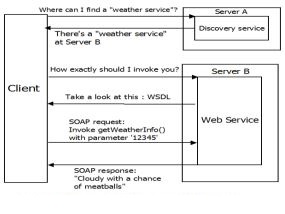
**CHAPTER 1**

**INTRODUCTION**

* 1. **BACKGROUND**

The concept of ***Web Services*** is not new and has been around for many years now. For a person who is unaware, a better way to understand it would be to consider the example of a user who might be interested in locating a public web service that gives the weather forecast in cities. Upon invocation, the web service will respond by giving information about which services are provided by which servers. Hence, now the user knows the location of the web service but doesn’t know how to invoke it. For this purpose, the web service needs to describe itself (tell us how the user should invoke it).The basic idea of a web service invocation involves sending of messages between a client and a server. For example, SOAP (Simple Object Access Protocol) specifies the format in which the requests are sent to the server and how the server should format the responses.

****

**Fig. 1.1 Web Service Discovery and Invocation**

Now-a-days one of the fastest growing industries is mobile industry. The developers from this industry are doing research and development on new platforms & user experience and improving the technology. One such technology is ***Android*** from Google which is supported for most of the present generation phones

**1.2 PROBLEM DEFINITION**

We have seen over the years that the process of manual attendance has been carried out across almost all educational institutions. Professors take the attendance in registers and calculate the percentage manually. The process is not only time consuming but also sometimes inefficient resulting in the false marking of attendance. In some colleges there are web based attendance management systems where professors take the attendance from their college web portal through their laptops.

Today, we need not maintain pen and paper based attendance registers. Following this thought, we have proposed an attendance monitoring system based on the concept of web services which is implemented as an Android mobile application that communicates with the database residing on a remote server. The mobile application would require connecting to the database using either General Packet Radio Service (GPRS) or Wi-Fi technology.

Professor uses his/her Android mobile phone to update the attendance. This application has a feature of intimating student parents with a message when their ward fails to maintain minimum attendance less than 75%.When the professors updates the attendance an immediate message will be sent to the mobile of parents with current percentage and the subject name.

**1.3 EXISTING SYSTEM**

In the existing system, there are mainly two types, One is manual attendance where the professor record attendance in attendance registers and the other is web based attendance monitoring system where the professors updates the attendance through laptop to their database through college web portal.

* 1. **DRAWBACKS OF EXISTING SYSTEM**
* In manual attendance system, professor needs to keep the record safely.
* The process is time consuming.
* Has a chance in the false marking of attendance.
* Professors need to calculate the percentage of each student.
* Professors need to create list of students with less attendance.
* No intimation to parents when their ward fails to maintain less attendance unless professors intimate them rarely.
* In web based attendance system, professors need to carry laptop for sake of taking attendance.
  1. **PROPOSED SYSTEM**
* Android application based attendance monitoring system can be developed in which professors takes attendance from his/her Android mobile phones.
* Application will be connected to different web services developed for different modules (login, subject list, student list etc).
* Each service is connected to database.
* Calculates attendance percentage of each student automatically and displays immediately.
* Message intimation to parents of student if they fail to maintain minimum attendance of less than 75%.
  1. **ADVANTAGES OVER EXISTING SYSTEM**
* Time saving.
* When compared to web portals, applications are more easy to use. So, taking attendance through this application will be easy instead of using laptops.
* No need to carry laptop for the sake of taking attendance as now-a- days where everyone are using smart phones. Therefore, when there is a simpler way to take attendance obviously the professors choose this application as 1st priority.
* Calculates attendance percentage of each student automatically, each time the professor upload attendance.
* Intimates parents when their ward fails to maintain minimum attendance with less than 75%.
* This application will be better to use which automatically do many things which should be done manually in the present existing system.
  1. **AIM(s) and OBJECTIVE(s)**

Our project aims to build this information system in a Mobile phone using the latest technology in the market namely Google Android SDK .This app is used to Upload attendance and to intimate students having minimum attendance.

* 1. **ORGANAISATION OF THE REPORT**

This section gives a brief representation of each chapter in this report

*Chapter 1- INTRODUCTION*

This chapter gives a brief introduction of this project and background of android, web services and it contains problem definition, aim & objectives of project, existing system, proposed system, advantages of proposed system, and disadvantages of existing system.

*Chapter 2–OVERVIEW OF THE PROPOSED SYSTEM /LITERATURE REVIEW*

This chapter gives a brief description of the whole proposed software system developed, system preliminary design, system planning and details of hardware and software requirements. A thorough review of literature with respect to the chosen field is projected.

*Chapter 3- DESIGN OF THE SYSTEM*

This chapter gives the base architecture of the system and it gives the architecture of different modules in the system.

*Chapter 4- IMPLEMENTATION OF SYSTEM/METHODLOGY*

This chapter should reflect development of the project such as: implementation, experimentation, optimization, evaluation etc and unit integration testing, Sample code, Screen shots of each modules etc.

*Chapter 5- CONCLUSION AND FUTURE ENHANCEMENTS*

This chapter summarizes the key aspects of the project (failures as well as successes) and states the conclusions. It also outlines about future work.

*Chapter 6- REFRENCES*

**2. Overview/Literature Review of Proposed System**

**2.1 OVERALL DESCRIPTION**

In the proposed system, the main concept is to develop an Android based attendance monitoring system, where the professor can upload the attendance through his/her android phone. Professor login with his username and password. He should select the respective subject and date and can get the students list. After marking the attendance and uploading, he will get the student attendance percentage list and the application will automatically send message to the parents of students with less attendance.

All the modules implemented in the applications requests a web service. Each request responses are parsed in the android and used as required.

**2.2 Software Requirement Specification**

**2.2.1 Software Requirements:**

* JAVA (JDK 1.6)
* Android SDK
* Eclipse IDE
* Operating System Windows 7 or 8

**2.2.2 Hardware Requirements:**

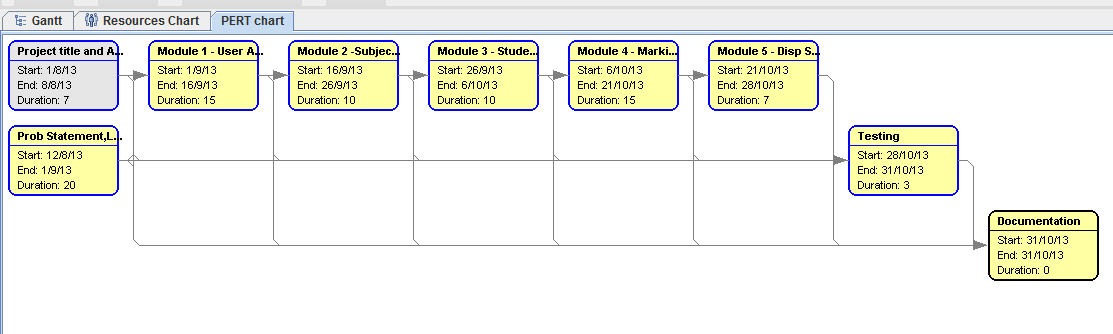
* RAM : 1GB and above
* Hard Disk : 20GB and above
* Processor : Dual Core

**2.3 MODULES**

We have identified 5 modules for this application.

1. User Authentication Module
2. Subject List Module
3. Student List Module
4. Marking Attendance Module
5. Display Student information and Message Sending Module

**2.4 SYSTEM PLANNING**

****

**Fig 2.4.1 PERT CHART OF SMART ATTENDANCE**

**2.5 LITERATURE REVIEW**

***2.5.1 Title:* Mobile Phone based Attendance System**

*Link:*<http://www.iosrjournals.org/iosr-jce/papers/Vol10-issue3/I01034850.pdf>

In this paper, the main aim is to develop an android application that helps the professor to take attendance through his/her mobile phone. This project will help the lecturers to reduce their workload by reducing the time and calculations required to update the attendance manually. Students and their parents will also view the attendance and curriculum details using the website.

This paper is similar to our project concept but in our project we are implement to some extra features like SMS or E-mail alert which will send an intimation to parents if their ward is absent for 2 days continuously. It will also intimate student with a warning message if their attendance percentage is less than 75%.

***2.5.2 Title:* Creating and Using SQLite Database for android applications**

*Link:* <http://www.sersc.org/journals/IJDTA/vol5_no2/8.pdf>

This paper presents the overview of the Android platform towards software development for mobile and non-mobile applications. Android platform includes the popular open source SQLite database which has been used with great success as on-disk file format that allows the developer to handle data in a simple way. It includes overview of android platform, android architecture and it has some information about creating a SQLite Database.

Our project requires a database as it has to store all the attendance details of students of each day. However not all the data is stored in the mobile database but just the attendance data.

