

**MINISTRY OF HIGHER EDUCATION  
AND SCIENTIFIC RESEARCH  
DIYALA UNIVERSITY  
COLLEGE OF ENGINEERING  
COMPUTER AND SOFTWARE  
ENGINEERING DEPARTMENT**



## **"DEVELOPMENT AN APPLICATION FOR ANDROID MOBILE PHONE (E-Bag)"**

**A Project Submitted To The Department Of Computer And Software  
Engineering\ College Of Engineering In Partial Fulfilment Of The  
Requirement For The Degree Of B.SC. In The Computer and Software  
Engineering.**

**BY:**

**Aws Mohammed Hameed**

**Zahraa Adnan Ghareeb**

**Alaa Yass Shwkt**

**SUP:**

**Saad M. Saleh**

**Royida A. Ibrahim**

May 2016

Rijab 1437

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿يُؤْتِي الْحِكْمَةَ مَنْ يَشَاءُ وَمَنْ يُؤْتَ الْحِكْمَةَ فَقَدْ أُوتِيَ

خَيْرًا كَثِيرًا وَمَا يَذَّكَّرُ إِلَّا أُولُو الْأَلْبَابِ﴾

(١٢٦٩ البقرة)

# الأهداء

الى من كان معي ووفقتني بكل خطوة من حياتي

الى ملاذي وسندي ومعيني

الله (جل وعلا) ..

الى من ارسله ربي رحمه لنا وصدي لنا للطريق

الحبيب محمد (صلى الله عليه وسلم) ..

الى الشفاء التي اكرمت لنا الدعاء

والتي لولاها لما امسكت قلوبنا

أمي الحبيبة ..

الى استاذي الاول وقودتي في الحياة

والدي الحنون ..

الى عائلتي المستقبلية الجميلة

ازواجنا الاحباء ( محفران )، ( عباس ) ..

الى الذين رافقتني قلوبهم وحنانهم

اخوتي وأصدقائي الاعزاء ..

الى الذي حاد أن يكون رسولا

أستاذتي الفاضلة ( رويدة الحيايلى ) ..

الى مراقبنا الشامخ .. حبا وتضحية ..

نهديكم هذا الجهد المتواضع

# شكر وتقدير

الحمد لله رب العالمين والصلاة والسلام على سيد المرسلين محمد صلى الله عليه وسلم  
الشكر والتعظيم لله العلي القدير الذي منّ علينا بنعمة الايمان والعلم  
ويشرفنا بالتقدم بالشكر والامتنان الجزيل الى الاساتذة الأفاضل "د. سعد محمد صالح"  
والاستاذة "رويدة عبد الله ابراهيم" لتفضلهما بالإشراف على هذا البحث وتوجيهاتهما  
القيمة في كل خطوة نخطوها وكل درج نسلكه.

ونتقدم بالشكر الى الاستاذ الفاضل أحمد خضير جميل لرعايته المستمرة ومساعدته..  
كما نتقدم بالشكر لرئيس قسم هندسة الحاسوب والبرمجيات د. علي جاسم عبود وجميع  
أساتذتنا الأفاضل على كل شيء. والى كل من شد ازرننا وساندنا وساعدنا .

الباحثين

اوسى ، زهراء ، الاء

## **SUPERVISORS CERTIFICATION**

We certify that this project entitled "**Development an Application for Android Mobile Phone (E-Bag)**", was prepared under our supervision at computer and software engineering department/ college of engineering by (**Aws Mohammed Hameed and Zahraa Adnan Ghareeb and Alaa Yaas Shwkt**) as a partial fulfilment of requirements for the degree of B.Sc. in computer and software engineering.

**Signature:**

**Name: Dr. Saad M. Saleh**

**Title: Lecturer**

**Date:     /     / 2016**

**signature:**

**Name: Royida A. Ibrahim**

**Title: Assistant Lecturer**

**Date:     /     / 2016**

In view of the available recommendations, I forward this project for debate by the examining committee.

**Signature:**

**Name : Dr. Ali J. Abboud**

**(Head of the department)**

**Title : Lecturer**

**Date :     /     / 2016**

## **CERTIFICATION OF THE EXAMINATION COMMITTEE**

We certify that we have read this project entitled "**Development an Application for Android Mobile Phone (E-Bag)**" and as examining committee examined the students by (**Aws Mohammed Hameed** and **Zahraa Adnan Ghareeb** and **Alaa Yaas Shwkt**) in its contents and that in our option it meets the standard of project for the degree of B.Sc. in Computer and Software Engineering.

**Signature :**

**Name :**

**Title :**

**(Member)**

**Date :**

**signature :**

**Name :**

**Title :**

**(Member)**

**Date :**

**Signature :**

**Name :**

**Title :**

**(Chairman )**

**Date :**

Approved for Computer and Software Engineering Department

**Signature:**

**Name : Dr. Ali J. Abboud**

**(Head of the department )**

**Title : Lecturer**

**Date : / / 2016**

## **Declaration**

We hereby declare that the project entitled "**Development an Application for Android Mobile Phone (E-Bag)**" is the result of our own work and includes nothing which is the outcome of work done in collaboration except as declared in the preface and specified in the text, and is not submitted a degree or diploma or other qualification at the Diyala University **or** other University or similar institution except as declared in the preface and specified in the next. We further state that no substantial part of the thesis has already been submitted, or is concurrently submitted for any such degree, diploma, or other qualification at the Diyala University **or** other any other University or similar institution except as declared in the preface and specified in the next.

**AWS MOHAMMED HAMEED**

**ZAHRAA ADNAN GHAREEB**

**ALAA YASS SHWKT**

# **TABLE OF CONTENTS**

<b>Abstract</b>	<b>VI</b>
<b>Chapter One: Introduction</b>	
1.1    Android (operating system)	12
1.2    Android software development	13
1.3    Official development tools	13
1.3.1  Android SDK	13
1.3.2  Android Debug Bridge	14
1.3.3  Android Debug Bridge	14
1.3.4  Fastboot	14
1.3.5  Android NDK	15
1.3.6  Android Open Accessory Development Kit	16
1.3.7  Native Go support	16
1.4    External hardware development	16
1.5    Android Developer Challenge	16
<b>Chapter Two: Android Persistency: SQL Databases</b>	
2.1    Structure of project	18
2.2    SQL Databases	18
2.2.1  Using SQLite on Android	18
2.2.2  Managing SQLite	19
2.3    Code of DatabaseHelper.java of E-bag	22



2.4	Code of LectureDetails.java of E-bag	26
2.5	Eclipse (software)	32
2.6	Database Tables	33
<b>Chapter Three: Showing the application and the results</b>		
3.1	Application's Windows	36
3.2	Checking the operation of the Application	40
<b>Chapter Four: Conclusions and future work</b>		
4.1	Conclusions	49
4.2	Future work	49
<b>References</b>		50

## **Abstract**

Customers in the world nowadays went to use the smart phone applications for faster access to various services' companies and other data resources. Recent studies have demonstrated the importance for any organization, private company and/or government agency to publish their profile and its services through smart phone applications. Further, many published studies have introduced the growing of number of smart phone users that get the services through websites applications. The progression in information technology and Electronic-learning represent an important motivation to developers to produce smart phone applications to facilitate the access to the required Electronic materials. According to over mentioned statements we have developed an Android smart phone application for the Computer and Software Engineering Department information. The application provides the information that frequently asked by the students such as information about courses and other important adverts. The application can also be applied to any other department and it can be extend to the whole college and university. We did it to facilitate the process of access to information relating to the curriculum and browsed and downloaded by students, and to access notifications by special plate announcements to the students by the department administration office.

# CHAPTER ONE

## INTRODUCTION

## **1. Introduction**

### **1.1. Android (operating system)**

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics. Android has the largest installed base of all operating systems of any kind. Android has been the bestselling OS on tablets since 2013, and on smartphones it is dominant by any metric.

Initially developed by Android, Inc., which Google bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. As of July 2013, the Google Play store has had over one million Android applications ("apps") published, and over 50 billion applications downloaded. An April–May 2013 survey of mobile application developers found that 71% of developers create applications for Android, and a 2015 survey found that 40% of full-time professional developers see Android as their priority target platform, which is comparable to Apple's iOS on 37% with both platforms far above others. At Google I/O 2014, the company revealed that there were over one billion active monthly Android users, up from 538 million in June 2013.

Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software, including proprietary software required for accessing Google services. Android is popular with technology companies that require a ready-made, low-cost and customizable operating system for high-tech devices. Its open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for

advanced users or bring Android to devices originally shipped with other operating systems. At the same time, as Android has no centralised update system most Android devices fail to receive security updates: research in 2015 concluded that almost 90% of Android phones in use had known but unpatched security vulnerabilities due to lack of updates and support. The success of Android has made it a target for patent litigation as part of the so-called "smartphone wars" between technology companies.

## **1.2. Android software development**

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in Java programming language using the Android software development kit (SDK), but other development environments are also available.

## **1.3. Official development tools**

### **1.3.1. Android SDK**

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows XP or later. As of March 2015, the SDK is not available on Android itself, but the software development is possible by using specialized Android applications. Until around the end of 2014, the officially supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. As of 2015, Android Studio, made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, INSTALLING SOFTWARE package(s) remotely).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

### 1.3.2. Android Debug Bridge

The Android Debug Bridge (ADB) is a toolkit included in the Android SDK package. It consists of both client and server-side programs that communicate with one another. The ADB is typically accessed through the command-line interface, although numerous graphical user interfaces exist to control ADB.

The format for issuing commands through the ADB is typically:

`adb [-d|-e|-s <serialNumber>] <command>`

For example, Android applications can be saved by the command backup to a file, whose name is backup.ab by default.

