

QR-CODE BASED AUTOMATIC ATTENDANCE & ONLINE COLLEGE MANAGEMENT SYSTEM

A Project Report

Submitted in partial fulfillment of the
Requirement for the award of the Degree

of

Bachelor of Technology

in

COMPUTER SCIENCE & ENGINEERING

BY

**MONICA DHAWAN (09-CSE-1049)
ANURAG MALIK (09-CSE-1015)**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ECHELON INSTITUTE OF TECHNOLOGY, FARIDABAD
(Affiliated to Maharshi Dayanand University, Rohtak)**

JUNE 2013



DECLARATION

We hereby declare that the work which is being presented in the B.Tech. Project Report entitled "**QR Code based Automatic Attendance and Online College Management System**", in partial fulfilment of the requirements for the award of the **Bachelor of Technology in Computer Science & Engineering** and submitted to **Maharshi Dayanand University, Rohtak** is an authentic record of our own work carried out during a period from January 2013 to May 2013 under the supervision of **Mr. Mukesh Chawla, Associate Professor, CSE Department**.

The matter presented in this Project has not been submitted by us for the award of any other degree elsewhere.

ANURAG MALIK (09-CSE-1015)

MONICA DHAWAN (09-CSE-1049)

ACKNOWLEDGEMENTS

We take this opportunity to thank all those who have helped us in completing the project successfully.

We would like to express our gratitude to **Mr. Mukesh Chawla**, Associate Professor (CSE), who as our tutor provided us with every possible support and guidance throughout the development of project. This project would never have been completed without his encouragement and support.

We would also like to show our gratitude to **Dr. C.K. Nagpal**, Principal (Echelon Institute of Technology, Faridabad) & **Dr. Manoj Wadhwa** (Head of Department - Computer Science & Engineering) for providing us with well-trained faculty and giving us all the required resources and a healthy environment to carry out our project work.

We are thankful to other Faculty Members and Lab Staff of the department whosoever has contributed in this project directly or indirectly.

ANURAG MALIK (09-CSE-1015)

MONICA DHAWAN (09-CSE-1049)



TO WHOM IT MAY CONCERN

This is to certify that project titled "**QR-Code based automatic attendance & online college management system**".

Submitted by:

1. **Anurag Malik** (09-CSE-1015)
2. **Monica Dhawan** (09-CSE-1049)

of B.Tech VIII Semester, Computer Science & Engineering, in partial fulfilment of the requirement for the degree of Bachelor of Technology in Computer Science & Engineering, **Maharshi Dayanand University, Rohtak**, during the academic year 2012-13 is their original endeavour carried out under my supervision and guidance and has not been presented anywhere else.

Dated :

Dr. Manoj Wadhwa
Head of Department,
Computer Science & Engineering

Mukesh Chawla
Associate Professor,
CSE Department

ABSTRACT

The Project titled “QR Code based Automatic Attendance & Online College Management System” is based upon the idea of improvements that can be made in the existing management systems functional in various colleges, universities and other such educational institutions throughout our country. Various existing systems used for the attendance management and other administrative functions are either pen paper based or are implemented using earlier technologies. Through this project we are determined to provide view of a better solution to attendance management problem and to create an integrated college management system that works on centralized database, which is not only accessible through website but can also be accessed through android devices, containing dedicated applications.

We started this project with the process of learning and studying new technologies including J2EE, Android and JavaScript. HTML/ CSS were implemented to design consistent look and feel for the website. AJAX served the purpose of asynchronous communication with the servers. Further android applications were also designed to communicate with the same web services (Servlets) that served the website.

Various features and functionalities of the project are as follows:

1. Consistent Look and Feel for website through HTML/CSS.
2. JavaScript/JQuery used for providing interactive user interface.
3. Special features and functionalities in user profiles based upon the idea of integrating routine college management tasks of different departments.
4. Android Applications for users, aimed at providing fast and real time updates.
5. Aimed at automating the process of attendance management which otherwise, is based upon pen paper based management.
6. Database was designed and normalized to improve efficiency in data management and enforce integrity.
7. Uncovers the possibilities, and ideas of how new technologies and devices can be implemented for real time data access and updates.

CONTENTS

TITLE	Pg. No.
1. INTRODUCTION.....	1
1.1 Objective	
1.2 Project Scope	
1.3 Document Conventions	
2. REQUIREMENT ANALYSIS.....	2
3. IMPLEMENTATION ENVIRONMENT.....	3
3.1 Technologies.....	3
3.2 Tools.....	5
4. PROJECT DESCRIPTION.....	6
4.1 Automatic Attendance System.....	6
4.2 Online College Management.....	8
5. SOFTWARE DESIGN.....	10
5.1 ER Diagram.....	10
5.2 Data Flow Diagrams.....	11
5.3 Class Diagrams.....	16
5.4 Use Case Diagram.....	22
5.5 Database Design.....	23
6. CODING.....	24
7. OUTPUT SCREENSHOTS.....	44
8. CONCLUSION.....	61
9. FUTURE SCOPE.....	62
10. REFERENCES.....	63

1. INTRODUCTION

1.1 OBJECTIVE

Objective of the project is to improve the existing college management system and to implement certain advanced smart devices like android based handheld devices and wall mount devices for fast and easy information gathering.

The project also aims at improving the existing attendance management system in colleges, other educational institutions. Through the use of smart imaging devices in classrooms attendance can be automatically fed to the servers and thus reduce the use of pen and paper to the minimum possible.

1.2 PROJECT SCOPE

Almost every institute, college or school in India is using pen-paper based attendance system where in students present in the class are marked present by the teachers themselves. This not only wastes paper but also consumes an ample amount of precious time too. The attendance taken from all classes are then manually feed into database over college servers.

Since most of the work is done by teachers and concerned administrators. Human made errors may easily drop in and later it takes another deal of large paper work & time to find out and correct these errors. A better technique is thus imminent and much needed.

Another problem with today's college management system is that it lacks fast and efficient way of information gathering. A student who wants to get details about his/ her fee, attendance, marks, etc. either need to stand in long queues or wait for the concerned authority to take action on his query.

To remove this delay and inefficiency, our project aims at providing instant info to a user, right in his mobile device, which can be used at the comfort of his/her home or classroom.

1.3 DOCUMENT CONVENTIONS

QR Code – Quick Response Code

WASCE – Web Sphere Application Server Community Edition

IDM DB2 – IBM Database

UID Cards – Universal Identity Cards

RFID – Radio-Frequency Identification

Admin – Administrators

Users – Students, Teachers, Professors, Admins, Parents

OS – Operating System

2. REQUIREMENT ANALYSIS

2.1 SOFTWARE REQUIREMENTS

- Windows XP / Vista / Windows 7, Linux OS
- Any Web Browser like IE, Firefox, Opera, etc.

2.2 HARDWARE REQUIREMENTS

- Internet Connection
- Peripherals : Mouse / Keyboard for user input
- Any handheld device running on Android version 2.0 or above.

3. IMPLEMENTATION ENVIRONMENT

3.1 TECHNOLOGIES

3.1.1 Java Platform, Enterprise Edition

Java Platform, Enterprise Edition or Java EE is Oracle's enterprise Java computing platform. The platform provides an API and runtime environment for developing and running enterprise software, including network and web services, and other large-scale, multi-tiered, scalable, reliable, and secure network applications. Java EE extends the Java Platform, Standard Edition (Java SE), providing an API for object-relational mapping, distributed and multi-tier architectures, and web services. The platform incorporates a design based largely on modular components running on an application server. Software for Java EE is primarily developed in the Java programming language.

Java EE includes several API specifications, such as JDBC, RMI, e-mail, JMS, web services, XML, etc., and defines how to coordinate them. Java EE also features some specifications unique to Java EE for components. These include Enterprise JavaBeans, Connectors, servlets, Java Server Pages and several web service technologies. This allows developers to create enterprise applications that are portable and scalable, and that integrate with legacy technologies. A Java EE application server can handle transactions, security, scalability, concurrency and management of the components that are deployed to it, in order to enable developers to concentrate more on the business logic of the components rather than on infrastructure and integration tasks.

Java Servlet

A servlet is a Java programming language class used to extend the capabilities of a server. Although servlets can respond to any types of requests, they are commonly used to extend the applications hosted by web servers, so they can be thought of as Java Applets that run on servers instead of in web browsers.

Servlets are most often used to:

- Process or store data that was submitted from an HTML form
- Provide dynamic content such as the results of a database query
- Manage state information that does not exist in the stateless HTTP protocol, such as filling the articles into the shopping cart of the appropriate customer

Technically speaking, a "servlet" is a Java class in Java EE that conforms to the Java Servlet API, a protocol by which a Java class may respond to requests. Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP protocol. Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML / JSON Data. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or URL rewriting.

A Servlet is an object that receives a request and generates a response based on that request. The basic Servlet package defines Java objects to represent servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environment. The package `javax.servlet.http` defines HTTP-specific subclasses of the generic servlet elements, including session management objects that track multiple requests and responses between the web server and a client. Servlets may be packaged in a WAR file as a web application.

Servlets can be generated automatically from Java Server Pages (JSP) by the Java Server Pages compiler. The difference between servlets and JSP is that servlets typically embed HTML inside Java code, while JSPs embed Java code in HTML.

JSP – Java Server Pages

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. To deploy and run Java Server Pages, a compatible web server with a servlet container, such as Apache Tomcat (as used in our project through WASCE – Apache Geronimo v3).

JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime; each JSP's servlet is cached and re-used until the original JSP is modified.

JSP allows Java code and certain pre-defined actions to be interleaved with static web mark-up content, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, use Java byte code rather than a native software format. Like any other Java program, they must be executed within a Java virtual machine (JVM) that integrates with the server's host operating system to provide an abstract platform-neutral environment.

JSP pages use several delimiters for scripting functions. The most basic is `<% ... %>`, which encloses a JSP *scriptlet*. A scriptlet is a fragment of Java code that is run when the user requests the page. Other common delimiters include `<%= ... %>` for *expressions*, where the value of the expression is placed into the page delivered to the user, and *directives*, denoted with `<%@ ... %>`.

3.2 TOOLS

3.2.1 Eclipse IDE (J2EE)- Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java. It can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Haskell, Perl, PHP, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Android and Scheme. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse 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.

Version Used – **Eclipse Indigo**

Release Date - 22 June 2011

Platform Version – 3.7

3.2.2 IBM DB2 - IBM DB2 is a relational model database server developed by IBM. **IBM DB2 Express-C** is a free to download, use and redistribute edition of the IBM DB2 data server, which has both XML database and relational database management system features. IBM DB2 Express C supports 32bit and 64 bit x86 Windows XP, Vista and Windows 7 & various Linux OS.

DB2 Express-C may be run on any sized system with any amount of processors and memory, however DB2 Express-C will limit total resource utilization:

Processor: 2 cores

Memory: 4 GB

Operating systems: Linux (32bit, 64bit) or Windows (32bit, 64bit).

3.3.3 WASCE – Web Sphere Application Server Community Edition (from now on WASCE) is a free, certified Java EE 6 application server for building and managing Java applications. It is IBM's supported distribution of **Apache Geronimo** that uses Tomcat for servlet container and Axis 2 for web services. Other difference from Apache Geronimo is that WASCE comes with DB2 and Informix database drivers, better XML parser libraries. WASCE comes with a plugin-in that can be used in the Eclipse IDE. Web Sphere Application Server Community Edition provides:

- Support for open standards and open source on a small-scale, Java EE platform.
- Centralized management with simplified deployment and standards-based security.
- Customization and advanced capabilities allow you to modify the server according to the needs of your business.

4. PROJECT DESCRIPTION

The project carries the idea of improvement in administration and management of college and other such educational institutions. Main objective of the project is to automate the process of attendance management system in colleges / universities. Unlike current scenario of pen paper based attendance management, the project is based upon the use of special devices in classrooms which can automatically feed the attendance of a class to the college servers or database.

Further another objective of the project is to provide an online college management system. It is a simple yet powerful one joint integrated platform that connects various departments of an institute such as accounting dept., employee administration, library management, etc.

Major objective of the project is to automate the routine management tasks throughout the various departments of a college or other similar educational institutions. We aim at centralizing the overall college management system through the use of special QR Cards and dedicated android devices in the campus, which provide the one stop solution to all management tasks and queries.

QR Cards can be used to uniquely identify each user and thus can be read and understood by android devices to provide instant data such as personal information, fee details, attendance, etc. of all users.

4.1 AUTOMATIC ATTENDANCE SYSTEM

4.1.1 CURRENT SCENARIO

Almost every institute, college or school in India is using pen-paper based attendance system where in students present in the class are marked present by the teachers themselves. This not only wastes paper but also consumes an ample amount of precious time too. The attendance taken from all classes are then manually feed into database over college servers. Since most of the work is done by teachers and concerned administrators. Human made errors may easily drop in and later it takes another deal of large paper work & time to find out and correct these errors. A better technique is thus imminent and much needed.

4.1.2 SOLUTION

One of the present age solutions this problem is to provide UID (Electronic Tags) to each student and then use RFID readers to identify each student present in the class and record their attendance automatically. Students are required to mark in their entry in the class by scanning their ID Cards and then again marking out for the final attendance being fed to the database.

The problem with this system is the use of bulky and costly RFID machines or any other similar devices for the attendance system. Since they make use of electronic tags as UIDs for each student and employee, the overall cost for this system is quite large. Further these devices provide no flexibility and are capable of serving only one or maximum two functions defined for them.

4.1.3 IMPROVEMENT

Our project aims at improving this scenario of attendance management in educational institutes, through the use of smart devices in place of traditional RFID readers and other such costly devices.

We use Android devices to serve the purpose of tag readers. Today, there is a wide range of handheld and wall mounted devices running on Android OS. These devices include various features like camera, internet connectivity, storage facility, touch screens, etc. Thus not only these devices can be used to scan tags for attendance purpose but also provide flexibility and various other options. We discuss the attendance management system in this module.

Quick Response Code

We have used QR Code tags as UIDs for each student and employee. QR code (abbreviated from Quick Response Code) is the trademark for a type of matrix barcode (or two-dimensional bar code). Bar codes are optical machine-readable labels attached to items that record information related to the item. These are almost free to generate and can be used to store variety of data including simple text data, urls, contact info, etc. these provide a far better replacement to the electronic tags, which were quite costly to fabricate.

A QR code is read by an imaging device, such as a camera, and formatted algorithmically until the image can be appropriately interpreted. Data is then extracted from patterns present in both horizontal and vertical components of the image. As a variety of industries utilize the QR code today, the applications for use can vary from product tracking, item identification, time tracking, document management and general marketing purposes.

4.1.4 FUNCTIONING

Each student and employee (teachers/ Professors/ Admins) is registered on the college website and a unique code is provided to them. These UIDs are then encoded onto QR Cards (ID Cards) which can be read by any android device.

Let's now consider, how automatic attendance system works to take attendance of all students present in a class –

- i.) Teacher / Professor scan his QR Card to begin the lecture.
- ii.) The android device marks the presence of teacher in class and sets a timer for students to scan and mark-in their presence in the class.
- iii.) All the ID Cards scanned during this time are stored into a list and no student is allowed to mark-in after this time, they are marked absent for this lecture.
- iv.) At the end of lecture, teacher must again scan his card to ensure that no student left class in between and thus attendance to be stored on the college server / database.
- v.) In case a student left in between of class, teacher is allowed to remove his name from the list of present students. He can then finalize the attendance.

4.2 ONLINE COLLEGE MANAGEMENT SYSTEM

Online College Management System aims at centralizing the overall management process and integrating various departments of the institute to save information directly on the online servers. Various departments of an institute can access information about an employee or a student from the centralized database. And this process is further simplified through the use of QR Cards and Android Devices throughout the campus.

Since each user (Student / Employee) carry a unique QR Card, which can be read and understood by android devices, we can use them for real time updating of data on the college server or more precisely its centralized database.

Students and employees can access their information stored in the database through special applications running on their handheld devices such as mobile phones, tables, etc. or through dedicated wall mounted devices installed at various places in the campus.

4.2.1 COLLEGE MANAGEMENT SYSTEM (WEBSITE)

The project contains an online college management system which can be accessed by different administrators and other users for easy information gathering. Administrators of different departments are allowed to access information about employees and students and make changes to their details stored on the centralized database of college.

Various intended functionalities of the management system are –

- i.) To represent basic hierarchy of the institute.
- ii.) To add new users
- iii.) To remove existing users
- iv.) To store information about all users
- v.) Options to view and update existing information of any user
- vi.) Accessibility constraints based upon user types

4.2.1.1 PERSPECTIVE USERS- Various perspective users and their roles in the project are as follows –

i.) Administrators

Administrators are referred to as the decision makers, and senior management of the institute like Director, Head of Departments. The online college management system provides the following functionalities to the Admins –

- i.) To register new employee/student and providing them unique IDs (QR Cards).
- ii.) To access his online profile
- iii.) To generate request for update or modification in profile
- iv.) To serve update requests from students/employees including teachers and professors.
- v.) To delete old employees and students from database
- vi.) To keep track of various accounting information of students and employees.
- viii.) To update changes in payroll, holidays, timetables, etc.
- xi.) To generate general notifications/ updates for the college
- x.) To generate reports and emails for faculty, students or parents.

ii.) Professors / Teachers

Professors and Teachers have the duty to teach students of various classes of the institute. They are generally masters of one or two subjects and thus handle one particular subject for a class.

Apart from basic teaching and class management, Teachers / Professors are allowed –

- i.) To access his/ her online profile
- ii.) To generate queries and update requests.
- iii.) To keep record of various students of his / her classes.
- iv.) To keep record of attendance and marks of his/ her class.
- v.) To schedule periodic tests and update marks of all students on the college website.
- vi.) To have communication with parents in case of a student's less attendance or poor performance, etc.
- vii.) To generate important notifications for his/ her classes or a particular student.
- viii.) To view latest updates and notifications created by Admins.

iii.) Students

Students of various classes are allowed to –

- i.) Access their online profiles
- ii.) To view their fee details, attendance reports, marks, etc.
- iii.) To create request for update or change in their profile info
- iv.) To view latest notifications and updates from teachers / Admins, etc.
- v.) To create online queries and send them to intended teacher / professor / admin.
- vi.) To print their reports

iv.) Visitors

Visitors are allowed to access and read general information about the college. They can get detailed information about various courses available in the college along with their fee details, etc. For further information they can get contact info from the website.

