CHAPTER 1

INTRODUCTION

1.1 Problem Definition

Our project aims to implement an android app that will help in making department paperless by automizing the department level manual tasks such as taking attendance and then recording it, calculating marks of regular assessment, maintaining profile and circulating notices for events.

1.2 Relevant Theory

DIS (Department Information System) is the unique system which will overcome the short coming of all existing system as described below. This session consists of details about existing similar systems along with their shortcomings.

- **Existing system**

1) E – Beat

E - Beat is an electronic beat constable's night patrolling attendance system to avoid manual recording of time &attendance. Shortcoming of E – Beat is very small memory storage as small as 64kb.

2) Mobile Phone Based Attendance System in J2ME

Professor enters the attendance to the mobile using keypad. It is a manual process. After saving the absentees or presenters in to the mobile, teacher can edit the attendance list in the mobile. Shortcomings of Mobile Phone Based Attendance System in J2ME are

It does not support the GPS technology to reduce the fake attendance. Regarding the above system and taking in to consideration the existing system and its drawbacks, we proposed the “DIS System on android”. In this proposed system, intention is to develop an android application which will eliminate all the drawbacks of the existing system, which consist of the following dynamic features:

- Time saving as attendance and all other work can be done just by checking the list of students and making it automated way.
- No needs to maintain several separate records and manual calculations.
- 24x7 availability of information.

- More secured than traditional attendance system.
- A step towards the futuristic e-schools

1.3 Literature Survey

It is carried out to find information on Android OS, Bluetooth case studies etc. The details are stated below:

1. Got information about Lecturers will login to the phone application and get connected to the server. After login, they will take attendance using mobile phone by the use of Bluetooth. After taking the attendance in the mobile, lecturer will send it over to the server using GPRS and JSON parsing algorithm technique[1].
2. Studied about what is Android, structure of android, Features of android, comparison between different mobile operating system. From this book we got the information about the layered architecture of android. This is required for Bluetooth implementation in Android[2].
3. Got information that programming in android is done in core java, forwhich we need framework called eclipse, downloading and Installing Eclipse,Downloading and Installing the Android SDK, Exploring the Android SDK ,& Using the Cell Phone's GPS Functionality[3].
4. Got information about Bluetooth connectivity, Bluetooth API and about mac id connectivity with mobile phones. The Bluetooth API documentation states that an application needs at least the BLUETOOTH permission to use the Bluetooth device[4].
5. Studied that for converting data formats we use JSON Parsing algorithm. Got information about working of JSON parsing algorithm and accessing remote database in android[5].

1.3.1 Android OS

Android is a Linux-based operating system designed primarily for touch screen mobile devices such as smart phones and tablet computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005 Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software, and

telecommunication companies devoted to advancing open standards for mobile devices. The first Android-powered phone was sold in October 2008.

Android is open source and Google releases the code under the Apache License. This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Additionally, Android has a large community of developers writing applications ("apps") that extend the functionality of devices, written primarily in a customized version of the Java programming language.

1.3.1.1 Advantages of Android

1. Android is open, because it is Linux based open source so it can be developed by anyone.
2. Easy access to the Android App Market: Android owners are people who love to learn the phone; with Google's Android App Market you can download applications for free.
3. Populist Operating System: Android Phones, different from the IOS is limited to the IPhone from Apple, then Android has many manufacturers, with their respective flagship gadget from HTC to Samsung.
4. USB full facilities. You can replace the battery, mass storage, Disk Drive, and USB tethering.
5. Easy in terms of notification: the operating system is able to inform you of a new SMS, Email, or even the latest articles from an RSS Reader.
6. Supports all Google services: Android operating system supports all of Google services ranging from Gmail to Google reader. All Google services can you have with one operating system, namely Android.
7. Install ROM modification: There are many customs ROM that you can use on Android phones, and the guarantee will not harm your device.

1.3.1.2 Disadvantages of Android

1. Connected to the Internet: Android can be said is in need of an active internet connection. At least there should be a GPRS internet connection in your area, so that the device is ready to go online to suit our needs.

2. Sometimes slow device company issued an official version of Android your own.
3. Android Market is less control of the manager, sometimes there are malware.

1.3.2 Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, Windows XP or later; for the moment one can develop Android software on Android itself by using [AIDE - Android IDE - Java, C++] app and [Android java editor] app. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plug-in, though Intelli J ,IDEA IDE (all editions) fully supports Android development out of the box, and Net Beans IDE also supports Android development via a plug-in. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely). Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .APK format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvikexecutable), resource files, etc [5].

- Advantage:

1. Simple to design application.
2. Provide demo of application.

- Disadvantage:

1. Time consuming.

1.4 Scope

The proposed system is the android application with central database. The proposed system will help lecturers to take the attendance automatically using Smart-phones and send to database directly using this application, can keep records of each and every students separately with each and every details of them. Now moving on teachers and Head of department, can send the notifications by using this application to students & teacher. In this project we will try to cover the attendance system for teachers and students, notification application for faculties. Student profile and teacher's profile too.

- In future we can add each and every module which includes in ERP system.
- We can have web base interface for the same application

1.5 Objective

The basic objective behind this application is to have completely automated and paperless department.

The main objectives of the systems are:

- The objective of the DIS app on android is to help organization to automating the whole manual processing of the existing system.
- The system should support multi-user environment.
- System should be fully automated.
- Various outputs should be available at the server side.
- There should be different login for teacher, students and Higher Authorities.
- The system should be capable of doing each and every task that come under department such as attendance record, student profiles, assignment assessment records, class test records, term work records, notification records etc.

CHAPTER 2

REQUIREMENT ANALYSIS AND SYSTEM DESIGN

Requirement analysis is an important part of software requirement life cycle. It forms the base for system design and development. Requirement analysis is the software-engineering task that bridges the gap between the software allocation and software design. It is the process of discovery, refinement, modeling and specification. Requirements establish an understanding of the user's needs, and also provide the final yardstick against which implementation success is measured. In requirement analysis phase we deal with the requirements of company related to project features, platform used for development, language etc.

Here we are exploring details about requirement specification, validation of those requirements and finally system requirements.

2.1 Requirement Specification

Requirement analysis is the detailed study of various operations performed by the system and their relationships within and outside system. It defines the boundaries of the system and decides whether it should relate to other system or not.

Complete understanding of software requirements is essential for the success of software development effort. No matter how well designed and well coded, the program is, poorly analyzed and specified program will disappoint the user and bring woe to the developer. The requirement analysis task is the process of discovery, refinement, modeling and specification. The software scope is initially established by the system designer and refined during software project. Models of required data, information and control flow and operational behavior are created.

Alternative solutions are analyzed and allocated to various software elements. Requirement analysis is thus a software engineering task that bridges gap between system level software allocation and software design. The requirements gathering process is intensified and focused specifically on software. Requirements for both the system and the software are documented and reviewed with the customer.

2.1.1 Normal Requirements

These are the requirements which are explicitly specified by the customer. Hence these requirements should be satisfied in order to validate the end product from the customer.

The normal requirements from the system are:

N1: Application should be easy to handle.

N2: Application should make record entry very fastly.

N3: Any updating in record should be made easily and correctly.

N4: Every registered member should get his own record.

N5: Students, Teachers and parents should be notified about every event.

2.1.2 Expected Requirements

These requirements are implicit type requirements. These are not stated by the customer but are expected to be satisfied by the end system.

The expected requirements from the system are:

E1: It should match stored macid with present student's mac id of mobile.

E2: Secure the database.

E3: Student should get notification as soon as teacher posts it.

E4: Detailed information of student and teacher should be displayed on his profile including all the achievements up to the present time.

2.1.3 Excited Requirements

These requirements are neither stated by customer nor expected. But to achieve total customer satisfaction the developer may include certain requirements which enhance the functionality of the product certain additions may enhance the functionality of the system for the customer.

X1: After an attendance app should give proper count.

X2: Remarks about attendance should be maintained and notice should be given to Students in case of attendance.

2.2 Non Functional Requirements

It describes non behavioral aspects of a system capturing the properties and constraints under which a system must operate. Requirements which are not specifically concerned with functionality of system. They place restriction on the product being developed and development process.

NF1: Graphical User Interface.

NF2: Laptop or Android mobile with Bluetooth.

2.3 Validation of Requirements

A validation criterion is probably the most important and ironically, the mostoften neglected section of software requirement specification. How do we recognize a successful implementation? What classes of tests must be conducted to validate the functions, performance and constraints? We neglect this section because completing its demands through understanding of software requirements, something that we often do not have at this stage. Yet, the specification of validation criteria acts as an implicit review of all the requirements. It is essential that time and attention to be given to this section.

The Table 2.1 below shows requirement number and the means to validate that requirement.

Table 2.1 Validation of Requirements

SR.NO	Validation of Requirements
N1	App will be easily handled by providing Tags.
N2	App will not get slower down while recording entries in database.
N3	Updating will be made on specified parameters only.
N4	Authentication will be provided.
N5	Notifications about events will be provided by sending mail or sms to user.
E1	App will match present student's mac id with stored mac id's by executing SQL queries for attendance.
E2	Database will be secured by unique ids and password for the user.
E3	Student will be notified about all the events posted by teacher.
E4	Detailed info of student and teacher will be displayed on his or her profiles including achievements.
X1	After an attendance app will give proper count.
X2	Remarks about attendance will be maintained and notice to student will be given in case of less attendance.

2.4 System Requirements

The system requirements are classified as:

2.4.1 Hardware Requirements

Project is divided into client side and server side implementation so the hardware requirements as follows:

- **Server side requirements**
  - 2GB RAM
  - 100GB Hard disk.
  - Processor P4 1.4 GHz and above
- **Client side/mobile device requirements**
  - 512 MB RAM
  - 1GB Internal Storage
  - Intel Atom Processor Z 2760

2.4.2 Software Requirements

- Server side requirements
  - Operating sytem-windows7/xp
  - Programming language-core Java
  - Database system-MYSQL
  - Web server-Glassfish 3 or above.
- Client side/mobile device requirements
  - Operating sytem-windows7/xp /Android
  - Programming laguage-core Java,eclipse,SDK.

Why Java?

Java was designed to be easy for the Professional programmer to learn and to use effectively. Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Why MYSQL?

The main features of MYSQL is portability, multilayered server design with independent modules, fully multi-threaded using kernel threads, provides transactional and non-transactional storage engines, Uses very fast B-tree disk tables with index compression.

2.5 Process Model

Incremental model is used as the process model in the system. Figure 2.1 shows the process model of the system.

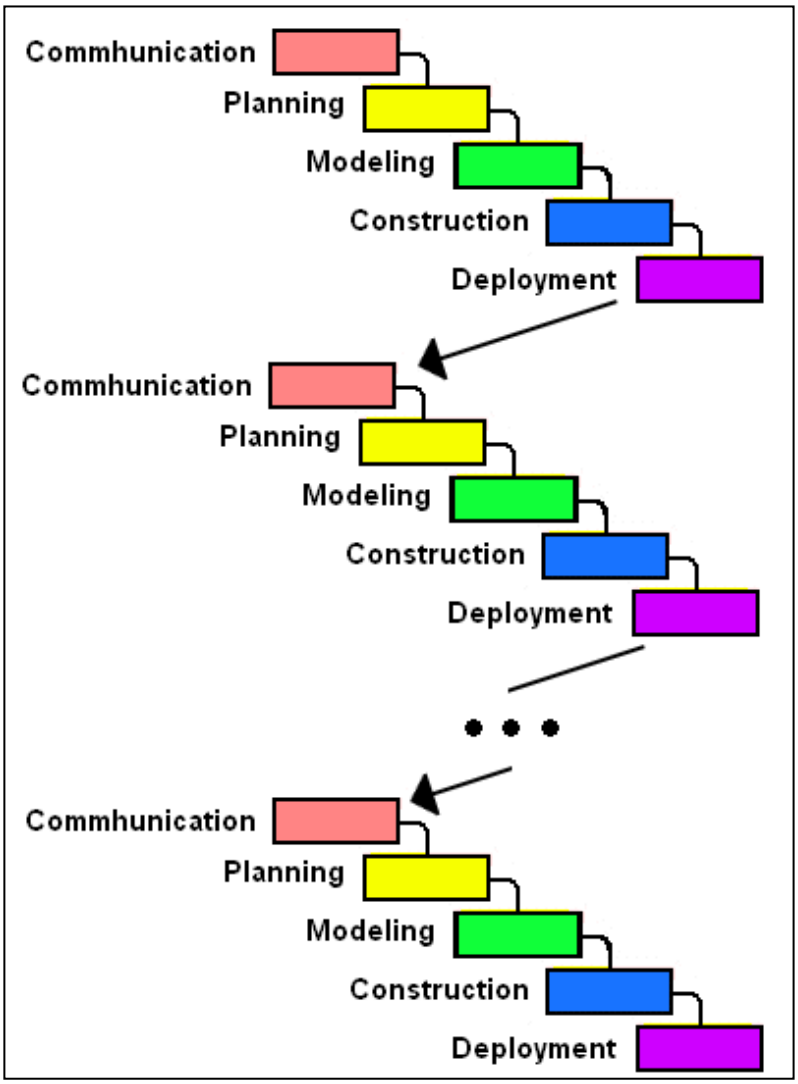


Fig. 2.1 Process Model

2.5.1 Incremental Model

The incremental model combines the elements of waterfall model applied in an iterative fashion. As in Figure 2.1 the incremental model applies linear sequences in a staggered fashion as calendar time progresses. Each linear sequence produces deliverable increments of the software. When an incremental model is used , the first increment if often a core product that is basic requirement are addressed but many supplementary features remain undelivered. The core

product is used by the customer. As a result of use and/or evaluation, a plan is developed for the next increment. The addresses the modification of the core product to better meet the needs of customer and delivery of additional feature and functionality. This process is repeated following the delivery of increment, until the complete product is produced. Figure 2.1depicts an incremental model that contains five phases:

Communication

Software development process starts with the communication between customer and developer .According to need of project , we gathered the requirements related to project.

Planning

It includes complete estimation and project scheduling and tracking. Also includes estimation of project cost and time.

Modeling

Tasks required building one or more representations of the application. It includes analysis and design.

Construction

It includes actual coding and testing.

Deployment

It includes delivery of the partially implemented project and takesfeedback.

2.5.2 Disadvantages of Other Process Models Compared to Incremental model

- **Waterfall Model:**

1. In this method, all the requirements of the software need to be specified upfront and there is no room for committing mistakes.
2. The project scope statement needs to be detailed in infinite depth from the start because changes are not possible when using waterfall methodology. This is because the only way to amend something which has been already developed is to go back

and start again. This will cause huge problems on projects where the project sponsors are indecisive and quickly causes scope creep.

3. Project communications with the client are extremely limited being either at the beginning or at the end of the development. In between, there is no way in which one can get feedback or potentially clarify any confusion over what the requirement actually means. The knock on effect is that it is up to the project team to make the key decisions on what requirements can be developed within the timeframes required, and what is developed later in a later deployment release by project planning in teams. This not only increases the overall time required to develop the software but also means that despite the team's best efforts, the customer may still be extremely unhappy with the end product delivered.

- **Spiral Model:**

1. Spiral models work best for large projects only, where the costs involved are much higher and system pre requisites involves higher level of complexity.
2. Spiral model needs extensive skill in evaluating uncertainties or risks associated with the project and their abatement.
3. Spiral models work on a protocol, which needs to be followed strictly for its smooth operation. Sometimes it becomes difficult to follow this protocol.
4. Evaluating the risks involved in the project can shoot up the cost and it may be higher than the cost for building the system.
5. There is a requirement for further explanation of the steps involved in the project such as breakthrough, blueprint, checkpoints and standard procedure.

- **Prototyping Model**

1. Producer might produce a system inadequate for overall organization needs
2. User can get too involved whereas the program cannot be to a high standard
3. Structure of system can be damaged since many changes could be made
4. Producer might get too attached to it (might cause legal involvement)
5. Not suitable for large applications
6. Over long periods, can cause loss in consumer interest and subsequent cancellation due to a lack of a market (for commercial products)

7. May slow the development process, if there are large numbers of end users to satisfy.

2.5.3 Advantages of Incremental Model

The incremental process model, like prototyping and other evolutionary approaches is iterative in nature. But unlike prototyping, the incremental model focuses on the delivery of an operational product with each increment. Early increments are stripped down version of the final product, but they do provide capability that serves the user and also provide the platform for evolution by the user. Incremental development is particularly useful when staffing is unavailable for a complete implementation by business deadline that has been established for the project. Early increments can be implemented with fewer people. If the core product is well received, additional staff can be added to implement next increment. In addition increment can be planned to manage technical risks. For example, a major system requires the availability of new hardware that is underdevelopment and whose delivery is uncertain. It might be possible to plan early increment in a way that avoids the use of hardware, thereby enabling partial functionality to be delivered to end users without in ordinate delay.