**2.5.3 *Title:* A Mobile Application to Access Remote Database using Web Services**

*Link:* <http://www.fcrit.ac.in/ncnte2012/library/comp_papers/paper3.pdf?.rxn=19732280>

In this paper, an Android based mobile application for Attendance Monitoring is presented. The application offers reliability, time savings and easy control. It can be used as a base for creating similar applications for tracking attendance in offices or any workplace. It can be also integrated in healthcare sector to keep track of nurse to patient visits by streamlining the time entry, time approval and management processes*.*

**CHAPTER 3**

**DESIGN**

**3.1 SYSTEM ARCHITECTURE**

Our project consists of 5 modules

1. User Authentication

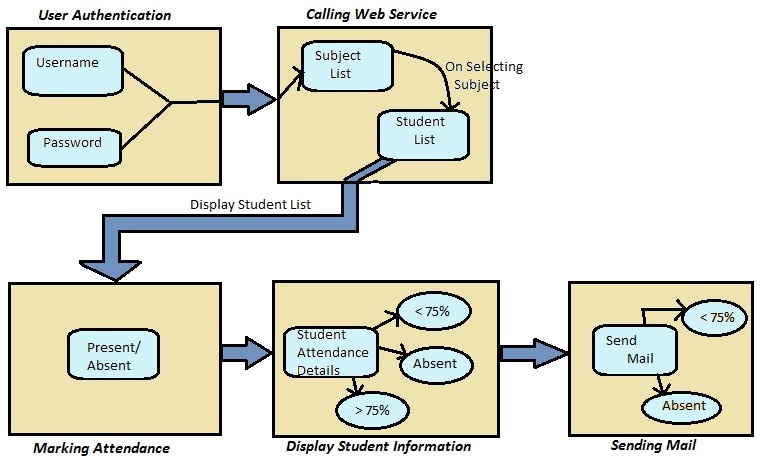
2. Subject List Module

3. Student List Module

4. Marking Attendance Module.

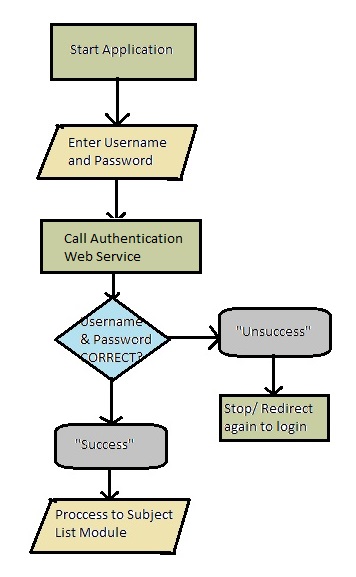
5. Display Student Information and Sending Mail Module.

**Fig. 3.1.1 SYSTEM ARCHITECTURE**

****

**3.1.1 User Authentication Module:**

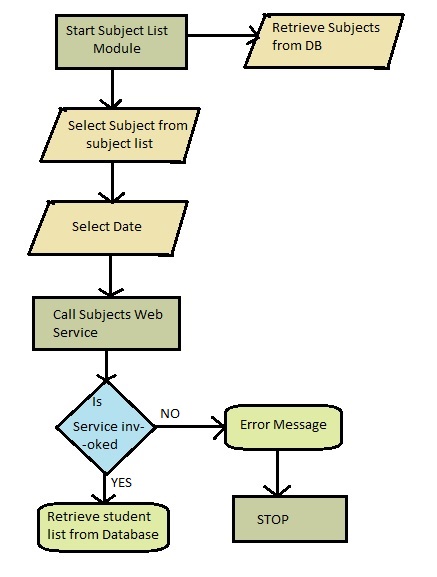
In User Authentication Module the user logins with his/her username and password. The username and password are requested as two parameters to the web service. The web service is connected with the database and checks whether the user is valid user or not. If the user is valid then the Subject list Activity will be displayed.



**Fig. 3.1.1.1 User Authentication Module**

**3.1.2 Subject List Module**

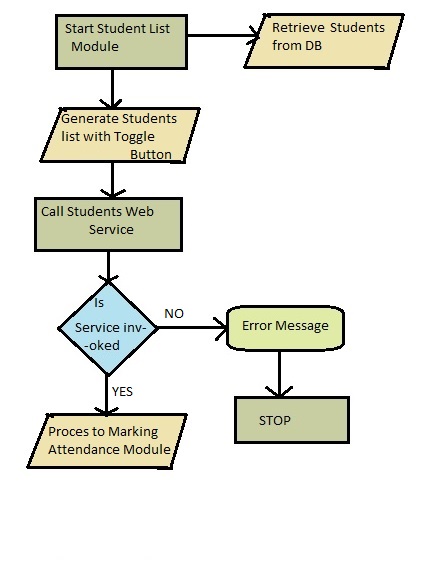
In this module, the user will be displayed with a spinner, which consists of respective subject list of the user. The subject list will be retrieved by requesting the database with the username. The user selects a subject and also selects the date and requests for student list.



**Fig. 3.1.2.1 Subject List Module**

**3.1.3 Student List Module**

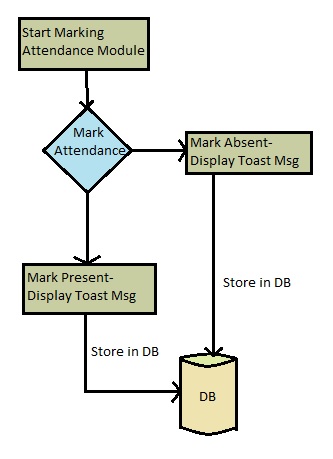
In the module, the students list will be retrieved from the database by using the login username and the subject selected. The student name are displayed along with a toggle button which value is default “present”.



**Fig 3.1.3.1 Student List Module**

**3.1.4 Marking Attendance Module**

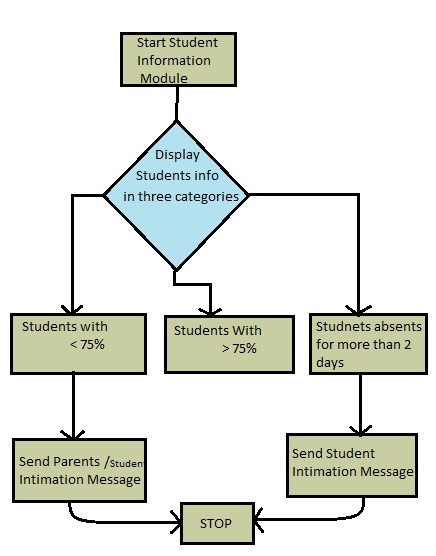
In this module, the professor marks the attendance. If a student is absent then the professor touches on the student name to mark it as absent.



**Fig 3.1.4.1 Marking Attendance Module**

**3.1.5 Display Student Information and Message Sending Module:**

In this module, when the professor submits the marked attendance, then the attendance is uploaded to the database. Each student attendance percentage is calculated and displays each student along with their attendance percentage. An Intimation message will be sent to the parent phone numbers of each student who are having attendance percentage less than 75%.



**Fig 3.1.5.1 Display Student Info and Message Sending Module**

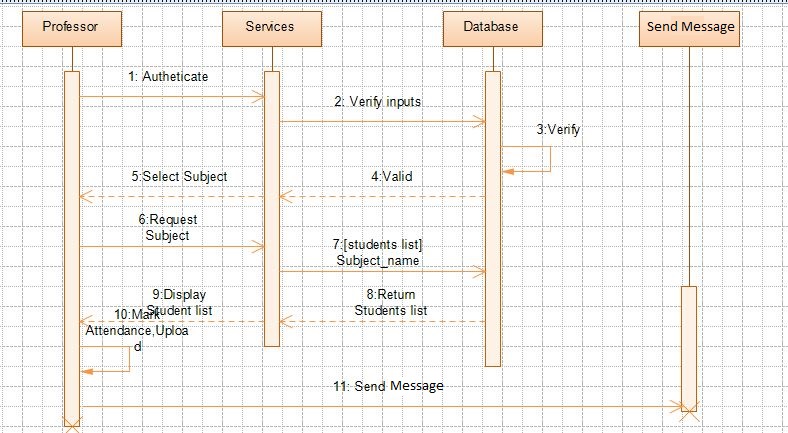
**3.2 UML DIAGRAMS**

The Unified Modelling Language (UML) is a standard language for writing software blue prints. The UML is a language for

* Visualizing
* Specifying
* Constructing
* Documenting the artifacts of a software intensive system.

1. It is mainly used in the analysis of applicaton.
2. It supports the entire Software Development life Cycle.
3. It supports diverse application areas.
4. Several CASE tools support it e.g. Rational,Together/j.
5. It is used to express the requirements of the software system we are developing.