4.2.2 ANDROID AT HELP (APPLICATION)

This module of the project is based upon improvising and simplifying the college management system through the use of android applications. These special dedicated applications can be used for routine information gathering and updating by any user. Since all users are provided by their Unique QR Cards, they can scan their cards and get instant access to their personal information, fee details, attendance reports, etc. as stored in the college database. Android Applications can connect to the web services running on the college servers.

4.2.2.1 FUNCTIONING

- 1.) Student scans his QR Cards on the android device
- 2.) Application decodes the QR Cards and asks for the password
- 3.) Login information entered is send to the web service for verification
- 4.) On successful verification student is allowed to use the application and retrieve information from the server.

5. SOFTWARE DESIGN

5.1 ER DIAGRAMS

ER DIAGRAM FOR COLLEGE MANAGEMENT SYSTEM

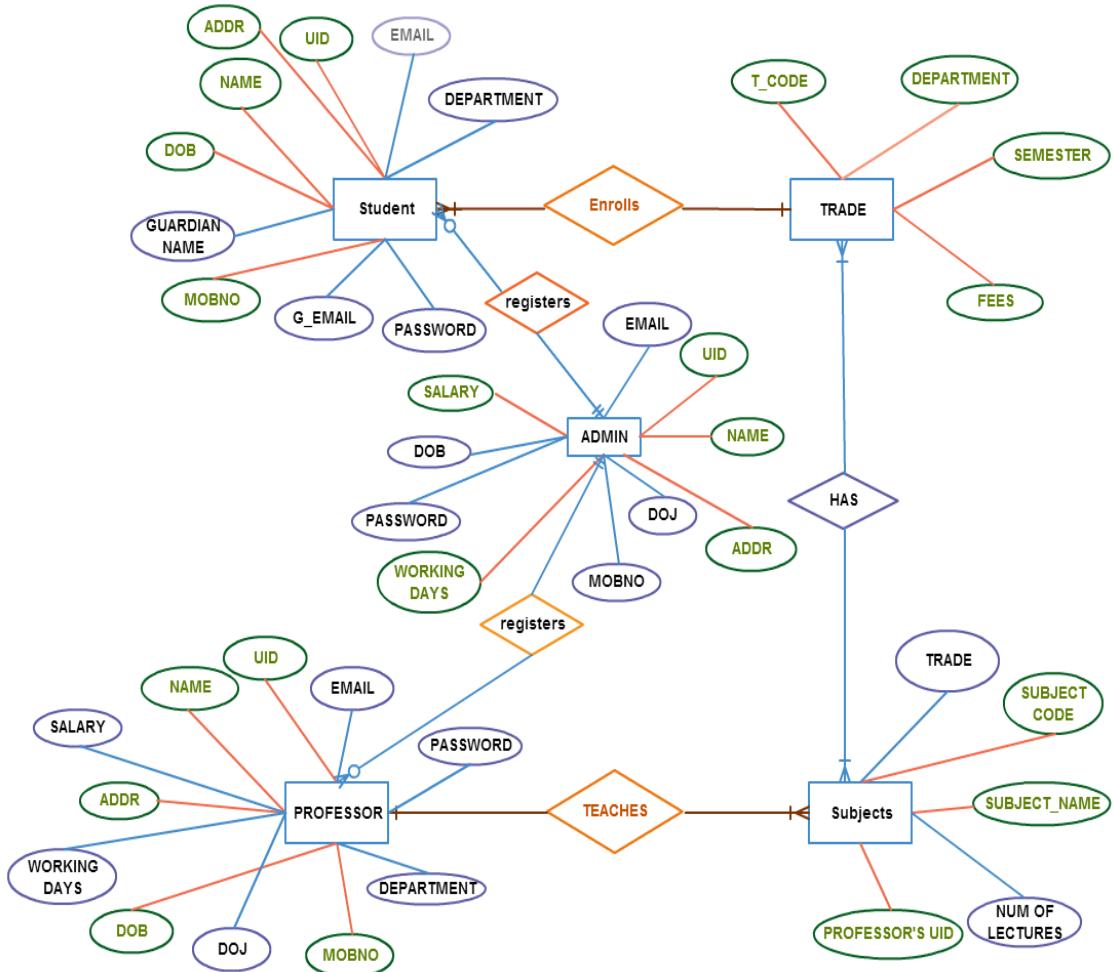


Fig. 5.1 ER Diagram

5.2 DATAFLOW DIAGRAMS

Attendance Management System

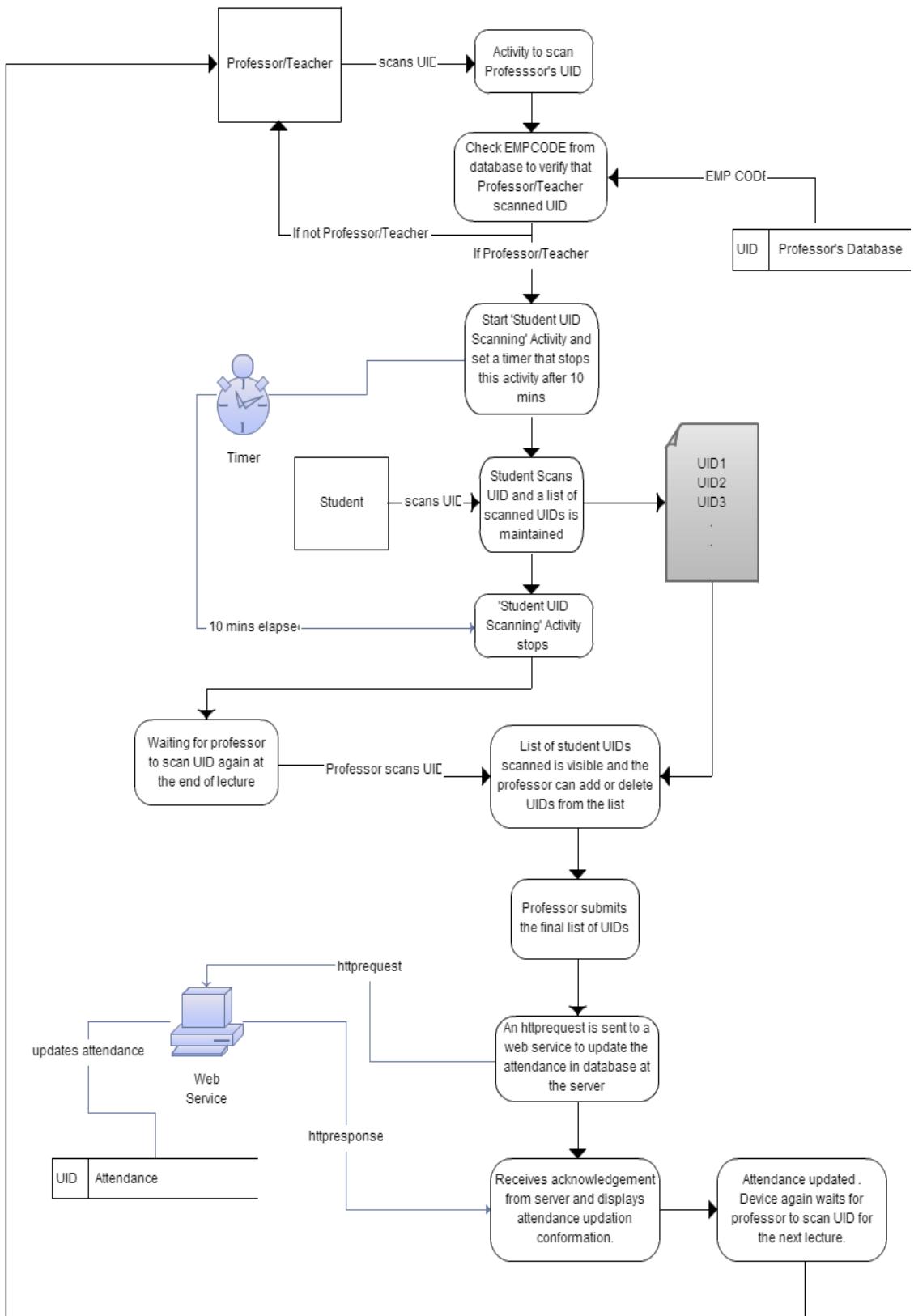


Fig. 5.2 Attendance Management System

New User Registration

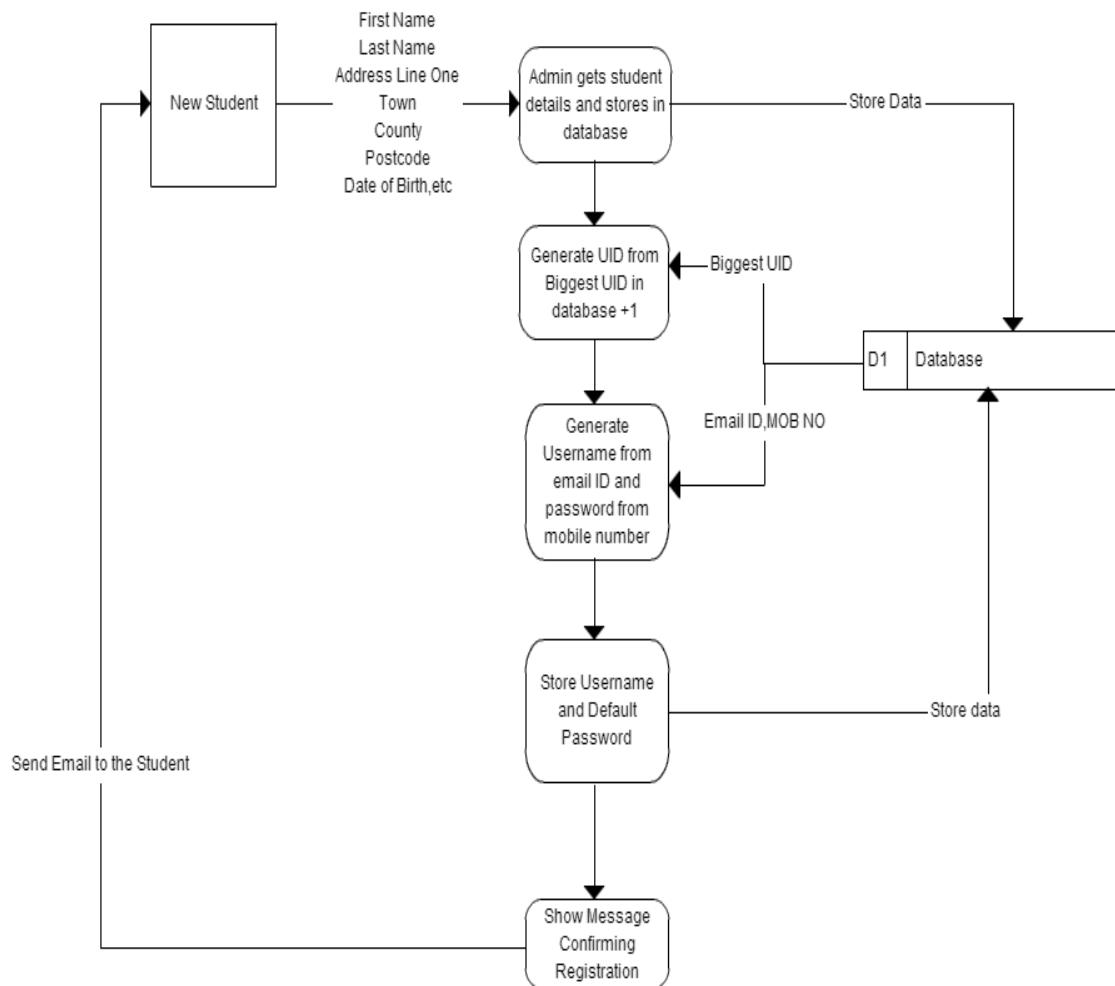


Fig. 5.3 Registration Process

User Login

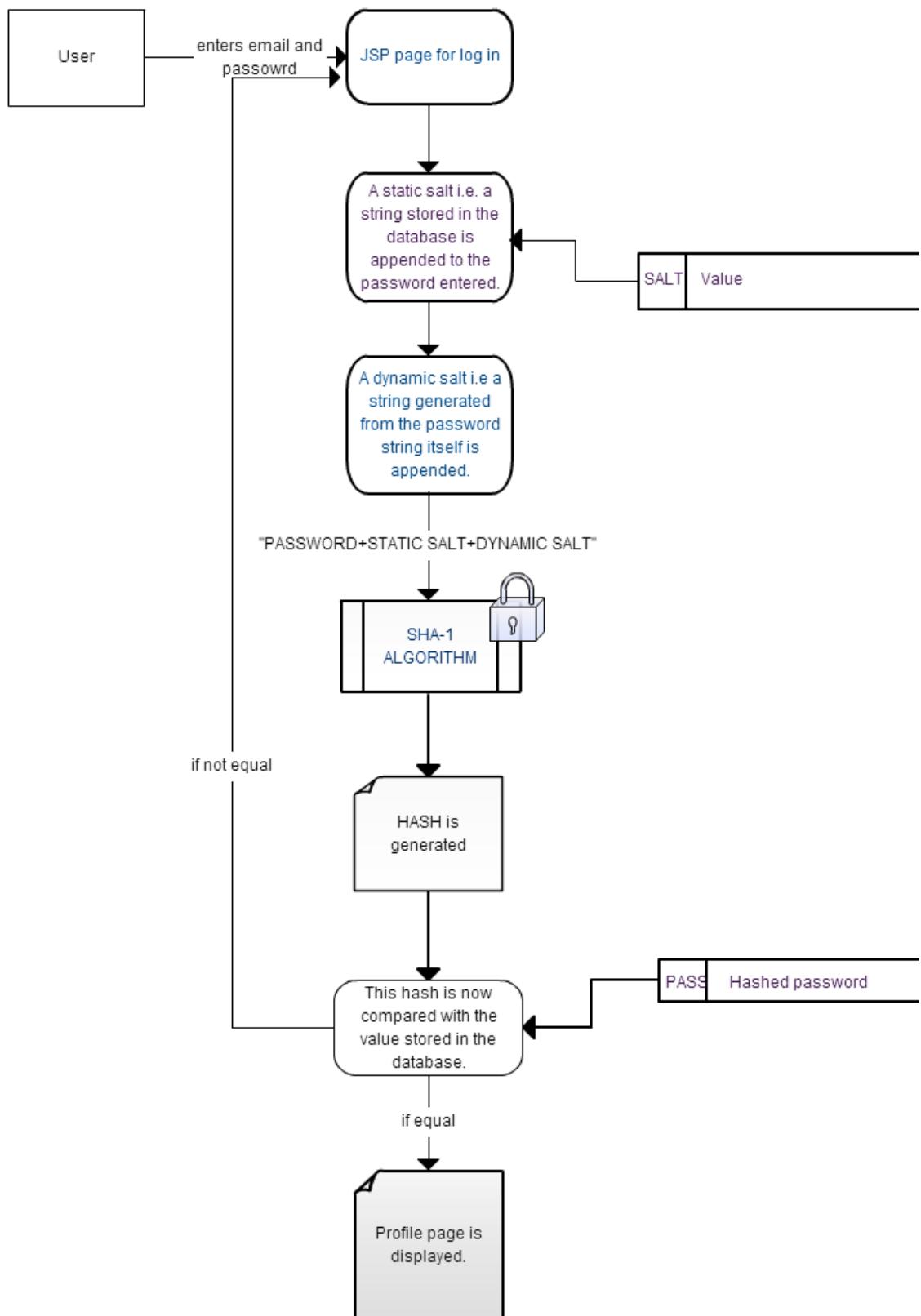


Fig. 5.4 User Login Process

Password Encryption

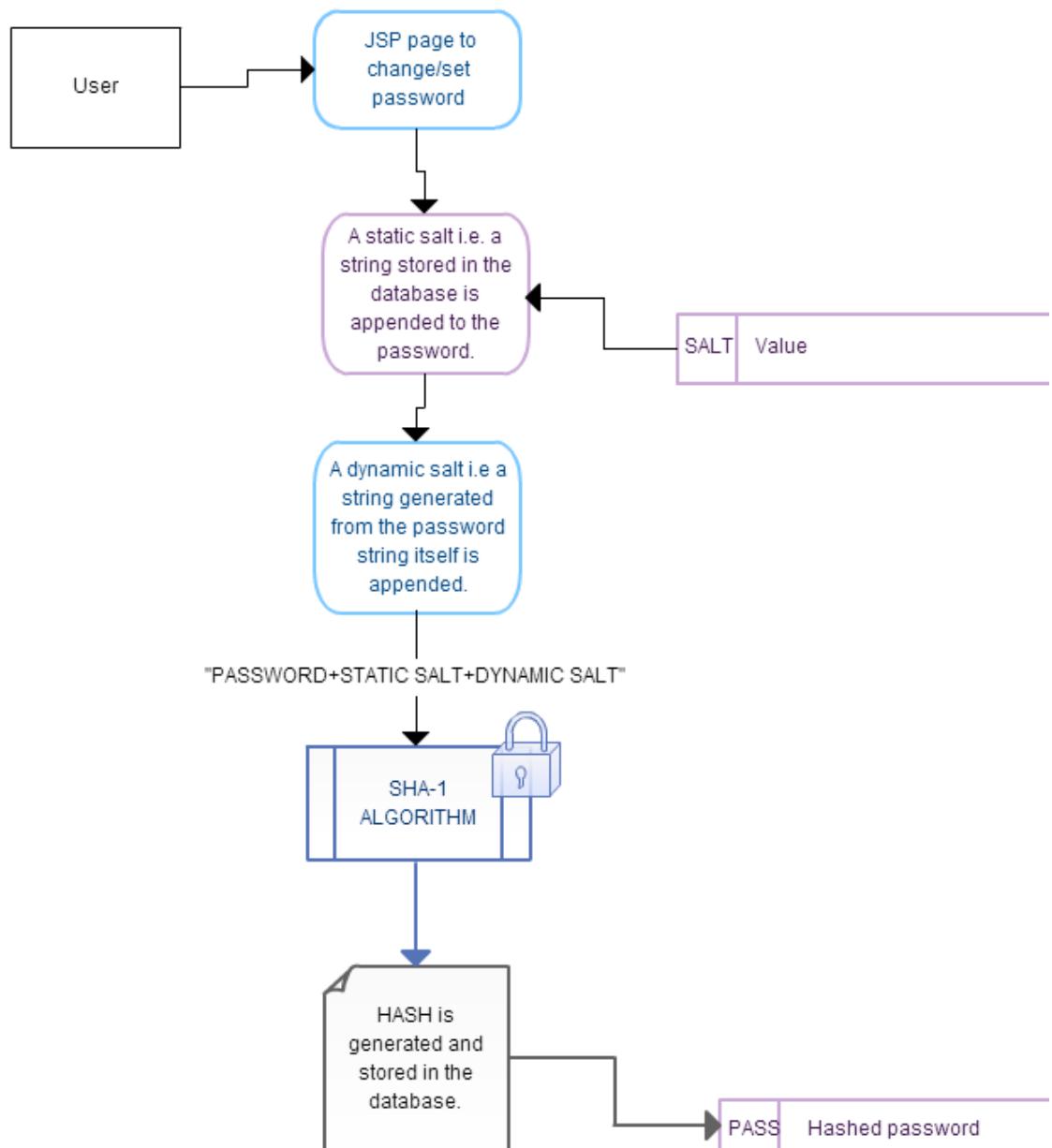


Fig. 5.5 Password Encryption while storing in database

Android Application for User handheld devices

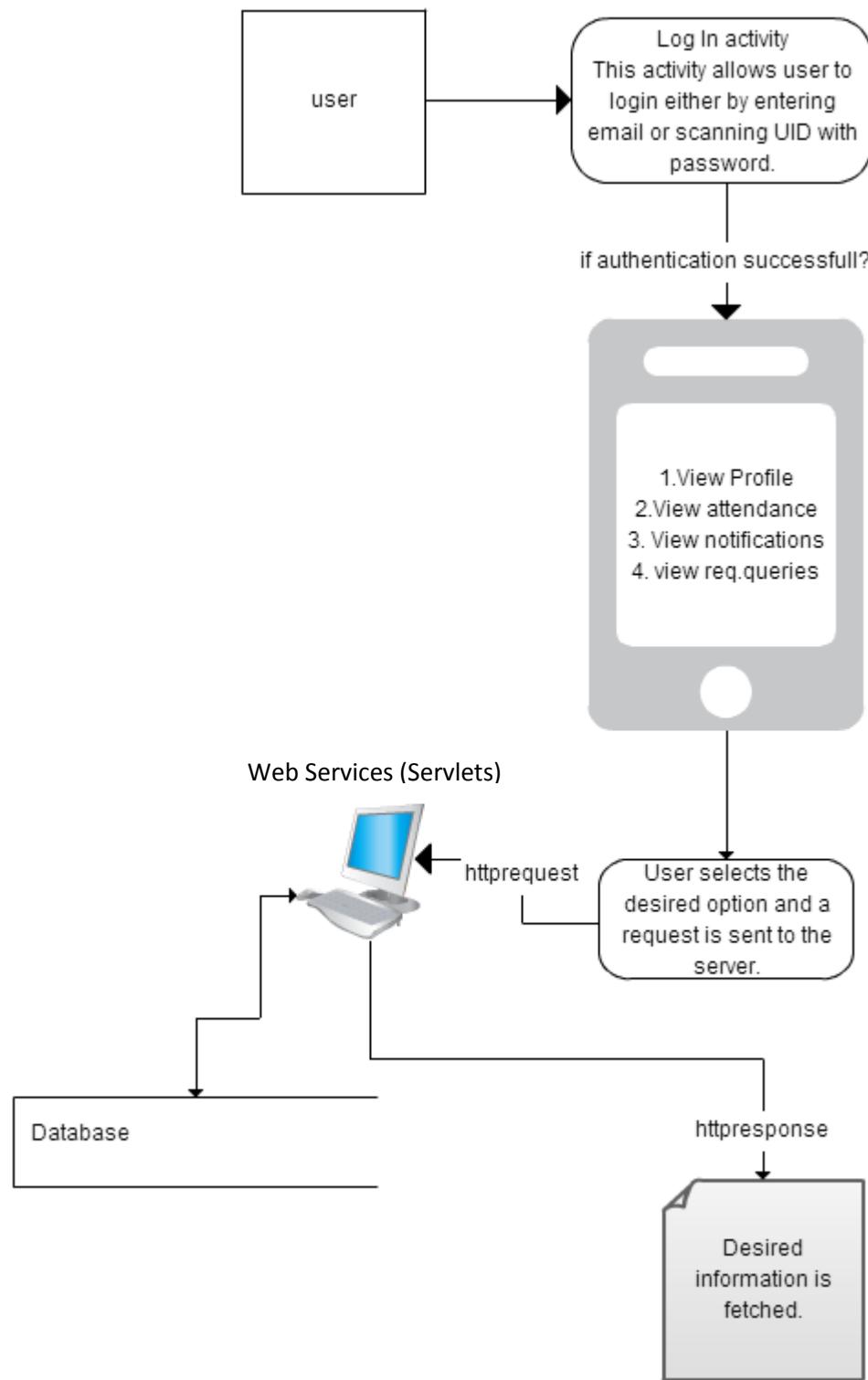
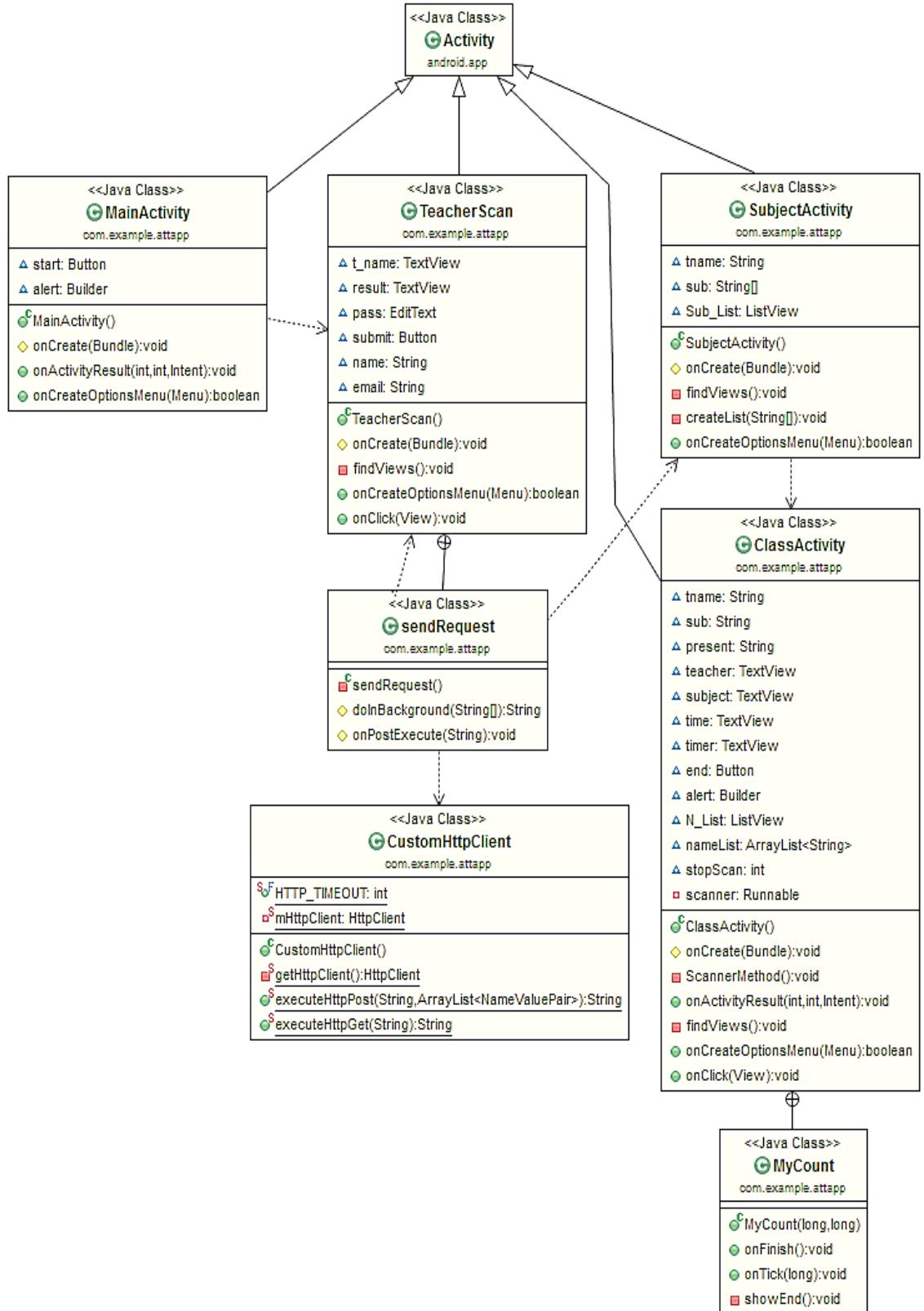


Fig. 5.6 Android Application for Users

5.3 CLASS DIAGRAMS

5.3.1 Android Application for Automatic Attendance

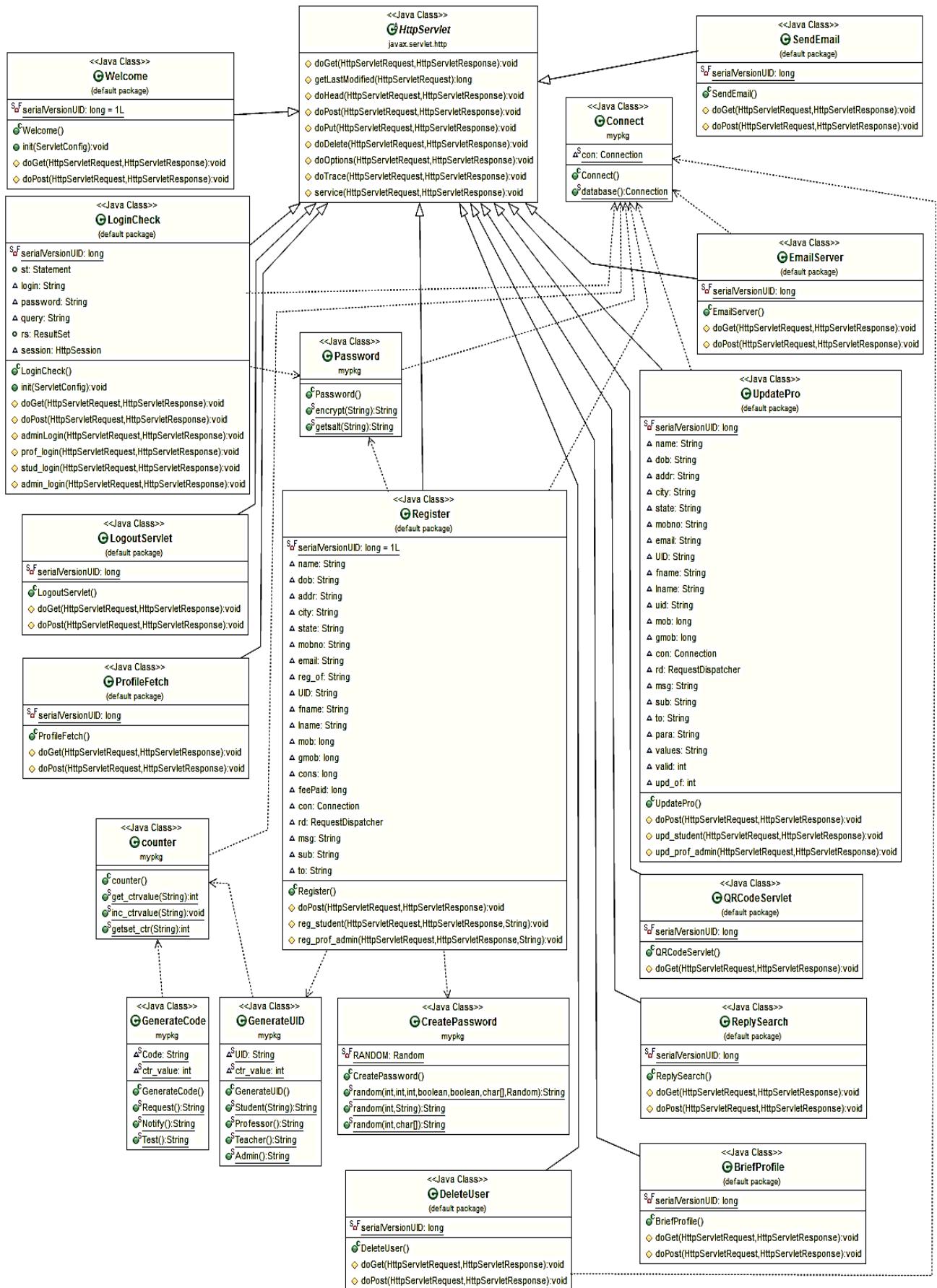
i.) User End Application Interface

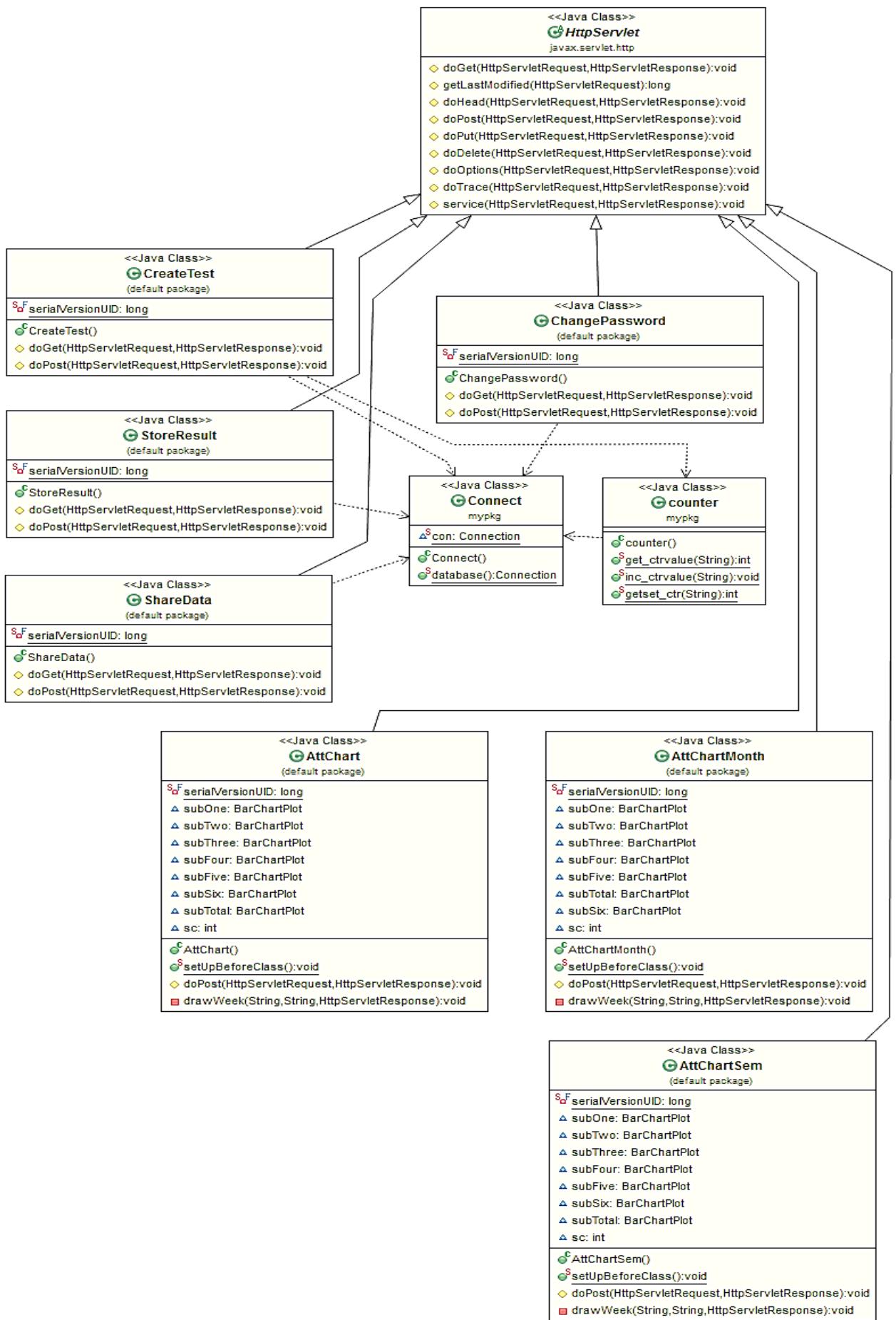


ii.) Interaction with the Server (Web Services - Servlets)



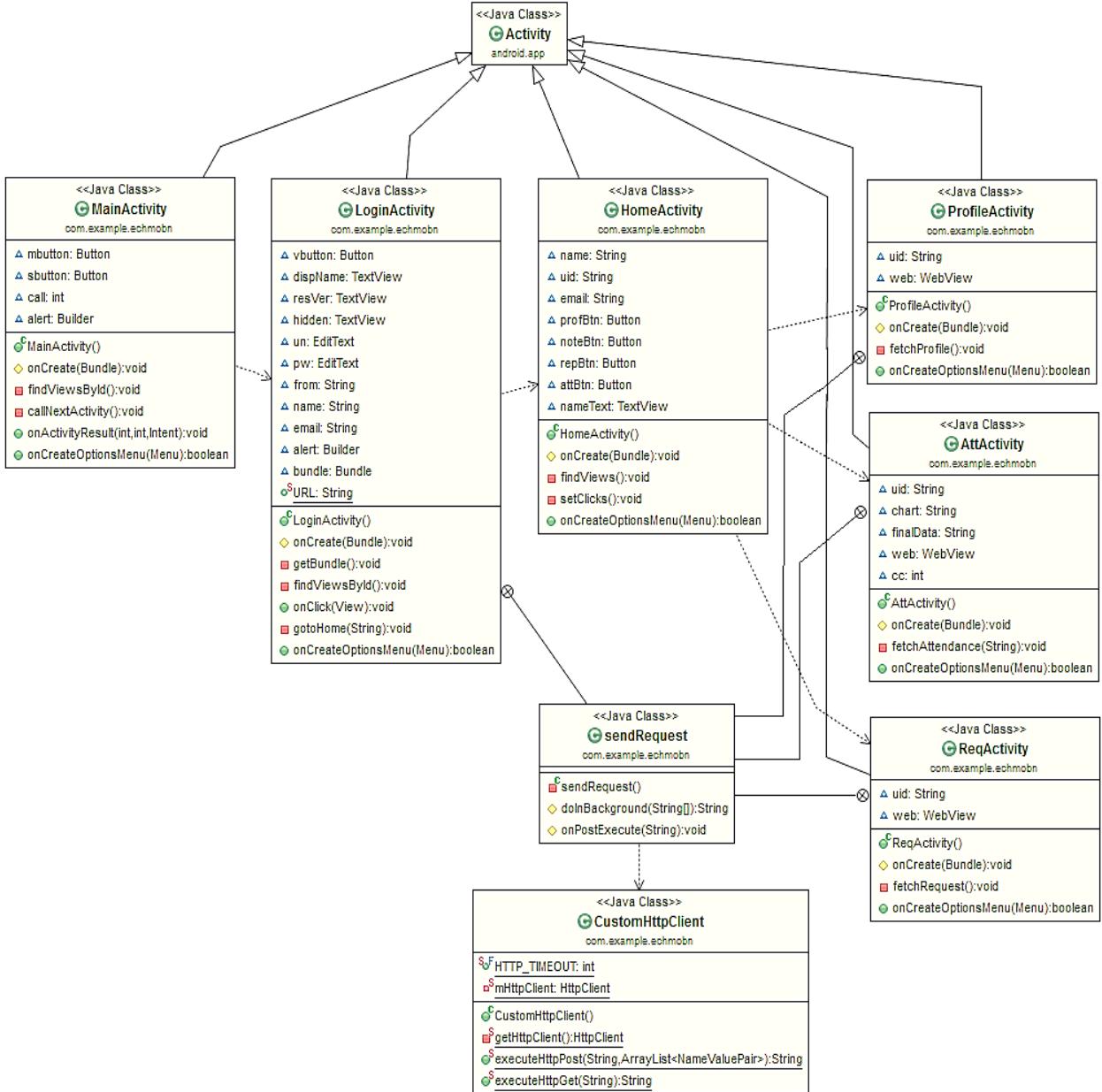
5.3.2 Online College Management System (Website)