2.5.4 Why Incremental Model?

1. For further increments database is updated.
2. It is simple management model.

2.6 Breakdown Structure (Module)

Following figure 2.2 shows break down structure of the project module wise.

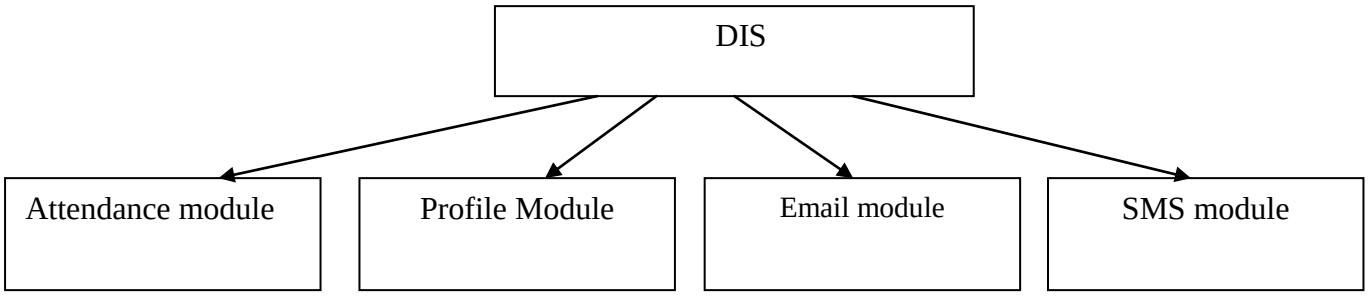


Fig. 2.2 Breakdown Structure (Module)

Details of each module are as follows

Attendance Module

This module is mainly for the faculties in the class who will be taking attendance. By having this module they are able to fill the attendance into the ERP server directly with the help of GPRS connectivity to the server.

Profile Module

In this module we will be making user interface for teachers ,students and higher authority. Make user interface more readable and more interactive so that the application must have good quality. And stand right on its standard.

Email Module

Those students who are not having that app downloaded that student will get the notifications by emails and for that connectivity we will require this module. In this module we will send the emails to their respective email id which will be present at the database.

SMS Module

In this module we will send the SMS to their respective Ph. no which will be present at the database.

2.7 Project Cost Estimation

The cost estimation of the project is done as follows:

2.7.1 Estimation of KLOC

The number of lines required for implementation of various modules can be estimated as follows:

Table 2.2 Estimation of KLOC

Sr. no.	Modules	KLOC
1.	User module	0.60
2	Attendance Entry module	0.45
3.	GPRS Connectivity	0.50
4.	Database module	0.40
5.	Email module	0.35
6.	SMS module	0.30
	Total	2.60

Thus, the total number of lines required is approximately 2.60 KLOC.

Efforts:

The efforts required in person per month are calculated as:

E=3.2* (KLOC) ^1.02

E=3.2* (2.60) ^1.02

E=8.68 person-month

Development Time (In Months)

The development time for project is estimated as:

D=E / N.

D=8 .68/3

D=2.89months.

Development Time for project

The total time for project is estimated as:

- Requirement analysis& design requires 3 months…….[1]
- Implementation & testing requires 2.25months…….[2]

  D=3+2.89……..(By adding value [1] and [2])

  D=5.89 months

Number of Persons

Three persons are required to complete the project within given time span successful.

CHAPTER 3

SYSTEM ANALYSIS AND RISK MANAGEMENT

3.1 Project Scheduling and Tracking

Project Scheduling and Tracking is important because in order to build a complex system, many software engineering tasks occur in parallel, and the result of work performed during one task may have a profound effect on work to be conducted in another task. These interdependencies are very difficult to understand without a detailed schedule.

3.1.1 Project Work Breakdown Structure (Analysis)

Figure 3.1 shows the Project Work Breakdown Structure. The proposed system is divided into four main tasks viz. requirement gathering& validation, planning, system design, risk analysis & management.

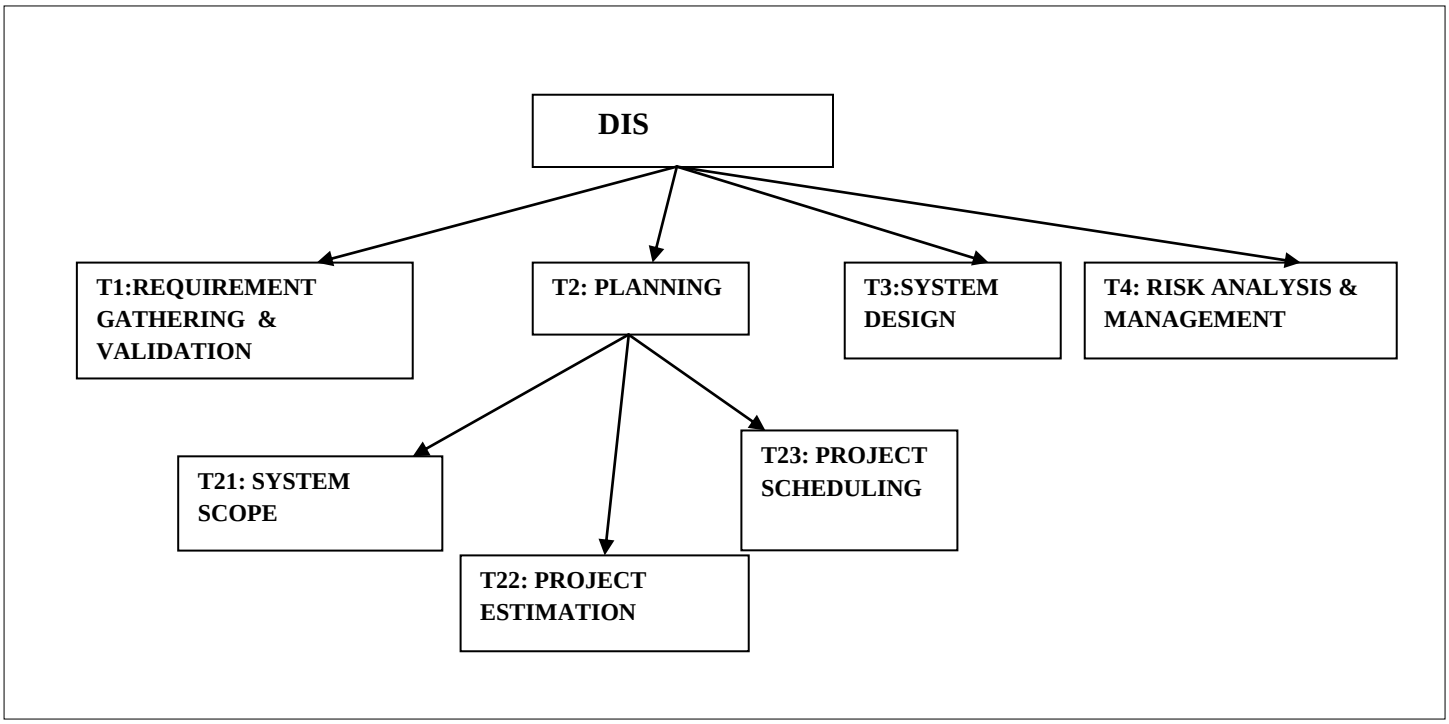


Fig. 3.1 Project Work Breakdown Structure (Analysis)

3.1.2 Project Work Breakdown Structure (Modules)

The module wise work break down structure is shown in Figure 3.2

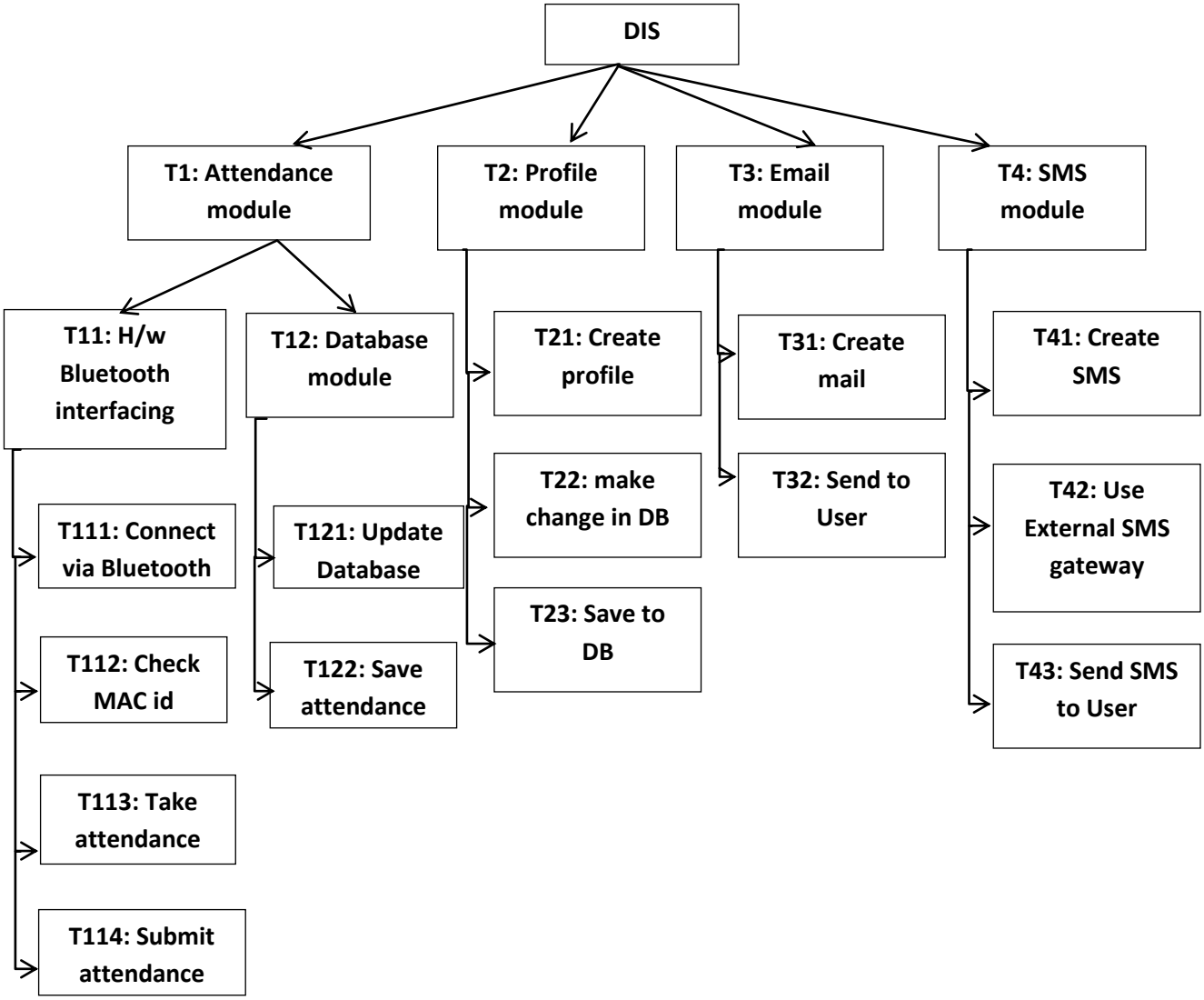


Fig. 3.2 Project Work Breakdown Structure (Modules)

3.2 Tasks

The tasks are categorized as per the below list to complete project work.

T1: Searching for project definitions.

T2: Required literature collection.

T3: Literature review.

T4: Give presentation of topic.

T5: Finalization of algorithms to be implemented.

T6: Allocation of responsibilities.

T7: Synopsis Submission.

T8: Requirements gathering and validation.

T9: System design.

T10: System analysis.

T11: UML Modeling

T12: Risk analysis and management.

T13: Completion of partial project report.

T14: Creating Initial GUI.

T15: Connectivity from mobile to server to store attendance.

T16: Handling of complaints by the server.

T17: Coding for registering to the server.

T18: Coding for android app for user.

T19: Coding for server storage.

T20: Implement algorithm for authenticated connection.

T21: Unit testing for Json parsing.

T22: Make required modifications.

T23: Testing and trouble shooting.

T24: Report generation.

T25: Deployment.

Each task is assigned to one or more team members, where

D1: KULKARNI AMRUTA C.

D2: KULKARNI PRACHI D.

D3: RODE AJINKYA V.

3.3 Project Schedule

The table 3.1 describes the schedule for project development and also highlights all the tasks to be carried out along with their duration, dependency and developer(s) assigned to accomplish the task.

Table 3.1 Project Schedule

Tasks	Days	Dependencies	Developers Assigned
T1	10	-	D1,D2,D3
T2	7	T1	D1,D2,D3
T3	8	T1,T2	D1,D2,D3
T4	5	T2	D1,D2,D3
T5	4	T4	D1,D2,D3
T6	4	T4,T5	D1,D2,D3
T7	7	T4,T5,t6	D1,D2,D3
T8	10	T2,T4	D1,D2,D3
T9	12	T2,T4,T8	D1,D2,D3
T10	9	T9	D1,D2,D3
T11	4	T8,T9,T10	D1,D2,D3
T12	5	T7,T9,T10	D1,D2,D3
T13	4	T8,T9,T11	D1,D2,D3
T14	14	T13	D1,D2
T15	8	T5,T9,T10	D1,D3
T16	7	T4,15,T11	D2,D3

Tasks	Days	Dependencies	Developers Assigned
T17	7	T9,T10	D1,D2,D3
T18	12	T13-T17	D1,D2
T19	7	T5,T9,T10	D1,D2,D3
T20	6	T14-T19	D1,D2,D3
T21	6	T14-T20	D1,D2,D3
T22	5	T13-T20	D1,D2,D3
T23	5	T19-T20	D1,D2,D3
T24	6	-	D1,D2,D3
T25	10	-	D1,D2,D3
Total	182		

3.3.1 Project Table

The Table 3.2 shows expected and actual Starting and Ending time of the tasks mentioned above:

Table 3.2 Project Table

Task no	Starting time		Ending Time		Developers Assigned	Remark
	Expected	Actual	Expected	Actual		
T1	01/07/13	01/07/13	10/07/13	12/07/13	D1,D2,D3	
T2	11/07/13	13/07/13	17/07/13	19/07/13	D1,D2,D3	
T3	18/07/13	20/07/13	25/07/13	28/07/13	D1,D2,D3	
T4	26/07/13	29/07/13	30/07/13	02/08/13	D1,D2,D3	
T5	31/07/13	03/08/13	03/08/13	07/08/13	D1,D2,D3	

T6	04/08/13	08/08/13	07/08/13	09/08/13	D1,D2,D3	
T7	08/08/13	10/08/13	14/08/13	16/08/13	D1,D2,D3	
T8	15/08/13	17/08/13	24/08/13	25/08/13	D1,D2,D3	
T9	25/08/13	26/08/13	05/09/13	07/09/13	D1,D2,D3	
T10	06/09/13	08/09/13	14/09/13	17/09/13	D1,D2,D3	
T11	15/09/13	18/09/13	18/09/13	21/09/13	D1,D2,D3	
T12	19/09/13	22/09/13	23/09/13	25/09/13	D1,D2,D3	
T13	24/09/13	26/09/13	30/09/13	09/10/13	D1,D2,D3	
T14	28/12/13	28/12/13	10/01/14	05/01/14	D1,D2	
T15	11/01/14	06/01/14	18/01/14	11/01/14	D1,D2,D3	
T16	19/01/14	12/01/14	25/01/14	16/01/14	D1,D2,D3	
T17	26/01/14	17/01/14	01/02/14	26/01/14	D2,D3	
T18	02/02/14	27/01/14	13/02/14	04/02/14	D1,D3	
T19	14/02/14	05/02/14	20/02/14	12/02/14	D1,D2,D3	
T20	21/02/14	13/02/14	26/02/14	23/02/14	D1,D2,D3	
T21	27/02/14	24/02/14	04/03/14	27/02/14	D1,D2,D3	
T22	05/03/14	28/02/14	09/03/14	09/03/14	D1,D2,D3	
T23	10/03/14	10/03/14	14/03/14	15/03/14	D1,D2,D3	
T24	15/03/14	16/03/14	20/03/14	27/03/14	D1,D2,D3	
T25	21/03/14	28/03/14	30/03/14	30/03/14	D1,D2,D3	