### 1.3.3. Fastboot

Fastboot is a diagnostic protocol included with the SDK package used primarily to modify the flash filesystem via a USB connection from host computer. It requires that the device be started in a boot loader or Secondary Program Loader mode, in which only the most basic hardware initialization is performed. After enabling the protocol on the device itself, it will accept a specific set of commands sent to it via USB using a command line.

### 1.3.4. Android NDK

Libraries written in C, C++ and other languages can be compiled to ARM, MIPS or x86 native code and installed using the Android Native Development Kit (NDK).

Native classes can be called from Java code running under the Dalvik VM using the `System.loadLibrary` call, which is part of the standard Android Java classes.

Complete applications can be compiled and installed using traditional development tools. However, according to the Android documentation, NDK should not be used solely for developing applications only because the developer prefers to program in C/C++, as using NDK increases complexity while most applications would not benefit from using it. The ADB debugger gives a root shell under the Android Emulator which allows ARM, MIPS or x86 native code to be uploaded and executed. Native code can be compiled using GCC or the Intel C++ Compiler on a standard PC. Running native code is complicated by Android's use of a non-standard C library (libc, known as Bionic). The graphics library that Android uses to arbitrate and control access to this device is called the Skia Graphics Library (SGL), and it has been released under an open source licence. Skia has backends for both Win32 and Unix, allowing the development of cross-platform applications, and it is the graphics engine underlying the Google Chrome web browser.

Unlike Java application development based on an IDE such as Eclipse, the NDK is based on command-line tools and requires invoking them manually to build, deploy and debug the apps. Several third-party tools allow integrating the NDK into Eclipse and Visual Studio.

### **1.3.5. Android Open Accessory Development Kit**

The Android platform introduces Android Open Accessory support, which allows external USB hardware (an Android USB accessory) to interact an Android-powered device in a special "accessory" mode. When an Android-powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. Android USB accessories are specifically designed to attach to Android-powered devices and adhere to a simple protocol (Android accessory protocol) that allows them to detect Android-powered devices that support accessory mode.

### **1.3.6. Native Go support**

Since version 1.4 of the Go programming language, writing applications for Android is supported without requiring any Java code, although with a restricted set of Android APIs.

## **1.4. External hardware development**

Development tools intended to help an Android device interact with external electronics include IOIO, Android Open Accessory Development Kit, Microbridge, Triggertrap, etc.

## **1.5. Android Developer Challenge**

The Android Developer Challenge was a competition to find the most innovative application for Android. Google offered prizes totaling 10 million US dollars, distributed between ADC I and ADC II. ADC I accepted submissions from January 2 to April 14, 2008. The 50 most promising entries, announced on May 12, 2008, each received a \$25,000 award to further development. It ended in early September with the announcement of ten teams that received \$275,000 each, and ten teams that received \$100,000 each.



# CHAPTER TWO

## ANDROID PERSISTENCY: SQL DATABASES



## 2. Android Persistency: SQL Databases

### 2.1 Structure of project

Summed up the project e-bag set layout, each of which has special functions for users to begin with Login Screen from which to work your sign a group of students using the site Parse to deal with databases through the Internet connection then we go to the main menu, which contains several lists such as bulletin board that notices section intraday come out and a list of the teaching staff, which is to set images belong to a summary curriculum vitae of each professor of the department and lists specific to each stage containing books and lectures section saved online as PDF files opened by a function within the application Adobe reader program application and checks for quality in the system though not exist will be summoned Cockle Bally's official website in order to be dealt with files with ease in addition to the internal data SQLite each section contains lectures details and details of the notices in the bulletin board base and we will discuss it in detail with the process of downloading lectures to the device unobtrusively not play files only by application.

### 2.2 SQL Databases

#### 2.2.1 Using SQLite on Android

**SQLite** is internal storage using to save data from type text or image. With a little care, SQLite can be used as a data store or full database on Android devices.

As with most platforms, Android gives us a few options to store data so that it persists even after we've terminated the application. Of the various ways we can do this, text files — either stored in the application's own files directory or to the phone's SD card — represent one approach. Preferences are also frequently used to store data because they can be both hidden from the user and persist as long as the application is installed. And while not strictly speaking in the same category, Assets can be useful for storing read-only data, too. Assets are essentially files that you bundle into the application package prior to compilation in the project assets folder, which can be accessed at runtime. I will take a closer look at these later.

Sometimes, however, we need to be able to carry out complex operations on persistent data, or the volume of data requires a more efficient method of management than a flat text file or preference entry will allow. This is where a mobile database comes in.

Android comes with SQLite (version 3.5.9+), which has been available on the platform since release 1.5 of the OS (the "Cupcake" release). For readers unfamiliar with SQLite, it is a self-contained, server-less, transactional SQL database. While it has its limitations, SQLite serves as a powerful tool in the Android developer's arsenal.

What I'll principally cover here is one way to use a SQLite database in Android, concentrating on its management; specifically, creation and update, rather than all the runtime operations.

### **2.2.2 Managing SQLite**

To begin with, we can manage SQLite using a class that extends SQLiteOpenHelper, which comes with a constructor and two required methods; on Create and onUpgrade. Naturally, the first of these is executed when the constructor is instantiated; it is here that, via the superclass, we provide four important pieces of data:

1. Context. This is the context of the application. It can be useful to set this in the constructor and store it locally for later use in other methods.
2. Database name. This is the filename (as a String) of the physical database file being accessed.
3. Cursor factory. The cursor factory in use, if any.
4. Database Version. This is the version of your database (as an integer), which I'll discuss in more detail later. Your initial version should be 1.

**For our example, we put these four pieces together and get the following:**

```
class DB extends SQLiteOpenHelper {  
  
    final static int DB_VERSION = 1;  
    final static String DB_NAME = "mydb.s3db";  
    Context context;  
  
    public DB(Context context) {  
        super(context, DB_NAME, null, DB_VERSION);  
        // Store the context for later use  
        this.context = context;  
    }  
}
```

The constructor does two things. First, it checks whether the database exists and, if not, will call the on Create method. Second, if the database does exist, it will check whether the existing database version number differs from the one implemented in the constructor, so as to determine if the database has been updated. If it has, the on Upgrade method will be called.

Additionally, as we now know that the on Create method is called only when the database does not exist, it can be used as a handy way to determine if you're dealing with a first run of the application following installation. As such, you can use this method to call any other methods that you need executed only on the first run, such as EULA dialogs.

Let's look at the database itself. For the purposes of this article, I'm just going to use a very simple employee database with a SQL creation script as follows:

```
CREATE TABLE employees (  
    _id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name TEXT NOT NULL,  
    ext TEXT NOT NULL,  
    mob TEXT NOT NULL,  
    age INTEGER NOT NULL DEFAULT '0'  
);
```

We can easily construct this by hard-coding and executing the creation SQL, line by line, in our code as follows:

```
@Override
public void onCreate(SQLiteDatabase database) {
    database.execSQL( "CREATE TABLE employees ( _id INTEGER PRIMARY KEY
"
    + "AUTOINCREMENT, name TEXT NOT NULL, ext TEXT NOT NULL, "
    + "mob TEXT NOT NULL, age INTEGER NOT NULL DEFAULT '0')");
}
```

As you can see, this can get quite unwieldy once the database reaches a certain size and complexity, so the ideal solution would be to bundle a SQL creation script as an asset file. To use this approach, you need to write a method that takes in a SQL script from the assets directory and parses it, executing it line by line:

```
@Override
public void onCreate(SQLiteDatabase database) {
    executeSQLScript(database, "create.sql");
}

private void executeSQLScript(SQLiteDatabase database,
                             String dbname) {
    ByteArrayOutputStream outputStream =
    new ByteArrayOutputStream();

    byte buf[] = new byte[1024];
    int len;
    AssetManager assetManager = context.getAssets();
    InputStream inputStream = null;

    try{
        inputStream = assetManager.open(dbname);
        while ((len = inputStream.read(buf)) != -1) {
            outputStream.write(buf, 0, len);
        }
        outputStream.close();
        inputStream.close();

        String[] createScript = outputStream.toString().split(";");
        for (int i = 0; i < createScript.length; i++) {
            String sqlStatement = createScript[i].trim();
            // TODO You may want to parse out comments here
            if (sqlStatement.length() > 0) {
                database.execSQL(sqlStatement + ";");
            }
        }
    } catch (IOException e) {
        // TODO Handle Script Failed to Load
    } catch (SQLException e) {
        // TODO Handle Script Failed to Execute
    }
}
```

While this is a more complex approach than simply executing each SQL statement for very simple databases, it quickly pays dividends once the database becomes more complex or you need to pre-populate it. You'll also see that I abstracted the creation into a separate method called execute SQL Script so that it can be reused in other situations.