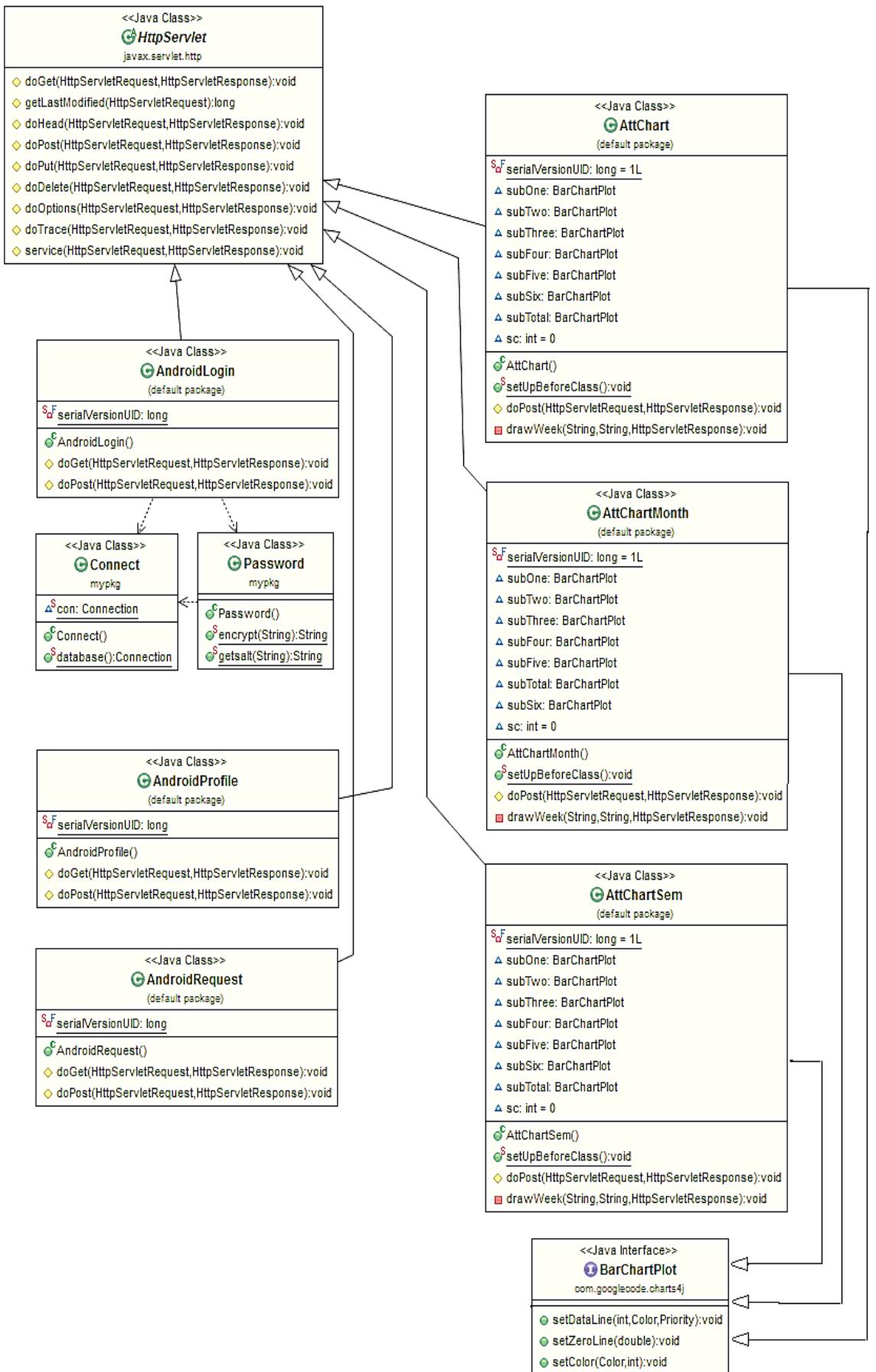


5.3.3 Android @ Help (Android Application)

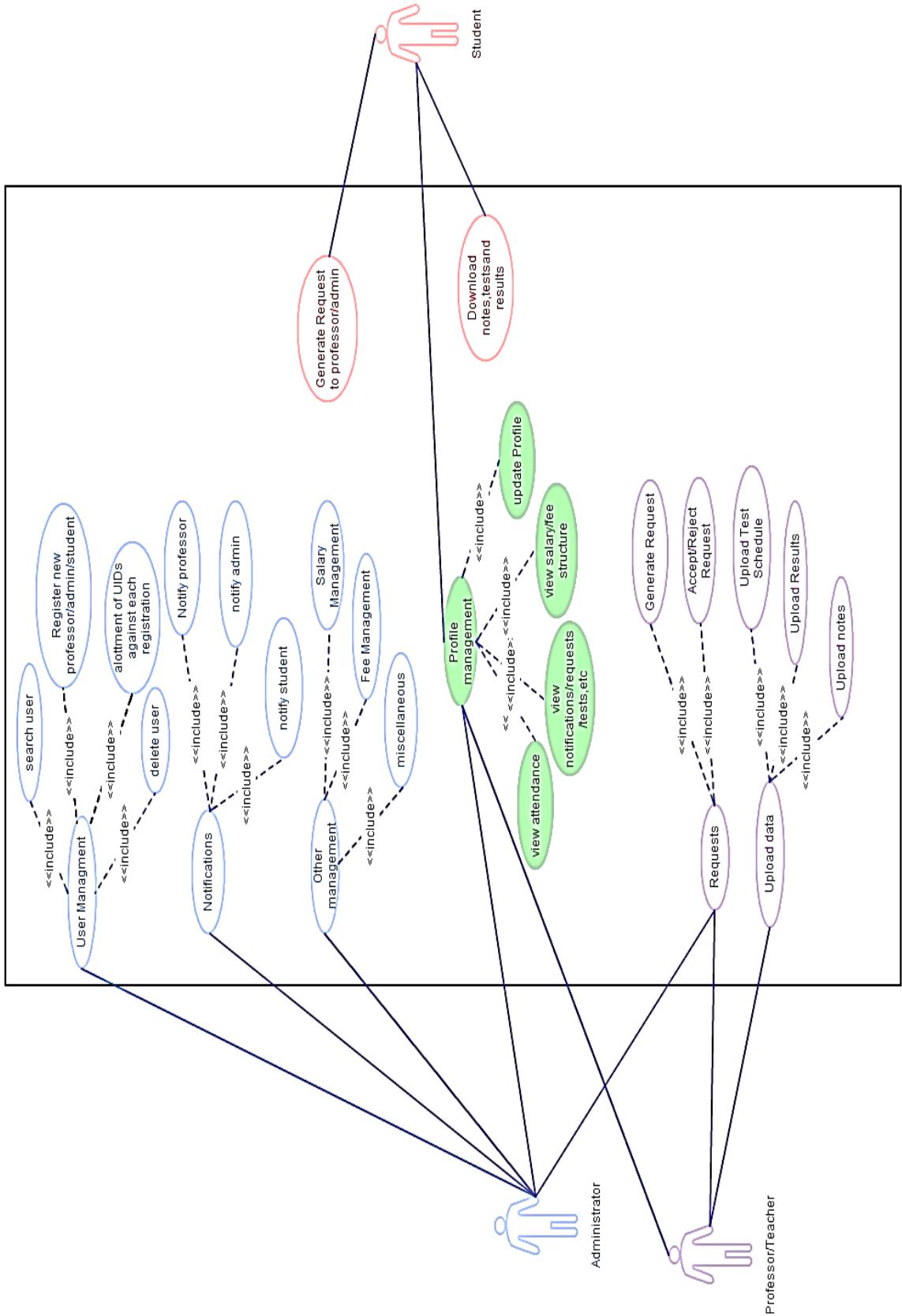
i.) User End Application Interface



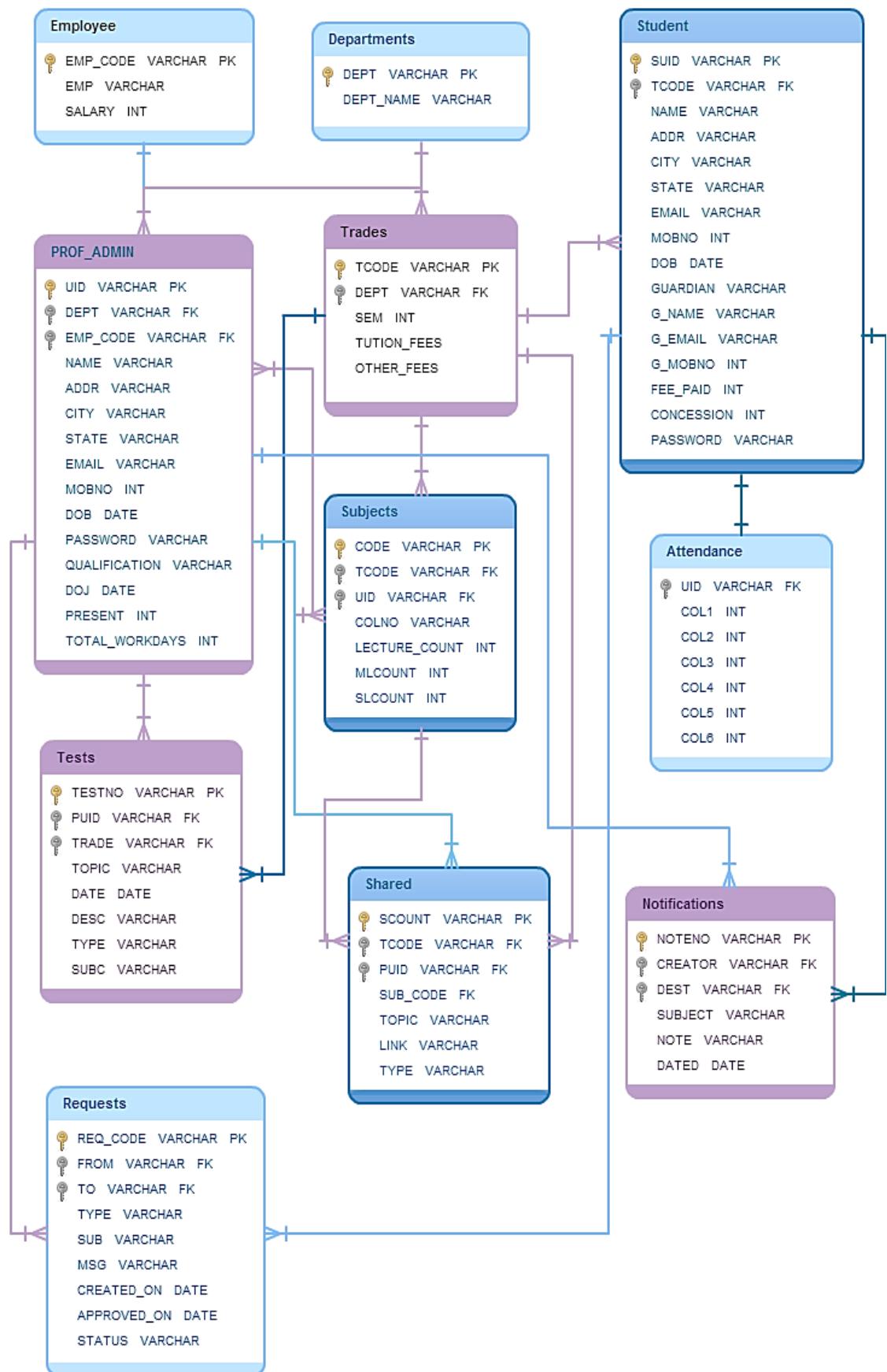
ii.) Interaction with the Server (Web Services – Servlets)



5.4 Use Case Diagram



5.5 Database Design



6. CODING

6.1 Attendance Management System (Android Application)

6.1.1 MainActivity.java

Main Activity presents the main screen of the attendance application to the user which simply allows the teacher to start a class after verification through his/ her QR Card. On click of the “Start Class” button on main screen, main activity calls the QR Code scanning activity.

```
start=(Button) findViewById(R.id.submitBtn);
start.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        try
        {
            Intent intent = new Intent("com.google.zxing.client.android.SCAN");
            intent.setPackage("com.google.zxing.client.android");
            intent.putExtra("com.google.zxing.client.android.SCAN_MODE", "QR_CODE_MODE");
            startActivityForResult(intent, 0);
        }
        catch(Exception e)
        {
            alert.setTitle("Info");
            alert.setMessage("Barcode Reader is not installed.\nPlease install free " +
                           "Barcode Scanner Application from Android market,provided by ZXING Team.");
            alert.setPositiveButton("OK", null);
            alert.show();
        }
    }
});
```

On successful scanning of teachers QR Card, a new activity is started asking for teacher’s password. In case, scanning fails then an error message is displayed and teacher must again scan his / her card.

```
public void onActivityResult(int requestCode, int resultCode, Intent intent) {
    if (requestCode == 0) {
        if (resultCode == RESULT_OK) {
            String data = intent.getStringExtra("SCAN_RESULT");
            // String format = intent.getStringExtra("SCAN_RESULT_FORMAT");
            String[] ss=data.split("#");
            Bundle bundle = new Bundle();
            bundle.putString("name",ss[0]);
            bundle.putString("email",ss[1].trim());

            Intent newIntent = new Intent(this.getApplicationContext(), TeacherScan.class);
            newIntent.putExtras(bundle);
            startActivity(newIntent);

        } else if (resultCode == RESULT_CANCELED) {
            alert.setTitle("Failed !");
            alert.setMessage("Failed to Scan Successfully. Try Again");
            alert.setPositiveButton("OK", null);
            alert.show();
        }
    }
}
```

6.1.2 TeacherScan.java

This Activity asks the teacher to provide his/her password. This password is same as that of his/her online profile on college management system. After successful verification, list of subjects that the teacher is currently teaching to various classes or departments is retrieved from the server and displayed on next activity/ view.

On verification failure, an error message is displayed and correct details are asked to be submitted.

```
public void onClick(View view) {
    String url="http://192.168.183.1:8080/EchWebsite/GetSubjects";
    sendRequest task = new sendRequest();
    task.execute(new String[] { url });
}
private class sendRequest extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls)
    {
        ArrayList<NameValuePair> postParameters = new ArrayList<NameValuePair>();
        postParameters.add(new BasicNameValuePair("email", email.toString()));
        postParameters.add(new BasicNameValuePair("password", pass.getText().toString()));

        String response = null;
        for (String url : urls) {
            try {

                response = CustomHttpClient.executeHttpPost(url, postParameters);
            } catch (Exception e) {
                result.setText(e.toString());
            }
        }
        return response;
    }
    protected void onPostExecute(String output) {

        if(output.charAt(0)=='V')
        {

            //result.setText(output);
            String[] subjs=output.split("#");
            Bundle bundle = new Bundle();
            bundle.putString("name", name);
            bundle.putStringArray("subjects", subjs);
            Intent newIntent = new Intent(TeacherScan.this, SubjectActivity.class);
            newIntent.putExtras(bundle);
            startActivity(newIntent);

        }
        else
            result.setText("Invalid Login Details");
    }
}
```

6.1.3 SubjectActivity.java

This activity gets the list of subjects that a teacher is associated with, and then displays it into a scrollable list view. The teacher can now select one of these subjects to start the class.

```
private void createList(String[] sub)
{
    ArrayList<String> noteList = new ArrayList<String>();
    for(int i=1;i<=sub.length-1;i++)
        noteList.add(sub[i]);
    ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,R.layout.list_item, noteList);
    Sub_List.setAdapter(arrayAdapter);

    Sub_List.setOnItemClickListener(new OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View view,int position, long id) {

            // selected item
            String subject = ((TextView) view).getText().toString().trim();

            // Launching new Activity on selecting single List Item
            Intent i = new Intent(getApplicationContext(), ClassActivity.class);
            // sending data to new activity
            Bundle bundle = new Bundle();
            bundle.putString("name", tname);
            bundle.putString("subject", subject);
            i.putExtras(bundle);
            startActivity(i);

        }
    });
}
```

6.1.4 ClassActivity.java

This activity is started when teacher initiates a class. The activity stores the start time and then allows the students to scan their QR Cards for a fixed amount of time. It repeatedly calls the QR code scanning activity and fetches details for each student. After the completion of fixed period no student is allowed to scan his card and thus is marked absent for this class.