3.3.2 Time Line Chart (Analysis& Design)

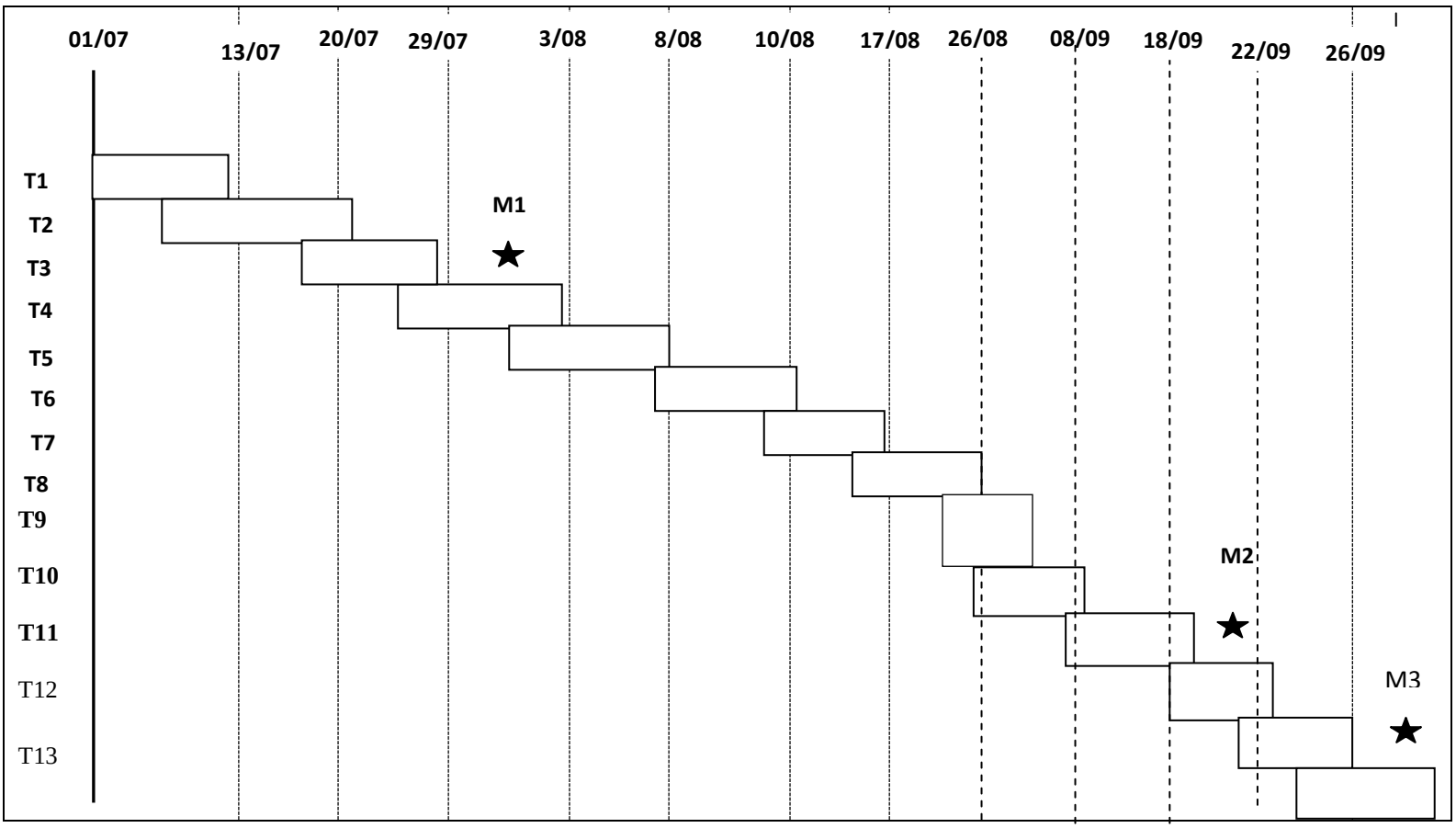


Fig. 3.3 Time Line Chart (I) for phase -1

★ Indicates Milestone

M1: Give presentation Topic

M2: UML modeling

M3: Partial Project definition Submission

3.3.3 Time Line Chart (Implementation):

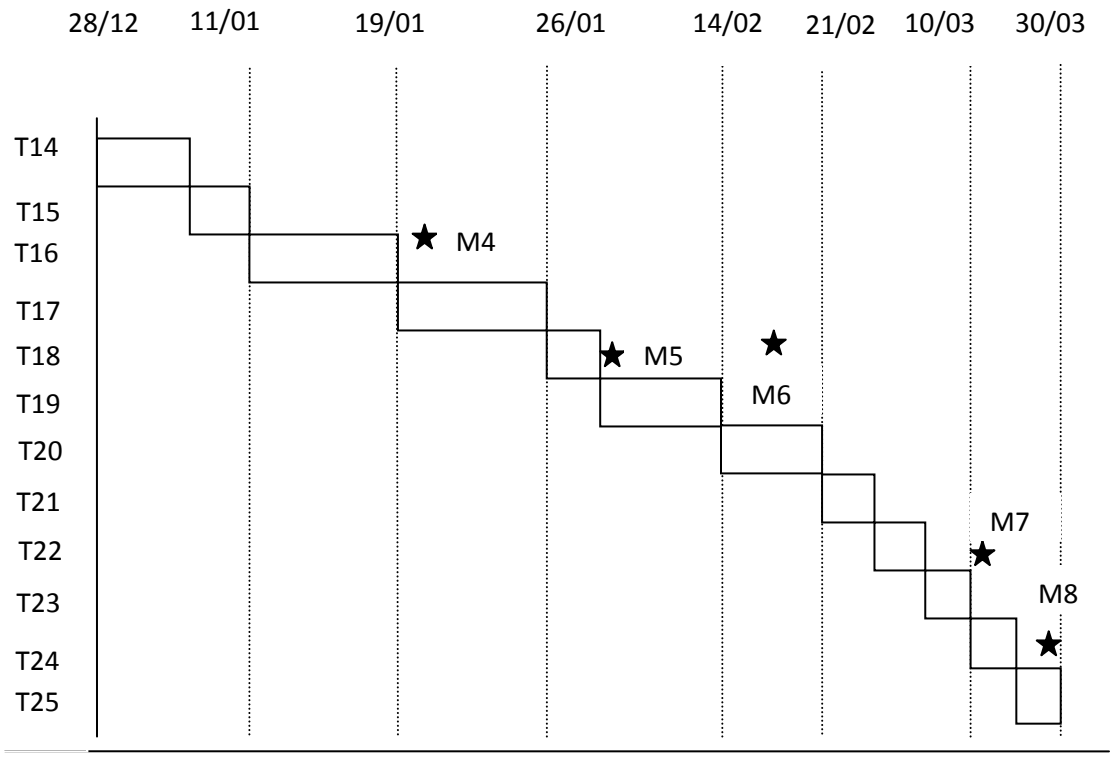


Fig. 3.4 Time Line Chart2 for Phase-2

★ **Indicates Milestone**

M4: Coding for registering to server

M5: Coding for server storage

M6: Implement algorithm for authenticated connection.

M7: Testing and troubleshooting

M8: Deployment of App

3.4 Analysis Model

Analysis modeling contains following modeling.

3.4.1 Behavioral Modeling

Behavioral diagrams depict the behavioral features of a system or business process. Behavioral diagrams include the following diagram types:

3.4.1.1 Use Case Diagram

A use case involves a sequence of interactions between the initiator and the system, possibly involving other actors. A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

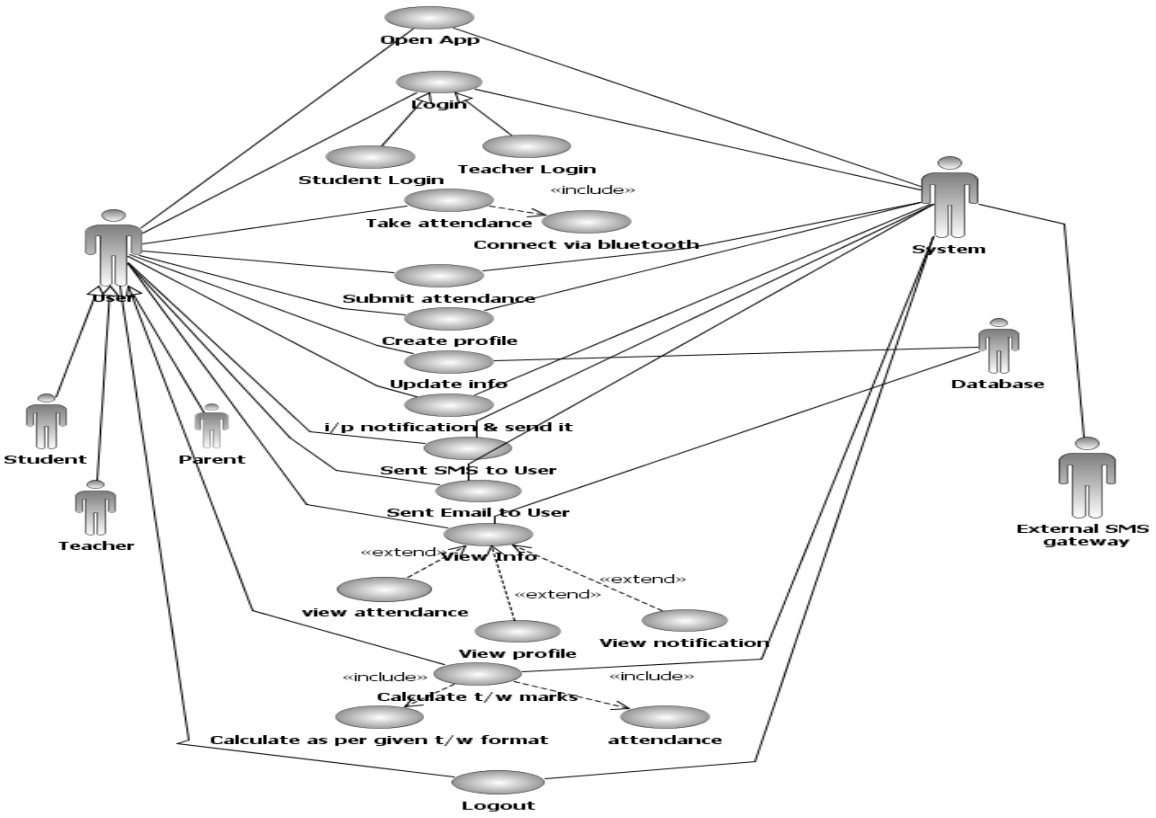


Fig. 3.5 Use Case Diagram

3.4.1.2 Class Diagram

In design specification it can be used to specify interfaces and classes that will be implemented in an object oriented program. The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

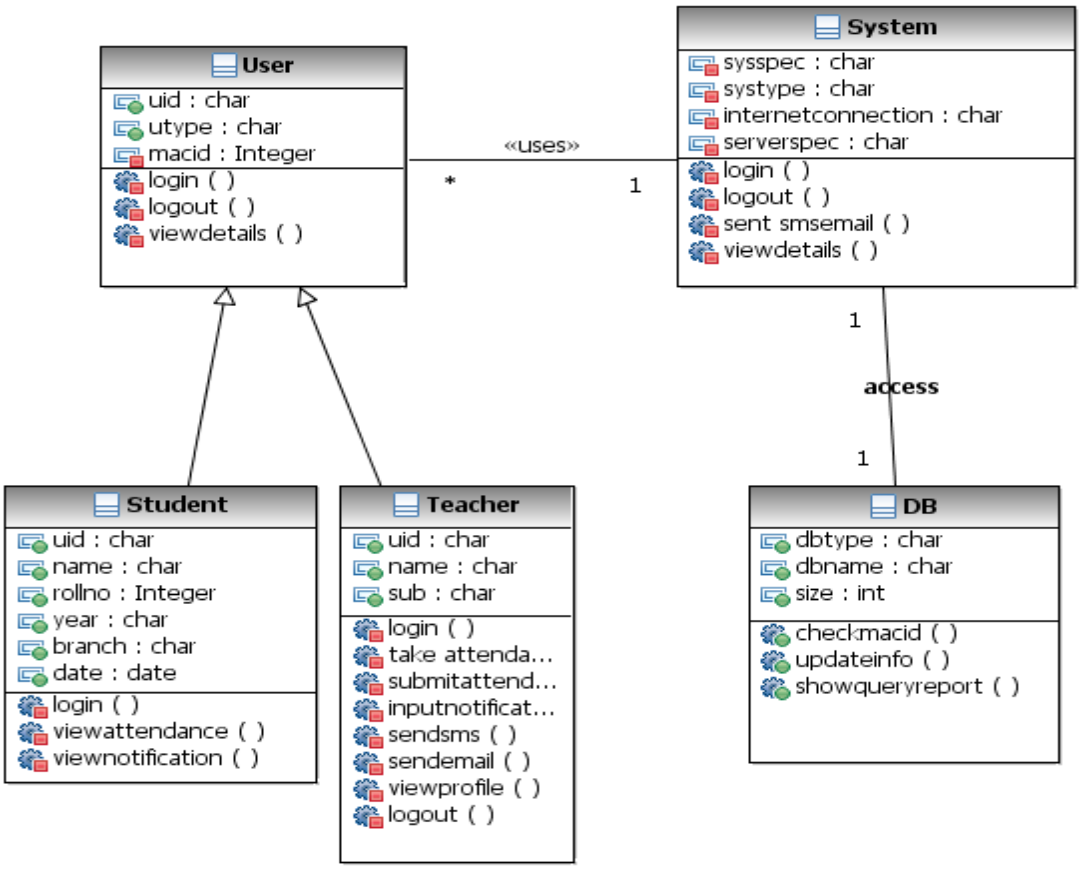


Fig. 3.6 Class Diagram

3.4.1.3 Sequence Diagram

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces. This type of diagram is best used during early analysis phases in design because they are simple and easy to comprehend. Sequence diagrams are normally associated with use cases.

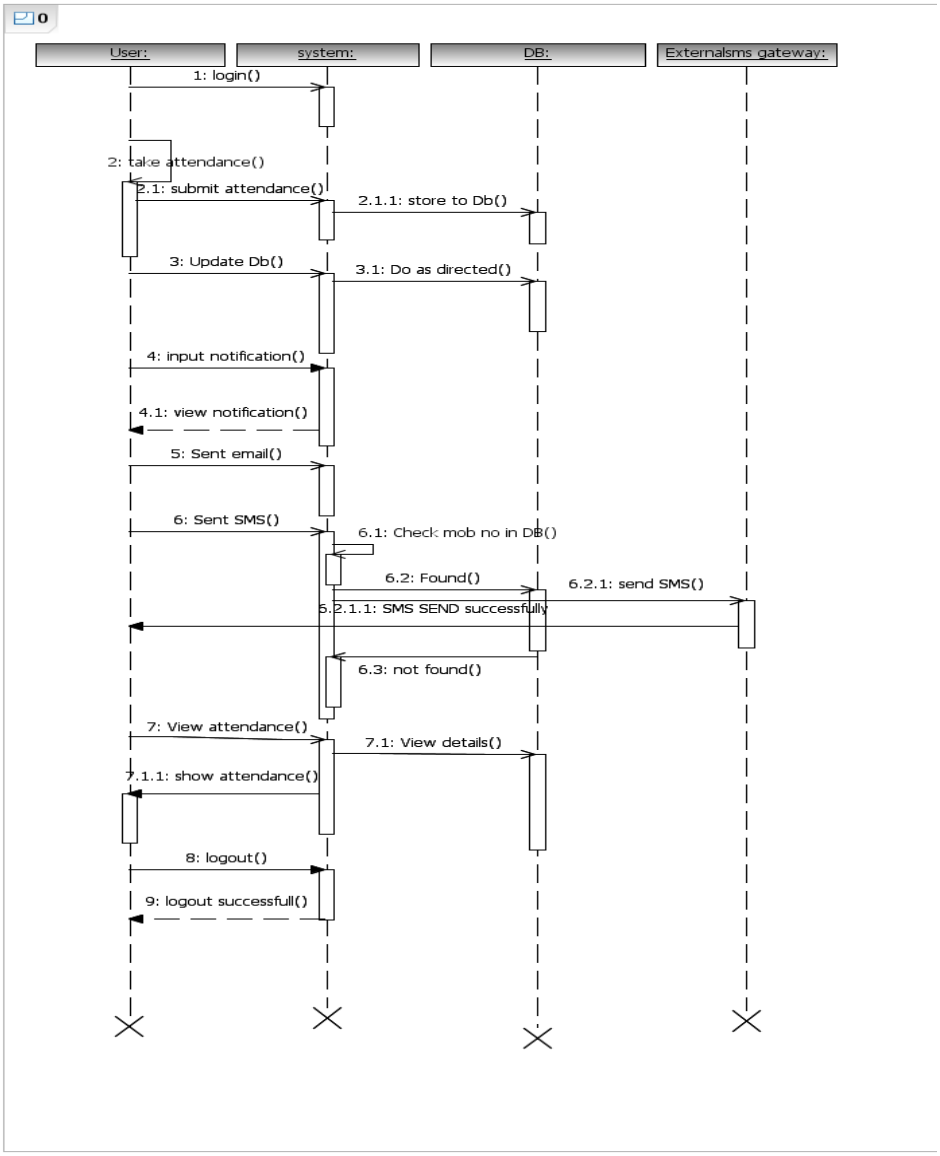


Fig. 3.7 Sequence Diagram

3.4.1.4 Activity Diagram

Activity diagrams are very similar to a flowchart because you can model a workflow from activity to activity.