## 2.3 Code of DatabaseHelper.java of E-bag

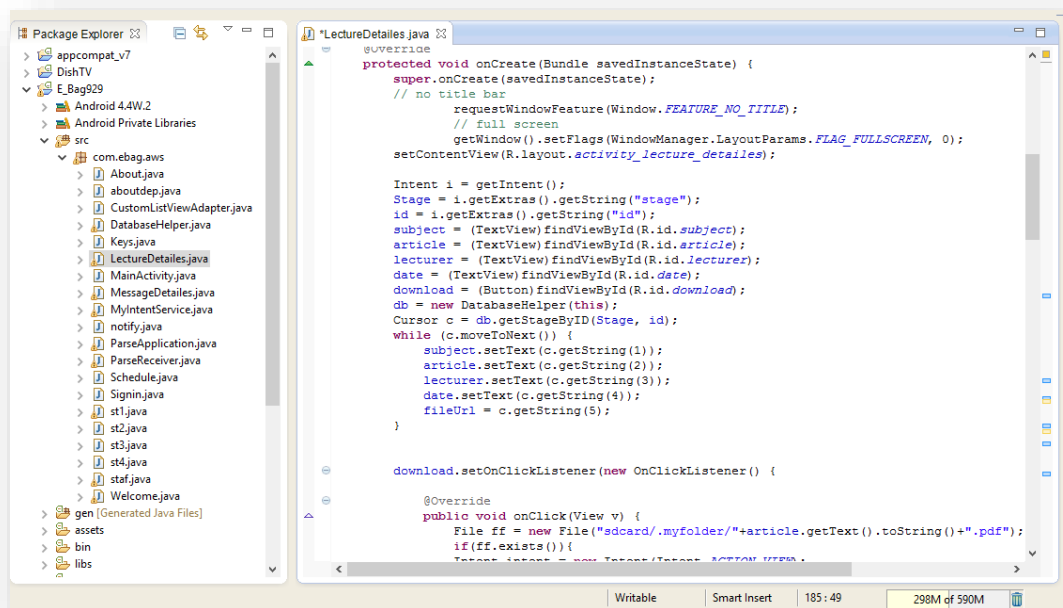


Fig 2.1 Code window of DatabaseHelper.java of E-bag

```
package com.ebag.aws;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import android.content.ContentValues;
import android.content.Context;
```

```
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "Ebag";
    private static Context context;
    // Table Names
    private static final String TABLE_NOTIFICATIONS = "Notification";
    private static String DB_PATH =
"/data/data/com.ebag.aws/databases/Ebag";
    private SQLiteDatabase db;

    public DatabaseHelper(Context paramContext) {
        super(paramContext, DATABASE_NAME, null, DATABASE_VERSION);
        context = paramContext;
    }

    long AddNotify(DatabaseHelper db,String title, String content,
String date)
    {
        SQLiteDatabase localSQLiteDatabase =
db.getWritableDatabase();
        ContentValues localContentValues = new ContentValues();
        localContentValues.put("title",title);
        localContentValues.put("content",content);
        localContentValues.put("date",date);
    }
}
```

```
        long id =localSQLiteDatabase.insert(TABLE_NOTIFICATIONS,
null, localContentValues);
        return id;
    }

    Cursor getAllNotify()
    {
        return getWritableDatabase().rawQuery("SELECT * FROM
"+TABLE_NOTIFICATIONS+"", null);
    }

    Cursor getStage(String stage)
    {
        return getWritableDatabase().rawQuery("SELECT * FROM
"+stage+"", null);
    }

    Cursor getStageByID(String stage,String id)
    {
        return getWritableDatabase().rawQuery("SELECT * FROM
"+stage+" where id="+id, null);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        try {
            //in this place you can creating required tables
        }

        catch (Exception e)
        {
            Log.i("Fail to Create Table in data base", e.toString());
        }

    }

    private boolean checkDataBase() {
        File checkDB = new File(DB_PATH);
        return checkDB.exists();
    }
}
```



```
public void createDatabase() throws IOException {
    boolean dbExist = checkDataBase();
    if (dbExist) {
        // do nothing
    } else {
        this.getReadableDatabase();
        try {
            copyDataBase();
        } catch (IOException e) {
            throw new Error("Error copying database");
        }
    }
}

public void copyDataBase() throws IOException
{
    InputStream myInput
    this.context.getAssets().open("Ebag");
    String outFileName = DB_PATH;
    OutputStream myOutput = new FileOutputStream(outFileName);
    byte[] buffer = new byte[1024];
    int length;
    while ((length = myInput.read(buffer)) > 0) {
        myOutput.write(buffer, 0, length);
    }

    myOutput.flush();
    myOutput.close();
    myInput.close();
}
```

```

    }

    @Override
    public synchronized void close() {
        if (db != null)
            db.close();
        super.close();
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {

    }}

```

## 2.4 Code of LectureDetails.java of E-bag

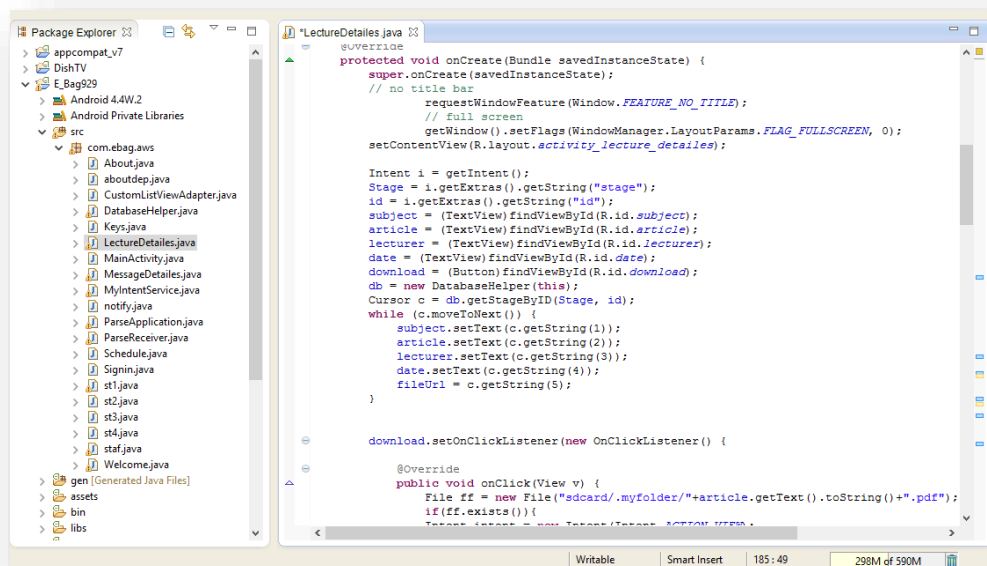


Fig 2.2 Code window of LectureDetails.java of E-bag

```
package com.ebag.aws;

import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.URL;
import java.net.URLConnection;
import android.app.Activity;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class LectureDetails extends Activity {
    Button download;
    TextView subject, article, lecturer, date;
    String Stage, id, fileUrl;
    DatabaseHelper db;
    public static final int DIALOG_DOWNLOAD_PROGRESS = 1;
    private ProgressDialog progressDialog;
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // no title bar
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    // full screen

    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
0);

    setContentView(R.layout.activity_lecture_details);

    Intent i = getIntent();
    Stage = i.getExtras().getString("stage");
    id = i.getExtras().getString("id");
    subject = (TextView) findViewById(R.id.subject);
    article = (TextView) findViewById(R.id.article);
    lecturer = (TextView) findViewById(R.id.lecturer);
    date = (TextView) findViewById(R.id.date);
    download = (Button) findViewById(R.id.download);
    db = new DatabaseHelper(this);
    Cursor c = db.getStageByID(Stage, id);
    while (c.moveToNext()) {
        subject.setText(c.getString(1));
        article.setText(c.getString(2));
        lecturer.setText(c.getString(3));
        date.setText(c.getString(4));
        fileUrl = c.getString(5);
    }

    download.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            File ff = new
File("sdcard/.myfolder/"+article.getText().toString()+".pdf");
            if(ff.exists()){
                Intent intent = new Intent(Intent.ACTION_VIEW);
                intent.setPackage("com.xodo.pdf.reader");
                intent.setDataAndType(Uri.fromFile(ff),
"application/pdf");

```

```

        intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
        startActivity(intent);
    }else{
        if (checkNetWork()) {
            new
DownloadFileAsync().execute(fileUrl);
        }
        //Toast.makeText(LectureDetailes.this, fileUrl,
1).show();
    }
});

}

class DownloadFileAsync extends AsyncTask<String, String, String>
{
    @Override
    protected String doInBackground(String... params) {
        // TODO Auto-generated method stub
        int count;

        try {

            URL url = new URL(params[0]);
            URLConnection connexion =
url.openConnection();

            connexion.connect();

            int lengthofFile =
connexion.getContentLength();
            Log.d("ANDRO_ASYNC", "Length of file: " +
lengthofFile);

            InputStream input = new
BufferedInputStream(url.openStream(),8192);
            File file = new File("sdcard/.myfolder");

```

```

        if (!file.exists()) {
            file.mkdir();
        }

        OutputStream output = new
FileOutputStream("sdcard/.myfolder/"+article.getText().toString()+".pdf");

        byte data[] = new byte[1024];

        long total = 0;

        while ((count = input.read(data)) != -1)
        {
            total += count;
            publishProgress("" + (int) ((total
* 100) / lengthOfFile));

            output.write(data, 0, count);
        }

        output.flush();
        output.close();
        input.close();
    } catch (Exception e) {
        // TODO: handle exception
    }

    return null;
}

@Override
protected void onPostExecute(String result) {
    // TODO Auto-generated method stub
    dismissDialog(DIALOG_DOWNLOAD_PROGRESS);
    // File ff = new
File("sdcard/.myfolder/"+article.getText().toString()+".pdf");
    // Intent i = new Intent(Intent.ACTION_VIEW);
    // i.setPackage("com.foobnix.pdf.reader");
    // i.setData(Uri.fromFile(ff));
    // startActivity(i);
}

```

```

@Override
protected void onPreExecute() {
    // TODO Auto-generated method stub
    super.onPreExecute();
    showDialog(DIALOG_DOWNLOAD_PROGRESS);
}

@Override
protected void onProgressUpdate(String... values) {
    // TODO Auto-generated method stub
    Log.d("ANDRO_ASYNC", values[0]);

    progressDialog.setProgress(Integer.parseInt(values[0]));
}

}

@Override
@Deprecated
protected Dialog onCreateDialog(int id) {
    // TODO Auto-generated method stub
    switch (id) {
        case DIALOG_DOWNLOAD_PROGRESS:
            progressDialog = new ProgressDialog(this);
            progressDialog.setMessage("Downloading file...");

            progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
;

            progressDialog.setCancelable(false);
            progressDialog.show();
            return progressDialog;
        default:
            return null;
    }
}

public boolean checkNetwork() {

```

```

        ConnectivityManager cm =
        (ConnectivityManager) getSystemService (Context.CONNECTIVITY_SERVICE);

        NetworkInfo ninfo = cm.getActiveNetworkInfo();
        if (ninfo != null && ninfo.isConnected()) {
            Log.d("network", "is connect");
            return true;
        } else {
            Log.d("network", "not connect");
            Toast.makeText(LectureDetails.this, "اتصال يوجد لا",
        بالنت", 1).show();
            return false;
        }
    }
}

```

## 2.5 Eclipse (software)

**Eclipse** is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages through the use of plugins , including: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Julia Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala , Clojure, Groovy, Scheme, and Erlang . It can also be used to develop packages for the software Mathematical. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.