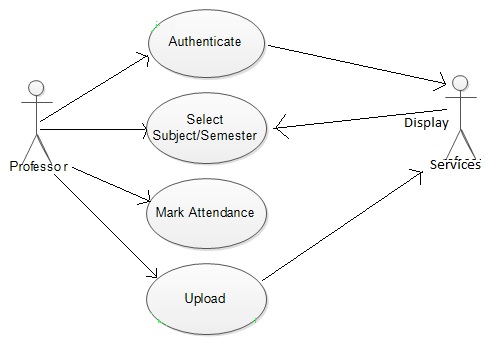
**3.2.1 Sequence Diagram**

****

**Fig 3.2.1.1 Sequence diagram for Smart Attendance**

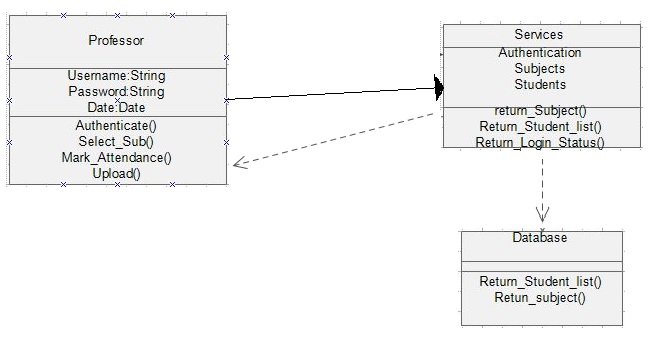
**3.2.2 Use Case Diagram**

A use case diagram is a graph of actors set of use cases enclosed by a system boundary, communication associations between the actors and users and generalization among use cases. The use case model defines the outside(actors) and inside(use case) of the system’s behaviour.

****

**Fig 3.2.2.1 Use Case Diagram for Smart Attendance**

**3.2.3 Class Diagram**

****

**Fig 3.2.3.1 Class Diagram for Smart Attendance**

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 Tools used in implementation**

The tools that we have used to implement the project are as follows

1. Eclipse IDE with Android Development Tools plug-in.
2. Microsoft Visual Studio 2010.

**4.1.1 Eclipse with ANDROID DEVELOPMENT TOOL Plug-In**

Android Development Tools (ADT) is a plug-in for the Eclipse IDE that is designed to give us a powerful, integrated environment in which to build Android applications.ADT extends the capabilities of Eclipse to let us quickly set up new Android projects, create an application UI, add packages based on the Android Framework API, debug our applications using the Android SDK tools, and even export signed (or unsigned) .apk files in order to distribute our application.

Developing in Eclipse with ADT is highly recommended and is the fastest way to get started. With the guided project setup it provides, as well as tools integration, custom XML editors, and debug output pane, ADT gives you an incredible boost in developing Android applications.

In our project, we used Eclipse IDE for creating the user interface for each module and also coding each module which each module request another program written in php.

## Creating an Android project

The ADT plug-in provides a New Project Wizard that we can use to quickly create a new Android project (or a project from existing code). To create a new project:

1. Select **File** > **New** > **Project**.
2. Select **Android** > **Android Project**, and click **Next**.
3. Select the contents for the project:

Enter a Project Name. This will be the name of the folder where our project is created.

Under Contents, select **Create new project in workspace**. Select your project workspace location.

Enter an Application name. This is the human-readable title for your application — the name that will appear on the Android device.

Enter a Package name. This is the package namespace (following the same rules as for packages in the Java programming language) where all your source code will reside.

Select Create Activity (optional, of course, but common) and enter a name for your main Activity class.

Enter a Min SDK Version. This is an integer that indicates the minimum API Level required to properly run your application. Entering this here automatically sets the min Sdk Version attribute in the [<uses-sdk>](http://developer.android.com/guide/topics/manifest/uses-sdk-element.html) of your Android Manifest file. If you're unsure of the appropriate API Level to use, copy the API Level listed for the Build Target you selected in the Target tab.

After all the coding is done we need to specify the permissions that our application needs to access from the phone like Internet access,wi-fi access, accessing phone contacts etc.

Running your application:

Before you can run your application on the Android Emulator, you **must** create an Android Virtual Device (AVD). An AVD is a configuration that specifies the Android platform to be used on the emulator.

Creating an AVD:

With ADT 0.9.3 and above, the Android SDK and AVD Manager provides a simple graphical interface for creating and managing AVDs.

To create an AVD with the AVD Manager:

Select **Window > Android SDK and AVD Manager**, or click the Android SDK and AVD Manager icon (a black device) in the Eclipse toolbar.

In the Virtual Devices panel, you'll see a list of existing AVDs. Click **New** to create a new AVD.

Fill in the details for the AVD.

Give it a name, a platform target, an SD card image (optional), and a skin (HVGA is default).

Click **Create AVD**.

Your AVD is now ready and you can close the AVD Manager. In the next section, you'll see how the AVD is used when launching your application on an emulator.

Running your application:

To run (or debug) your application, select **Run** > **Run** (or **Run** > **Debug**) from the Eclipse main menu. The ADT plug-in will automatically create a default launch configuration for the project.

**4.1.2 Microsoft Visual Studio 2010**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop  [console](http://en.wikipedia.org/wiki/Console_application) and [graphical user interface](http://en.wikipedia.org/wiki/Graphical_user_interface) [applications](http://en.wikipedia.org/wiki/Application_software) along with [Windows Forms](http://en.wikipedia.org/wiki/Windows_Forms) or [WPF](http://en.wikipedia.org/wiki/Windows_Presentation_Foundation) applications, [web sites](http://en.wikipedia.org/wiki/Web_site), [web applications](http://en.wikipedia.org/wiki/Web_application), and [web services](http://en.wikipedia.org/wiki/Web_service) in both [native code](http://en.wikipedia.org/wiki/Native_code) together with [managed code](http://en.wikipedia.org/wiki/Managed_code) for all platforms supported by [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [Windows Mobile](http://en.wikipedia.org/wiki/Windows_Mobile), [Windows CE](http://en.wikipedia.org/wiki/Windows_CE), [.NET Framework](http://en.wikipedia.org/wiki/.NET_Framework), [.NET Compact Framework](http://en.wikipedia.org/wiki/.NET_Compact_Framework) and [Microsoft Silverlight](http://en.wikipedia.org/wiki/Microsoft_Silverlight).

Visual Studio includes a [code editor](http://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](http://en.wikipedia.org/wiki/IntelliSense) as well as [code refactoring](http://en.wikipedia.org/wiki/Code_refactoring). The integrated [debugger](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building [GUI](http://en.wikipedia.org/wiki/GUI) applications, designer, class designer, and [database schema](http://en.wikipedia.org/wiki/Database_schema) designer.

However in our application we just use this IDE for developing web service for user authentication module. It connects with a My-SQL database which authenticates the users for using the application.

**4.2 Sample Code**

**4.2.1 Module 1 (User Authentication Module)**

**4.2.1.1 Java code in Android**

public class LoginActivity extends Activity {

private static final String NAMESPACE = "http://tempuri.org/";

private static final String URL = "http://praneethambati-001-site1.smarterasp.net/Service.asmx";

private static final String SOAP\_ACTION = "http://tempuri.org/faculty";

private static final String METHOD\_NAME = "faculty";

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

Button login = (Button) findViewById(R.id.btn\_login);

login.setOnClickListener(new View.OnClickListener() {

public void onClick(View arg0) {

loginAction();

}

});

}

private void loginAction(){

SoapObject request = new SoapObject(NAMESPACE,METHOD\_NAME);

EditText userName = (EditText) findViewById(R.id.tf\_userName);

String username = String.valueOf(userName.getText().toString());

EditText userPassword = (EditText) findViewById(R.id.tf\_password);

String password = String.valueOf(userPassword.getText().toString());

/\*

request.addProperty("username","pranu");

request.addProperty("password","pranu");

\*/

//Pass value for userName variable of the web service

PropertyInfo unameProp =new PropertyInfo();

unameProp.setName("username");//Define the variable name in the web service method

unameProp.setValue(username);//set value for userName variable

unameProp.setType(String.class);//Define the type of the variable

request.addProperty(unameProp);//Pass properties to the variable

//Pass value for Password variable of the web service

PropertyInfo passwordProp =new PropertyInfo();

passwordProp.setName("password");

passwordProp.setValue(password);

passwordProp.setType(String.class);

request.addProperty(passwordProp);

SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);

envelope.dotNet = true;

envelope.setOutputSoapObject(request);

HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);

try{

androidHttpTransport.setXmlVersionTag("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");

androidHttpTransport.call(SOAP\_ACTION,envelope);

SoapObject response=(SoapObject)envelope.bodyIn;

String result=response.getProperty(0).toString();

Log.i("info","Received :" + result);

//SoapPrimitive response = (SoapPrimitive)envelope.getResponse();

System.out.println("response:"+response.toString());

if(result.contentEquals("success"))

{

Intent it=new Intent(AndroidLoginExampleActivity.this,Subjects.class);

it.putExtra("user",username.toString());

startActivity(it);

Toast.makeText(getApplicationContext(),"Login Successful",Toast.LENGTH\_LONG).show();

}

else if(result.contentEquals("invalid"))