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organisational processes (i.e. workflows).Activity diagrams show the overall flow of control. rounded rectangles represent actions:

- Diamonds represent decisions.
- Bars represent the start (split) or end (join) of concurrent activities.
- A black circle represents the start (initial state) of the workflow.
    - An encircled black circle represents the end (final state).

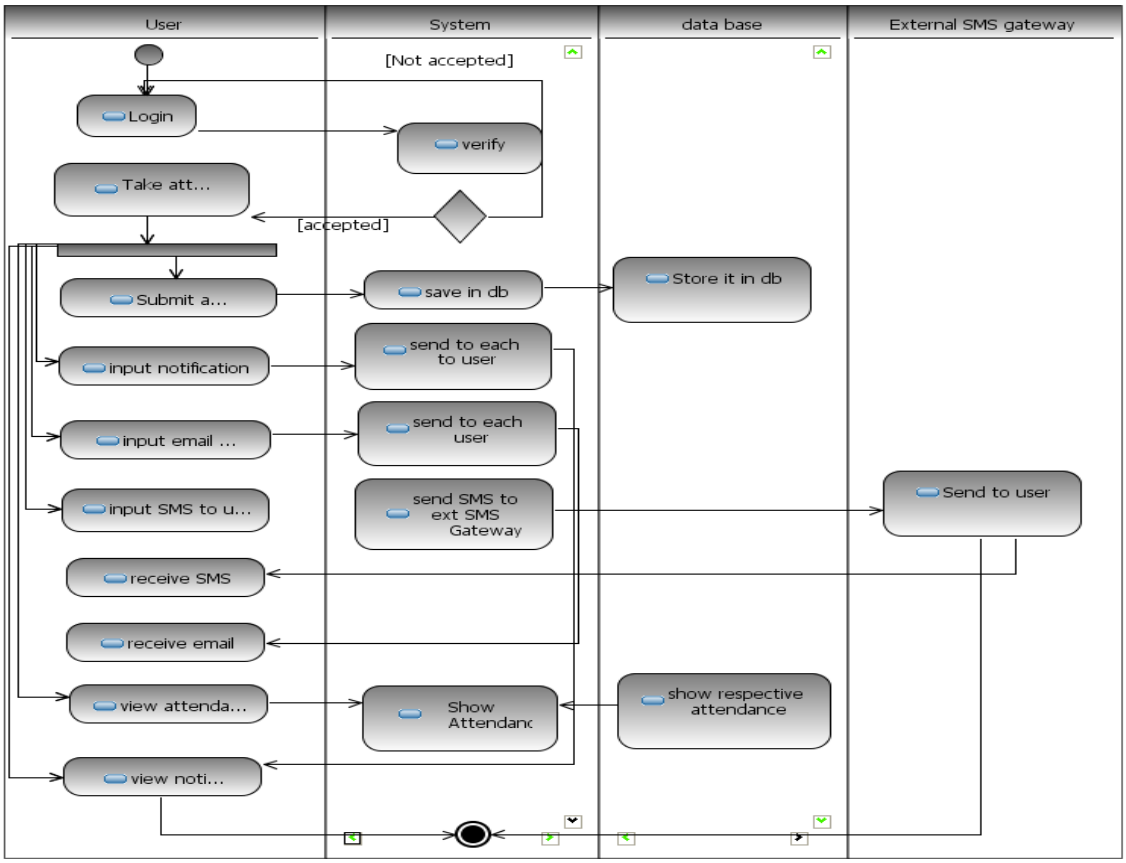


Fig. 3.8 Activity Diagram

3.4.1.5 State Chart Diagram

State chart diagrams model the dynamic behavior of individual classes or any other kind of object. They show the sequences of states that an object goes through, the events that cause a transition from one state to another and the actions that result from a state change.

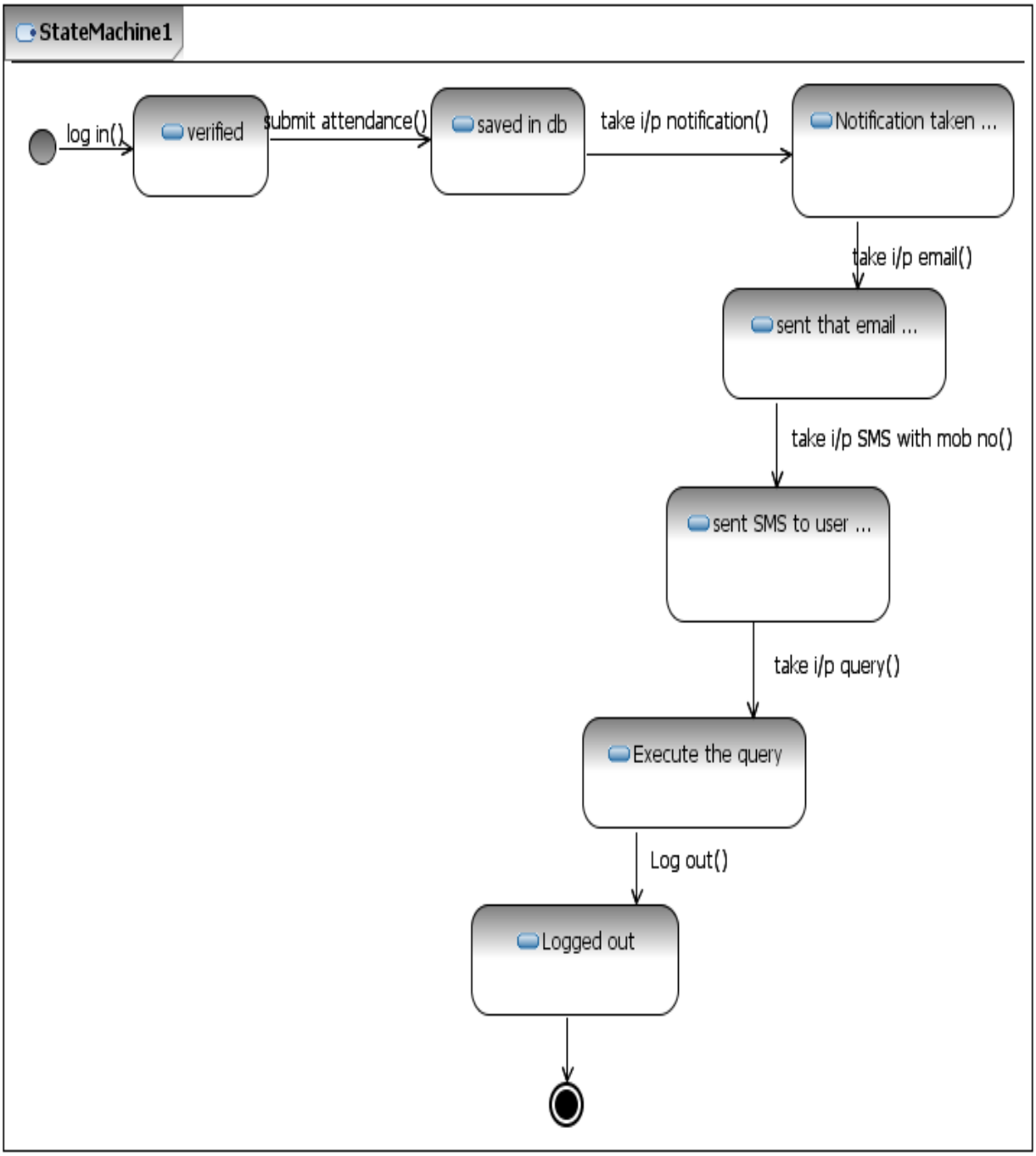


Fig. 3.9 State Chart Diagram

3.4.2 Functional Modeling

A function model or functional model in systems engineering and software engineering is a structured representation of the functions (activities, actions, processes, operations) within the modeled system or subject area**.**

3.4.2.1 Data Flow Diagram

Data flow diagram (DFD) is also called as 'Bubble Chart' is a graphical technique, which is used to represent information flow, and transformers those are applied when data moves from input to output. DFD represents system requirements clearly and identify transformers those becomes programs in design. DFD may further partitioned into different levels to show detailed information flow e.g. level 0, level 1 etc.

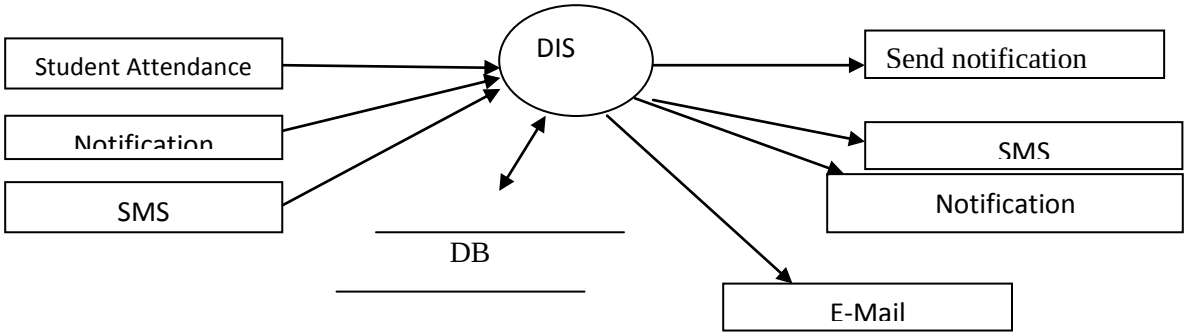


Fig. 3.10 Data Flow Diagram (level 0)

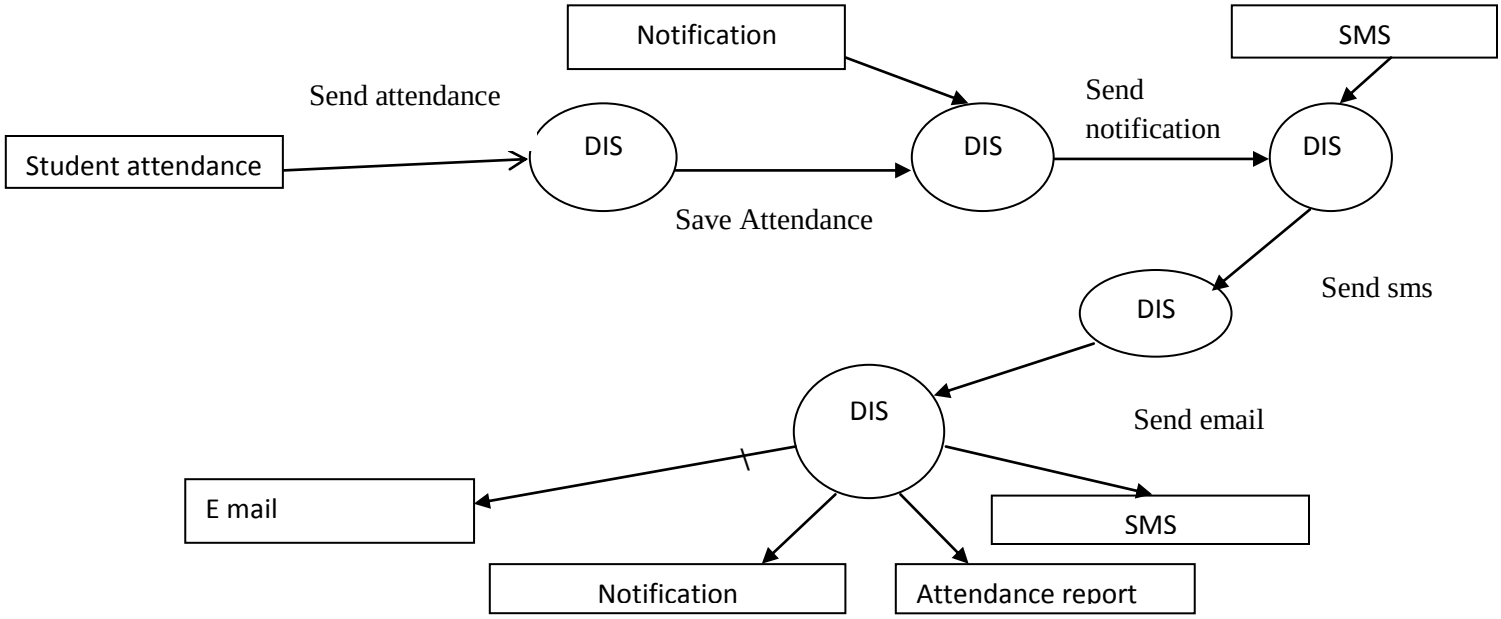


Fig. 3.11 Data Flow Diagram (Level 1)

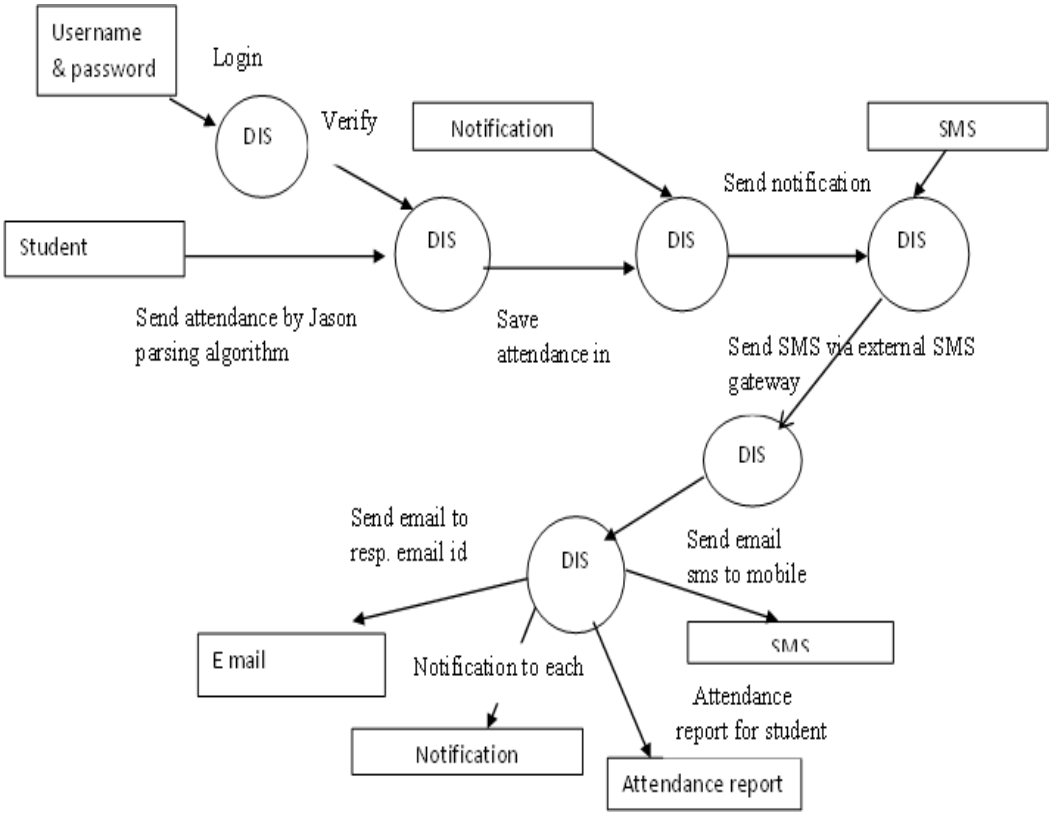


Fig 3.12 Data Flow Diagram (Level 2)

3.4.2.2 Control Flow Diagram:

A flow diagram can be developed for the process control system for each critical activity. Process control is normally a closed cycle in which a sensor provides information to a process control software application through a communications system. The application determines if the sensor information is within the predetermined (or calculated) data parameters and constraints. The results of this comparison are fed to an actuator, which controls the critical component. This feedback may control the component electronically or may indicate the need for a manual action.