Released under the terms of the Eclipse Public License, Eclipse SDK is free and open-source software (although it is incompatible with the GNU General Public License). It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

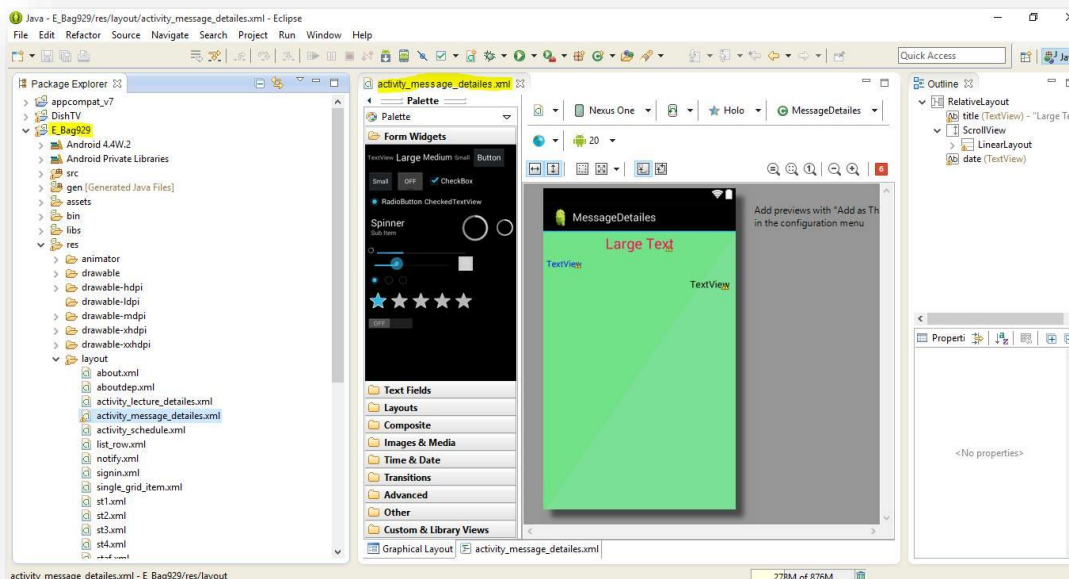


Fig 2.3 Eclipse project working

### 2.6 Database Tables

Stage table contain an important sequence in database, subject of lecture, name of lecture, lecture professor name ,date when lecture uploaded and link who uploaded lecture in website or any electronic cloud.

