this**);  dialog.setTitle("Confirm delete");  dialog.setMessage("Are you sure to delete? This action cannot be undone");  dialog.setIcon(R.drawable.*logo*);  dialog.setPositiveButton("Confirm", **new** DialogInterface.OnClickListener() {  @Override  **public** **void** onClick(DialogInterface dialog, **int** which) {    uService.deletehw(hwname, teachername);  initlist(teachername);  }  });    dialog.setNegativeButton("Cancel", **new** DialogInterface.OnClickListener() {  @Override  **public** **void** onClick(DialogInterface dialog, **int** which) {  }  });  dialog.create();  dialog.show();    }  } |

Code : initviewdeletehw() function

The function will call 2 methods from **DbService.java**, the first is **querydata** method which is to show only the teacher’s homework in the ListView while the **deletehw** method will actually run the SQL statement to delete the homework from the database.

|  |
| --- |
| /\* Teachers' Functions for Deleting Homework \*/    **public** **void** deletehw(String hwname, String teachername)  {  SQLiteDatabase sdb=dbHelper.getReadableDatabase();  String sql="delete from homework where hwName='"+ hwname +"' AND hwTeacher='"+ teachername +"'";  sdb.execSQL(sql);  }    /\* Teachers' Functions for Deleting Homework End \*/ |

Code : deletehw method