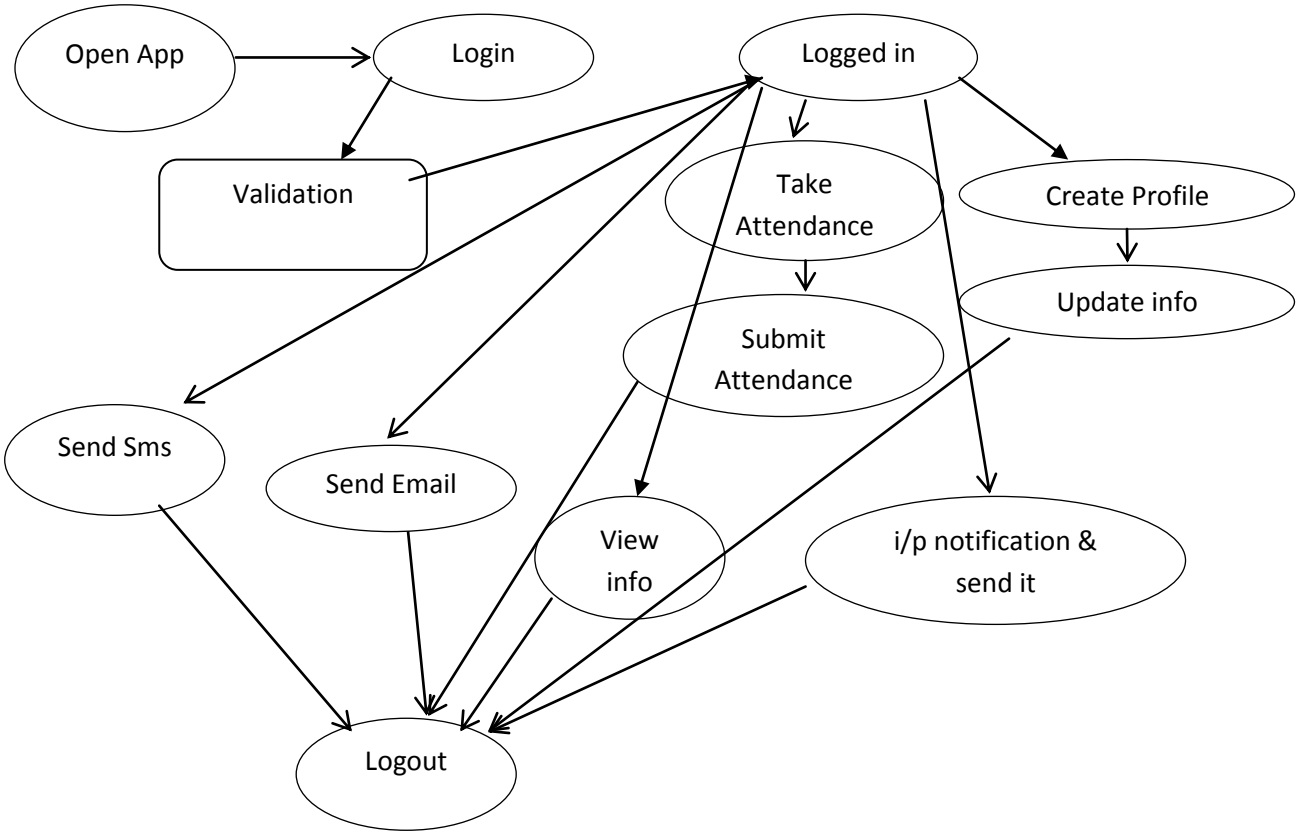


Fig. 3.13 Control Flow Diagram

3.4.3 Architectural Modeling

It represents the overall framework of the system. It contains both structural and behavioral elements of the system. Architectural model can be defined as the blue print of the entire system. Package diagram comes under architectural modeling.

3.4.3.1 Component Diagram

Component diagrams provide a physical view of current model. A component diagram shows organizations and dependencies among software components, including source code components, binary code components, and executable components. These diagrams also show externally-visible behavior of components by displaying interfaces of the components.

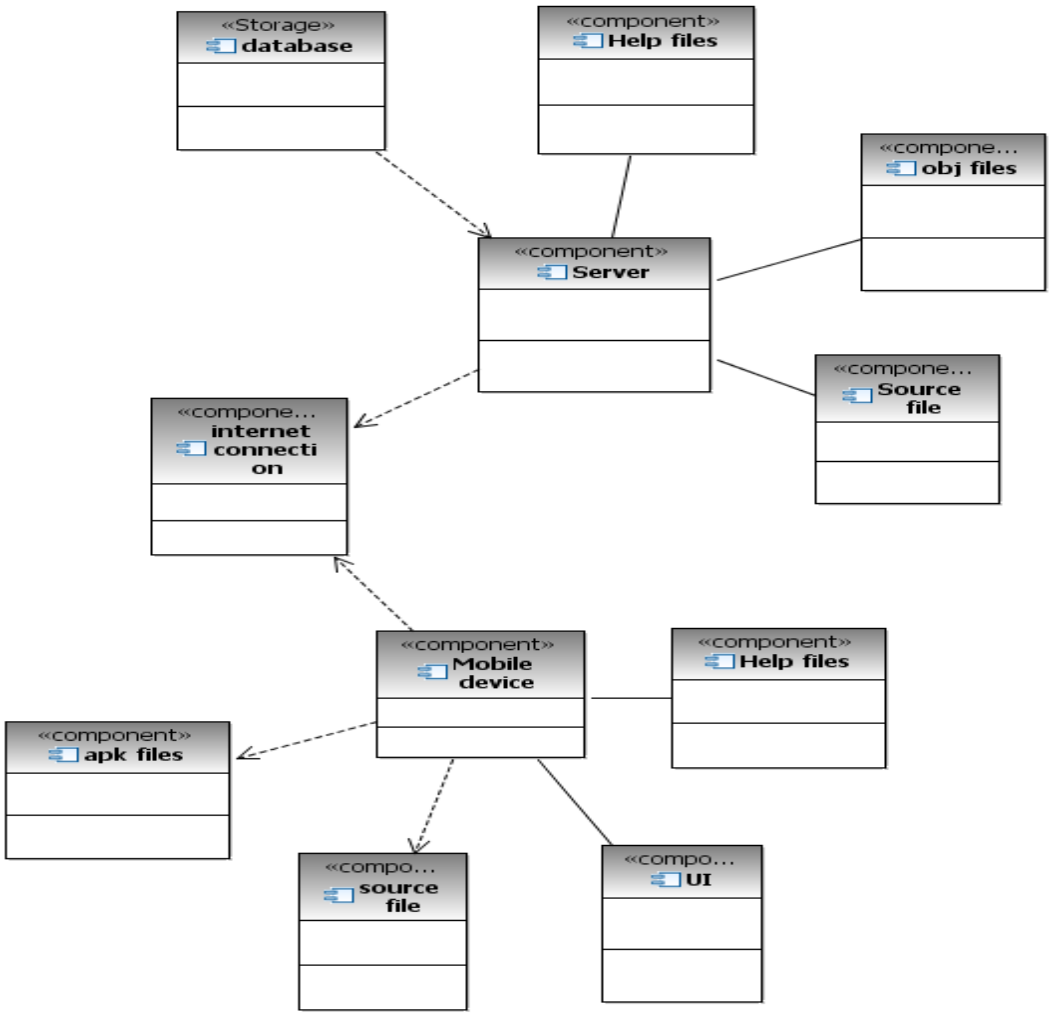


Fig. 3.14 Component Diagram

3.4.3.2 Deployment Diagram

A deployment diagram shows the allocation of processes to processors in the physical design of a system. A deployment diagram may represent all or part of the process architecture of a system.

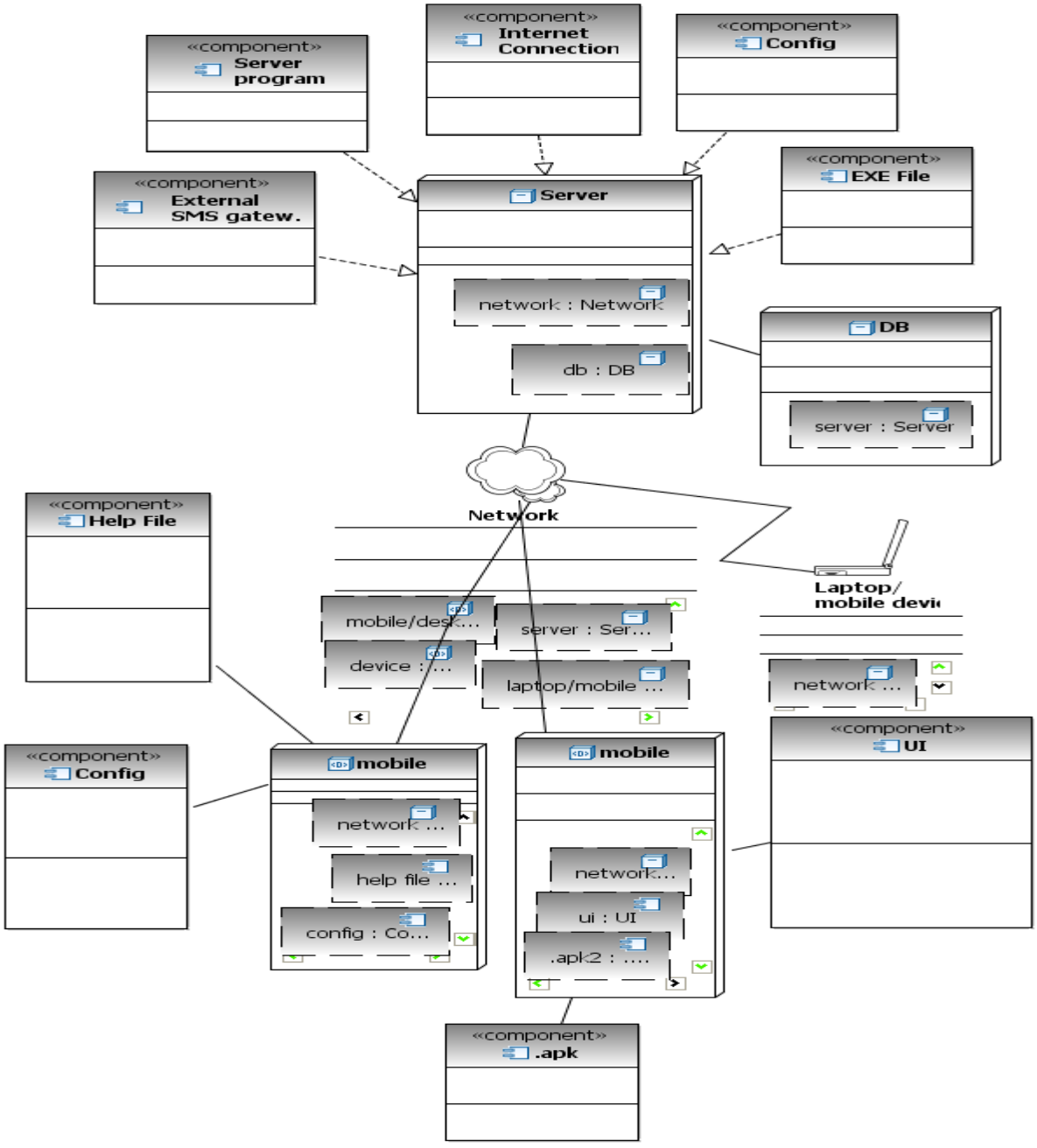


Fig. 3.15 Deployment Diagram

3.5 Mathematical Modeling

A decision problem is a problem whose output is a single Boolean value: YES or NO. It defines three classes of decision problems:

• ***P*** is the set of decision problems that can be solved in polynomial time. Intuitively, P is the set of problems that can be solved quickly.

• ***NP*** is the set of decision problems with the following property: If the answer is YES, then there is a *proof* of this fact that can be checked in polynomial time. Intuitively, NP is the set of decision problems where we can verify a YES answer quickly if we have the solution in front of us.

•***co-NP*** is the opposite of NP. If the answer to a problem in co-NP is NO, then there is a proof of this fact that can be checked in polynomial time.

For example, the circuit satisfiability problem is in NP. If the answer is YES, then any set of *m* input values that produces TRUE output is a proof of this fact; we can check the proof by evaluating the circuit in polynomial time. It is widely believed that circuit satisfiability is *not* in P or in co-NP, but nobody actually knows. Every decision problem in P is also in NP. If a problem is in P, we can verify YES answers in polynomial time recomputing the answer from scratch! Similarly, any problem in P is also in co-NP.

Perhaps the single most important open questions in theoretical computer science, if not all of mathematics, is whether the complexity classes P and NP are actually different.

NP-hard, NP-easy, and NP-complete

A problem _ is **NP-hard** if a polynomial-time algorithm for _ would imply a polynomial-time algorithm for *every problem in NP*. In other words:

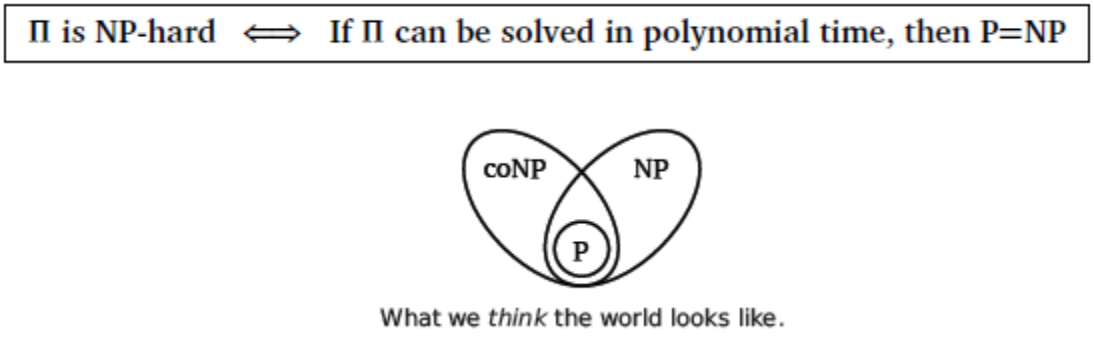


Fig. 3.16 P, NP and CO-NP Problem

P = NP means that for all problems whose solutions can be efficiently verified, the solutions can be efficiently generated too. It is widely believed that P 6= NP. This problem is listed as one of the seven most important unsolved problems in mathematics. There is a $1 million prize for anyone who proves P = NP or P!= NP!

From above study it is identified that this problem definition comes in P class.This problem has following inputs, processes, outputs, and rules.