A list of students present in class is then displayed in a scrollable view on the activity screen. Now, at the end of class teacher can submit this list to the server and feed their attendance for this lecture.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_class);

    Bundle bundle = this.getIntent().getExtras();
    tname = bundle.getString("name").trim();
    sub = bundle.getString("subject").trim();
    findViews();

    teacher.setText(tname);
    subject.setText(sub);

    Calendar c = Calendar.getInstance();

    SimpleDateFormat df = new SimpleDateFormat("HH:mm:ss dd-MM-yyyy");
    String currentTime = df.format(c.getTime());
    time.setText(currentTime);
    nameList = new ArrayList<String>();
    end.setOnClickListener(this);

    MyCount count=new MyCount(15000,1000);
    count.start();
    ScannerMethod();
}

private void ScannerMethod()
{
    this.runOnUiThread(scanner);
}
private Runnable scanner= new Runnable() {
    @Override
    public void run() {
        try
        {
            Intent intent = new Intent("com.google.zxing.client.android.SCAN");
            intent.setPackage("com.google.zxing.client.android");
            intent.putExtra("com.google.zxing.client.android.SCAN_MODE", "QR_CODE_MODE");
            startActivityForResult(intent, 0);
        }
        catch(Exception e)
        {
            alert=new AlertDialog.Builder(ClassActivity.this);
            alert.setCancelable(true);
            alert.setTitle("Failed to read UID");
            alert.setMessage("Try Again");
            alert.show();
        }
    }
};

public void onActivityResult(int requestCode, int resultCode, Intent intent) {
    if (requestCode == 0) {
        if (resultCode == RESULT_OK) {
            String feed = intent.getStringExtra("SCAN_RESULT").trim();
            String[] data=feed.split("#");
            namelist.add(data[0]);
            present +=data[1].trim()+"#";
            ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,R.layout.list_item, nameList);
            N_List.setAdapter(arrayAdapter);
            try {
                Uri notification = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
                Ringtone r = RingtoneManager.getRingtone(getApplicationContext(), notification);
                r.play();
            } catch (Exception e) {}

            if(stopScan!=1) ScannerMethod();
        } else if (resultCode == RESULT_CANCELED) {
            alert.setTitle("Please scan again..");
        }
    }
}

```

The Attendance Management application is responsible for interacting with the web services working online, the final task of feeding attendance in the database is done by these web services (servlets). Two of the major servlets functional on the server are as follows –

- GetSubjects.java – this servlets is responsible for validating the login details of a teacher who has scanned his QR card and password through the attendance application. On successful validation, a list of subjects taught by this teacher is returned by this servlet. Whereas in case of validation failure, an error message is displayed and login details are required to be corrected.

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String subjects="Valid";
    String email=request.getParameter("email");
    String pass="";
    try {
        pass = Password.encrypt(request.getParameter("password")).trim();
    } catch (Exception e1) {
        e1.printStackTrace();
    }
    PrintWriter out=response.getWriter();
    try{
        Connection con=Connect.database();
        Statement st = con.createStatement();
        String query="SELECT * FROM Anurag.PROF_ADMIN WHERE EMAIL='"+email
                    +"'AND PASSWORD='"+pass+"'AND (EMP_CODE ='P' OR EMP_CODE='T')";
        ResultSet rs=st.executeQuery(query);
        if(rs.next())
        {
            String PUID=rs.getString("UID");
            Statement st1=con.createStatement();
            String query1="SELECT SUB_NAME FROM Anurag.SUBJECTS WHERE UID='"+PUID+"'";
            ResultSet rs1=st1.executeQuery(query1);
            while (rs1.next())
            {
                subjects += "#"+rs1.getString("SUB_NAME");
            }
            out.println(subjects);
        }
        else
            {out.println("Invalid");}
    }
    catch(Exception e)
    {
        System.out.println("caught "+e);
    }
}
}

```

- UpdateAttendance.java – this servlet carries out the task of finally submitting the attendance of all students that were present in a class. The servlet is called when a teacher click on the ‘End Class’ button from the attendance application.

6.2 Online College Management System

6.2.1 College Management Website

College management website consists of a collection of jsp and html pages, which provides the front end user interface on the browser. At back end, these pages can interact with servlets and other jsp pages to feed / fetch data to and from the database.

Various important pages hyperlinked on the college website are as follows:

- Home.jsp – this page displays the homepage of the website. It provides navigation options to other pages along with the option for teacher, students and admins to login into their profiles. Visitors can traverse details about the college and its courses through navigation bar.
- AdminPro.jsp – this page is dynamically loaded with details of an administrator after he is successfully logged in. It displays his various personal details along with latest college notifications, updates, and events.
Further admin can see and take action on requests and queries from other users. He can search profiles of other users, and make changes in them.

Other functionalities associated with an admin profile include – create notifications for individual as well as universal users, to register new users, to delete existing users, to search and update user profiles, to generate reports and to carry fees and payroll management.

- StudPro.jsp – this page is dynamically loaded with details of a student after his successful login. It displays his personal details as stored on the college database. Profile contains option to update his profile and password. Students can see detailed information about their attendance in various subjects. Other functionalities provided to a student are – create queries and requests, see status of earlier queries, new notifications about tests and results, to see his marks / assignments / results etc. online, to see various note and links shared by various teachers and professors, to send leave application through email to the concerned administrator.
- TeacherPro.jsp - this page is dynamically loaded with details of a student after his successful login. It displays his personal details as stored on the college database. Profile contains option to update his profile and password. Other functionalities provided to a teacher / professor include create new test and assignments, to update new results, to share new notes and links, to take action on requests and queries made by various students, etc. Other options available to a teacher are to create requests / queries to concerned administrator and to see status of earlier queries.
- Updateform.jsp – any user can fill this update form to make request for change in his/ her earlier profile details. This page displays current details of a user and allows him to make changes in any of these details. Finally on submission of this form a profile update request is sent to the concerned authority to take action on it.

- Notification.jsp – this page is only accessible to administrators through their profiles and allows them to create universal notifications related to new update and events related to college. Notifications created by various administrators are displayed in profiles of all users.
- RegForm.jsp – this page serves as the registration form and is only accessible to authorised administrators. It allows them to register new users including admins, teachers / professors, students. After successful registration user details are sent to their email ids.
- Attendance.jsp – this page displays a details view of a student's attendance in various subjects over various time intervals including current week, month and the overall semester.
- Passforgot.jsp – this page is accessible to all users in case they have misplaced or lost their login details. New login details can be generated through this page only after successful validation of the applicant. It requires them to provide their registered email id, on which new details are sent.
- Courses.jsp – It contains details about various courses available in the college and allows visitors to read them online.

List of few important servlets (web services) functional on the website are as follows:

- Welcome.java – this servlet serves as the welcome page and is automatically called when a browser makes http request for the college website. It is responsible for setting various session variables and attributes in order to identify each user uniquely and thus provide modified results according to the user type.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    HttpSession session=request.getSession(true);
    session.setAttribute("loggedin",new String("false"));
    session.setAttribute("name",new String("Guest"));
    session.setAttribute("loginFail",new String("false"));
    session.setAttribute("afterLogout",new String("false"));

    response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); // HTT
    response.setHeader("Pragma", "no-cache"); // HTTP 1.0.
    response.setDateHeader("Expires", 0);

    RequestDispatcher rd = request.getRequestDispatcher("Home.jsp");
    rd.forward(request, response);
```

- Register.java – this servlet is called using a post request from the RegForm.jsp, that is when admin creates a new user. It simply stores the new user details in the database and creates an acknowledgment for its successful completion. Below is a code snippet depicting how a student is registered and his details are stored in database. Similar would be the registration of new admins and teachers with their respective details.

```

protected void reg_student(HttpServletRequest request, HttpServletResponse response, String passwrd) throws Exception {
    String gname=request.getParameter("gfname")+request.getParameter("lname");
    String gemail=request.getParameter("gemail");
    String gmobno=request.getParameter("gmobno");
    String trade=request.getParameter("trade");
    cons=Long.parseLong(request.getParameter("cons"));
    feePaid=Long.parseLong(request.getParameter("feePaid"));
    gmob=Long.parseLong(gmobno);

    con=Connect.database();
    con.setAutoCommit(false);
    Statement st=con.createStatement();
    String query="SELECT TCODE FROM Anurag.DEPARTMENT,Anurag.TRADES WHERE Anurag.DEPARTMENT.DEPT_NAME='"+trade
        +" AND Anurag.DEPARTMENT.DEPT=Anurag.TRADES.DEPT AND Anurag.TRADES.SEM=1";
    ResultSet rs=st.executeQuery(query);
    rs.next();
    String tcode=rs.getString("TCODE");
    String UID=GenerateUID.Student(trade);

    query="INSERT INTO Anurag.STUDENT VALUES('"+UID+"','"+name+"','"+addr+"','"+city+"','"+state+"','"+dob+"','"+mob+
        "','"+email+"','"+tcode+"','"+feePaid+"','"+gname+"','"+gemail+"','"+gmob+"','"+passwrd+"','"+cons+"')";
    st=con.createStatement();
    st.executeUpdate(query);

    query="INSERT INTO Anurag."+trade+"_ATTENDANCE(SUID) VALUES('"+UID+"')";
    st.executeUpdate(query);
    query="INSERT INTO Anurag."+trade+"_MATTENDANCE(SUID) VALUES('"+UID+"')";
    st.executeUpdate(query);
    query="INSERT INTO Anurag."+trade+"_SATTENDANCE(SUID) VALUES('"+UID+"')";
    st.executeUpdate(query);

    rd.forward(request, response);
}

```

Finally it dispatches an email to the user's email id containing his/her login details.

```

request.setAttribute("to",new String(email));
request.setAttribute("sub",new String("Registration Successfull !"));
request.setAttribute("msg",new String("Hello "+name+", are now successfully registered with "
    + " Echelon Institute of Technology." +
    "\n Below are your login details and you will receive your QR Card soon enough.\n" +
    "Continue Login with following Details : \n Username :" +email+ "\n Password: " +pass
    +"\n You may change your password for further use." ));
request.setAttribute("reqFrom",new String("Register"));
rd = this.getServletContext().getRequestDispatcher("/email.jsp");

```

- LoginCheck.java – this servlet is responsible for checking login details of various users. On successful validation of login details, it transfers control to respective profiles of the user and it also sets the session attributes related to this user. Otherwise it displays a login failure message and deviates them to another login page to feed the correct login details again.

Below is a code snippet of how a student login details are verified, similar would be the case for administrators and teachers with their corresponding appropriate details.

```

protected void stud_login(HttpServletRequest request, HttpServletResponse response)
{
    login=request.getParameter("uname");
    try {
        password=Password.encrypt(request.getParameter("password").toString()).trim();
    } catch (Exception e1) {
        e1.printStackTrace();
    }
    try {

        Connection con=Connect.database();
        st = con.createStatement();
        query="SELECT * FROM Anurag.STUDENT WHERE EMAIL='"+login+"'AND PASSWORD='"+password+"'";
        rs=st.executeQuery(query);
        if(rs.next())
        {
            session.setAttribute("loggedin",new String("STUD"));
            session.setAttribute("uid", new String(rs.getString("SUID")));
            session.setAttribute("name", new String(rs.getString("NAME")));
            session.setAttribute("loginFail", new String("false"));
            System.out.println("correct email");
            RequestDispatcher rd = request.getRequestDispatcher("Home.jsp");
            rd.forward(request, response);
        }
        else
        {System.out.println("wrong email");
        session.setAttribute("loginFail", new String("true"));
        RequestDispatcher rd = request.getRequestDispatcher("loginPage.jsp");
        rd.forward(request, response);}
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
}

```

- GenerateUID.java – this class is responsible for creating new uid for each new user, required at the time of registration.

```

public class GenerateUID {

    static String UID;
    static int ctr_value;

    public static synchronized String Student(String Trade) throws Exception
    {
        int year = Calendar.getInstance().get(Calendar.YEAR);

        ctr_value=counter.getset_ctr(Trade);
        String years=Integer.toString(year);
        UID="S"+years.charAt(2)+years.charAt(3)+"-"+Trade+"-"+ctr_value;
        return UID;
    }

    public static synchronized String Professor() throws Exception
    {
        ctr_value=counter.getset_ctr("P");
        UID="P-"+ctr_value;
        return UID;
    }

    public static synchronized String Teacher() throws Exception
    {
        ctr_value=counter.getset_ctr("T");
        UID="T-"+ctr_value;
        return UID;
    }

    public static synchronized String Admin() throws Exception
    {
        ctr_value=counter.getset_ctr("A");
        UID="A-"+ctr_value;
        return UID;
    }
}

```

- Password.java – this class is responsible for encrypting user passwords, which are then stored in the database and provides security against any hack.

```
public static synchronized String encrypt(String plaintext) throws Exception
{
    String text=plaintext+Password.getsalt(plaintext);
    MessageDigest md = null;
    try
    {
        md = MessageDigest.getInstance("SHA"); //step 2
    }
    catch(NoSuchAlgorithmException e)
    {
        throw new Exception(e.getMessage());
    }
    try
    {
        md.update(text.getBytes("UTF-8")); //step 3
    }
    catch(UnsupportedEncodingException e)
    {
        throw new Exception(e.getMessage());
    }

    byte raw[] = md.digest(); //step 4
    String hash = (new BASE64Encoder()).encode(raw); //step 5
    return hash; //step 6
}
```

- ChangePassword.java – this servlet is responsible for making changes in user passwords as are stored in the database. Users are allowed to change their passwords after validation of their older passwords as follows :

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

try {
    String uid=request.getParameter("uid").trim();
    String newpass=request.getParameter("newp").toString().trim();
    String oldp=Password.encrypt(request.getParameter("oldp").toString().trim());
    String newp=Password.encrypt(newpass);
    String q;
    ResultSet rt;

    Connection con=Connect.database();
    Statement st=con.createStatement();
    if(uid.charAt(0)=='S')
    {
        q="SELECT PASSWORD FROM Anurag.STUDENT WHERE suid like'%" +uid+"%'";
        rt=st.executeQuery(q);
        rt.next();
        if(rt.getString("PASSWORD").trim().equals(oldp))
        {
            q="UPDATE Anurag.STUDENT SET PASSWORD='"+newp+"' WHERE suid like '%"+uid+"%'";
            st.executeUpdate(q);
            response.getWriter().print("<h2>Password Changed Succesfully</h2>");
        }
        else
        {response.getWriter().print("<h2>Invalid Details : Try Again</h2>");}
    }
    else
    {
        q="SELECT PASSWORD FROM Anurag.PROF_ADMIN WHERE uid like'%" +uid+"%'";
        rt=st.executeQuery(q);
        rt.next();
        if(rt.getString("PASSWORD").trim().equals(oldp))
        {
            q="UPDATE Anurag.PROF_ADMIN SET PASSWORD='"+newp+"' WHERE uid like '%"+uid+"%'";
            st.executeUpdate(q);
            if(uid.charAt(0)=='A')
            {q="UPDATE Anurag.ADMIN SET PASSWORD = '"+newp+"' WHERE uid like '%"+uid+"%'";
            st.executeUpdate(q);}

            response.getWriter().print("Password Succesfully Changed");
        }
        else
        {response.getWriter().print("Invalid Details");}
    }
}

} catch (Exception e) {
// TODO Auto-generated catch block
response.getWriter().print("Error Occured : Try Again");
e.printStackTrace();
}
}

```

- Notifier.java – this servlet is responsible for creating new notifications and is accessible from admin profile only.

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String type=request.getParameter("type");
    String sender=request.getParameter("creator").trim();
    String dest=request.getParameter("dest").trim();
    String note=request.getParameter("msg").trim();
    String subj=request.getParameter("sub").trim();
    String[] rec=new String[20];
    String freq="";
    int i=0;
    if(type.equals("Gen")){
        if(dest.contains("all")){
            if(dest.contains("dpt")) {rec[i++]="ADPT";}
            else
                {if(dest.contains("c")){rec[i++]="ACSE";}
                if(dest.contains("e")){rec[i++]="AECE";}
                if(dest.contains("m")){rec[i++]="AME";}
                if(dest.contains("i")){rec[i++]="AIT";}
            }
        }
        else
        {
            if(dest.contains("a")){
                if(dest.contains("dpt")) {rec[i++]="DPTA";}
                else
                    {if(dest.contains("c")){rec[i++]="CSEA";}
                    if(dest.contains("e")){rec[i++]="ECEA";}
                    if(dest.contains("m")){rec[i++]="MEA";}
                    if(dest.contains("i")){rec[i++]="ITA";}
                }
            }
            if(dest.contains("s")){
                if(dest.contains("dpt")) {rec[i++]="DPTS";}
                else
                    {if(dest.contains("c")){rec[i++]="CSES";}
                    if(dest.contains("e")){rec[i++]="ECES";}
                    if(dest.contains("m")){rec[i++]="MES";}
                    if(dest.contains("i")){rec[i++]="ITS";}
                }
            }
        }
        i=i-1;
        freq="#";
        for(int x=0;x<=i;x++)
        {freq +=rec[x]+";";}
        freq=freq.substring(1,freq.length()-1);
    }
    else {freq=dest; }

    try {
        Connection con=Connect.database();
        Statement st=con.createStatement();
        String q="Insert into Anurag.NOTIFICATION (CREATOR,DEST,NOTE,DATED,SUBJECT) VALUES ('"+sender
                +"','"+freq+"','"+note+"','"+date.now().trim()+"','"+subj+"')";
        st.executeUpdate(q);
        response.getWriter().println("Notification Successfully Sent");
    } catch (SQLException e) { response.getWriter().println("Some Error Occured : Try Again");
        e.printStackTrace(); }
}

```

- AttChart.java – this servlet is responsible for generating weekly attendance chart when called from a student profile page. Similar is the working of AttChartMonth.java, AttChartSem.java which creates monthly and semester attendance reports respectively.