{

Toast.makeText(getApplicationContext(),"Invalid login details,Try again",Toast.LENGTH\_LONG).show();

}

System.out.println("Response:"+result.toString());

}

catch(Exception e){

}

}

}

**4.2.1.2 XML code for login USER INTERFACE**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:background="@drawable/images"

android:gravity="center"

android:orientation="vertical" >

<EditText

android:id="@+id/tf\_password"

android:layout\_width="189dp"

android:layout\_height="wrap\_content"

android:layout\_above="@+id/btn\_login"

android:layout\_alignParentRight="true"

android:layout\_marginBottom="21dp"

android:layout\_marginRight="17dp"

android:ems="10"

android:inputType="textPassword" />

<EditText

android:id="@+id/tf\_userName"

android:layout\_width="194dp"

android:layout\_height="wrap\_content"

android:layout\_above="@+id/tf\_password"

android:layout\_alignLeft="@+id/tf\_password"

android:layout\_marginBottom="24dp"

android:ems="10" />

<Button

android:id="@+id/btn\_login"

android:layout\_width="106dp"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="168dp"

android:text="Login" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/tf\_userName"

android:layout\_alignParentLeft="true"

android:text="Username :"

android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/tf\_password"

android:layout\_alignParentLeft="true"

android:text="Password :"

android:textAppearance="?android:attr/textAppearanceLarge" />

</RelativeLayout>

**4.2.1.3 Web Service for Login**

[WebMethod]

public string faculty(string username, string password)

{

try

{

string connString = "SERVER=MYSQL5002.Smarterasp.net" + ";" +

"DATABASE=db\_9ab3ca\_pranu;" +

"UID=9ab3ca\_pranu;" +

"PASSWORD=pintu848;" +

"PORT=3306;";

MySqlConnection cnMySQL = new MySqlConnection(connString);

MySqlCommand cmdMySQL = cnMySQL.CreateCommand();

MySqlDataReader reader;

cmdMySQL.CommandText = "select \* from faculty WHERE username='" + username + "' AND password='" + password + "'";

cnMySQL.Open();

reader = cmdMySQL.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(reader);

Object o = dt.Rows[0]["username"];

Object i = dt.Rows[0]["password"];

if (o == username && i == password)

{

return "success";

}

else

{

return "unsuccess";

}

cnMySQL.Close();

}

catch

{

return "invalid";

}

}

**4.2.2 Module 2 (Subject List Module)**

**4.2.2.1 Java code in Android**

public class Subjects extends Activity {

String tv1=null;

int day,year,month;

DatePicker dp;

Button btnChangeDate;

String fDate;

TextView dateview;

static final int DATE\_DIALOG\_ID=999;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.subjects);

setCurrentDateOnView();

addListenerOnButton();

Bundle bdl=getIntent().getExtras();

final String user=bdl.getString("user");

TextView user1= (TextView) findViewById(R.id.textView3);

user1.setText(user);

InputStream is = null;

String result = "";

//the year data to send

ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();

nameValuePairs.add(new BasicNameValuePair("username",user));

//http post

try{

HttpClient httpclient = new DefaultHttpClient();

HttpPost httppost = new HttpPost("http://praneethambati-001site1.smarterasp.net/subjects.php");

httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

HttpResponse response = httpclient.execute(httppost);

HttpEntity entity = response.getEntity();

is = entity.getContent();

}catch(Exception e){

Log.e("log\_tag", "Error in http connection "+e.toString());

}

//convert response to string

try{

BufferedReader reader = new BufferedReader(new InputStreamReader(is,"iso-8859-1"),8);

StringBuilder sb = new StringBuilder();

String line = null;

while ((line = reader.readLine()) != null) {

sb.append(line + "\n");

}

is.close();

result=sb.toString();

}catch(Exception e){

Log.e("log\_tag", "Error converting result "+e.toString());

}

//parse json data

ArrayList<String> arrProducts = new ArrayList<String>();

try{

JSONArray jArray = new JSONArray(result);

for(int i=0;i<jArray.length();i++){

JSONObject json\_data = jArray.getJSONObject(i);

String str =jArray.getJSONObject(i).getString("subject");

arrProducts.add(str);

Log.i("log\_tag","user: "+json\_data.getString("username")+

", sub: "+json\_data.getString("subject") );

}

}catch(JSONException e){

Log.e("log\_tag", "Error parsing data "+e.toString());

}

final Spinner spi=(Spinner) findViewById(R.id.spinner1);

ArrayAdapter<String> productAdapter = new ArrayAdapter<String>(

Subjects.this,android.R.layout.simple\_spinner\_item, arrProducts); productAdapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spi.setAdapter(productAdapter);

spi.setOnItemSelectedListener(new OnItemSelectedListener() {

public void onItemSelected(AdapterView<?> parentView, View selectedItemView,int position, long id) {

final String tv2=spi.getSelectedItem().toString();

System.out.println("value:"+tv2);

Toast.makeText(getApplicationContext(), "You selected: " + tv2, Toast.LENGTH\_LONG).show();

Button submit = (Button) findViewById(R.id.button1);

submit.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

Intent it= new Intent(Subjects.this,Students.class);

it.putExtra("subject", tv2);

it.putExtra("username", user);

it.putExtra("date",fDate);

startActivity(it);

}});

}

Button logout= (Button) findViewById(R.id.button2);

logout.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) { Intent it=new Intent(Subjects.this,AndroidLoginExampleActivity.class);

startActivity(it);

Toast.makeText(getApplicationContext(), "You have been Logged out successfully!", Toast.LENGTH\_LONG).show();

}});

}

public void setCurrentDateOnView() {

dateview = (TextView) findViewById(R.id.textView4);

dp = (DatePicker) findViewById(R.id.datePicker1);

final Calendar c = Calendar.getInstance();

year = c.get(Calendar.YEAR);

month = c.get(Calendar.MONTH);

day = c.get(Calendar.DAY\_OF\_MONTH);

//display date in text view// Month is 0 based, just add 1

dateview.setText(new StringBuilder().append(day).append("-").append(month + 1).append("-").append(year).append(" "));

// set current date into datepicker

dp.init(year, month+1, day, null);

}

public void addListenerOnButton() {

btnChangeDate = (Button) findViewById(R.id.btnChangeDate);

btnChangeDate.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

showDialog(DATE\_DIALOG\_ID);

}

});

}

@Override

protected Dialog onCreateDialog(int id) {

switch (id) {

case DATE\_DIALOG\_ID:

// set date picker as current date

return new DatePickerDialog(this, datePickerListener, year, month,day);

}

return null;

}

private DatePickerDialog.OnDateSetListener datePickerListener = new DatePickerDialog.OnDateSetListener() {

//when dialog box is closed, below method will be called.

public void onDateSet(DatePicker view, int selectedYear,

int selectedMonth, int selectedDay) {

year = selectedYear;

month = selectedMonth+1;

day = selectedDay;

dateview.setText(new StringBuilder()

// Month is 0 based, just add 1

.append(day).append("-").append(month ).append("-")

.append(year).append(" "));

// set selected date into datepicker also

dp.init(year, month, day, null);

fDate=day+"-"+month+"-"+year;

}

};

**4.2.2.2 XML code for User Interface**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/images"

android:gravity="center\_vertical"

android:orientation="vertical" >

<Button

android:id="@+id/button1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:text="Submit" />

<Spinner

android:id="@+id/spinner1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/textView1"

android:layout\_marginLeft="14dp"

android:layout\_toRightOf="@+id/textView1" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentTop="true"

android:layout\_marginLeft="14dp"

android:layout\_marginTop="107dp"

android:text="Subject:"

android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentTop="true"

android:text="Welcome , "

android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/spinner1"

android:layout\_alignParentTop="true"

android:text="Large Text"

android:textAppearance="?android:attr/textAppearanceLarge" />

<DatePicker

android:id="@+id/datePicker1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/spinner1"

android:layout\_below="@+id/spinner1"

android:layout\_marginTop="132dp"

android:visibility="invisible" />

<Button

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_alignParentTop="true"

android:background="@android:drawable/btn\_default\_small"

android:text="Logout"

android:visibility="visible" />

<Button

android:id="@+id/btnChangeDate"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_centerVertical="true"

android:text="Select Date" />

<TextView

android:id="@+id/textView5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBaseline="@+id/textView4"

android:layout\_alignBottom="@+id/textView4"

android:layout\_toLeftOf="@+id/spinner1"

android:text="Date :"

android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView

android:id="@+id/textView4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/datePicker1"

android:layout\_below="@+id/datePicker1"

android:layout\_marginTop="33dp"

android:text="Large Text"

android:textAppearance="?android:attr/textAppearanceLarge" />

</RelativeLayout>

**4.2.2.3** **Php script for retrieving subject list**

<?php

mysql\_connect("MYSQL5002.Smarterasp.net","9ab3ca\_pranu","pintu848");

mysql\_select\_db("db\_9ab3ca\_pranu");