INPUTS

The inputs can be represented as in the form of set as follows

I= {I1, I2,I3,I4}……..(1)

Where I1=studentattendance

I2= Notices,

I3=message

I4=Profile (personal details)

PROCESSES

Processes can be represented as in the form of set as follows:

P= {P1,P2,P3,P4,P5,P6,P7,P8,P9,P10}……(2)

Where P1= Take attendance

P2= Send attendance,

P3=Save data,

P4=JsonParsingAlgorithm,

P5=Show attendance,

P6=Send the notification,

P7=Show notification,

P8=Send SMS,

P9=Show profile,

P10=Send Email

OUTPUT

Outputs can be shown in set as follows

O= {O1, O2, O3, O4, 05}…….(8)

Where O1= Attendance Report

O2=Notification,

O3=Email message

O4= SMS message

O5=Profile details

3.6 Venn Diagram

Venn diagram represents the complete view of the project with dependencies between the inputs, processes, rules imposed on them and finally the output of the project .this complete scenario is represented in the following Figure .3.17

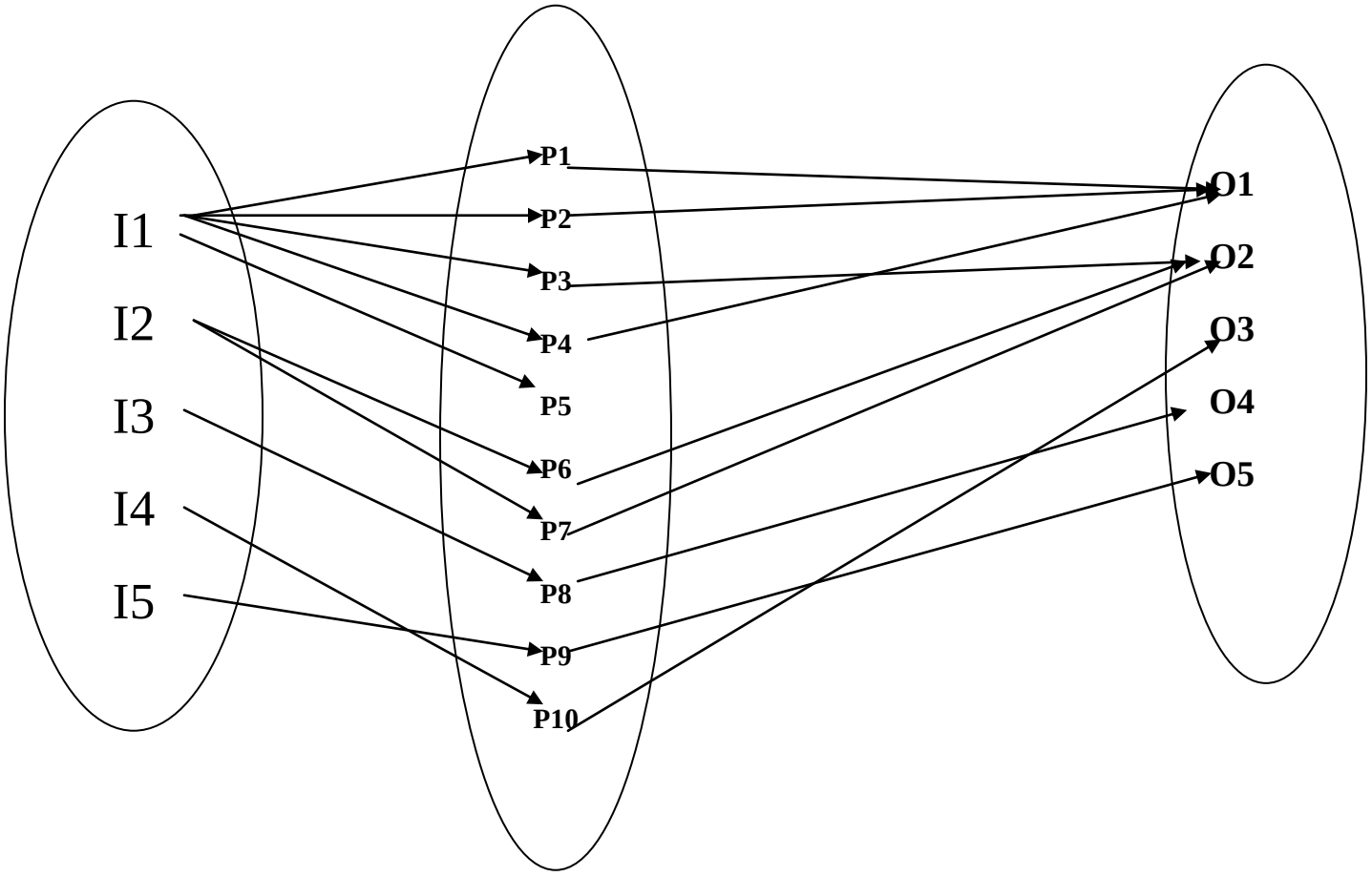


Fig. 3.17 Venn Diagram for the Project System

Process state diagram

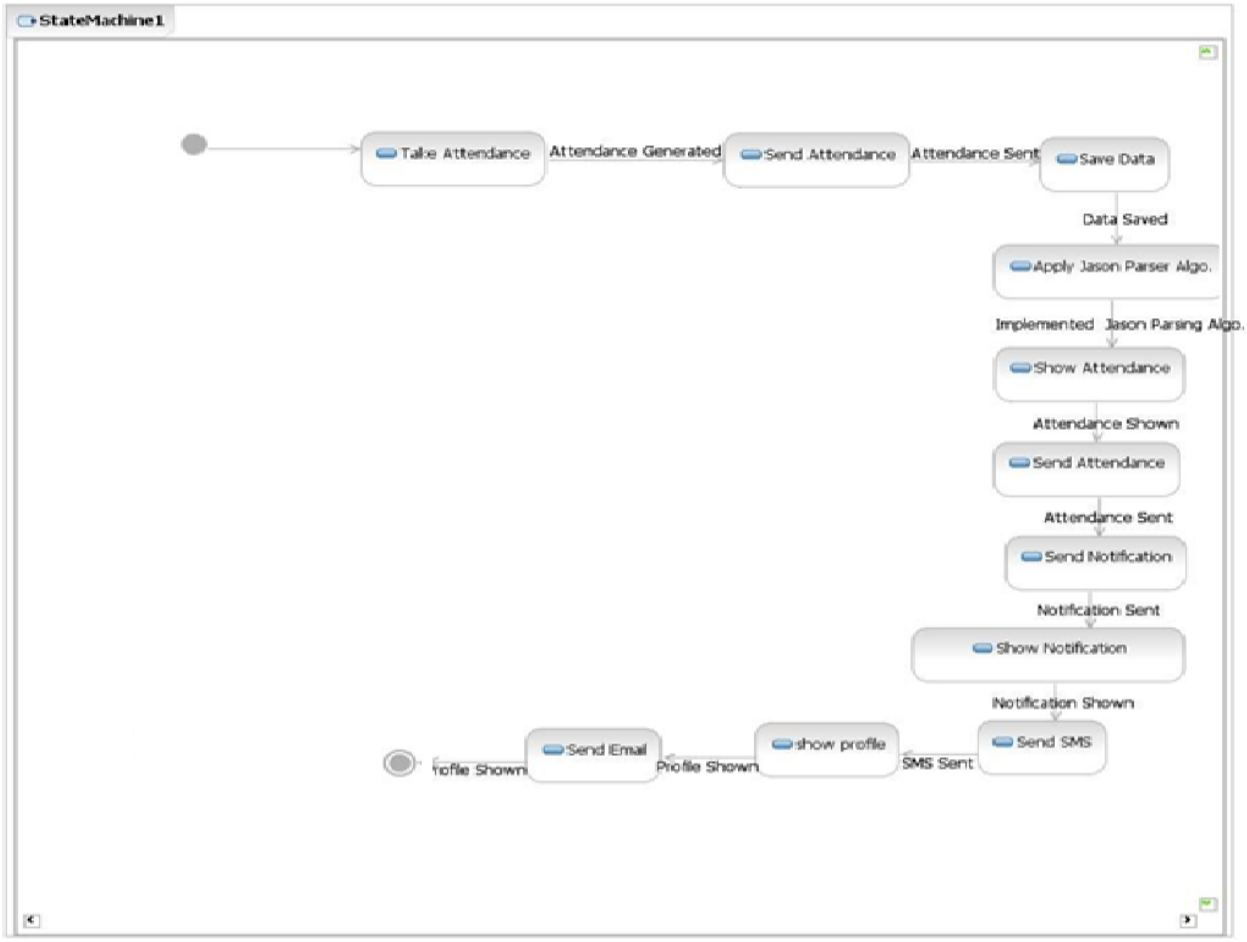


Fig. 3.18 Process State Diagram

- Time Complexity

1) Start App
2) Time complexity for taking attendance- O(n)
3) Time complexity for sending attendance(JSON Parsing Algorithm)- O(n)
4) Time complexity for Saving attendance- O(1)
5) Time complexity for Profile- O(n)
6) Time complexity for sending mail or SMS- O(n)
7) Time complexity for Event notifier- O(n)
8) Close App.

   Total time complexity of system=O (n).

3.7 Risk Management

Risk management involves anticipating risks that might affect the project schedule or quality of the software being developed and taking action to avoid these risks. It is the identification, assessment, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability and/or impact of unfortunate events or to maximize the realization of opportunities.

Effective risk management makes it easier to cope with problems and to ensure that these do not lead to unacceptable budget or schedule slippage.

3.7.1 Risk Identification

Following types of risks are to be identified:

3.7.1.1 Product Size Related

R1: Project does not complete on time. Extra lines of code may cause wastage of memory.

3.7.1.2 Business Impact

R2: Delay in project delivery (violation in time constraints) can hamper the customer economically.

R3: If System is not efficient than the existing system, it will cause economic losses.

3.7.1.3 Customer Related

R4: As the customer is not the technical person, so it may cause a problem while understanding the exact requirements of the stakeholder.

R5: If the user asks for change and gives any unexpected changes in later stages that is difficult to alter the entire system.

3.7.1.4 Development Environment Related

R6: Lack of training on the development tools may cause an inefficiency and difficulty in project completion.

3.7.1.5 Staff Size Related

R7:Improper planning about completion of the project.

R8: Illiterate to project management skills.

R9: Low level technical and project experience of the software engineers who are working on the project.

R10: As the scope of project is too large, it may be difficult to complete it in the given span of time with 3 persons.

3.7.1.6 Technical Risks

R11: Improper communication of client machine with server.

R12: Crashing of database may fail the system leading to maintain backup the database.

R13: System may fail to provide desired efficiency.

3.8 Strategies Used to Reduce Risks

S1. Formulation and follow of the project plan.

S2. Keep assigned work under threshold value.

S3. Efficient feature extraction techniques should be used.

S4. Regular meeting with customer reduce the risks to some extent, design system with flexibility and maintain necessary documentation for the same.

S5. Redefine software process at higher degree.

S6. Proper training on required technical tools for development of project reduces risks.

S7. Ensures that all the members are participating in the design, so that everyone knows And contribute to planning for completion of the project.

S8. Study and understanding of the project definition, study of languages used for system development and all its related software.

S9. Provide proper guidance to software developer by organizing expert lectures.

S10. Check for correct establishment of connection between client and server.

S11. Time constraints must be followed to avoid economical risks.

S12: Backup should be maintained.

S13: More time should be spending on testing and improvement.

3.9 Risk Projection

Table 3.3 lists all possible risks which may occur at any stage during development of project. Table also clearly shows the impact of risks and RMMM (Risk Mitigation Monitoring and Management) plan to deal with any such risks.

Table 3.3 Risk Table

Risk	Category	Probability	Impact	RMMM Plan
R1	Product Size	More	High	S1
R2	Business impact	More	High	S2,S11
R3	Business impact	More	High	S3,S11
R4	Customer related	Less	Low	S4
R5	Customer related	Less	High	S5
R6	Development Environment related	Less	High	S6
R7	Staff Size	More	High	S7
R8	Staff Size	More	High	S8
R9	Staff Size	Less	Low	S9
R10	Staff Size	Less	Low	S13
R11	Technology	Less	Low	S2,S10
R12	Technology	Moderate	High	S12
R13	Technology	High	High	S11,S13

CHAPTER 4

IMPLEMENTATION DETAILS

4.1 Introduction

The details of each module are described here with the help of sequence diagram of object oriented modeling and design. A sequence diagram is graphical view of scenario that shows object interaction in time based sequence. What happens first, what happens next? Sequence Diagram establishes the roles of objects and help to provide essential information to determine class responsibilities and interfaces. This type of diagram is best used early analysis phases in design because they are simple and easy to comprehend. Sequence Diagram are normally associated with use cases.

4.2 Implementation Steps for Modules

Following detail gives the implementation details of the developed application modules.

4.2.1 Attendance Module

Steps

1. Firstly, Lecturer will be registered by admin.
2. Lecturer will login into App.
3. He will select one of the ways for taking attendance.
4. If ‘Bluetooth attendance’ is selected then all the active Bluetooth will be detected and attendance will be submitted to database.
5. If ‘manual attendance’ is selected then teacher will highlight present roll no & submit attendance to database.