```

private void drawWeek(String uid, String res, HttpServletResponse response) throws IOException
{
    int total=0,totlec=0;
    double pert=0.0f;
    String[] slp=uid.split("-");
    String q;
    DecimalFormat df = new DecimalFormat("#.##");
    try {
        javax.naming.InitialContext ctx= new javax.naming.InitialContext();
        javax.sql.DataSource ds =(javax.sql.DataSource)ctx.lookup("java:comp/env/jdbc/MyDataSource");
        java.sql.Connection con = ds.getConnection();
        java.sql.Statement st = con.createStatement();
        q="SELECT SUB_NAME,COLNO,LECTURE_COUNT FROM Anurag.SUBJECTS,Anurag.STUDENT WHERE SUID='"+uid
          +" AND Anurag.SUBJECTS.TCODE=Anurag.STUDENT.TCODE";
        java.sql.ResultSet rs = st.executeQuery(q);
        while(rs.next()){
            java.sql.Statement str = con.createStatement();
            String w="SELECT "+rs.getString("COLNO").trim() +" FROM Anurag."+slp[1]+"_ATTENDANCE where uid='"+uid+"'";
            java.sql.ResultSet rsr = str.executeQuery(w);
            while(rsr.next()){
                int att=Integer.parseInt(rsr.getString(1).trim());
                total +=att;
                int lcount=Integer.parseInt(rs.getString("LECTURE_COUNT").trim());
                totlec +=lcount;
                double per;
                if(lcount!=0)
                {per=(att*100)/lcount;}
                else
                    per=0;

                Color col=newColor("C67171");
                switch(sc){
                    case 0: col=newColor("C67171"); break;
                    case 1: col=newColor("63B8FF"); break;
                    case 2: col=newColor("6495ED"); break;
                    case 3: col=newColor("A2CD5A"); break;
                    case 4: col=newColor("BC8F8F"); break;
                    case 5: col=newColor("CCCCCC"); break;
                }
            }
            BarChartPlot sub = Plots.newBarChartPlot(Data.newData(Integer.parseInt(df.format(per))), col,rs.getString("SUB_NAME").trim());
            switch(sc++){
                case 0: subOne=sub;break;
                case 1: subTwo=sub;break;
                case 2: subThree=sub;break;
                case 3: subFour=sub;break;
                case 4: subFive=sub;break;
                case 5: subSix=sub;break;
            }
            res +=<tr><td>"+rs.getString("SUB_NAME").trim()+"</td><td>"+lcount+"</td><td>"+att+"</td><td>" +per+"</td></tr>";
        }
        if(totlec!=0){ pert=(total*100)/totlec; }
        else
            pert=0;
        subTotal = Plots.newBarChartPlot(Data.newData(Integer.parseInt(df.format(pert))), newColor("388E8E"), "OverAll");
        res +=<tr><td>Over All</td><td>"+totlec+"</td><td>"+total+"</td><td>"+pert+"</td></tr>";
    }
    catch (Exception e) {
        response.getWriter().println(e.toString());
    }

    BarChart chart = GCharts.newBarChart(subOne,subTwo,subThree,subFour,subFive,subSix,subTotal);
    AxisStyle axisStyle = AxisStyle.newAxisStyle(BLACK, 10, AxisTextAlignment.CENTER);
    AxisLabels score = AxisLabelsFactory.newAxisLabels("Percentage ", 50.0);
    score.setAxisStyle(axisStyle);
    AxisLabels year = AxisLabelsFactory.newAxisLabels("Subjects", 50.0);
    year.setAxisStyle(axisStyle);
    chart.addYAxisLabels(AxisLabelsFactory.newNumericRangeAxisLabels(0, 100));
    chart.addXAxisLabels(year);
    chart.setSize(350,200);
    chart.setBarWidth(25);
    chart.setSpaceWithinGroupsOfBars(2);
    chart.setDataStacked(false);
    chart.setTitle("Weekly Attendance", FIREBRICK, 14);
}

```

```

        chart.setGrid(100, 10, 3, 2);
        chart.setBackgroundFill(Fills.newSolidFill(WHITE));
        LinearGradientFill fill = Fills.newLinearGradientFill(0, WHITE, 100);
        fill.addColorAndOffset(WHITE, 0);
        chart.setAreaFill(fill);
        String url = chart.toURLString();

        res += "</table></td><td colspan=\"2\"><img src=\""+url+"\"></img></td></tr></table>";

        response.getWriter().println(res);

    }

```

- QRCodeServlet.java – this servlet is responsible for generating QR Codes for each user, these QR Codes are visible in each user profile.

```

public class QRCodeServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
                         HttpServletResponse response) throws ServletException, IOException {

        String qrtext = request.getParameter("qrText");

        ByteArrayOutputStream out = QRCode.from(qrtext).to(ImageType.PNG).withSize(250, 250).stream();

        response.setContentType("image/png");
        response.setContentLength(out.size());

        OutputStream outStream = response.getOutputStream();

        outStream.write(out.toByteArray());

        outStream.flush();
        outStream.close();
    }
}

```

6.2.2 User End Android Application

This Android application allows students to view their details as stored on the database, further it allows them to see latest notifications, their attendance reports and they can also see status of their requests and queries.

Few of the important activities of the application are as below:

- `MainActivity.java` – this activity displays the main screen of application and allows the user to select one of the following two options to login into their profiles and access their information stored on the server –
 - i.) Manual Login – this option asks the student to provide his/her login details i.e. their email ids and passwords, which are also used to login onto their online profiles through college website. After successful login, home screen is displayed in the application.
 - ii.) Automatic Login – this option requires the user to scan his /her QR card and then provide his password. Required details are then automatically fetched from the card. After successful login, home screen is displayed.
- `HomeActivity.java` - this activity provides the following four options to a student user :
 - i.) To view his profile (`ProfileActivity.java`) – through this option a student can view various details associated with his profile, as are stored on the server / database. In case some updates are required he can generate am update request from his online profile.
 - ii.) To view his attendance report (`AttActivity.java`) – this option allows the students to view their attendance reports in a detailed and well-illustrated manner through the use of tables and charts. It displays their weekly, monthly and semester attendance of all their subjects.
 - iii.) To view latest Notifications & Updates (`NoteActivity.java`) – this option allows the user to view various notifications and updates related to new events and administrative decisions, etc. and also any new tests and results updated by their teachers /professors.
 - iv.) To view request/ queries status (`ReqActivity.java`) – a student can use his/her online profile to create requests and queries to various administrators and teachers. The status of these requests and queries can be viewed easily through this option in the android application.

The application in android devices is capable of interacting with the server through certain dedicated web services (Servlets) functional on the server. A list of few important servlets and their functionalities are as follows:

- `AndroidLogin.java` – this servlet is responsible for verifying the login details that are submitted by a user, only after successful validation students are allowed to use application otherwise an error message is displayed and login page is displayed again.

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String email=request.getParameter("email");
    String pass="";
    try {
        pass = Password.encrypt(request.getParameter("password")).trim();
    } catch (Exception e1) {
        e1.printStackTrace();
    }
    PrintWriter out=response.getWriter();
    try{
        Connection con=Connect.database();
        Statement st = con.createStatement();
        String query="SELECT * FROM Anurag.STUDENT WHERE EMAIL='"+email+"' AND PASSWORD='"+pass+"'";
        ResultSet rs=st.executeQuery(query);
        if(rs.next())
        {
            out.println(rs.getString("SUID"));
        }
        else
            {out.println("Invalid");}
    }
    catch(Exception e){}
}

```

- AndroidProfile.java – this servlet serves as the profile fetcher, when user clicks on the view profile button in his/her application then this servlet is called with his UID and thus his profile is fetched from the database and displayed in the application.
- AndroidRequest.java – this servlet is responsible for displaying a list of queries and request that are related/ created by a user through his online user profile. The user can thus see status of his various queries I they are accepted or got rejected by concerned administrator. On click of the ‘View Request /Queries status’ button in application, a http post request is made to this servlet which returns a tabular list of related data.