$q=mysql\_query("SELECT \* FROM subjects WHERE username='".$\_REQUEST['username']."'");

while($e=mysql\_fetch\_assoc($q))

$output[]=$e;

print(json\_encode($output));

mysql\_close();

?>

**4.2.3 MODULE 3 (Student List Module)**

**4.2.3.1 Java Code for Android**

public class Students extends Activity {

//toggle status

String strStatus;

//toggle button

ToggleButton tgl;

//student names that will be retrieved from database will be stored in stu\_names array

String[] stu\_names = null;

String[] upload\_stu=null;

String[] upload\_sta=null;

boolean[] status;

TextView dateDisp;

ArrayList<String> limits = new ArrayList<String>();

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.students);

//getting values of selected subjects and faculty name from subjects page

Bundle bdl=getIntent().getExtras();

final String subject=bdl.getString("subject");

final String user=bdl.getString("username");

final String date=bdl.getString("date");

TextView user1= (TextView) findViewById(R.id.pranu);

user1.setText(subject);

dateDisp=(TextView) findViewById(R.id.textView1);

dateDisp.setText(date);

System.out.println("Received Date:"+date);

InputStream is = null;

String result = "";

//request the subject and username values to the php script

ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();

nameValuePairs.add(new BasicNameValuePair("subject",subject));

nameValuePairs.add(new BasicNameValuePair("username",user));

//http post

try{

HttpClient httpclient = new DefaultHttpClient();

HttpPost httppost = new HttpPost("http://praneethambati-001-site1.smarterasp.net/students.php");

httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

HttpResponse response = httpclient.execute(httppost);

HttpEntity entity = response.getEntity();

is = entity.getContent();

}catch(Exception e){

Log.e("log\_tag", "Error in http connection "+e.toString());

}//convert response to string

try{

BufferedReader reader = new BufferedReader(new InputStreamReader(is,"iso-8859-1"),8);

StringBuilder sb = new StringBuilder();

String line = null;

while ((line = reader.readLine()) != null) {

sb.append(line + "\n"); } is.close();

result=sb.toString();

}catch(Exception e){

Log.e("log\_tag", "Error converting result "+e.toString());

}

//parse json data

final ArrayList<String> arrProducts = new ArrayList<String>();

try{

final ListView stulist = (ListView) findViewById(R.id.ListView1);

// Each row in the list stores student name and its status

List<HashMap<String,String>> aList = new ArrayList<HashMap<String,String>>();

final JSONArray jArray = new JSONArray(result);

for(int i=0;i<jArray.length();i++){

JSONObject json\_data = jArray.getJSONObject(i);

String str =jArray.getJSONObject(i).getString("name");

arrProducts.add(str);

stu\_names= new String[arrProducts.size()];

stu\_names[i]=json\_data.getString("name"); System.out.println(i+json\_data.getString("name")+",,,"+strStatus);

Log.i("log\_tag","user: "+json\_data.getString("name")+ ", sub: "+json\_data.getString("subject")+", faculty: "+json\_data.getString("faculty")

);

HashMap<String, String> hm = new HashMap<String,String>();

hm.put("txt" , stu\_names[i])

aList.add(hm);

// Keys used in Hashmap

String[] from = {"txt"};

// Ids of views in listview\_layout

int[] to = { R.id.tv\_item};

// Instantiating an adapter to store each items

// R.layout.listview\_layout defines the layout of each item

SimpleAdapter adapter = new SimpleAdapter(getBaseContext(), aList, R.layout.st\_layout, from, to);

stulist.setAdapter(adapter);

}

OnItemClickListener itemClickListener = new OnItemClickListener() {

public void onItemClick(AdapterView<?> lv, View item, int position, long id) {

ListView lView = (ListView) lv;

SimpleAdapter adapter = (SimpleAdapter) lView.getAdapter();

HashMap<String,Object> hm = (HashMap) adapter.getItem(position);

/\*\* The clicked Item in the ListView \*/

RelativeLayout rLayout = (RelativeLayout) item;

/\*\* Getting the toggle button corresponding to the clicked item \*/

tgl = (ToggleButton) rLayout.getChildAt(1);

String strStatus = "";

if(tgl.isChecked()){

tgl.setChecked(false);

strStatus = "Absent";

}else{

tgl.setChecked(true);

strStatus = "Present";

// status[position]=true;

}

Toast.makeText(getBaseContext(), (String) hm.get("txt") + " : " + strStatus, Toast.LENGTH\_SHORT).show();

System.out.println("Status:"+hm.get("txt")+":"+strStatus+".."+status);

limits.add(hm.get("txt")+".."+strStatus);

System.out.println(limits);

}};

stulist.setOnItemClickListener(itemClickListener);

// JSONArray jA=new JSONArray(result);

}catch(JSONException e){ Log.e("log\_tag", "Error parsing data "+e.toString());

}

//Uploading Attendance to db

Button b=(Button) findViewById(R.id.button1);

b.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

Intent it=new Intent(Students.this,DispInfo.class);

it.putExtra("user", user);

it.putExtra("date", date);

it.putExtra("subject", subject);

it.putStringArrayListExtra("stu",limits);

startActivity(it);

}

});

**4.2.3.2 XML code for USER INTERFACE**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

<TextView

android:id="@+id/pranu"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="TextView" />

<Button

android:id="@+id/button1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_alignParentLeft="true"

android:text="Button" />

<ListView

android:id="@+id/ListView1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_above="@+id/button1"

android:layout\_alignParentRight="true"

android:layout\_below="@+id/pranu" >

</ListView>

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_alignParentTop="true"

android:text="Medium Text"

android:textAppearance="?android:attr/textAppearanceMedium" />

</RelativeLayout>

**4.2.3.3 PHP code for getting students list**

<?php

mysql\_connect("MYSQL5002.Smarterasp.net","9ab3ca\_pranu","pintu848");

mysql\_select\_db("db\_9ab3ca\_pranu");

$q=mysql\_query("SELECT \* FROM students WHERE faculty='".$\_REQUEST['username']."' and subject='".$\_REQUEST['subject']."'");

while($e=mysql\_fetch\_assoc($q))

$output[]=$e;

print(json\_encode($output));

mysql\_close();

?>

**4.2.4 MODULE 5(Display Student Information & Message Sending)**

**4.2.4.1 Java Code for Android**

public class DispInfo extends Activity {

String subject;

ArrayList<String> lessName = new ArrayList<String>();

ArrayList<String> lessPhone = new ArrayList<String>();

ArrayList<Integer> eachpercentage = new ArrayList<Integer>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.dispinfo);

ListView list = (ListView) findViewById(R.id.listView1);

ArrayList<HashMap<String, String>> mylist = new ArrayList<HashMap<String, String>>();

Bundle bdl=getIntent().getExtras();

subject=bdl.getString("subject");

final String user=bdl.getString("user");

final String date=bdl.getString("date");

final ArrayList<String> stu=bdl.getStringArrayList("stu");

StringBuilder s=new StringBuilder();

s.append("["+stu.get(0)+",");

for(int i=1;i<stu.size()-1;i++)

{

s.append(stu.get(i)+",");

}

s.append(stu.get(stu.size()-1)+"]");

InputStream is = null;

String result = "";

ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();

nameValuePairs.add(new BasicNameValuePair("username",user));

nameValuePairs.add(new BasicNameValuePair("subject", subject));

nameValuePairs.add(new BasicNameValuePair("date", date));

nameValuePairs.add(new BasicNameValuePair("stu",s.toString()));

try{

HttpClient httpclient = new DefaultHttpClient();

HttpPost httppost = new HttpPost("http://praneethambati-001-site1.smarterasp.net/upload\_atten9.php");

httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

HttpResponse response = httpclient.execute(httppost);

HttpEntity entity = response.getEntity();

is = entity.getContent();

}catch(Exception e){

Log.e("log\_tag", "Error in http connection "+e.toString());

}

//convert response to string

try{

BufferedReader reader = new BufferedReader(new InputStreamReader(is,"iso-8859-1"),8);

StringBuilder sb = new StringBuilder();

String line = null;

while ((line = reader.readLine()) != null) {

sb.append(line + "\n");}

is.close();

result=sb.toString();

}catch(Exception e){

Log.e("log\_tag", "Error converting result "+e.toString());

}

ArrayList<String> studentName = new ArrayList<String>();

ArrayList<Integer> percentages = new ArrayList<Integer>();

ArrayList<String> phones = new ArrayList<String>();

try{

JSONArray jArray = new JSONArray(result);

for(int i=0;i<jArray.length();i++){

JSONObject json\_data = jArray.getJSONObject(i){

//getting all student names in studentName

String student =jArray.getJSONObject(i).getString("student");

studentName.add(student);

//getting each student percentage in percentages

int percentage =jArray.getJSONObject(i).getInt("percen");

percentages.add(percentage);