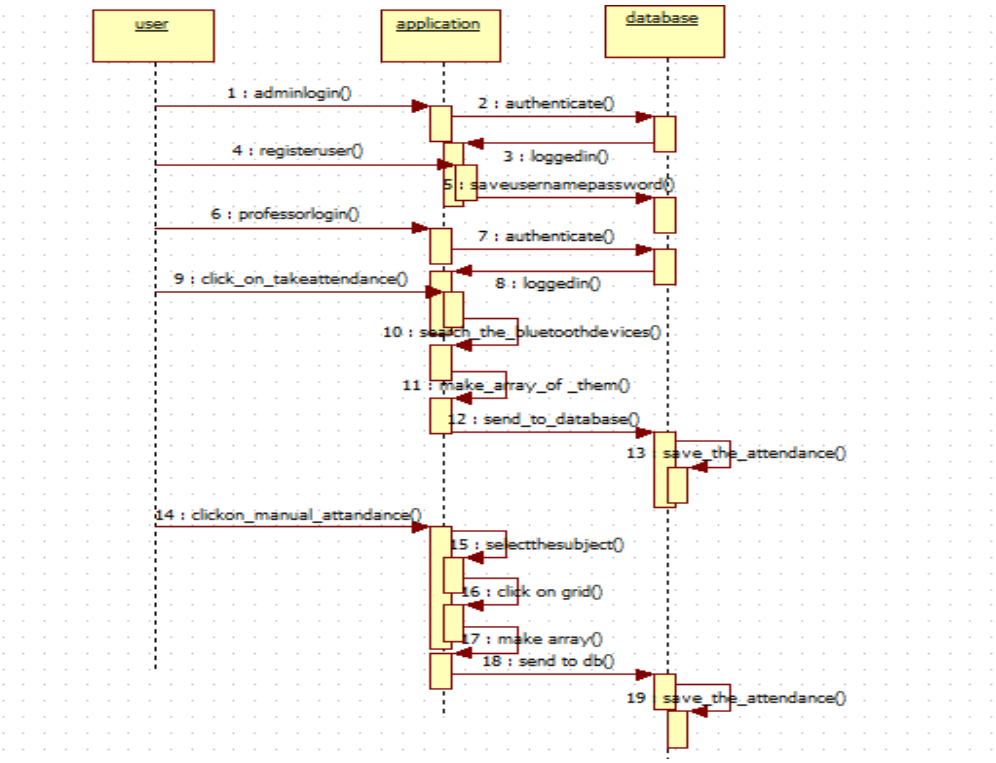


Fig. 4.1 Sequence Diagram for Attendance Module

Code for Attendance Module

```
public void onClick(View arg0) {
        // TODO Auto-generated method stub
        StudentActivity main = (StudentActivity) getActivity();
        Main1.studentLogINFragment();
        Toast.makeText(getActivity(),"You are Logged out Successfully..",
Toast.LENGTH_SHORT).show();
}
});
public void setCurrentDateAndTime() {
Calendar c = Calendar.getInstance();
intmonthNo = c.get(Calendar.MONTH) + 1;
intdayNo = c.get(Calendar.DAY_OF_MONTH);
intyearNo = c.get(Calendar.YEAR);
int hour = c.get(Calendar.HOUR);
```

```
int min = c.get(Calendar.MINUTE);
crntdate.setText("" + dayNo + "-" + monthNo + "-" + yearNo);
yyyy.setText("" + yearNo);
}
ArrayAdapter<String>yearAdapter = new
ArrayAdapter<String>(getActivity(),android.R.layout.simple_spinner_item,subStringArray);
yearAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
subject.setAdapter(yearAdapter);
}
});
}
}).start();
}
public void day_attendance(intdd, EditTextedt)throws NumberFormatException {
if (edt.getText().toString().length() < 1) {
edt.setError("Enter Day");
day = "";
} else if (edt.getText().toString().length() > 2) {
edt.setError("Invalid Day");
day = "";
} else {
if (Integer.parseInt(edt.getText().toString()) <= 0
|| Integer.parseInt(edt.getText().toString()) >dd) {
edt.setError("Enter Valid Day");
day = "";
} else {
day = edt.getText().toString();
}
}
}
publicArrayList<BranchYearInfo>parseJsonSub(String result) {
```

```
ArrayList<BranchYearInfo>mainObj = new ArrayList<BranchYearInfo>();
try {
JSONArraymainMEnu = new JSONArray(result);
for (int i = 0; i <mainMEnu.length(); i++) {
JSONObjectjsonObj = mainMEnu.getJSONObject(i);
BranchYearInfo menu = new BranchYearInfo();
menu.setSubject_id(jsonObj.getString("Subject_id"));
menu.setSubject_name(jsonObj.getString("Subject_Name"));
mainObj.add(menu);
}
} catch (Exception e) {
}
returnmainObj;
}
publicArrayList<BranchYearInfo>parseJson(String result) {
ArrayList<BranchYearInfo>mainObj = new ArrayList<BranchYearInfo>();
try {
JSONArraymainMEnu = new JSONArray(result);
for (int i = 0; i <mainMEnu.length(); i++) {
JSONObjectjsonObj = mainMEnu.getJSONObject(i);
BranchYearInfo menu = new BranchYearInfo();
menu.setSubject_id(jsonObj.getString("Sub_id"));
menu.setBranch_id(jsonObj.getString("Branch_id"));
menu.setYear_id(jsonObj.getString("Year_id"));
mainObj.add(menu);
}
} catch (Exception e) {
}
returnmainObj;
}
@Override
```

```
public void onClick(View arg0) {
// TODO Auto-generated method stub
        switch (arg0.getId()) {
caseR.id.studattence_btnok:
day = dd.getText().toString();
month = mm.getText().toString();
        yr = yyyy.getText().toString();
if (day.equals("") || month.equals("") || yr.equals("")) {
Toast.makeText(getActivity(), "Enter Correct Date..",
Toast.LENGTH_SHORT).show();
} else {
date = yr + "-" + month + "-" + day;
selectedSub = subject.getSelectedItem().toString();
StudentActivity main = (StudentActivity) getActivity();
sidBundl = main.getloginStudentInfoBundle();
sid = sidBundl.getString("StudID");
finalStudentInfoNetworkobj = new StudentInfoNetwork();
new Thread(new Runnable() {
public void run() {
System.out.println("sid=" + sid + " date=" + date+ " selectedSub=" + selectedSub);
studAttStatus = obj.getStudentStatus(sid, date,selectedSub);
getActivity().runOnUiThread(new Runnable() {
public void run() {
ArrayList<StudentInfo>mainInfo = parseJsonATTendance(studAttStatus);
Iterator itr = mainInfo.iterator();
while (itr.hasNext()) {
StudentInfo info = (StudentInfo) itr.next();
attendance = info.getStatus();
}
System.out.println("DDD" +attendance);
if (attendance.equals("")) {
```

```
attendnce.setText("Not Submited");
} else {
attendnce.setText(attendance);
attendance = "";
}
});
}
}).start();
}
```

4.2.2 Profile Module

Steps

1. Lecturer, student & Admin will login into their account.
2. If student is logged in then his complete profile will appear on screen.
3. If professor is logged in then professor will click on button “student info”.
4. Professor will then select exam-id and student-id & then selected students profile will be shown.
5. If admin will logged in, then admin will click on button “professor info” or “student info”.
6. Admin will select professor id or student id and selected students or professors profile will appear.

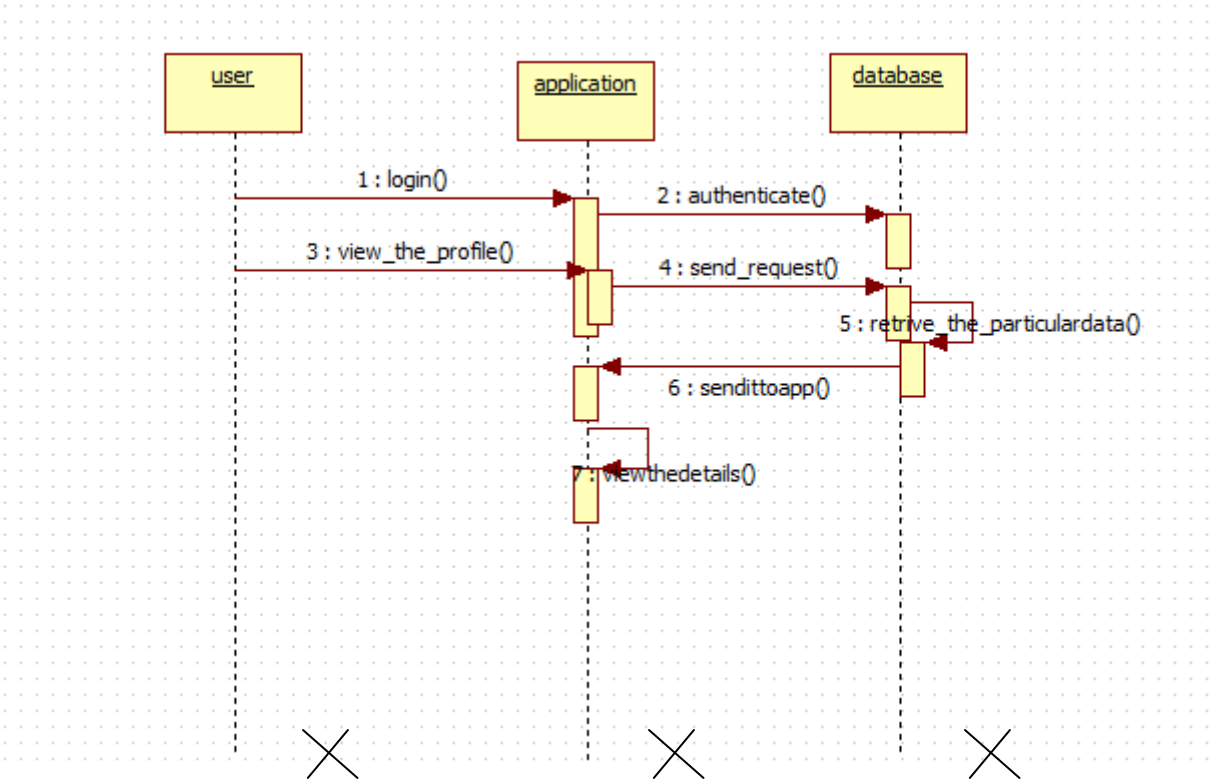


Fig. 4.2 Sequence Diagram for Profile Module

Code for Profile Module

```
public String getAdminid_Reg() {
        returnadminid_Reg;
}

public void setAdminid_Reg(String adminid_Reg) {
        this.adminid_Reg = adminid_Reg;
}
String Adminlogin;
String Lectlogin;
String Studlogin;
public String getAdminlogin() {
        returnAdminlogin;
}
```

```
public void setAdminlogin(String adminlogin) {
        Adminlogin = adminlogin;
}

public String getLectlogin() {
        returnLectlogin;
}

public void setLectlogin(String lectlogin) {
        Lectlogin = lectlogin;
}

public String getStudlogin() {
        returnStudlogin;
}

public void setStudlogin(String studlogin) {
        Studlogin = studlogin;
}

public String getSub_id() {
        returnsub_id;
}

public String getYear_id() {
        returnyear_id;
}

public void setYear_id(String year_id) {
        this.year_id = year_id;
```

```
}

public String getBranch_id() {
        returnbranch_id;
}

public void setBranch_id(String branch_id) {
        this.branch_id = branch_id;
}

public void setSub_id(String sub_id) {
        this.sub_id = sub_id;
}

public String getSubjectname() {
        returnsubjectname;
}

public void setSubjectname(String subjectname) {
        this.subjectname = subjectname;
}

public String getYearname() {
        returnyearname;
}

public void setYearname(String yearname) {
        this.yearname = yearname;
}

publicintgetAdminId() {
```

```
        returnadminId;
    }

    public void setAdminId(intadminId) {
        this.adminId = adminId;
    }

    public String getAdmin_Fname() {
        returnadmin_Fname;
    }

    public void setAdmin_Fname(String admin_Fname) {
        this.admin_Fname = admin_Fname;
    }

    public String getAdmin_Lname() {
        returnadmin_Lname;
    }

    public void setAdmin_Lname(String admin_Lname) {
        this.admin_Lname = admin_Lname;
    }

    public String getAdmin_City() {
        returnadmin_City;
    }

    public void setAdmin_City(String admin_City) {
        this.admin_City = admin_City;
    }
```

```
        public String getAdmin_LogId() {
                returnadmin_LogId;
        }

        public void setAdmin_LogId(String admin_LogId) {
                this.admin_LogId = admin_LogId;
        }

        public String getAdmin_Password() {
                returnadmin_Password;
        }

        public void setAdmin_Password(String admin_Password) {
                this.admin_Password = admin_Password;
        }

        public String getBranchname() {
                returnbranchname;
        }

        public void setBranchname(String branchname) {
                this.branchname = branchname;
        }
}
```

4.2.3 SMS Module

Steps

1. Professor will login into App.
2. Professor will click button “send SMS”.
3. Professor will enter mobile number to whom SMS to be send.
4. Then he will type text and send the SMS.

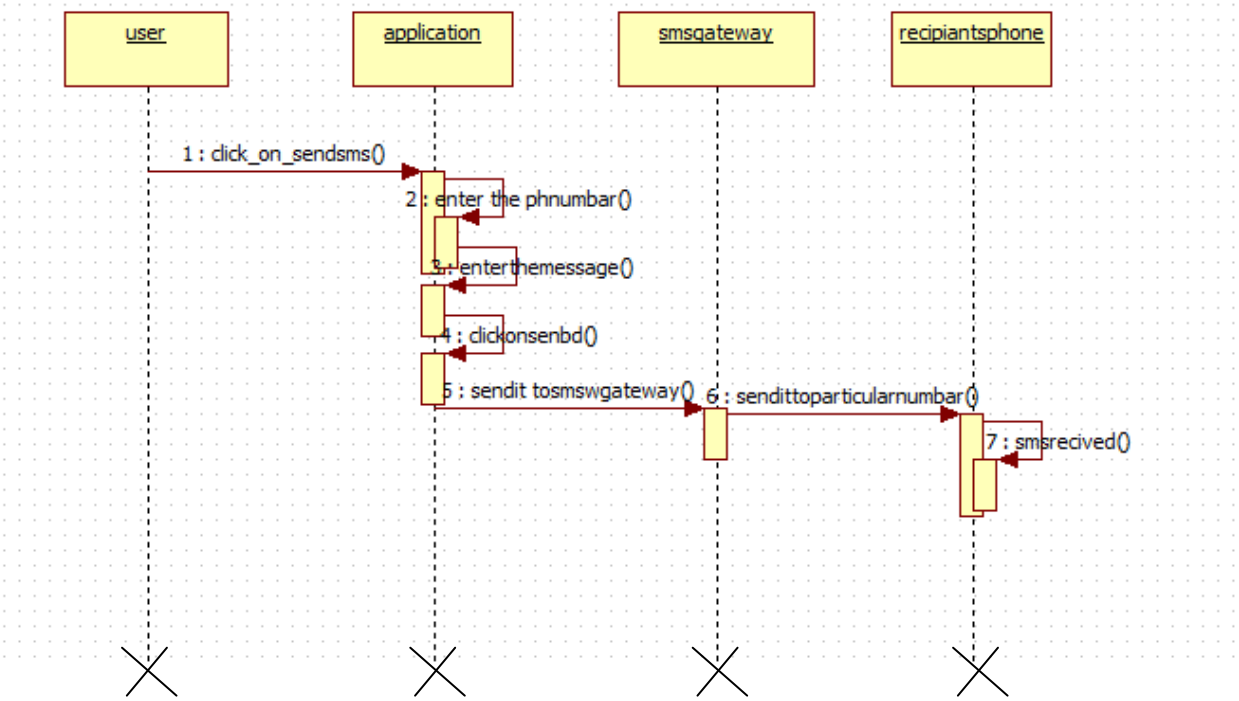


Fig. 4.3 Sequence Diagram for SMS Module

Code for SMS Module

```
public class SendSms extends Activity{


        EditTextmobileno;
        EditText message;
        Button send;
        String num;


        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.sendsms);
        mobileno=(EditText)findViewById(R.id.mobileno);
        message=(EditText)findViewById(R.id.message);
        send=(Button)findViewById(R.id.send);
```

```
mobileno.addTextChangedListener(new TextWatcher() {
@Override
public void onTextChanged(CharSequence arg0, int arg1, int arg2, int arg3) {
// TODO Auto-generated method stub
}
@Override
public void beforeTextChanged(CharSequence arg0, int arg1, int arg2,int arg3) {
// TODO Auto-generated method stub
}
@Override
public void afterTextChanged(Editable arg0) {
// TODO Auto-generated method stub
contact_valid(mobileno);
}
});
send.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
String mobile=mobileno.getText().toString();
String msg=message.getText().toString();
try {
SmsManagersmsManager=SmsManager.getDefault();
smsManager.sendTextMessage(mobile, null, msg, null, null);
Toast.makeText(getApplicationContext(), "SMS Sent!!!", Toast.LENGTH_SHORT).show();
} catch (Exception e) {
Toast.makeText(getApplicationContext(),"SmsSend Fail Try again
later!!!",Toast.LENGTH_SHORT).show();
}
}
});
}
```

```
@Override
publicbooleanonCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.main, menu);
return true;
}
private void contact_valid(EditTextedt) {
// TODO Auto-generated method stub
if (edt.getText().toString().length() < 10) {
edt.setError("Enter Correct Number");
num = "";
} else if (edt.getText().toString().length() > 10) {
edt.setError("Invalid Mobile Number");
num = "";
} else if(edt.getText().toString().equals(" "))
{
edt.setError("Please Enter Mobile Number");
num= "";
}else {
num = edt.getText().toString();
}
}
}
```

4.1.4 Email Module

Steps

1. Professor will login into App.
2. Professor will click button “send Email”.
3. Then Professor will select synched account.
4. Once the professor is logged in into his synched account, he will send email thro’ his account with attachments if any.