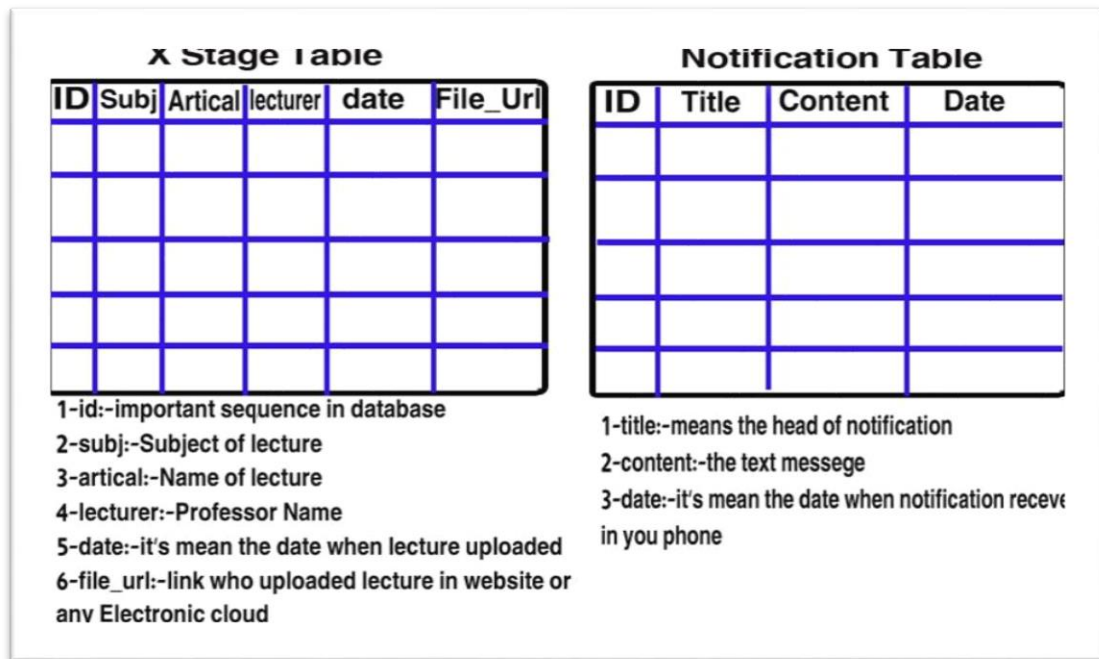


Fig 2.4 Shows SQLite Table structure

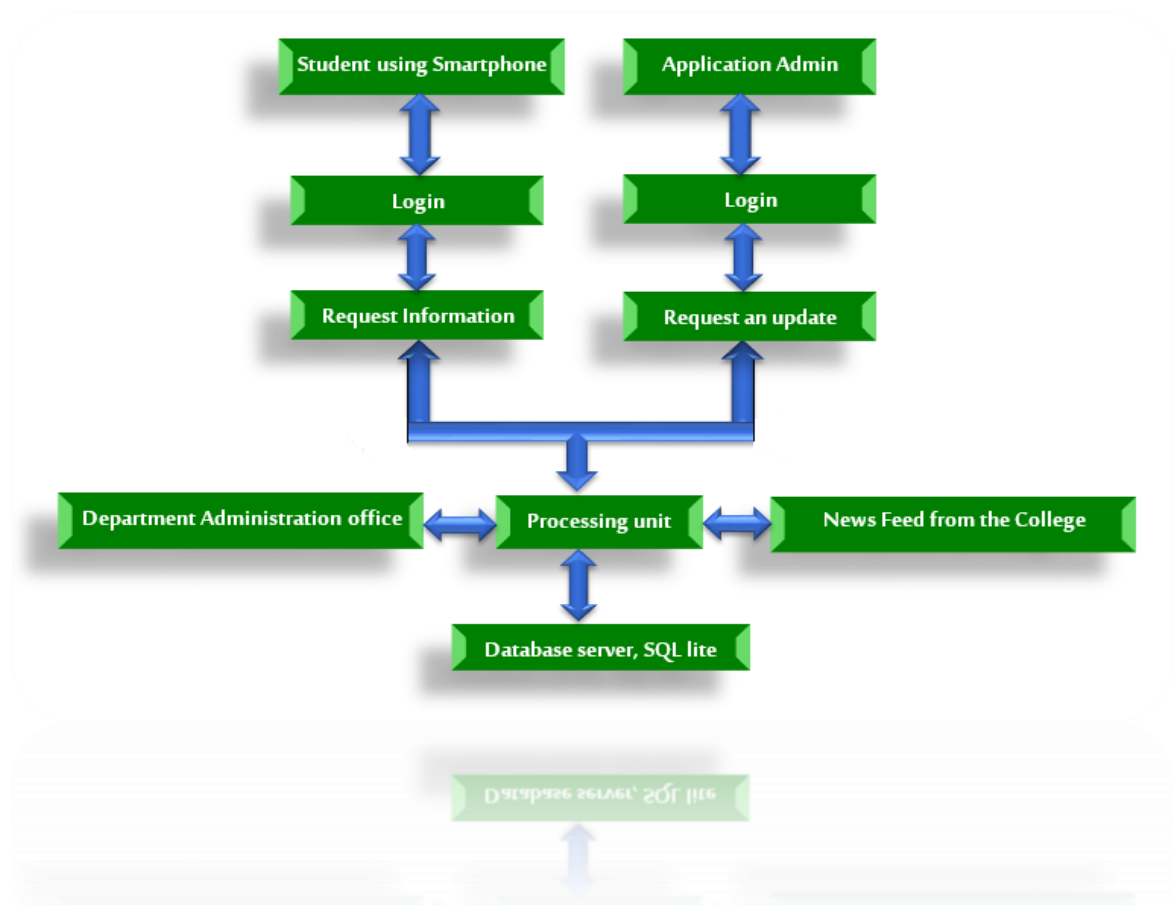


Fig. 2.5 Block Diagram for application

# **CHAPTER THREE**

**SHOWING THE**

**APPLICATION**

**AND**

**THE RESULTS**

### 3. Showing the application and the results

#### 3.1 Application's Windows

##### Execution of project when app is running

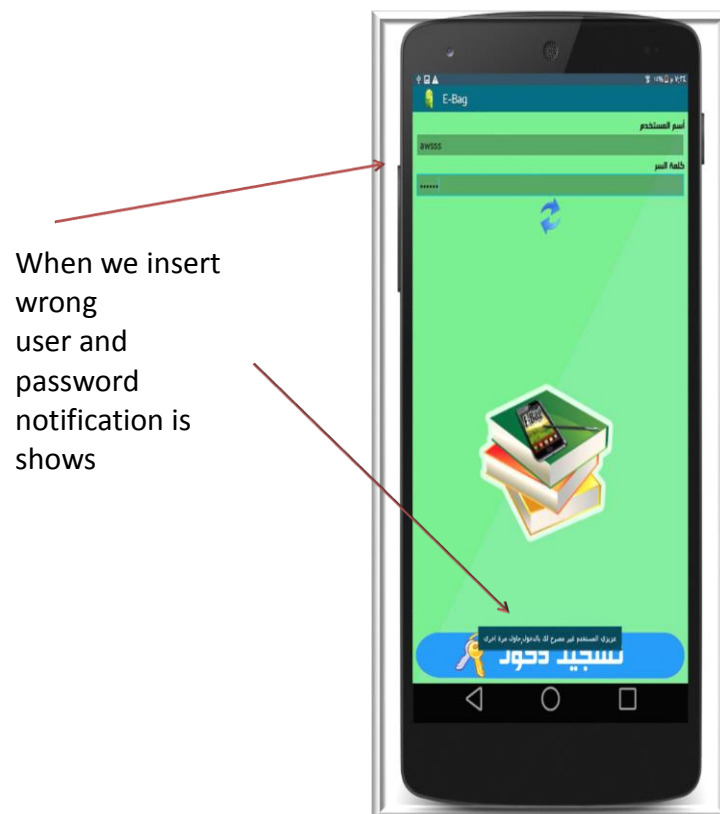


Fig. 3.1 login interface



Fig.3.2 Enter right user &amp; password



Fig.3.3 Screenshot for Main Menu

Notification  
Table



Fig. 3.4 Notification table layout



Fig. 3.5 Staff information



Fig. 3.6 Lectures table layout



Fig. 3.7 Schedule item

### 3.2 Checking the operation of the Application

We create a php code to upload photo of schedule and upload in a free Hosting

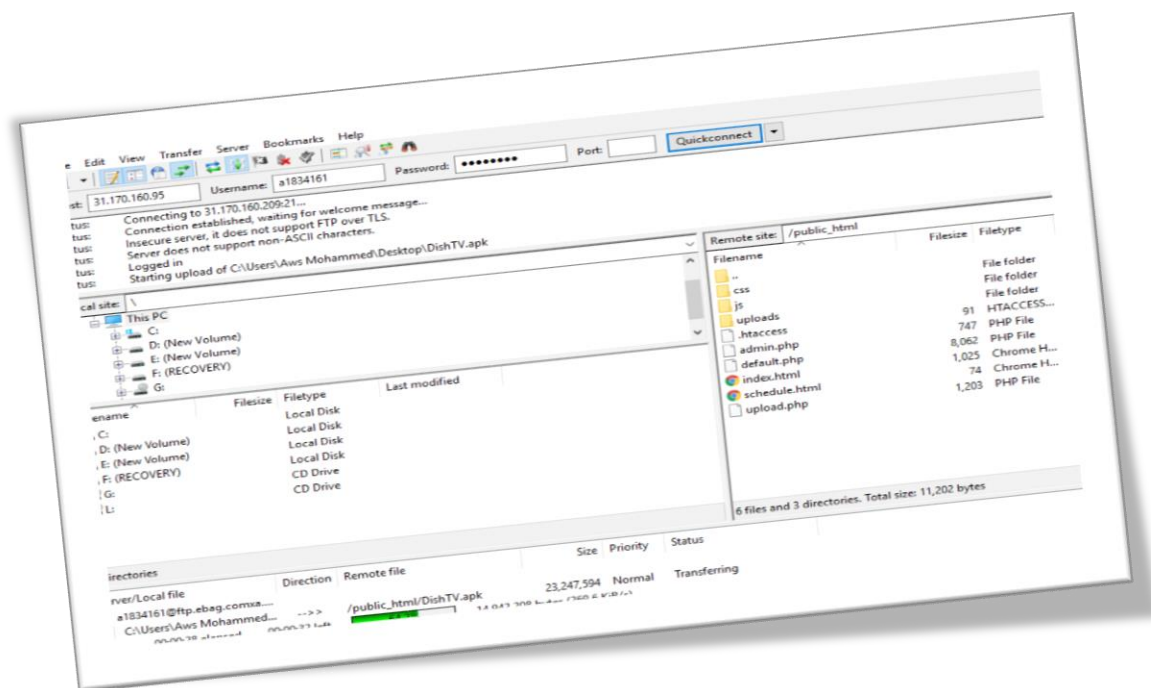


Fig.3.8 File zella upload website

**user:admin**

**pass:1992**



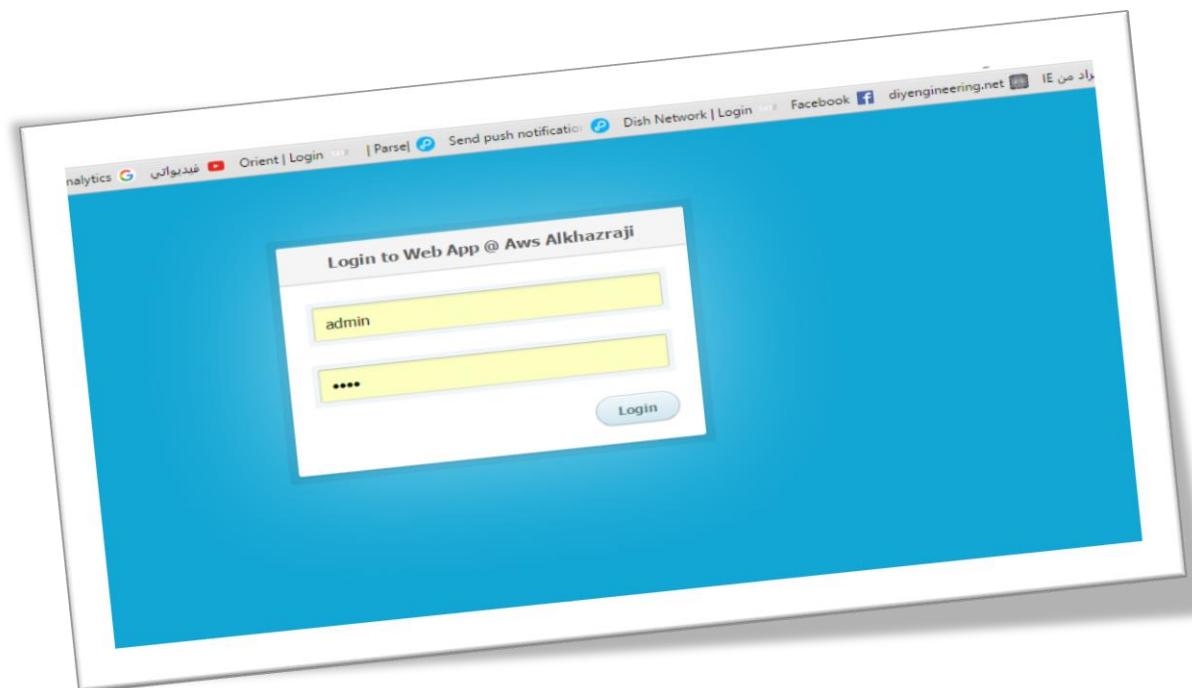


Fig.3.9 Login website



Fig.3.10 Select a photo to uploaded

The image shows a screenshot of a software application's schedule layout. The interface is titled "Schedule" and contains a large table with multiple columns and rows. The table is organized into several sections, with headers in Arabic and English. The data includes dates, times, and various categories of events or tasks. The layout is color-coded with yellow and blue headers, and the table itself has a grid structure with alternating colors for readability. The overall design is professional and detailed, typical of a project management or scheduling tool.