//getting each student phone num in phones

String phone=jArray.getJSONObject(i).getString("phone");

phones.add(phone);

Log.i("log\_tag","user: "+json\_data.getString("username")+", sub: "+json\_data.getString("subject")+", day: "+json\_data.getString("day")+", month: "+json\_data.getString("month")+", year: "+json\_data.getString("year")+", student name"+json\_data.getString("student")+",percen:"+json\_data.getString("percen")+", phone no:"+json\_data.getString("phone"));

HashMap<String, String> map = new HashMap<String, String>();

map.put("student", studentName.get(i)); map.put("percentages", percentages.get(i)+"%");

mylist.add(map);

if(percentages.get(i)<=75)

{

String lName=studentName.get(i);

String lPhone=phones.get(i);

int eachatten1=percentages.get(i);

lessName.add(lName);

lessPhone.add(lPhone);

eachpercentage.add(eachatten1);

}

}

}catch(JSONException e){

Log.e("log\_tag", "Error parsing data "+e.toString());

}

SimpleAdapter mSchedule = new SimpleAdapter(this, mylist, R.layout.row,new String[] { "student", "percentages"}, new int[] { R.id.FROM\_CELL, R.id.TO\_CELL});

list.setAdapter(mSchedule);

sendSMS();

}

public void sendSMS() {

for(int i=0;i<lessName.size();i++)

{

String phoneNumber = lessPhone.get(i);

String StuName=lessName.get(i);

int StuPercen=eachpercentage.get(i);

String message = "This is to inform that your ward \""+StuName+"\"is having less attendance with "+ StuPercen+"% in "+subject+" subject .So,Kindly warn your ward to maintain minimum attendance.REGARDS: DCP PROJECT!.";

SmsManager smsManager = SmsManager.getDefault();

ArrayList<String> parts = smsManager.divideMessage(message);

smsManager.sendMultipartTextMessage(phoneNumber, null, parts, null, null);

}

}

}

**4.2.4.2 XML code for User Interface**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

<ListView

android:id="@+id/listView1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" >

</ListView>

</RelativeLayout>

**4.2.4.3 Php Code for Uploading Attendance and getting Student Percentage**

<?php

mysql\_connect("MYSQL5002.Smarterasp.net","9ab3ca\_pranu","pintu848");

mysql\_select\_db("db\_9ab3ca\_pranu");

$user=$\_REQUEST['username'];

$subject=$\_REQUEST['subject'];

$date1=$\_REQUEST['date'];

$stu=$\_REQUEST['stu'];

//Breaking date to day,month,year for feature use

$date2=explode("-",$date1);

$day=$date2[0];

$month=$date2[1];

$year=$date2[2];

//collect all student names

$tot=mysql\_query("SELECT name,phone FROM students where faculty='$user' and subject='$subject'");

while($row = mysql\_fetch\_array($tot))

{

$a=mysql\_num\_rows($tot);

//inserting all students with present as default

$instot=mysql\_query("insert into attendance values('$user','$subject','$row[0]','Present','$day','$month','$year','$row[1]')");

}

// Breaking the students from received from STU variable eg: [Naresh..absent,Ravali..Absent]

$stu1=explode(",",$stu);

$j=0;

$stu2=explode("[",$stu1[0]);

$stu3=explode("]",$stu1[sizeof($stu1)-1]);

for($i=0;$i<=sizeof($stu1)-2;$i++)

{

$id[$i]=$stu1[$i+1];

}

$id[sizeof($stu1)-2]=$stu2[1];

$id[sizeof($stu1)-1]=$stu3[0];

for($i=0;$i<sizeof($id);$i++)

{

$sp[$i]=explode("..",$id[$i]);

}

//checking each stu names and status trailssssss

for($i=0;$i<sizeof($id);$i++)

{

$stu\_name[$i]=$sp[$i][0];

$stu\_status[$i]=$sp[$i][1];

}

// update the students with absent which will replace the previous present value

for($i=0;$i<sizeof($id);$i++)

{

$jstupdate=mysql\_query("UPDATE attendance SET stu='$stu\_name[$i]',status='$stu\_status[$i]' WHERE stu='$stu\_name[$i]' and username='$user' and subject='$subject' and day='$day' and month='$month' and year='$year'");

}

$ab=array();

$pcou=array();

$totcou=array();

$per=array();

$per2=array();

$output;

$tot1=mysql\_query("SELECT \* from attendance where username='$user' and subject='$subject' and day='$day' and month='$month' and year='$year' ");

while($row1=mysql\_fetch\_array($tot1))

{

$ab[$row1[2]]=$row1[2];

}

$ab1=array\_values($ab);

for($i=0;$i<count($ab1);$i++)

{

$present=mysql\_query("SELECT stu FROM attendance where username='$user' and subject='$subject' and day<='$day' and month<='$month' and year<='$year' and status='Present' and stu='$ab1[$i]' ");

$pcou[$ab1[$i]]=mysql\_num\_rows($present);

$totclasses=mysql\_query("SELECT \* FROM attendance where username='$user' and subject='$subject' and stu='$ab1[$i]' and day<='$day' and month<='$month' and year<='$year' ");

$totcou[$ab1[$i]]=mysql\_num\_rows($totclasses);

}

$pcou1=array\_values($pcou);

$totcou1=array\_values($totcou);

for($i=0;$i<count($pcou1);$i++)

{

for($j=0;$j<count($totcou1);$j++)

{

$per[$i]=round((($pcou1[$i]/$totcou1[$j])\*100),2);

}

}

for($i=0;$i<count($ab1);$i++)

{

$fi=mysql\_query("SELECT \* FROM attendance where username='$user' and subject='$subject' and stu='$ab1[$i]' and day<='$day' and month<='$month' and year<='$year'");

$per1[$ab1[$i]]=$per[$i];

}

$abcd=mysql\_query("SELECT \* FROM attendance where username='$user' and subject='$subject' and day='$day' and month='$month' and year='$year'");

$r=mysql\_query("SELECT \* FROM percentage where username='$user' and subject='$subject' ");

if(mysql\_num\_rows($r)==0)

{

while($row=mysql\_fetch\_array($abcd))

{

mysql\_query("INSERT INTO percentage (username,subject,day,month,year,student,phone) values('$user','$subject','$day','$month','$year','$row[2]','$row[7]')");

}

}

$per2=array\_keys($per1);

for($k=0;$k<count($per);$k++)

{

mysql\_query("UPDATE percentage SET percen='$per[$k]',day='$day',month='$month',year='$year' where student='$per2[$k]' and subject='$subject'");

}

$q=mysql\_query("SELECT \* FROM percentage WHERE username='$user' and subject='$subject' and day='$day' and month='$month' and year='$year'");

while($g=mysql\_fetch\_assoc($q))

{

$output1[]=$g;

}

if(isset($output1))

{

print(json\_encode($output1));

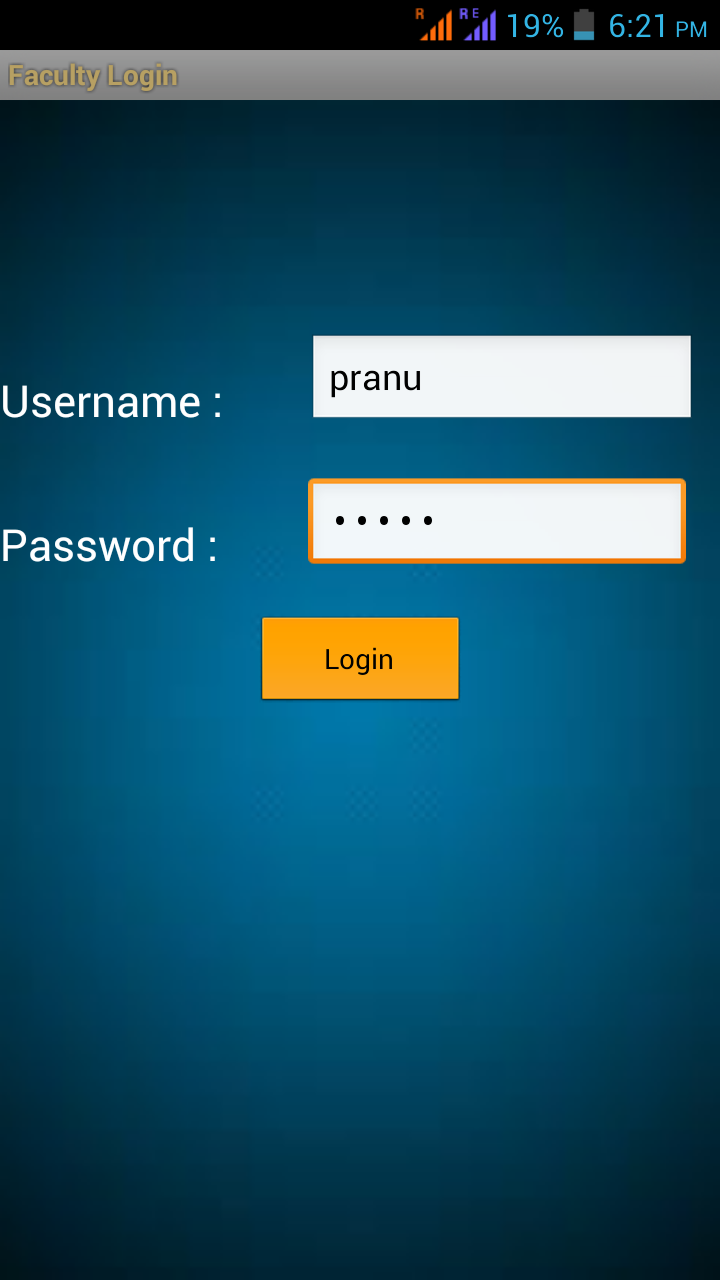
}

mysql\_close();