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String uid=request.getParameter("uid");
    String buffer=<table width='95%' align='center'><tr bgcolor='#c0c0c0'><th>Request Code</th>+
    "<th>Requested To</th><th>Details</th><th>Status</th></tr>";
    java.sql.ResultSet rs;
    try{
        javax.naming.InitialContext ctx= new javax.naming.InitialContext();
        javax.sql.DataSource ds =(javax.sql.DataSource)ctx.lookup("java:comp/env/jdbc/MyDataSource");
        java.sql.Connection con = ds.getConnection();
        java.sql.Statement st = con.createStatement();
        String q="SELECT * FROM Anurag.REQUESTS WHERE Anurag.REQUESTS.FROM='"+uid+"'";
        rs = st.executeQuery(q);
        while(rs.next()){
            buffer += " <tr><td>" +rs.getString("REQ_CODE")+"</td><td>" +rs.getString("TO")
            +"</td><td>" +rs.getString("SUB")+
            "</td><td>" +rs.getString("STATUS")+"</tr>";
        }
        buffer=buffer+"</table>";
        response.getWriter().println(buffer);
    }
    catch (Exception e) {
    }
}

```

6.3 Database Design

6.3.1 Database Name – COLLEGE

6.3.2 Tables - Description of various tables and their fields used in the project are as follows:

i.) ADMINS – this table is used to store the list of email ids and passwords of all administrators. Fields associated with this table are –

Key	Name	Data type	Length	Nullable
★	EMAIL	CHARACTER	40	No
	PASSWORD	CHARACTER	30	No
	UID	CHARACTER	30	No

ii.) PROF_ADMIN – this table is used to store the details of all professors / teachers and administrators in the college. Fields associated with this table are as follows –

Key	Name	Data type	Length	Nullable
★	UID	CHARACTER	20	No
	NAME	CHARACTER	30	No
	ADDR	CHARACTER	100	No
	CITY	CHARACTER	30	No
	STATE	CHARACTER	30	No
	DOB	DATE	4	No
	MOBNO	BIGINT	8	No
	EMAIL	CHARACTER	40	No
	QUALIFICATION	CHARACTER	30	No
	DEPT	CHARACTER	10	No
	EMP_CODE	CHARACTER	10	No
	DOJ	DATE	4	No
	PRESENT	BIGINT	8	No
	TOTAL_WORKDAYS	BIGINT	8	No
	PASSWORD	CHARACTER	40	No

iii.) STUDENT – this table is used to store details of all students in the college belonging to different departments and semester. Fields associated with this table are as follows –

Key	Name	Data type	Length	Nullable
★	SUID	CHARACTER	20	No
	NAME	CHARACTER	30	No
	ADDR	CHARACTER	100	No
	CITY	VARCHAR	30	No
	STATE	CHARACTER	30	No
	DOB	DATE	4	No
	MOBNO	BIGINT	8	No
	EMAIL	CHARACTER	30	No
	TCODE	CHARACTER	10	No
	FEE_PAID	BIGINT	8	No
	GUARDIAN	CHARACTER	30	No
	G_EMAIL	CHARACTER	30	No
	G_MOBNO	BIGINT	8	No
	PASSWORD	VARCHAR	50	No
	CONCESSION	BIGINT	8	Yes

iv.) SUBJECTS – this table consists of details about all subjects related to various departments and semesters in the college. This table contain info like subject name, its trade, professor associated with it, its total lecture count, etc. Fields associated with this table are as follows –

Key	Name	Data type	Length	Nullable
★	CODE	CHARACTER	10	No
	SUB_NAME	CHARACTER	50	No
	TCODE	CHARACTER	10	No
	UID	CHARACTER	20	Yes
	COLNO	CHARACTER	10	No
	LECTURE_COUNT	BIGINT	8	No
	MLCOUNT	BIGINT	8	Yes
	SLCOUNT	BIGINT	8	Yes

v.) DEPARTMENT – this table stores code and name for different departments in the college. It contain following two fields –

Key	Name	Data type	Length	Nullable
★	DEPT	CHARACTER	10	No
	DEPT_NAME	CHARACTER	50	No

vi.) EMPLOYEE – this table contain various employee codes for different type of employees in the college management working at different designations and in various departments. It also consists of their salary details. Fields associated with this table are-

Key	Name	Data type	Length	Nullable
★	EMP_CODE	CHARACTER	2	No
	EMP	CHARACTER	20	No
	SALARY	BIGINT	8	No

vii.) TRADES – this table is used to store trade codes for different trades in the college i.e. codes to distinguish between various department and their current semesters. Fields associated with this table are –

Key	Name	Data type	Length	Nullable
★	TCODE	CHARACTER	10	No
	DEPT	CHARACTER	10	No
	SEM	BIGINT	8	No
	TUTION_FEES	BIGINT	8	No
	OTHER_FEES	BIGINT	8	No

viii.) COUNTERS – this table is used to store values of various counters used throughout the website management. These counters assist in generation of new UIDS, request codes, test codes, notifications, etc. Various fields stored in this table are –

Key	Name	Data type	Length	Nullable
	CTR_NAME	CHARACTER	30	No
	VALUE	BIGINT	8	No

ix.) REQUESTS – this table is used to keep track of various requests and queries generated by different users and consists of info about its sender, receiver, status, etc. Fields associated with this table are as follows –

Key	Name	Data type	Length	Nullable
★	REQ_CODE	CHARACTER	30	No
	"FROM"	CHARACTER	30	No
	"TO"	CHARACTER	30	No
	"TYPE"	CHARACTER	20	No
	SUB	CHARACTER	100	No
	MSG	CHARACTER	200	No
	CREATED_ON	DATE	4	No
	APPROVED_ON	DATE	4	Yes
	STATUS	CHARACTER	20	No

x.) NOTIFICATION – this table is used to store details associated with various notifications i.e. its creator, intended recipients, date of creation, etc. Fields associated with this table are –

Key	Name	Data type	Length	Nullable
	NOTENO	INTEGER	4	No
	CREATOR	CHARACTER	20	No
	DEST	VARCHAR	50	No
	NOTE	CHARACTER	120	No
	DATED	DATE	4	No
	SUBJECT	CHARACTER	50	Yes

xi.) TESTS – this table consist of details about all tests/ assignments and results created by various teacher / professors. It consists of info such as test code, its creator, intended trade (class), topic / subject of test, date of test, result status, link to updated results, etc. Fields contained in this table are as follows –

Key	Name	Data type	Length	Nullable
	TESTNO	CHARACTER	10	No
	TOPIC	CHARACTER	30	No
	DESC	CHARACTER	100	No
	DATE	CHARACTER	20	No
	PUID	CHARACTER	20	No
	TRADE	CHARACTER	5	No
	SUBC	CHARACTER	10	No
	"TYPE"	CHARACTER	5	Yes

xii.) SHARED – it is used to store links to various notes and websites (external links), that are shared by various teachers and professors meant for their classes to study. Fields associated with this table are –

Key	Name	Data type	Length	Nullable
	SCOUNT	INTEGER	4	No
	TOPIC	CHARACTER	30	No
	TCODE	CHARACTER	10	No
	LINK	CHARACTER	120	No
	PUID	CHARACTER	20	No
	"TYPE"	CHARACTER	10	No
	SUB_CODE	CHARACTER	10	No

xiii.) CSE_ATTENDANCE / ECE_ATTENDANCE / ME_ATTENDANCE / IT_ATTENDANCE – these tables is used to store weekly attendances of all students in their respective subjects depending upon their respective departments. Fields associated with these tables are –

Key	Name	Data type	Length	Nullable
	SUID	CHARACTER	20	No
	SUB1	BIGINT	8	No
	SUB2	BIGINT	8	No
	SUB3	BIGINT	8	No
	SUB4	BIGINT	8	No
	SUB5	BIGINT	8	No
	SUB6	BIGINT	8	No

xiv.) CSE_MATTENDANCE / ECE_MATTENDANCE / ME_MATTENDANCE / IT_MATTENDANCE – these tables is used to store monthly attendances of all students in their respective subjects depending upon their respective departments. Fields associated with these tables are –

Key	Name	Data type	Length	Nullable
	SUID	CHARACTER	20	No
	SUB1	BIGINT	8	No
	SUB2	BIGINT	8	No
	SUB3	BIGINT	8	No
	SUB4	BIGINT	8	No
	SUB5	BIGINT	8	No
	SUB6	BIGINT	8	No

xv.) CSE_SATTENDANCE / ECE_SATTENDANCE / ME_SATTENDANCE / IT_SATTENDANCE – these tables is used to store attendances over whole semester, of all students in their respective subjects depending upon their respective departments. Fields associated with these tables are –

Key	Name	Data type	Length	Nullable
	SUID	CHARACTER	20	No
	SUB1	BIGINT	8	No
	SUB2	BIGINT	8	No
	SUB3	BIGINT	8	No
	SUB4	BIGINT	8	No
	SUB5	BIGINT	8	No
	SUB6	BIGINT	8	No

7. OUTPUT SCREENSHOTS

7.1 Automatic Attendance – Android Application

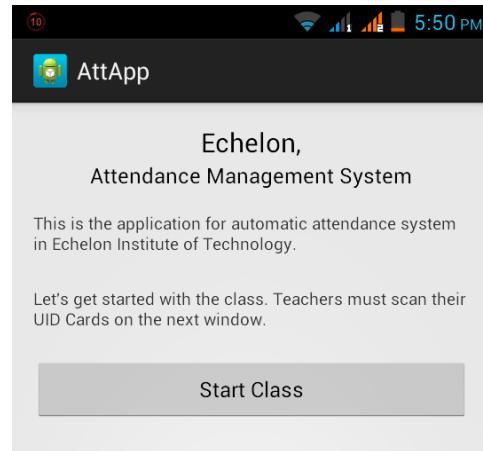


Fig. 7.1 Attendance Application (MainActivity)

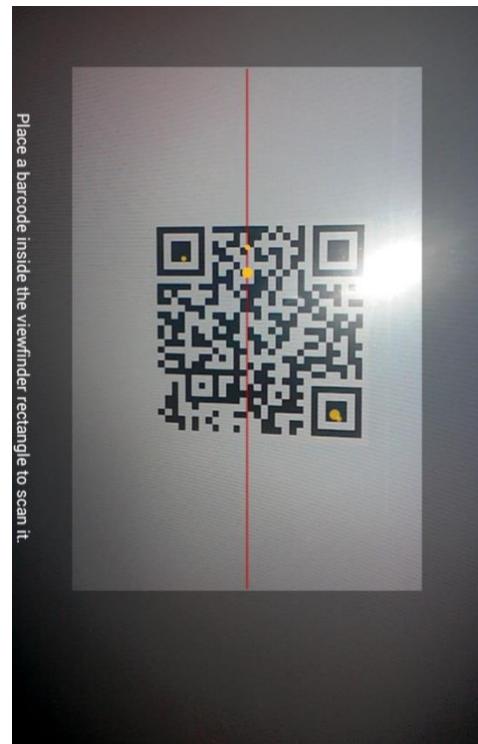


Fig. 7.2 QR Code Scanning (MainActivity)

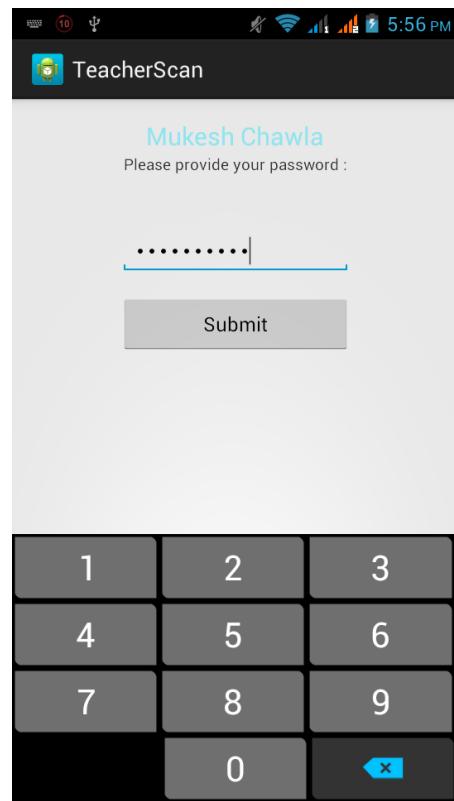


Fig. 7.3 Teacher/Professor Login Page (TeacherScan)

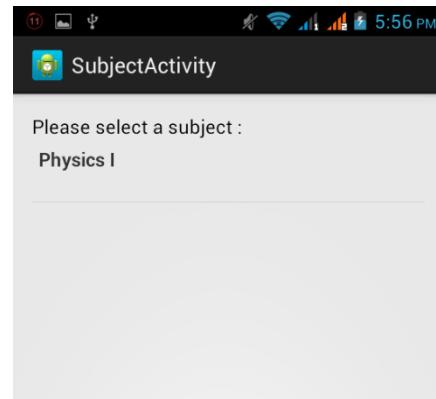


Fig. 7.4 List of subjects associated with a teacher (SubjectActivity)

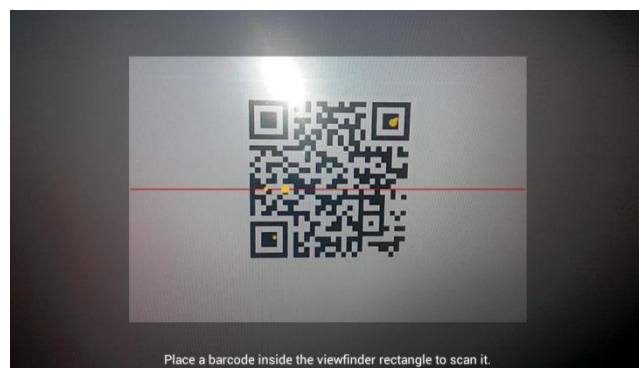


Fig. 7.5 Students scanning their QR Cards (ClassActivity)

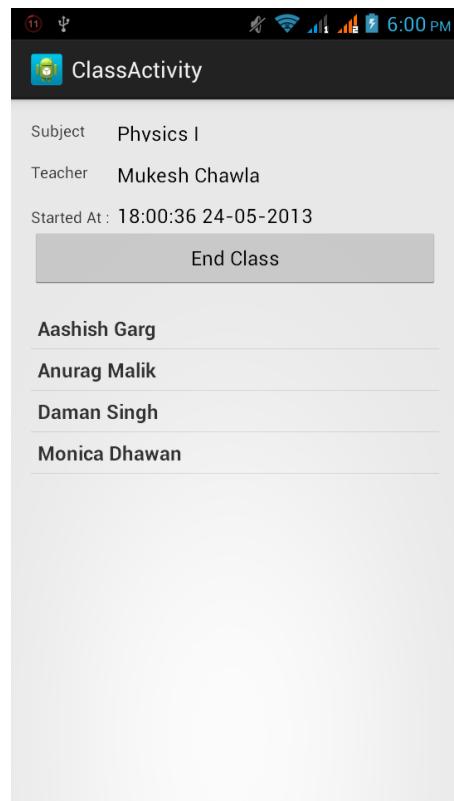


Fig. 7.6 List of students present in the class (ClassActivity)

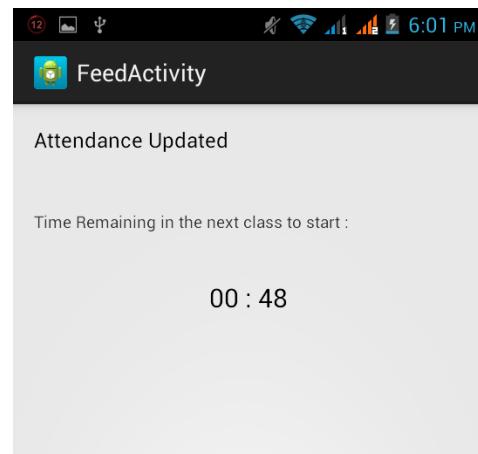


Fig. 7.7 Attendance sent to the server (FeedActivity)

7.2 Online College Management System (Website)

Welcome Guest , Login

ECHELON INSTITUTE OF TECHNOLOGY

About

Echelon Institute of Technology, Faridabad was established in the year 2007 on a sprawling campus spread over 17.5 acres. In the Year 2007-08, it had an intake capacity of 240 seats in CSE, IT, ECE and AEI (60 seats each). The institute was inaugurated by Shri R.R. Shah, IAS (Retd.), Member Secretary, Planning Commission, Govt. of India on 19th August 2007. The landmark feature of session 2007-08 was the lecture series which was inaugurated by Prof. Yashpal, the grand old man of science and followed by Dr. Naresh Gupta, CEO, Adobe India.

Mission & Vision

ECHELON is all set to produce highly skilled, trained, competent and confident engineers having passion for technology with positive attitude and an urge too become a successful professional and citizen.

Our Mission -

- * To provide quality education
- * To promote research and development activities in close association with industries and other institutions of high repute.
- * To help our students develop their skills and passion and graduate as responsible citizens of the country.

Login As :

Teacher Student

USERNAME (UID/EMAIL ID) :
Enter UID Here

PASSWORD :
Enter Password Here

Forgot Password

Login

Announcements

Echelon welcomes Sh. Ashok Thakur I.A.S., Secretary, MHRD, Govt of India on Apr 23rd, 2013.

The Institute is organizing 2nd National Conference NCNPECC-2013 with an objective to bring together the Scientists and Students.

Dr. Naresh Gupta, CEO ADOBE India, will be present amongst us on 5th of May to discuss the various facets of "Making an organisation".

Admissions Open

Events / Opportunities

NEWS & UPDATES

Date sheet for Second Sessional examinations April- May 2013 available for download.

Symposium is going to be held this friday on "Embedded System and its Industrial application".

A Hands-on Training on the ongoing trends of hacking. Basically objective is to get aware and secure our self by the recent technological flaws and

[Admin Login](#) | [Home](#) | [Student](#) | [Parents](#) | [Contact Us](#)

Director's Word : We, at Echelon Institute of Technology, are strongly committed to produce engineers who are men of vision and have high technical competence to evolve cost-effective business solutions as per the requirement of emerging market trends.	Extras Placements 2013 Placement Team Admission Broucher Echelonian Magazine	Sitemap Admissions Contacts Bus Routes Visit Us Jobs in EIT
--	---	---

@ Copyright of Echelon Institute of Technology.
Kabulpur,XYZ
Faridabad

Fig. 7.8 Home Page

Login As :

Teacher Student

USERNAME (UID/EMAIL ID) :
Enter UID Here

PASSWORD :
Enter Password Here

[Forgot Password](#)

Admin Login

USERNAME (UID/EMAIL ID) :
Enter UID Here

PASSWORD :
Enter Password Here

[Forgot Password](#) | [Close](#)

Fig. 7.9 Login boxes for Teacher / Students & Administrators

ECHELON INSTITUTE OF TECHNOLOGY

Hello User, You have been Succesfully Logged out.

You can now proceed to login again or go to Homepage

LOGIN AS : Admin Teacher Student

USERNAME :

PASSWORD :

[Forgot Password](#)

Fig. 7.10 Login Validation Failure

Welcome Power Admin , Logout

ECHELON INSTITUTE OF TECHNOLOGY

Admin Profile

Power Admin

Change Password | Update

UID – A-1
 Contact – 1000000001
 Email – admin@gmail.com
 Address – HNo. 100, Street 10 ,Delhi ,Delhi
 Date of Birth – 1990-12-23
 Joined On – 2013-03-13
 Qualifications–PHd, IIT Delhi
 Department–ECE



Fig. 7.11 Admin Profile (Upper Half)

USER MANAGEMENT

± Queries / Requests New						
Request Code	Sender	Details	Status	Options		
Req-1	S13-CSE-1	Change My Details	Pending	View	Accept	Reject
Req-12	S13-CSE-5	Profile Update Request	Rejected	View	Accept	Reject
Req-2	S13-CSE-2	Leave Application	ACCEPTED	View	Accept	Reject
Req-6	S13-ECE-1	Provide Fee Receipt	ACCEPTED	View	Accept	Reject
Req-7	P-1	Salary Receipt	REJECTED	View	Accept	Reject
Req-8	P-1	Update Address	Accepted	View	Accept	Reject
Req-9	S13-CSE-1	Fest Updates	Accepted	View	Accept	Reject

Fig. 7.21 Options to take action on requests / queries from various users

± View / Update User Profile

± Notification / Email

CONNECT

Create : Notification Email

TO* : Anurag Malik

SUBJECT* : Subject

MESSAGE * : Message Body

Send

User Management

Register New User	Search User	Delete User
-------------------	-------------	-------------

Reports / Updates

View Reports	Create Notification	Send Email
--------------	---------------------	------------

Others

Payroll Management	Fee Structure
Timetable Management	Academic Calender

Fig. 7.12 Other options Available in Admin Profile



ECHELON INSTITUTE OF TECHNOLOGY

Registration Form

Student

Personal Details

TITLE

Mr. Mrs.

FIRST NAME*

LAST NAME

FATHER'S NAME*

DATE OF BIRTH*

MM / DD / YYYY

GENDER*

Male Female

COURSE APPLIED FOR :*

Major

Trade

FEES PAID

in Rs.

in Rs.

CONTACT DETAILS*

Mobile

Email-Id

ADDRESS*

Street Address

Address Line 2

City

State / Province / Region

India

Postal / Zip Code

Country

In Case Of Emergency We Should Contact

GUARDIAN NAME :

First Name*

Last Name

CONTACT DETAILS*