Fig.3.11 Schedule layout

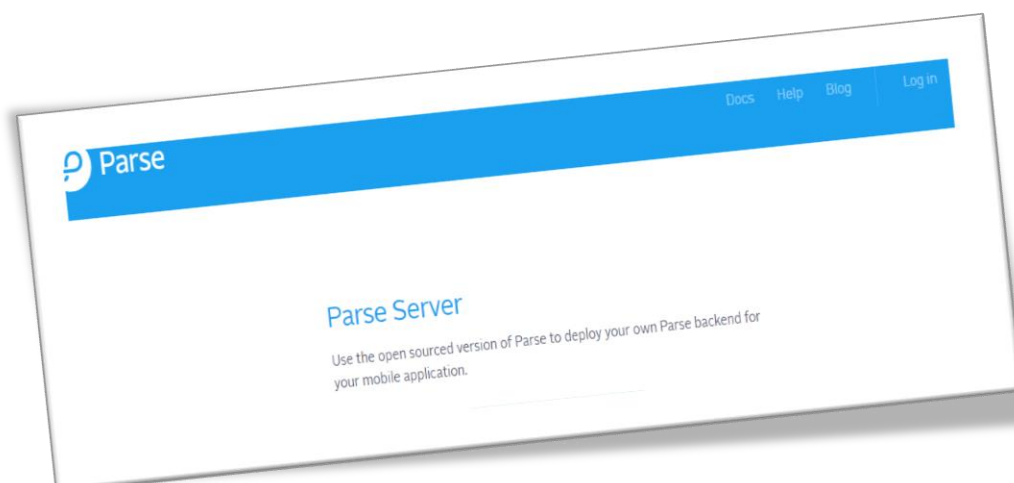
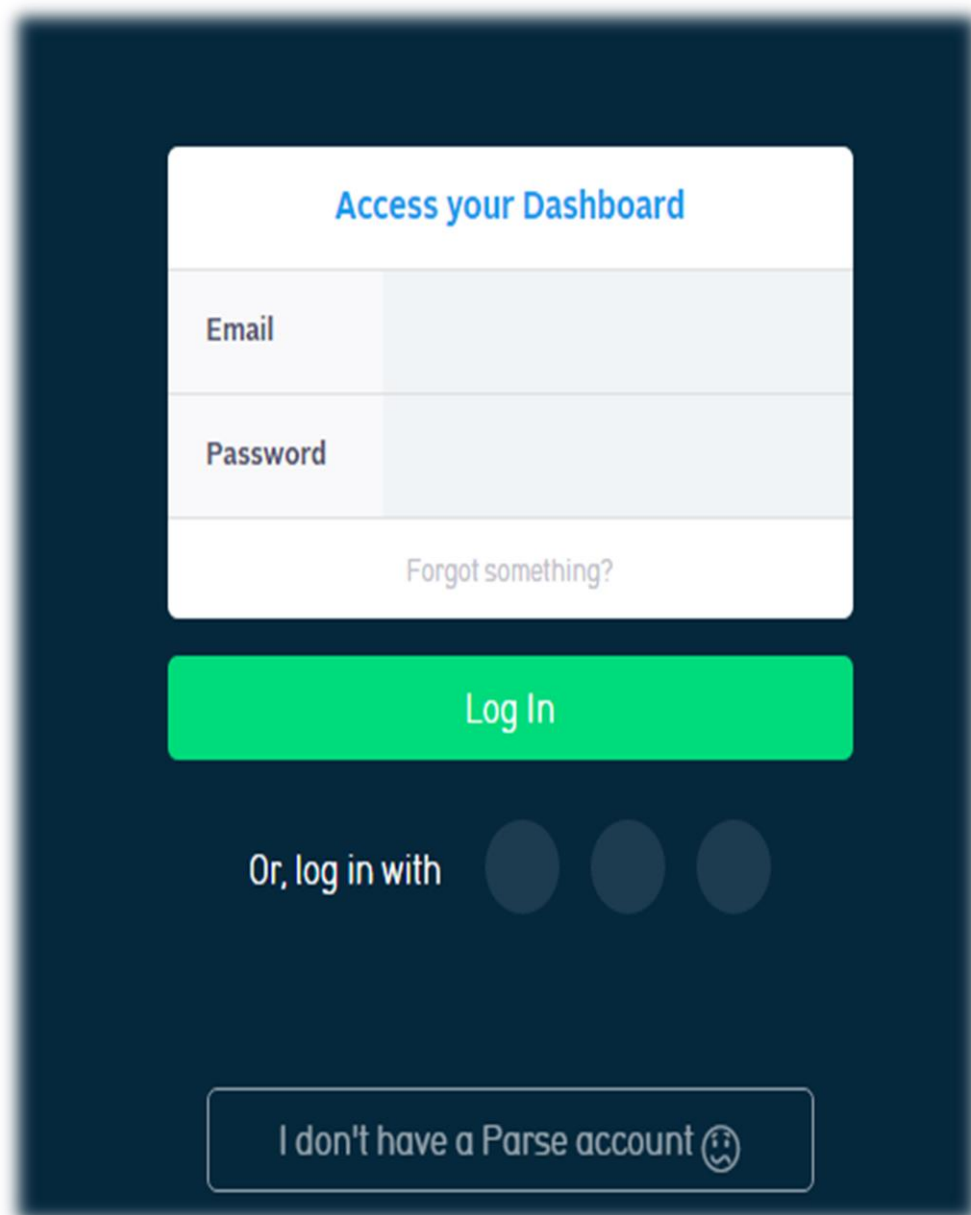





Fig.3.12 Add user and send notification join to parse.com

**Sign up in parse.com**

The image shows a sign-up menu for Parse.com. It features a dark blue background with a white login form. The form has a title 'Access your Dashboard' in blue. Below the title are two input fields: 'Email' and 'Password'. A link 'Forgot something?' is located below the password field. A large green 'Log In' button is positioned below the form. Below the button, the text 'Or, log in with' is followed by three dark blue circular icons. At the bottom, a button with the text 'I don't have a Parse account' and a sad face icon is displayed.

Access your Dashboard	
Email	<input type="text"/>
Password	<input type="password"/>
<a href="#">Forgot something?</a>	

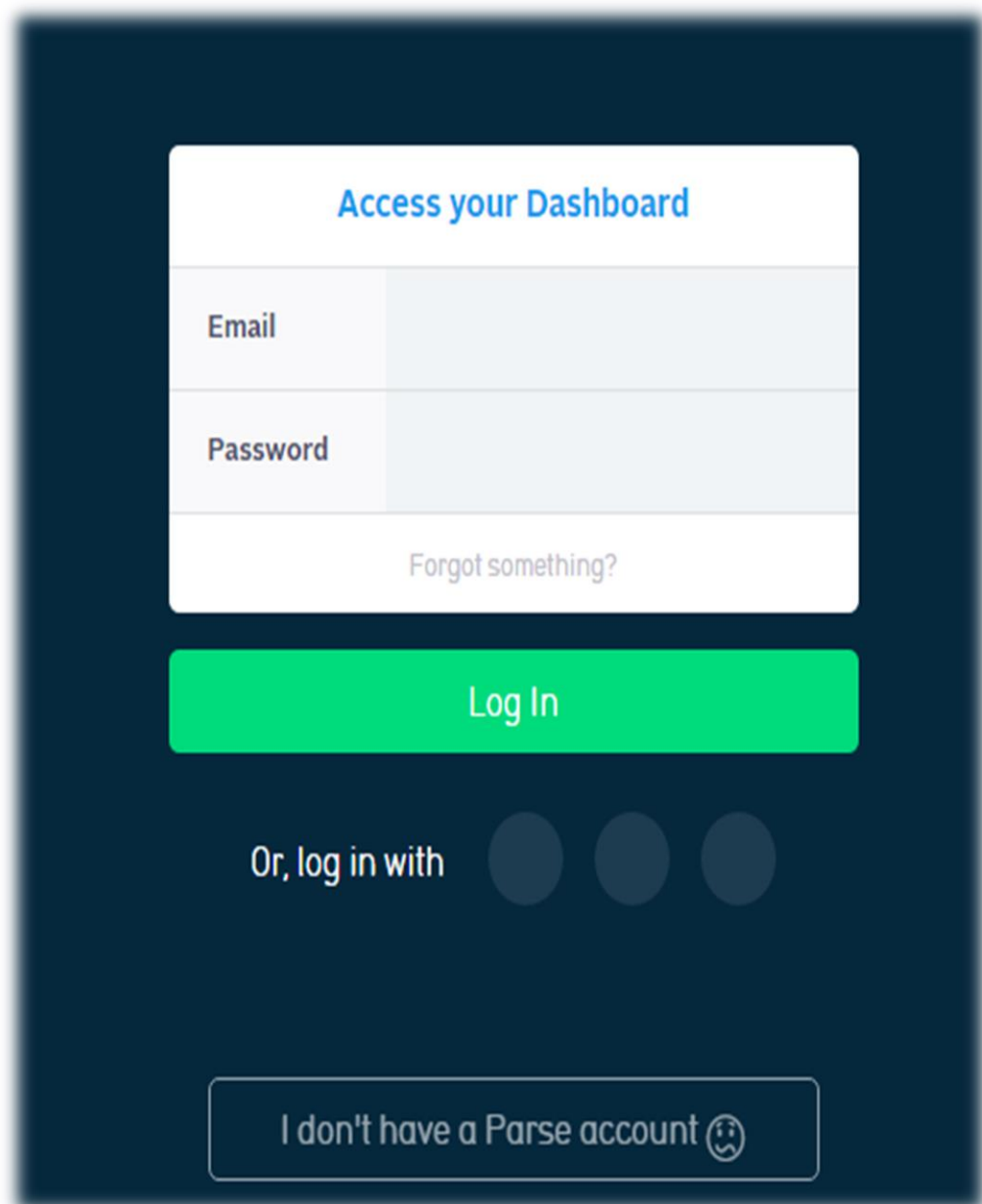
[Log In](#)

Or, log in with   

[I don't have a Parse account 😞](#)

Fig.3.13 Sign up menu

### Sign in User and Password



The image shows a login interface for Parse.com. It features a dark blue background with a white login form in the center. The form has a title 'Access your Dashboard' in blue. Below the title are two input fields: 'Email' and 'Password', both with light blue borders. A link 'Forgot something?' is located below the password field. A large green 'Log In' button is positioned below the form. Below the button, the text 'Or, log in with' is followed by three dark blue circular icons representing different login methods. At the bottom, a button with the text 'I don't have a Parse account' and a sad face icon is displayed.