?>

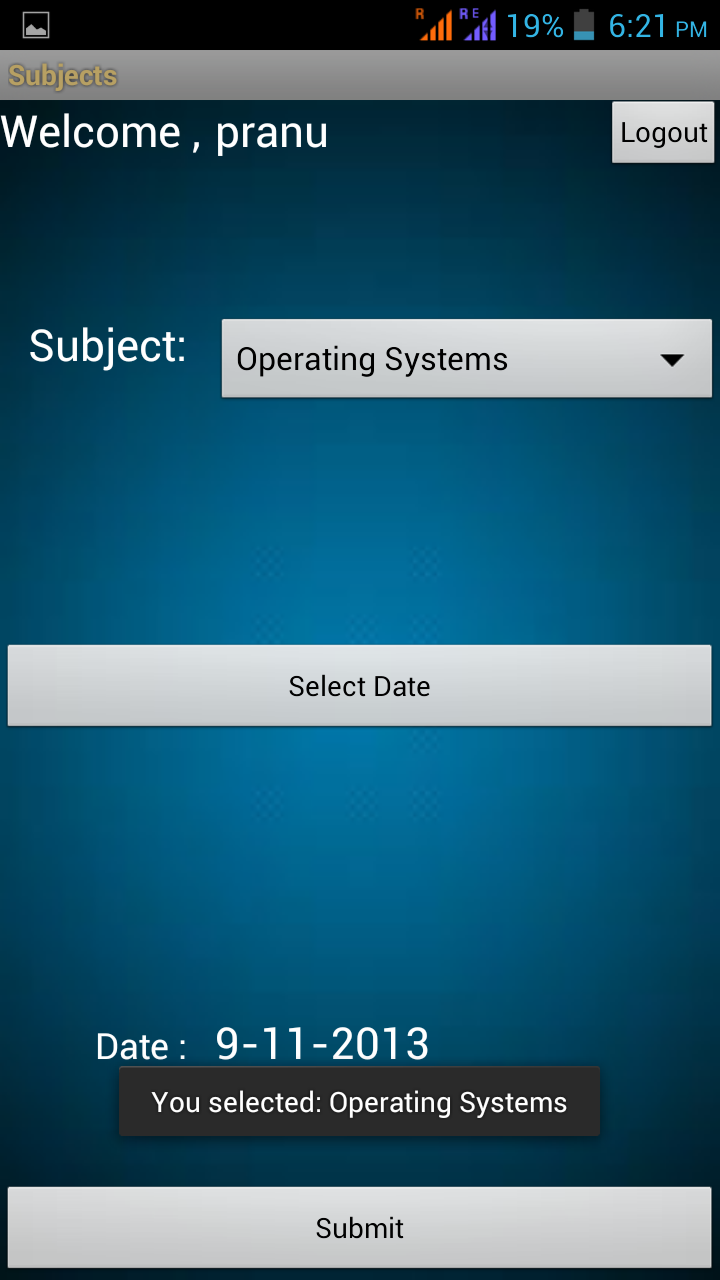
**4.3 OUTPUT AND SCREEN SHOTS**

**4.3.1 Module -1 (USER AUTHENTICATION MODULE)**

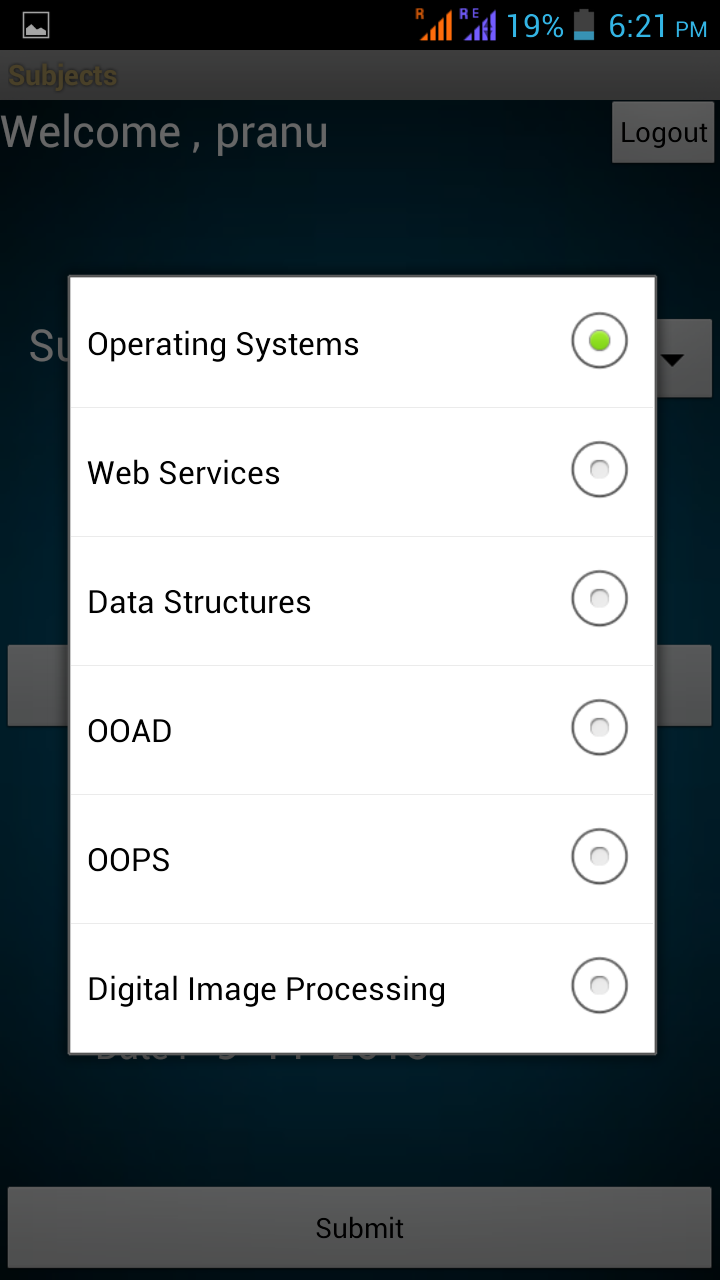
****

**Fig 4.3.1.1 Screen shot displaying user authentication**

**4.3.2 Module-2 (SUBJECT LIST MODULE)**

****

**Fig 4.3.2.1 Screen shot displaying after user Login (Selecting subject & Date)**



**Fig 4.3.2.2 Screen shot displaying Subject list in spinner list**

****

**Fig 4.3.2.3 Screen shot displaying date picker**

**4.3.3 Module-3 (Student List Module)**

****

**Fig 4.3.3.1 Screen shot displaying Student list with default PRESENT**

**4.3.4 Module-4 (Marking Attendance Module)**

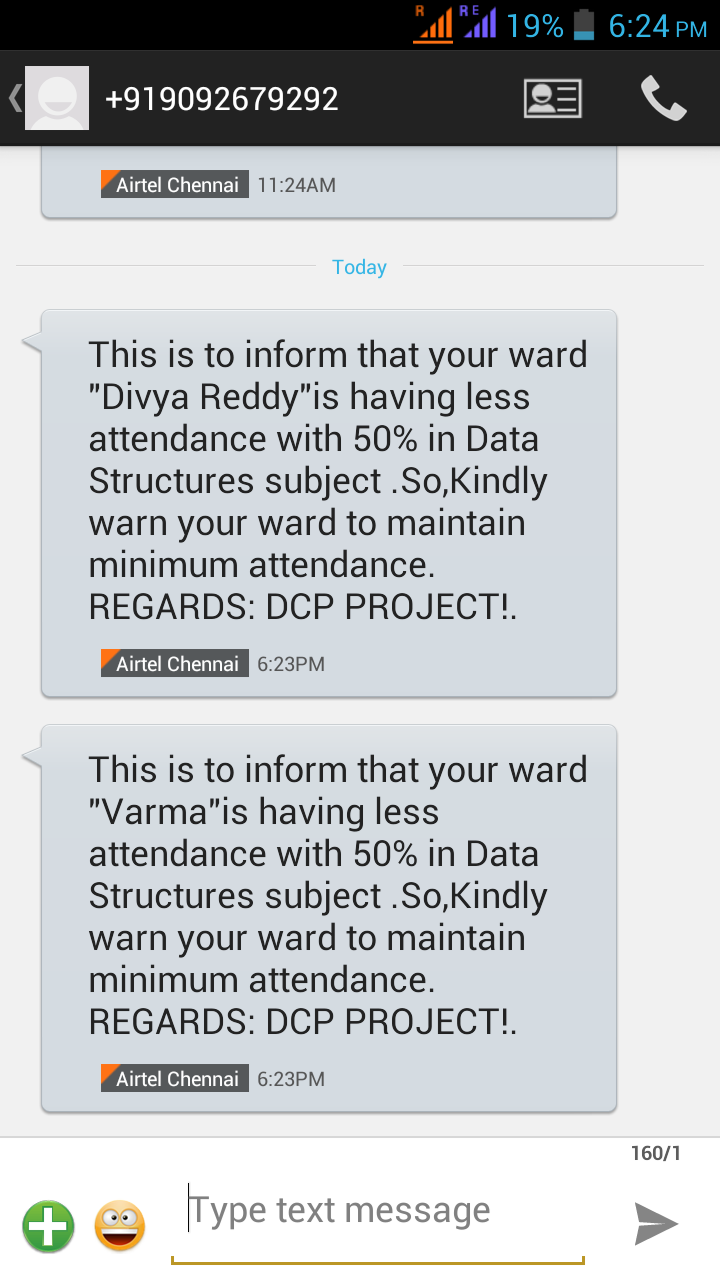
****

**Fig 4.3.4.1 Screen shot displaying students list after marking attendance**

**4.3.5 Module-5 (Display Student Information & Message Sending)**

****

**Fig 4.3.5.1 Screen shot Displaying each Student Information**

****

**Fig 4.3.5.2 Screen shot displaying the Intimation message sent to Parent Mobile if students attendance is less than 75%**

**4.4 TESTING**

**4.4.1 TESTING OVERVIEW:**

* Testing is a process of executing a program with intent of finding an error.
* Testing presents an interesting anomaly for the software engineering.
* The goal of the software testing is to convince system developer and the customers that the software is good enough for operational use. Testing is a process intended to build confidence in the software.
* Testing is a set of activities that can be planned in advance and conducted systematically.
* Software testing is often referred to as verification & validation.

**4.4.2 DIFFERENT TYPES OF TESTING**

**Types of testing:**

* Unit Testing
* Integration Testing
* Functional Testing
* System Testing

**4.4.2.1 Functional Test:**

Functional tests provides systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation and user manuals.

Functional testing is centred on the following items:

Valid Input : Identified classes of valid input must be accepted.

Invalid Input : Identified classes of invalid input must be rejected.

Functions : Identified functions must be exercised.

Output : Identified classes of application outputs must be exercised

Systems/Procedures: Interfacing systems or procedures must me invoked.

Organizations and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify business process lows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**4.4.2.2 System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flow, emphasizing pre-driven process links and integration points. System testing falls within the scope of black box testing and as such, should require no knowledge of the inner design of the code or logic.

**4.4.2.3 White Box Testing**

White-box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit.

**4.4.2.4 Black Box Testing**

Black box testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a defective source document, such as specification or requirements document. It is a testing in which the software under testing is treated as a black box. We cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works. It’s also called as behavioural tests as the focus is on functional requirements of the software.

**4.4.2.5 Acceptance Testing**

Acceptance testing is a test conducted to determine if the requirements of specification or contract are met. Software developers often distinguish acceptance testing by the system provider from acceptance testing by the customer (the user or client) prior to accepting transfer of ownership. In the case of software, acceptance testing performed by the customer is known as user acceptance testing.

**4.4.3 TEST CASES**

**4.4.3.1 MODULE-1 (User Authentication Module)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Correct Username & Password | Valid user/Process to retrieve subject list and display next module | Msg: Login Successful/ Processed to next module with subject list | **Pass** |
| **2** | Correct Username & Wrong Password | Invalid User/Show toast msg as Invalid user,try again | Msg: Shows toast msg as invlaid user, try again | **Pass** |
| **3** | Wrong Username & Correct Password | Invalid User/Show toast msg as Invalid user,try again | Msg: Shows toast msg as invlaid user, try again | **Pass** |
| **4** | Wrong Username & Wrong Password | Invalid User/Show toast msg as Invalid user,try again | Msg: Shows toast msg as invlaid user, try again | **Pass** |