Mobile

Email-Id

Submit

Fig. 7.13 Registration Form

Search User

Enter Name Or UID below :

Search by : NAME UID

CSE

Following users were found :

Anurag Malik (S13-CSE-1)

Monica Dhawan (S13-CSE-2)

Mukul Krishnatray (S13-CSE-3)

Gaurav Dhingra (S13-CSE-4)

Vikram Sharma (S13-CSE-5)

Siddharth Dhawan (S13-CSE-6)

Fig. 7.14 Search Page (depicting auto complete feature)

Search User

Enter Name Or UID below :

Search by : NAME UID

Anurag Malik

Name	Anurag Malik				
Classs	CSE 1	Roll NO	S13-CSE-1	Date of Birth	1990-12-23
Contact no.	8010566342	Email ID	anuraginmdu@gmail.com		
Guardian's Name	Vijay Malik				
Contact no.	9863809844	Email ID	vijaymalik@gmail.com		
Address	D 702, HPCL Society				
City	Greater Noida	State	Uttar Pradesh	PinCode	201310H
Fee Details					
Tution Fees:	10000				(in Rs.)
Others :	5000				
Total Fees:	15000				
Concession :	1000				
Paid Fees:	15000				
Due Fees:	29000				

[Print Details](#)

[Update User Profile](#)

[Delete User Profile](#)

Fig. 7.15 Profile Update & Delete Options available to Administrators

Create Notification

TO : All Administrators Teachers Students

DEPARTMENT : All CSE ECE ME IT

SUBJECT*:

Farewell Details

Subject Line

MESSAGE*:

Farewell for the final year is scheduled on 24th June, 2013. Third year students must submit their participation fees till 10th June, to Mr. Akhilesh Yadav.

URGENCY :

Important

[Send](#)

Fig. 7.16 Notification Page (Accessible only to Administrators)

Teacher Profile

Mukesh Chawla

Change Password | Update

UID – P-1
Dept – CSE
Designation- Professor
Contact – 9080706050
Email – mukeshk.chawla@gmail.com
Address – HNo. 10, EIT Hostel ,Delhi ,
New Delhi
Joined On – 2012-03-12
Qualification – PHd, IIT Delhi
Salary– 45000



Updates

Test / Assignment

Update Results

Share

Notifications

Exams Rescheduled (By- Power Admin Dated:2013-04-10)

Sessions has been postponed to new week.

Personal Notifications

Fig. 7.17 Teacher / Professor Profile

View Updates

Test / Assignment

Update Results

Share

± New Test / Assignment

± List of Scheduled Test and Assignments

Scheduled Tests/Assignments

Class Code No. Topic

Description

Dated

CSE 1 Sem T1003 Optics (Physics I) Full Chapter Test. Marks to be added in finals.

2013-03-05

Fig. 7.18 Options to create new tests and see list of earlier tests

View Updates

Test / Assignment

Results

Share

± Update New Results

± Previous Results

Previous Tests / Assignments Status

Class Code No.

Subject

Result Status

Last Updated On

Options

CSE 1 Sem R1003

Physics I

Updated

2013-4-10

View

Fig. 7.19 Options to update new results and see status of other results

View Updates	Test / Assignment	Update Results	Notes / Links
<p>± Share Notes/Links</p> <p>CLASS : <input type="text"/></p> <p>Subject / Class <input type="text"/></p> <p>Share : <input type="radio"/> Notes <input checked="" type="radio"/> Link</p> <p>TOPIC* : <input type="text"/></p> <p>SHARED LINK* : <input type="text"/></p> <p>document shared at this link</p> <p><input type="button" value="Share"/></p>			

Fig. 7.20 Options to share new notes and links and see details of other shared data

± Received Queries / Requests Status New						
Request Code	Sender	Details	Status	Options		
Req-10	S13-CSE-3	Update notes	ACCEPTED	<input type="button" value="View"/>	<input type="button" value="Accept"/>	<input type="button" value="Reject"/>
Req-11	S13-CSE-1	Hello Sir	ACCEPTED	<input type="button" value="View"/>	<input type="button" value="Accept"/>	<input type="button" value="Reject"/>
Req-3	S13-CSE-1	Update Attendance	ACCEPTED	<input type="button" value="View"/>	<input type="button" value="Accept"/>	<input type="button" value="Reject"/>
Req-4	S13-CSE-2	Mark Sheet	Pending	<input type="button" value="View"/>	<input type="button" value="Accept"/>	<input type="button" value="Reject"/>

Fig. 7.21 Options to take action on requests / queries from various students

<p>± Request/ Leave Application</p> <p>± Received Queries / Requests Status New</p> <p>± My Queries / Requests New</p>																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Request Code</th> <th>Requested To</th> <th>Details</th> <th>Status</th> <th colspan="2">Options</th> </tr> </thead> <tbody> <tr> <td>Req-7</td> <td>A-1</td> <td>Salary Receipt</td> <td>REJECTED</td> <td><input type="button" value="View"/></td> <td></td> </tr> </tbody> </table>						Request Code	Requested To	Details	Status	Options		Req-7	A-1	Salary Receipt	REJECTED	<input type="button" value="View"/>	
Request Code	Requested To	Details	Status	Options													
Req-7	A-1	Salary Receipt	REJECTED	<input type="button" value="View"/>													
<p>Sir, I am in need of a salary receipt on urgent basis.</p>																	
Req-8	A-1	<input type="button" value="Update Address"/>	<input type="button" value="Accepted"/>	<input type="button" value="View"/>													

Fig. 7.22 Other options available in teacher profile

Change Password

OLD PASSWORD*	<input type="password"/>
NEW PASSWORD*	<input type="password"/>
CONFIRM PASSWORD*	<input type="password"/>
<input type="button" value="Submit"/>	

Fig. 7.23 Change Password Option in User Profiles

Student Profile

Anurag Malik

[Change Password](#) | [Update](#)

UID – S13-CSE-1

Contact – 8010566342

Email – anuraginmdu@gmail.com

Address – D 702, HPCL Society ,Greater Noida ,Uttar Pradesh

Semester –1

Stream-CSE

Date of Birth –1990-12-23

Guardian Name –Vijay Malik

Guardian Email–vijaymalik@gmail.com

Guardian Contact Num –9863809844



QR Code of student

[Updates](#)

[View Attendance](#)

[Fee Details](#)

[Results/ Notes](#)

Notifications

Low Attendance (By- Power Admin , Date: 2013-04-09)

Please maintain attendance above min 75%

Personal Notifications

Test / Results

A Test has been Scheduled for subject CSE-01 on topic Optics by Mukesh Chawla on 2013-03-05

Full Chapter Test. Marks to be added in finals.. Test code -1003

Fig. 7.23 Student Profile

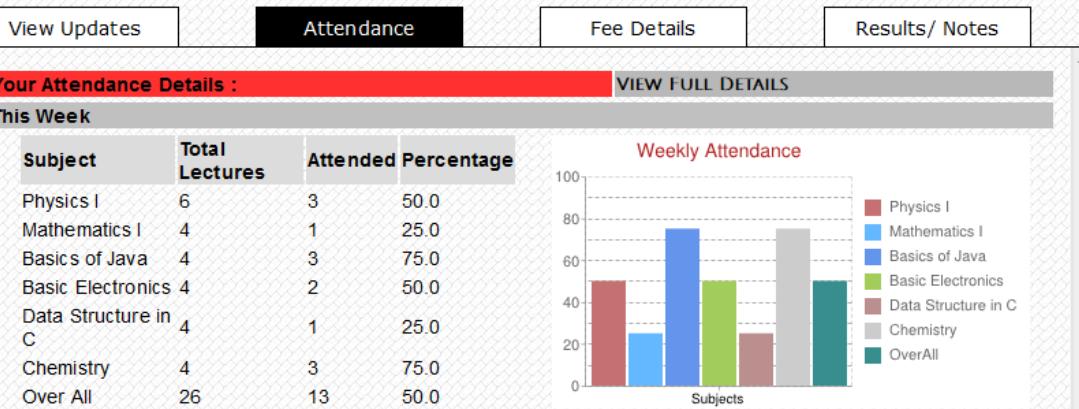


Fig. 7.24 Student Attendance Report (In Profile – Weekly)

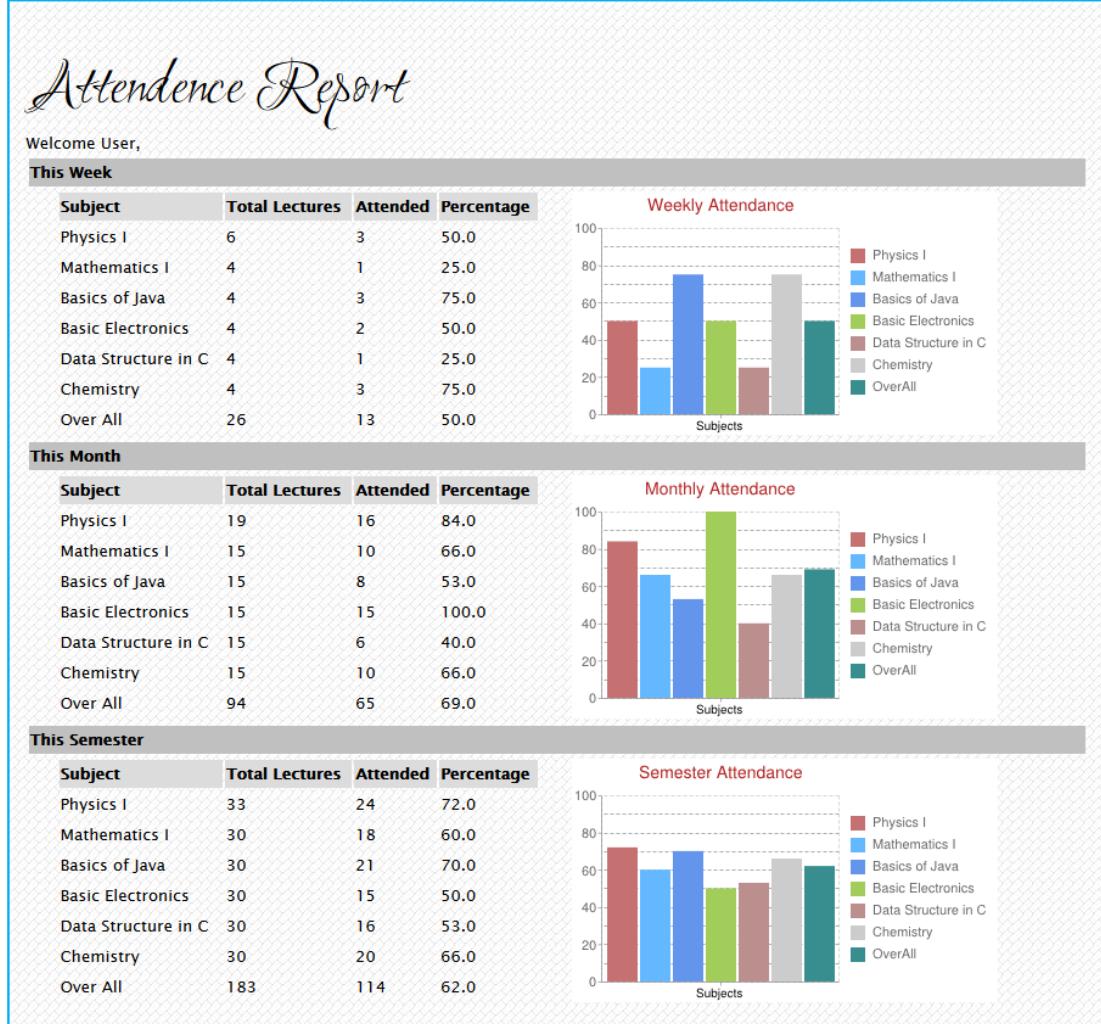


Fig. 7.25 Detailed Attendance Report

View Updates	View Attendance	Fee Details	Results/Notes		
Latest Results :					
Test Code	Subject	Result Status	Updated By	Last Updated On	Options
R1001	Mathematics I	Updated	Vikas Nagpal	2013-4-8	View
R1002	Mathematics I	Pending	- - -	- - -	- - -
R1003	Physics I	Updated	Mukesh Chawla	2013-4-10	View
Latest Notes / Links :					
On Topic	Shared by	Shared At	Type		
Optics (Physics I)	Mukesh Chawla	http://www.dropbox.com?top=optics	Notes		
Differentiation (Mathematics I)	Vikas Nagpal	http://www.diffworld.com	Notes		
Electro magnetics (Physics I)	Mukesh Chawla	www.w3schools.com	Link		
Algebra (Mathematics I)	Vikas Nagpal	http://www.eeasyalgebra.com	Link		

Fig. 7.25 Option to see Latest Results and Notes/ Links

± Request / Leave Application

CONNECT

Create : Request Leave Application

TO* : Power Admin

SUBJECT* : Going Out of town for a week

MESSAGE* : Sir, Kindly grant me leave for next week, as i have to attend a marriage anniversary in Himachal and thus will not be able to attend classes.
Thanking You.
Anurag Malik
S13-CSE-1

Send

± Queries / Requests New

Fig. 7.26 Creating new request/leave application in Student Profile

Update Profile

Your Current Profile Details :

Name – Anurag Malik
Contact – 8010566342
Email – anuraginmdu@gmail.com
Address – D 702, HPCL Society ,Greater Noida ,Uttar Pradesh
Date of Birth –1990-12-23
Guardian Name –Vijay Malik
Guardian Email–vijaymalik@gmail.com
Guardian Contact Num –9863809844

Fill New Details Here :

YOUR NAME :	FULL NAME	
DATE OF BIRTH	DD-MM-YYYY	
CONTACT DETAILS*	Mobile	Email-Id
ADDRESS		
GUARDIAN NAME :		
CONTACT DETAILS*	Full Name*	example@gmail.com
Send To :	Mobile	Email-Id
Update		

Fig. 7.27 Profile Update option in user profiles

ECHELON INSTITUTE OF TECHNOLOGY

Bachelors in Technology**Masters in Business Administration**

Bachelors in Technology

Computer Science and Engineering**Mechanical Engineering****Electronics & Communication Engineering****Information Technology**

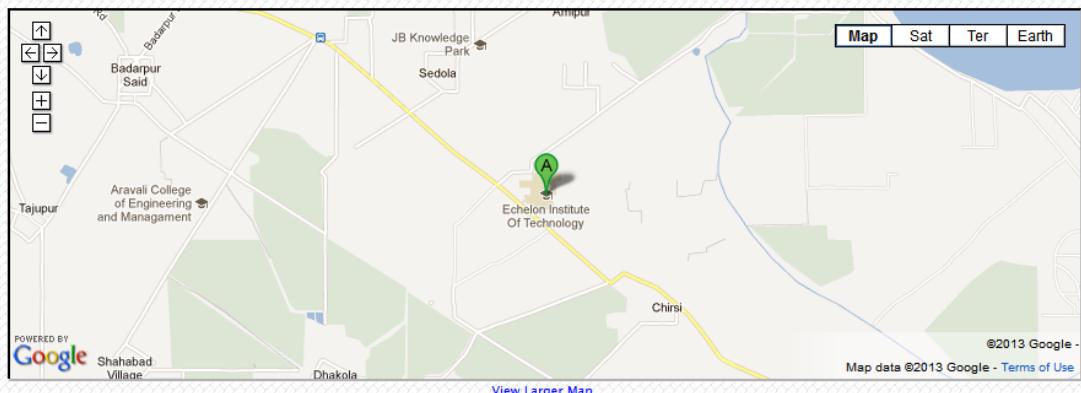
Department of Electronics and Communication Engineering

Apart from a good foundation in Electronics, the Electronics and Communication Engineering students are provided in-depth knowledge to design, fabricate, maintain, supervise and manufacture electronic equipment relevant to entertainment media, health care, IT industry, communication and defence. They work with microprocessors, microcontrollers and fiber optics for applications in automation.

Fig. 7.28 Courses details page

Contact Details

Way to Echelon Institute of Technology



Or Call Us :

Echelon Institute of Technology

Kabulpur, Jasana-Manjhawali Road, Naharpur

Faridabad-121101

Haryana, India

Phone: 0129-2201060

Mob: 9718312584

Fax: +91-129-2201040

info@echeloninstitute.com

For Admission queries:

Call: 9718312586, 9718312587

Email:admissions@echeloninstitute.com

Fig. 7.29 Contact details page

7.3 Android @ Help (Android application)

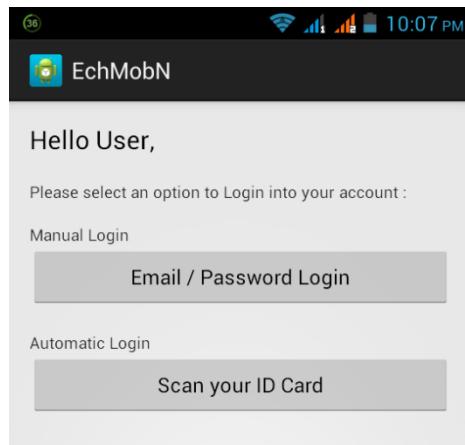


Fig. 7.30 Main Screen (MainActivity)

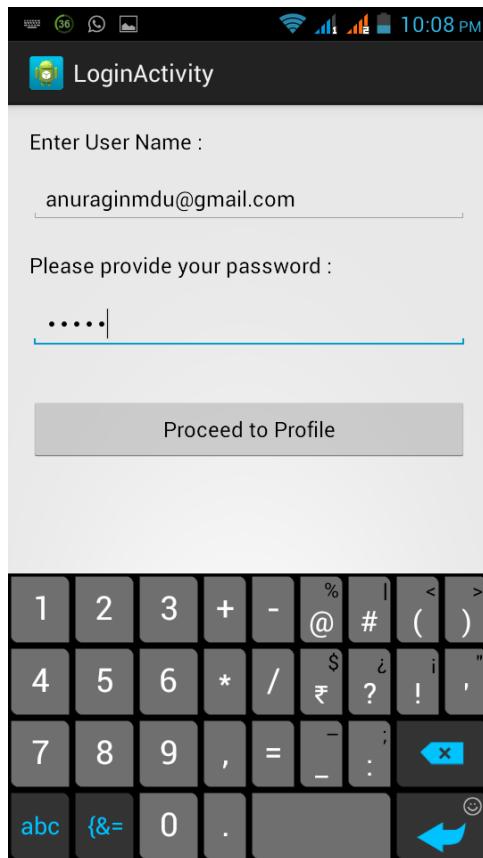


Fig. 7.31 Manual Login (LoginActivity)

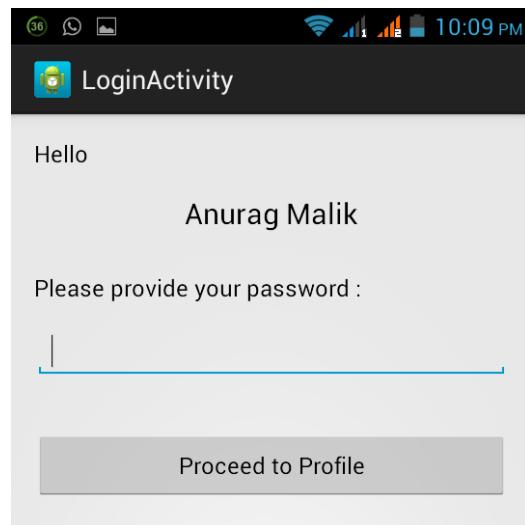


Fig. 7.32 Automatic Login (LoginActivity)

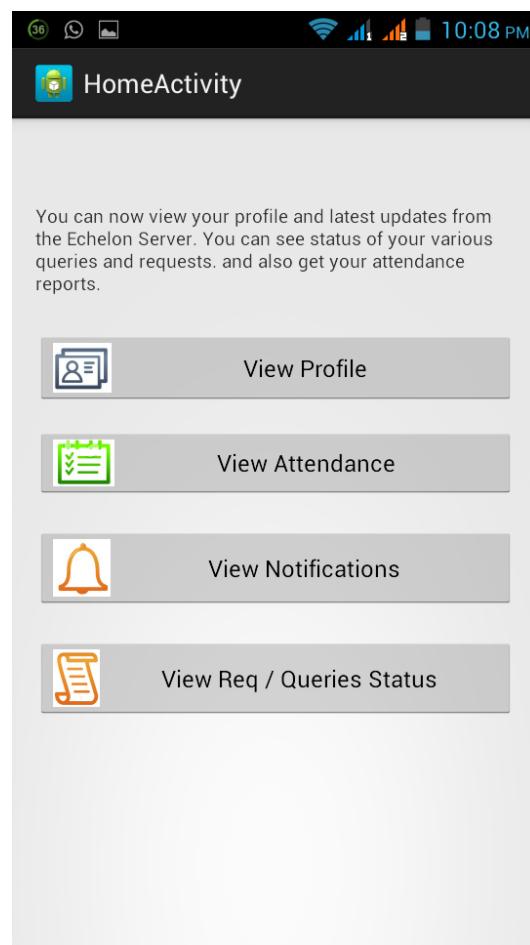


Fig. 7.33 Home Screen (HomeActivity)

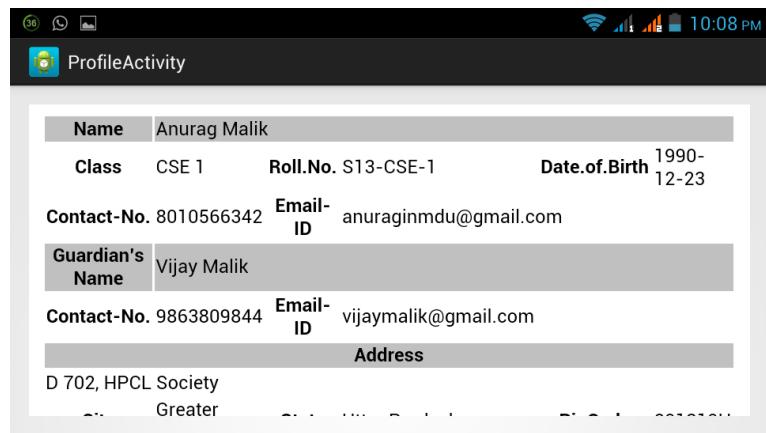


Fig. 7.34 Student Profile (ProfileActivity)

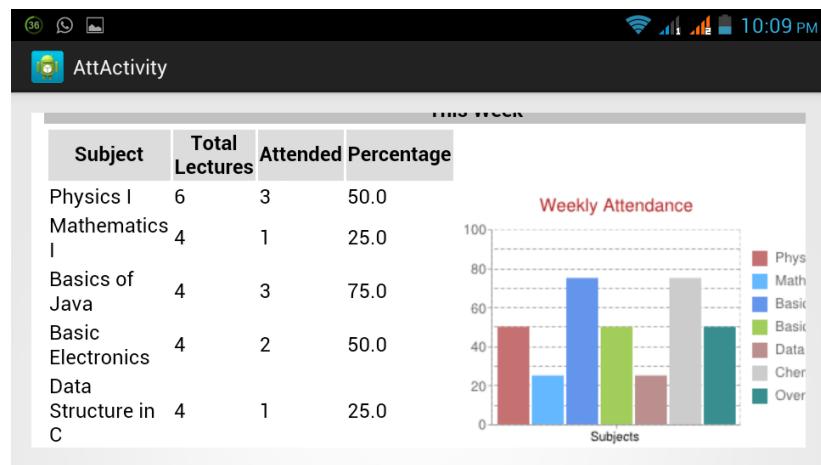


Fig. 7.35 Attendance Report (AttActivity)

The screenshot shows a mobile application interface titled "StatusActivity". It displays a table of request status.

Request Code	Requested To	Details	Status
Req-1	A-1	Change My Details	Pending
Req-11	P-1	Hello Sir	ACCEPTED
Req-3	P-1	Update Attendance	ACCEPTED
Req-9	A-1	Fest Updates	Accepted

Fig. 7.36 Status of various request / queries (ReqActivity)

8. CONCLUSION

The project titled “**QR code based Automatic Attendance & Online College Management System**” was successfully completed and served its functionality of providing an efficient way of attendance management in colleges and other such institutions. Further it was capable of providing online management options related to routine work in colleges and other such institutions where different departments are to work in an integrated manner. The said project provides us with a futuristic view of how smart devices like android tablets / handheld / wall-mounted devices can be used in our management systems for faster and reliable data accessibility. Different departments can be integrated together to work on centralized database and thus keep the user data updated and provide real time results.

The above said project discuss few improvements that can be made in our existing college management system and thus create an easy and faster way of data updating and deliverance whenever required. Use of advanced mobile devices and applications can be made to provide real time results to users associated with a college and other such institutions.

Further, it also provides us with a view of its future scope and capabilities that can be later used to generate a responsive design of management systems in educational and other similar institutions in our country.

9. FUTURE SCOPE

The given project holds scope for future improvements and modifications. With the advances in technology new features and functionalities can be added. Since the project is developed in a modular way it provides an extensible and flexible development design.

Following is an overview of future improvements in the project –

- Capability of working with multiple distributed databases.
- Capability of working with different types of existing web services ex- php, etc.
- Applications for Symbian, IOs Devices
- Modified applications installed in dedicated devices for special functionalities. Such as special devices in Libraries can be used for automatic book issuing.
- Multiple language support for both android apps and online website.
- Devices installed with features like speakers, which can be used for real time announcements throughout the campus.

10. REFERENCES

BOOKS

Andreas Hutfless, Gopal Indurkhy, Joaquin Picon, Martin Weiss, Regis Coqueret; Design and Implement Servlets, JSPs, and EJBs for IBM Web Sphere Application Server, An IBM RedBook Publication, August 2000.

Charles P Brown , Ueli Wahli, Steven Calello, Rafael Coutinho, Patrick Gan, Cedric Hurst, Maan Mehta; Experience Java EE! Using Web Sphere Application Server Community Edition 2.1, An IBM RedBook Publication, February 2009

Whei-Jen Chen, John Chun, Naomi Ngan, Rakesh Ranjan, Manoj K. Sardana; DB2 Express-C: The Developer Handbook for XML, PHP, C/C++, Java, and .NET, An IBM RedBook Publication, August 2006

ONLINE ARTICLES & TUTORIALS

- IBM Tutorials –
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v8/topic/com.ibm.db2.udb.doc/core/c0008278.htm>
 - <http://www-01.ibm.com/software/webservers/appserv/was/>
 - http://www.ibm.com/developerworks/websphere/tutorials/0703_george/
 - http://www.ibm.com/developerworks/data/library/techarticle/0203zikopoulos/0203_zikopoulos.html
- <http://stackoverflow.com/questions/7970916/connecting-eclipse-websphere-and-db2>
- www.tutorialspoint.com/jsp/
- www.apl.jhu.edu/~hall/java/Servlet-Tutorial/