Fig.3.14 Registration site parse.com

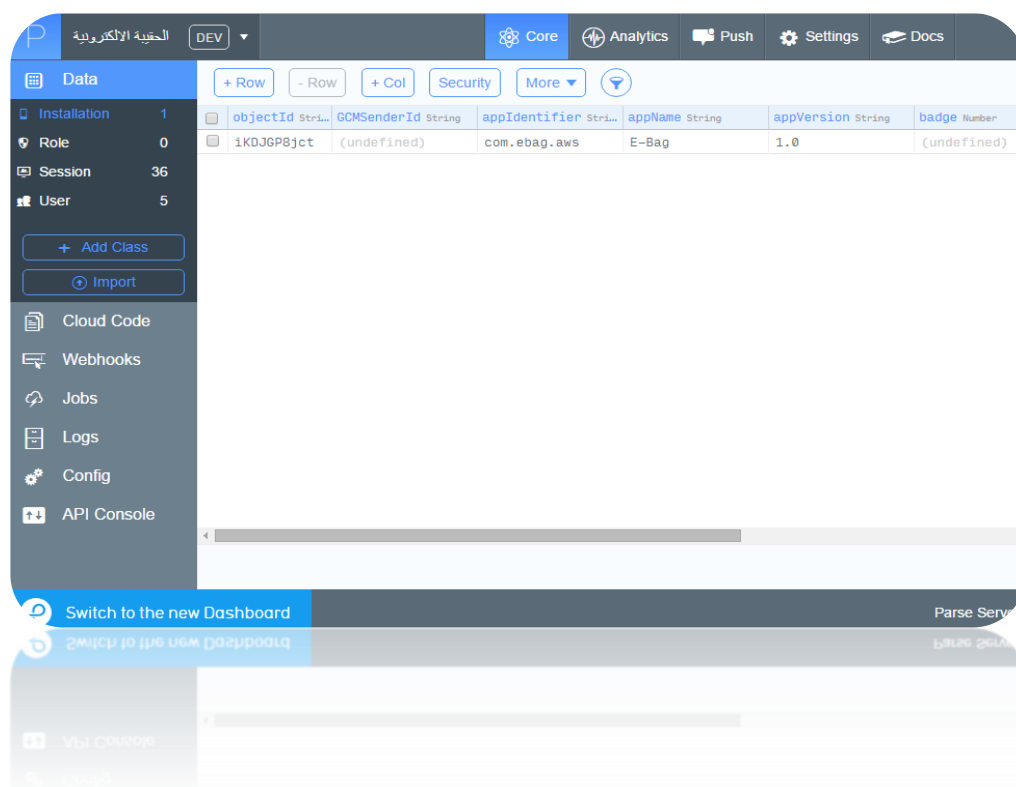


Fig.3.15 Main Dashboard: Basic control panel for the user(main-cove-user)

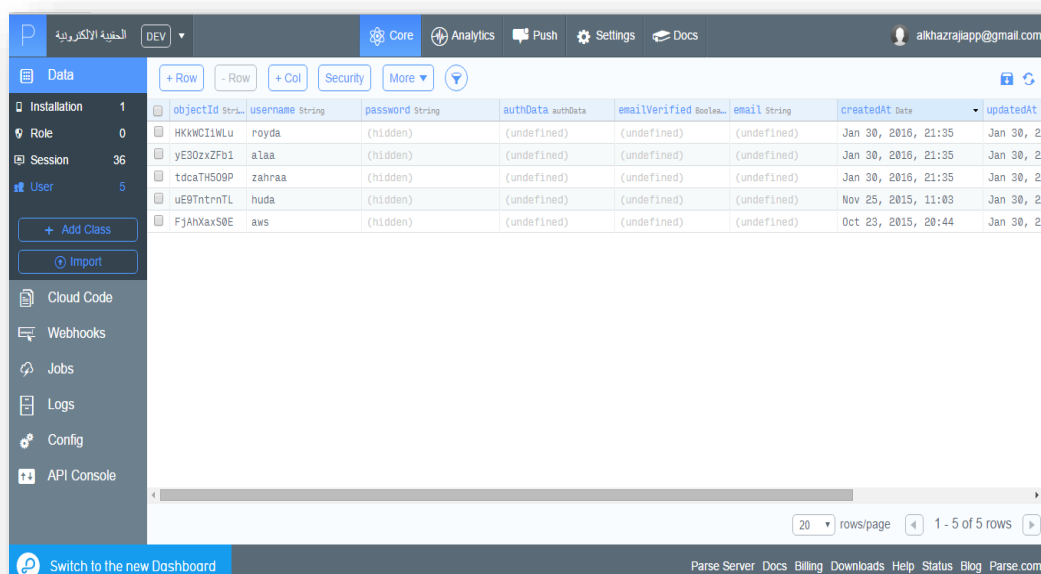


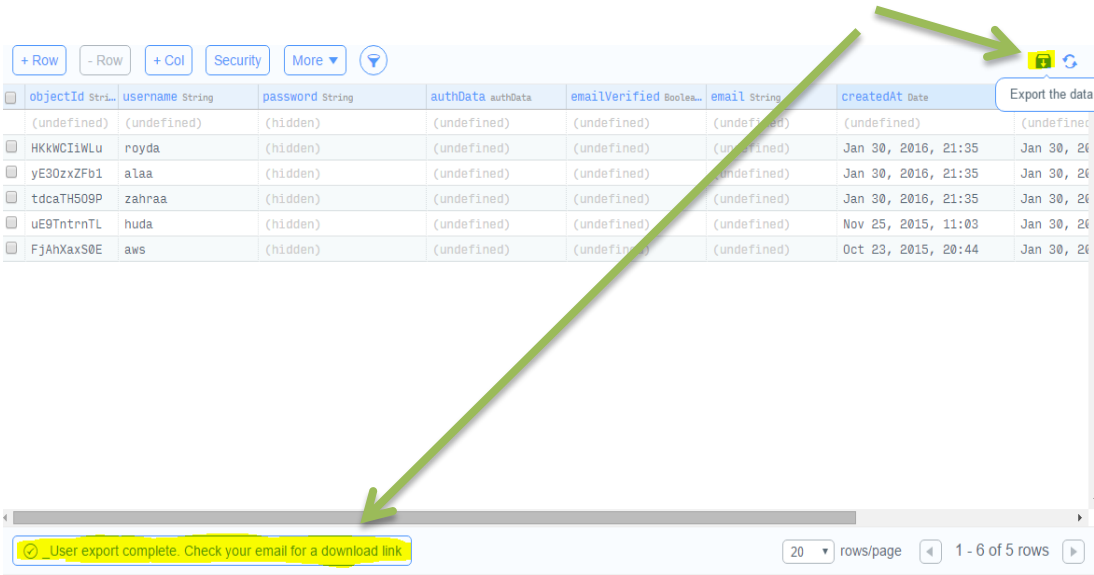
Fig.3.16 Edit user menu

Add New User



+ Row	- Row	+ Col	Security	More			
objectId	username	password	authData	emailVerified	email	createdAt	updatedAt
(undefined)	(undefined)	(hidden)	(undefined)	(undefined)	(undefined)	(undefined)	(undefined)
HKkWCiIwLu	royda	(hidden)	(undefined)	(undefined)	(undefined)	Jan 30, 2016, 21:35	Jan 30, 2016, 21:35
yE30zxZFb1	alaa	(hidden)	(undefined)	(undefined)	(undefined)	Jan 30, 2016, 21:35	Jan 30, 2016, 21:35
tdcaTH509P	zahraa	(hidden)	(undefined)	(undefined)	(undefined)	Jan 30, 2016, 21:35	Jan 30, 2016, 21:35
uE9TntrnTL	huda	(hidden)	(undefined)	(undefined)	(undefined)	Nov 25, 2015, 11:03	Jan 30, 2016, 21:35
FjAhXaxS0E	aws	(hidden)	(undefined)	(undefined)	(undefined)	Oct 23, 2015, 20:44	Jan 30, 2016, 21:35

Fig.3.17 New user through Add row=+row



+ Row	- Row	+ Col	Security	More				Export the data
objectId	username	password	authData	emailVerified	email	createdAt	updatedAt	
(undefined)	(undefined)	(hidden)	(undefined)	(undefined)	(undefined)	(undefined)	(undefined)	(undefined)
HKkWCiIwLu	royda	(hidden)	(undefined)	(undefined)	(undefined)	Jan 30, 2016, 21:35	Jan 30, 2016, 21:35	
yE30zxZFb1	alaa	(hidden)	(undefined)	(undefined)	(undefined)	Jan 30, 2016, 21:35	Jan 30, 2016, 21:35	
tdcaTH509P	zahraa	(hidden)	(undefined)	(undefined)	(undefined)	Jan 30, 2016, 21:35	Jan 30, 2016, 21:35	
uE9TntrnTL	huda	(hidden)	(undefined)	(undefined)	(undefined)	Nov 25, 2015, 11:03	Jan 30, 2016, 21:35	
FjAhXaxS0E	aws	(hidden)	(undefined)	(undefined)	(undefined)	Oct 23, 2015, 20:44	Jan 30, 2016, 21:35	