**Table 4.4.3.1 Test Cases for module 1**

**4.4.3.2 MODULE-2 (Subject List Module)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| **1** | Requesting subjects list with login username | Should generate subject list in the spinner list when login is successful | Generating correct subject list according to login username | **Pass** |
| **2** | Request subject list with wrong login username | Not Possible | Result is NULL | **Pass** |
| **3** | Select Date | Display selected Date in TextView | Displaying selected date in  TextView | **Pass** |
| **4** | Date Not Selected | Do Not display anything in date TextView | Displays Current Date in textview but does not store | **Fail** |

**Table 4.4.3.2 Test Cases for Module-2**

**4.4.3.3 MODULE-3 (Student List Module)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| **1** | Requesting students list with login username,selected subject | Should generate student list in the list view | Generating correct student list according to login username and subject | **Pass** |
| **2** | Request student list with wrong login username,wrong subject | Not Possible | Result is NULL | **Pass** |

**Table 4.4.3.3 Test Cases Module 3**

**4.4.3.4 MODULE-4 (Marking Attendance)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| **1** | Marking Attendance as Absent | Displays Toast with Student name and status | Displaying Toast with student name and status | **Pass** |
| **2** | Marking Attendance as Present | Displays Toast with Student name and status | Displaying Toast with student name and status | **Pass** |

**Table 4.4.3.4 Test Cases for Module-4**

**4.4.3.5 MODULE-5 (Display student Info & Message sending)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| **1** | On Uploading attendance | Display Each student with attendance percentage | Displaying Student names with attendance Percentages | **Pass** |
| **2** | Sending Message if less than 75% attendance | Sends message to parents mobile with name, subject,  percentage | Sending message to parents numbers with name, subject,  percentage | **Pass** |
| **3** | Sending message if greater than 75% attendance | Does not send any message | Not sending any message | **Pass** |

**Table 4.4.3.5 Test Cases for Module-5**

**CHAPTER 5**

**CONCLUSION AND FUTURE ENHANCEMENTS**

**5.1 CONCLUSION**

In this Project, we have proposed an android application based attendance monitoring system where the professor can take the attendance from his/her smart phone. We have followed the real time approach for developing this application. In the real world android application development, the developers basically create web services and use those services from the application. We also followed the same procedure while implementing all modules. We can develop this type of applications for our university as most of the faculty are using smart phones and it’s easy to use, it has an extra feature of intimating parents with a message etc.

**5.2 FUTURE ENHANCEMENTS**

In future, we have a concept to develop this application without GPRS or Wi-Fi connection. By using Shared Preferences in android we can store the details of each user details locally in the mobile database and can use it even without internet connection. For this, however , we need internet connection for the first time users to login and store their details in the phone database.

**CHAPTER 6**

**REFERENCES**

[1] Grant Allen, **Beginning Android 4,** 1st Edition, Apress, 2011

[2] Sunguk Lee, “Creating and using Database for Android application”, International Journal of Database Theory and Application, Vol: 5, No. 2, Year: June, 2012.

[3] **Karan Balkar,Reyomi Roy,Preeyank Pable,M. Kiruthika and Shweta Tripathi**, “A Mobile Application to Access Remote Database using Web Services”,Delhi,2010.

[4] Shraddha S. Chawhan, Mangesh P. Girhale, Gunjan Mankar,“

Mobile Phone Based Attendance System”, IOSR Journal of Computer Engineering, Volume 10, Issue 3, and Page: PP 48-50, Year: Mar. - Apr. 2013.

# [5] [I-Ching Hsu](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.I-Ching%20Hsu.QT.&searchWithin=p_Author_Ids:37317887800&newsearch=true), “Mobile ubiquitous attendance monitoring system using wireless sensor networks”, IEEE Xplore, Volume:4 , V4-533 - V4-537, Shanghai, 2010.

[6] <http://www.codeproject.com/Articles/463338/Sending-a-SMS-Message-from-an-Android-Application> (10th October 2013).

[7] <http://m-zeeshanarif.blogspot.in/2013/05/android-connecting-to-server-mysql.html> (20th August 2013).

[8] <http://msdn.microsoft.com/en-us/library/8wbhsy70(v=vs.90).aspx> (20th August 2013)

[9] <http://www.c-sharpcorner.com/uploadfile/88b6e5/how-to-call-web-service-in-android-using-soap/> (25th august 2013)