✔ User export complete. Check your email for a download link

20 rows/page 1 - 6 of 5 rows

Fig.3.18 Download user Documents

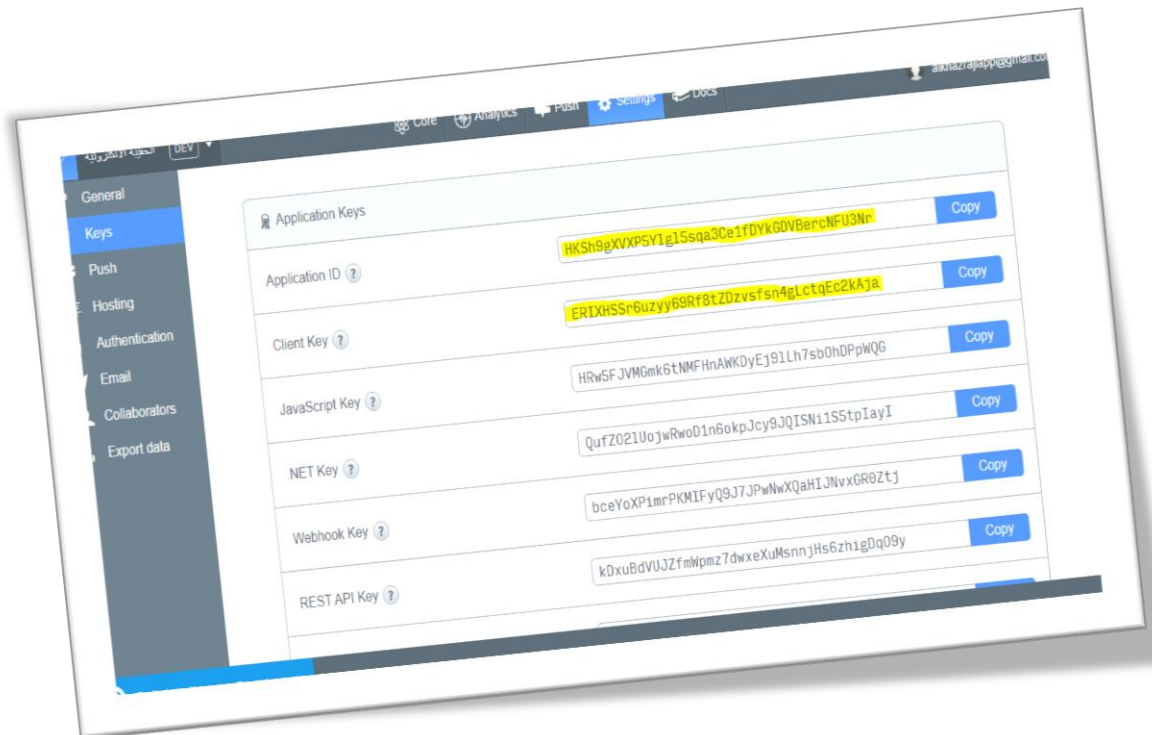


Fig.3.19 Application ID and Key in parse

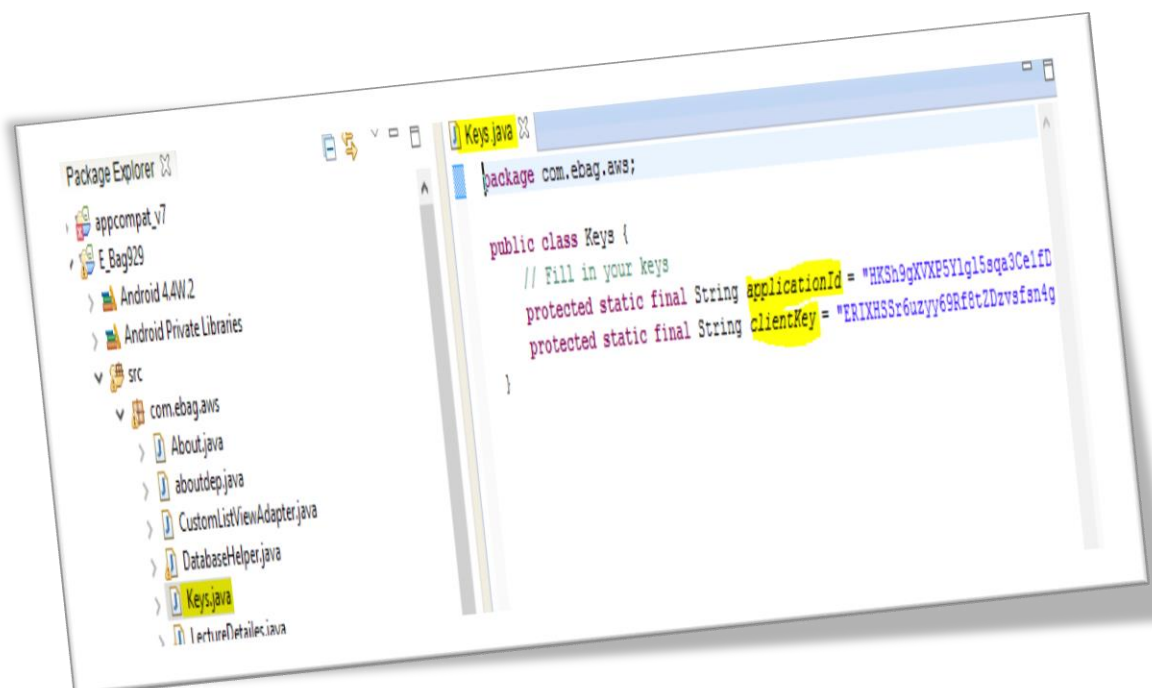


Fig.3.20 Application ID and Key in eclipse

# **CHAPTER FOUR**

## **CONCLUSION**

**&**

## **FUTURE WORK**



## **4. CONCLUSION & Future work**

### **4.1. Conclusions**

It is better to buy space on the Internet and linked to an integrated site contains several features special account to raise teaching lectures And calculate the last decision as part of our work to the project earlier to lift the weekly schedule and notifications

The special account under the chairmanship of the section to be added students and change the pin numbers to enter through them or the student can change his password as needed and not added this feature. So that the project will be integrated on the Service and Environment Web Android mobile phone and tablet

We can develop the project in order to work on devices iPhone and iPhone devices windows if we use C language environment and C#. As it is to add a record absences updated electronically and the addition of the college news service **Rss** previously raised on the Cockle shop college application

As are additions special privileges estimates student over four years and scored an integrated student if the application has been developed to an advanced level as we build a chat between teaching and students are controlled by the head of the department as well as deletions and urban areas to communicate more between the administration and students

### **4.2. Future work**

As a suggestion for future work, the following point can be considered:

Doing a web serves to hole the project (for all Lecture and head of department) every one of them his owner work.

## REFERENCES

- 1) "Android Code Analysis". Retrieved June 6, 2012.
- 2) Jack Wallen (2015-03-06). "How to create a full backup of your Android device without root". Retrieved 2016-01-29.
- 3) "SDK Tools | Android Developers". Developer.android.com. Retrieved October 28, 2015.
- 4) "The Android Source Code: Governance Philosophy". *source.android.com*. December 17, 2014. Retrieved January 25, 2015.
- 5) "Google's iron grip on Android: Controlling open source by any means necessary". *Ars Technica*. Retrieved December 8, 2013.
- 6) "Announcing the Android 1.0 SDK, release 1". *September 9, 2008*. Retrieved September 21, 2012.
- 7) *Shah, Agam (December 1, 2011)*. "Google's Android 4.0 ported to x86 processors". *Computerworld. International Data Group*. Retrieved February 20, 2012.
- 8) "android/platform/system/core/toolbox/". *Archived from the original on February 9, 2014*.
- 9) "dd command from NetBSD as an example". *Archived from the original on March 19, 2014*.
- 10) *Wilson Rothman (October 24, 2012)*. "Why iPad is stomping Android tabs 24 to 1 – Technology on". *Nbcnews.com*. Retrieved November 9, 2012.
- 11) *Software Development Discussion Paper: An overview of mobile development in the context of current technology*.
- 12) VisionMobile, Plum Consulting, "European App Economy", September 2013
- 13) Nowotarski et al., "Increasing Allowance Rates by Selectively Targeting Patent Class" IPwatchdog, April 6, 2011
- 14) Florian Mueller, "Lodsys sues 7 app developers in Eastern Texas, disagrees with Apple; Android also targeted", FOSS Patents blog, May 31, 2011
- 15) <http://www.androidbegin.com/tutorial/android-parse-com-simple-login-and-signup-tutorial/>  
<http://developer.android.com>
- 16) <http://www.eclipse.org>
- 17) <https://www.parse.com>
- 18) <http://www.iitk.ac.in/ccnew/index.php/8-servers/102-advantages-of-using-a-server>
- 19) [http://www.answers.com/Q/Advantages\\_and\\_disadvantages\\_of\\_server\\_based\\_networks](http://www.answers.com/Q/Advantages_and_disadvantages_of_server_based_networks).

# الخلاصة

العملاء في العالم في الوقت الحاضر قامو باستخدام تطبيقات الهواتف الذكية لسرعة الوصول إلى شركات الخدمات المختلفة وموارد البيانات الأخرى. وقد أثبتت الدراسات الحديثة على أهمية أي مؤسسة أو شركة خاصة أو وكالة حكومية لنشر ملف التعريف الخاص بهم وخدماتها من خلال تطبيقات الهواتف الذكية. أن التقدم في تكنولوجيا المعلومات الالكترونية والتعلم تمثل الدافع المهم للمطورين لإنتاج تطبيقات الهواتف الذكية لتسهيل الوصول إلى المواد الإلكترونية المطلوبة. وفقا لكثير البيانات المذكورة أعلاه قمنا بتطوير تطبيق الهواتف الذكية لقسم هندسة الحاسبات والبرامجيات في جامعة ديالى يوفر هذا التطبيق المعلومات التي يتكرر طرحها من قبل الطلاب مثل المعلومات عن المناهج الدراسية وتصفحها وتحميلها وطباعتها من قبل الطلاب وصول اشعارات بلوحة الاعلانات الخاصة بالقسم من قبل رئاسة القسم والوصول لمعلومات عن الكادر التدريسي للقسم حيث يتم الحصول على اي معلومات من اي مكان فقط بوجود الانترنت نحصل على المعلومات ، حيث يعمل المشروع كواجهة لمكتبة و اشعارات القسم وغيرها من المميزات التي يقدمها القسم للطلاب الجامعي. ويمكن أن تطبق أيضا هذا التطبيق على أي دائرة أخرى، او اي قسم في الكلية او اي كلية في الجامعة.



وزارة التعليم العالي والبحث العلمي

جامعة ديالى

كلية الهندسة

قسم الحاسوب والبرمجيات

## " تطوير تطبيق للماترنة المحمول لأنظمة الاندرويد (الحقيبة الالكترونية)"

مشروع مقدم إلى كلية الهندسة / جامعة ديالى - كجزء من متطلبات نيل  
درجة البكالوريوس في هندسة الحاسوب والبرمجيات

أعداد الطلاب:

اوس محمد حميد

زهراء عدنان شوكت

الاء ياس شوكت

بأشراف:

م.د. سعد محمد صالح

م.م. رويدة عبدالله إبراهيم