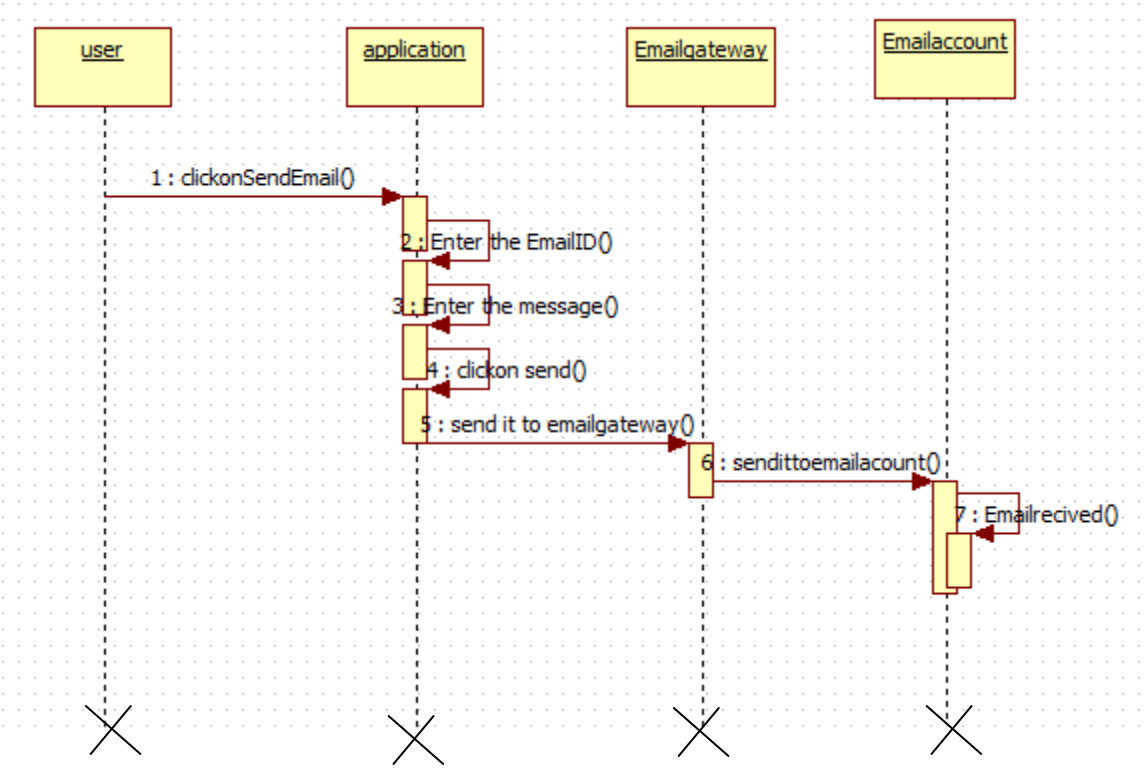


Fig. 4.4 Sequence diagram for email Module

Code for Email Module

```
public class SendEmail extends Activity
{
        Button attachment;
        Button button;
        EditTextdestinationAddress;
        EditTextsbj;
        EditTextmessageBody;
        private static final int PICK_FROM_GALLERY = 101;
        protected static final File FOLDER_NAME = null;
        protected static final int SELECT_PICTURE = 1;
intcolumnIndex;
   String email, subject, message, attachmentFile;
   Uri URI = null;
```

```
String strMessageBody;
String subjects;
String to;
    FileOutputStreamfout = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.sendemail_activity);

        sbj = (EditText) findViewById(R.id.subject);
        messageBody = (EditText) findViewById(R.id.messageBody);
        button = (Button) findViewById(R.id.button);
        attachment=(Button)findViewById(R.id.attach);
        destinationAddress = (EditText) findViewById(R.id.destinationAddress);

        //createFile();

        attachment.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {

                    openGallery();
            }
            });
        button.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View view) {
```

```
String subject = sbj.getText().toString();
String message = messageBody.getText().toString();
String to = destinationAddress.getText().toString();
Intent emailActivity = new Intent(Intent.ACTION_SEND);
emailActivity.putExtra(Intent.EXTRA_EMAIL, new String[] { to });
emailActivity.putExtra(Intent.EXTRA_SUBJECT, subject);
if (URI != null) {
emailActivity.putExtra(Intent.EXTRA_STREAM, URI);
            }
emailActivity.putExtra(android.content.Intent.EXTRA_TEXT, message);
startActivity(Intent.createChooser(emailActivity, "Sending email..."));
emailActivity.putExtra(Intent.EXTRA_TEXT, message);
emailActivity.setType("message/rfc822");
startActivity(Intent.createChooser(emailActivity, "Select your Email Provider :"));
Toast.makeText(getApplicationContext(), "Message send sucessfully to mail_id:"+to,
Toast.LENGTH_SHORT).show();
}
});
}
        protected void onActivityResult(intrequestCode, intresultCode, Intent data) {
if (requestCode == PICK_FROM_GALLERY &&resultCode == RESULT_OK) {
            /**
             * Get Path
             */
            Uri selectedImage = data.getData();
String[] filePathColumn = { MediaStore.Images.Media.DATA };
Cursor cursor = getContentResolver().query(selectedImage,filePathColumn, null, null, null);
cursor.moveToFirst();
columnIndex = cursor.getColumnIndex(filePathColumn[0]);
attachmentFile = cursor.getString(columnIndex);
Log.e("Attachment Path:", attachmentFile);
```

```
 URI = Uri.parse("file://" + attachmentFile);
cursor.close();
}
 }
public void openGallery() {
 Intent intent = new Intent();
intent.setType("image/*");
intent.setAction(Intent.ACTION_GET_CONTENT);
intent.putExtra("return-data", true);
startActivityForResult(
Intent.createChooser(intent, "Complete action using"),
   PICK_FROM_GALLERY);
}
}
```

CHAPTER 5

TESTING AND ANALYSIS

Testing is an integral part of any system or project. The various objectives of Testing:

1. To uncover the errors in function logic or implementation for the software.
2. To verify that software needs the specific requirement.
3. To verify that software has been implemented according to the predefined standard.

5.1 Unit Testing

Here we have performed module level testing, checking for each input to be tested. In computer programming, unit testing is a procedure used to validate that individual units of the source code are working properly. A Unit is the smallest testable part of an application. In procedural programming, a unit may be an individual program, function, procedure etc. While in Object Oriented Programming, the smallest unit is a method, which may belong to a base or super class, abstract class or derived/child class.

Ideally, each test case is independent from others; mock or fake objects as well as test harness can be used to assist testing a module in isolation. Unit testing is typically done by developers and not by software testers or end users.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a script, return contract that the piece of code must satisfy. As a result, affords several benefits.

Unit testing allows the programmer to modify code at a later date and make sure the module still works correctly (i.e. Regression Testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault it can be quickly identified and fixed.

Readily available unit test make it easy for the programmer to check whether a piece of code still working properly. Good unit test design produces test cases that cover all paths through the unit with attention paid to look conditions.

In continuous unit testing environment, through the inherent practice of sustained maintenance, unit tests will continue to accurately reflect the intended use of the executable and code in the phase of any change. Depending upon established development practices and unit test coverage, up-to-the-second accuracy can be maintained.

Unit testing helps to eliminate uncertainty in the units themselves and can be used in a bottom up testing style approach. By testing the parts of a program first and then testing the sum of its paths, integration testing becomes much easier.

5.2 Integration Testing

Integration testing is the phase of software testing in which individual software module are combined and tested as a group. It follows unit testing and precedes system testing.

Integration testing takes as it input modules that have been unit tested, groups them in larger aggregates, applies test defined in an integration test plan to those aggregates and delivers as its output the integrated system ready for system testing.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. This “design items”, i.e. assemblages(or group of units) are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameters and data inputs. Simulated usage of shared data areas and inter process communication is tested and individual sub systems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure or process activation and this is done after testing individual modules i.e. unit testing.

5.3 Test Cases

For the developed system, the test case are generated and tested manually. The details about module wise test cases are given in table 5.1.

Table 5.1 Unit Test Cases

Test ID	Description	Expected Result	Actual Result	Status
TC01	Profile module checking	All the details of registered member if name is correct	Profile displayed	Pass
TC02	SMS module checking	SMS to specified no. should be sent	SMS sent to specified no	Pass
TC03	Email Module checking	Email should be sent to specified user	Email sent	Pass
TC 04	Attendance module checking	Active Bluetooth devices should be detected and attendance with sub, date & timing should be sent to server	Active Bluetooth devices detected and attendance with date, time & sub is stored to server	Pass
TC05	Registration, log in & profile activity checking	Each user should get registered by admin only and all the entries should be made in relevant tables after registration	All users are registered by admin only and entries are made in relevant tables	Pass
TC06	Attendance and profile module integration	Attendance should be synched with profile with date and time	Attendance is synched with profile	Pass
TC07	SMS	SMS should not be sent if balance is not sufficient.	SMS is not sent	Fail
TC08	Email	Email should not be sent if internet connection is not available.	Email is not sent	Fail

Table 5.2 Integration Test Cases

Test Id	description	Expected result	Actual Result	status
TC1	Registration ,login & profile activity checking	Each user should get registered by admin only .user will login by username and password given by admin only	Registrations are done by admin & after Registration all the entries are made in relevant tables and those changes are reflected in profile of user. In the app each user can login and perform his or her task.	pass
TC2	Attendance and profile module checking	After submitting attendance by professor it should be saved to database with date, time, professor name and subject	After submitting the attendance we get the all required field present at the server database	pass
TC3	Report generation	At the server side the report should get generated with total attendance and out of attendance record with each students details	We are getting the reports like total attendance out of attendance and average attendance with students details	pass

5.4 Result Analysis

We have tested this application in the unit as well as integration testing and got the following results:-

Attendance

We are getting the attendance record present at the server side and we have generated the results at the server side according to query fired on report table .For each and every individual subject ,date ,time, and student we are having the record present at the database.

Profile

After admin have registered the user with his or her personal details the user can login to the app and can view his or her details any time and admin can view the details of each and every user by app .Professor can view the details of each student. Students can view his marks as well as the attendance of particular day.

SMS

Professors have to type the phone number and message to be sent and clicking on ok then we are getting the SMS received at the specified phone number. The SMS will be sent through the phone number of the professor and professor will be charged for that particular SMS.

Email

Same like SMS the professor type the Email id of the recipients and message to be send and then whichever the email account sink with the phone the Email will get send by that email account only. And recipients will get the Email.

5.5 Performance Analysis

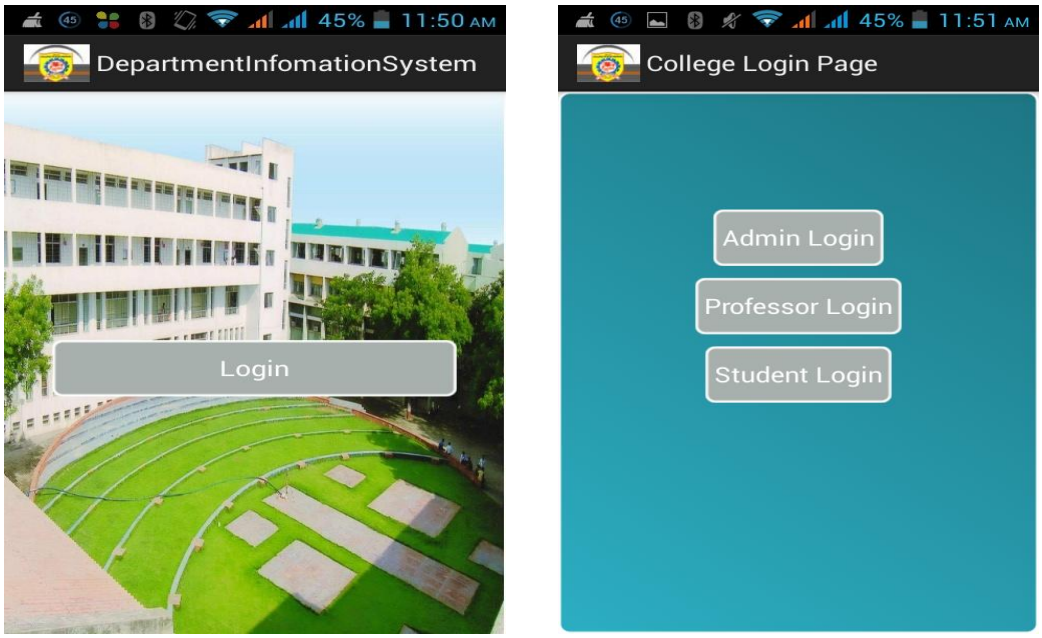
We have compared our system with some existing systems like E-beat; Mobile based attendance system using J2ME. The performance of all systems are summarized as below in Table 5.2.

Table 5.3 Performance Analysis

Sr.No	Parameters	E-beat	Mobile based attendance on J2ME	Developed Application
1	Memory Usage	Very small	Medium	As per user requirement
2	Temperature affect	$70^{O}C$	NOT Applicable	NOT Applicable
3	Operating System	Electronic device	J2ME	Android
4	Time requirement	More	medium	Less
5	Use of GPRS	No	Yes	Yes

5.6 Snapshots

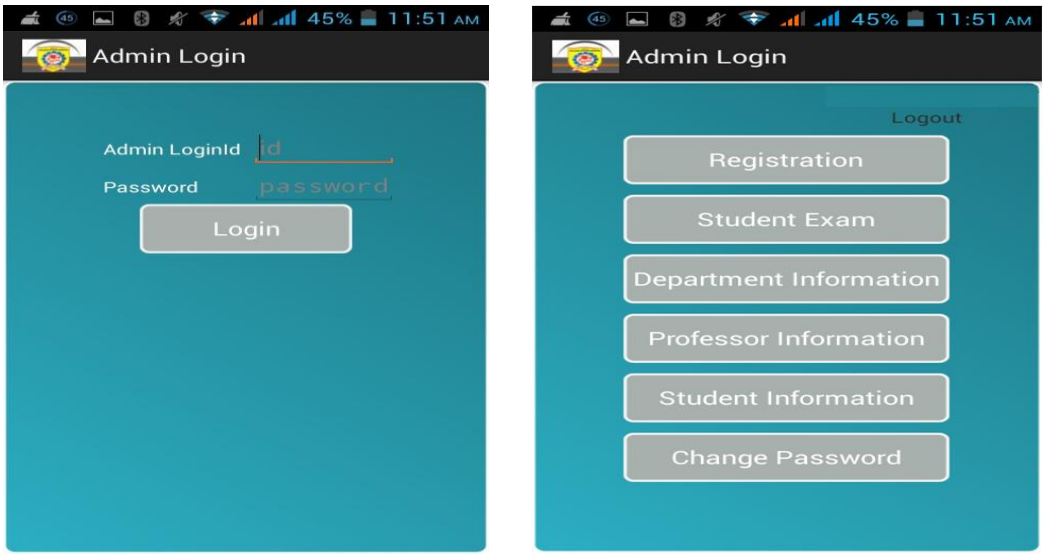
The following figure 5.1(a) shows the initial GUI for DIS Developed system. After clicking 'login' button it goes into Fig 5.1(b) tree levels of login i.e. Admin , Professor , student login.



(a) Initial GUI **(b) User Login**

Fig. 5.1 Initial GUI

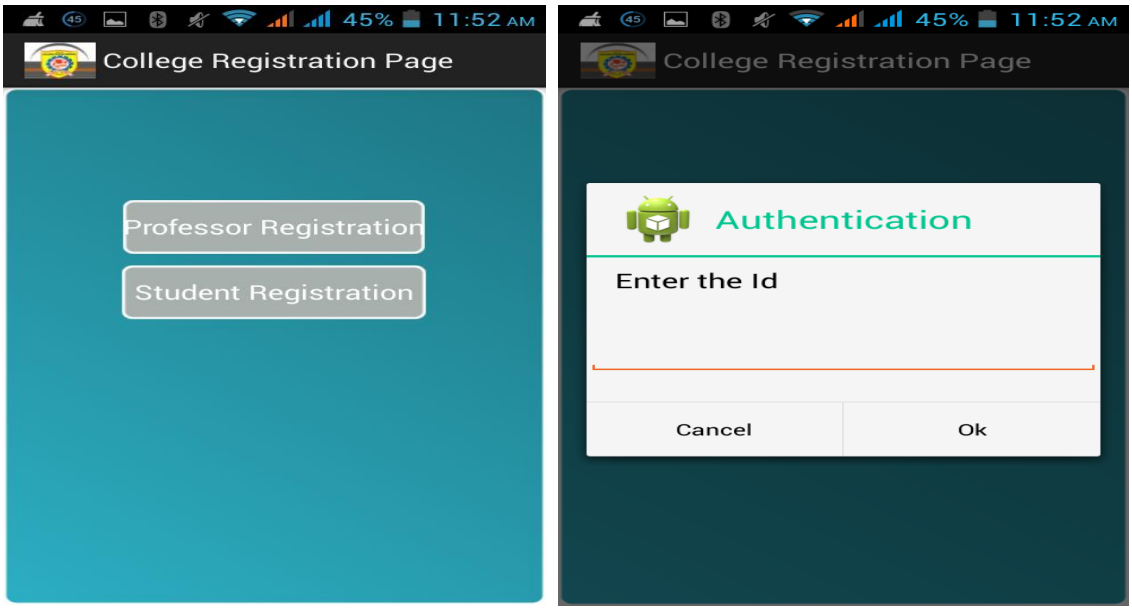
After clicking ‘admin login’, admin will enter his/her username and password as shown in Fig 5.2(a). After the login, Admin can perform different tasks like Registration, profile view, Department, Professor, Student information manipulation etc. as shown in fig 5.2(b).



(a)Before Admin Login **(b) After Admin Login**

Fig. 5.2 Admin Login Details

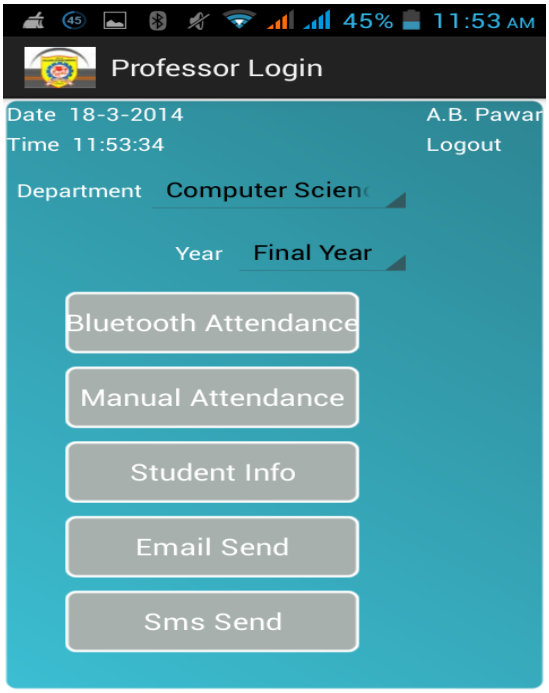
Admin will register professor as well as student as shown in Fig 5.3(a) and using authentication window system will authenticate the admin as shown in Fig 5.3(b).



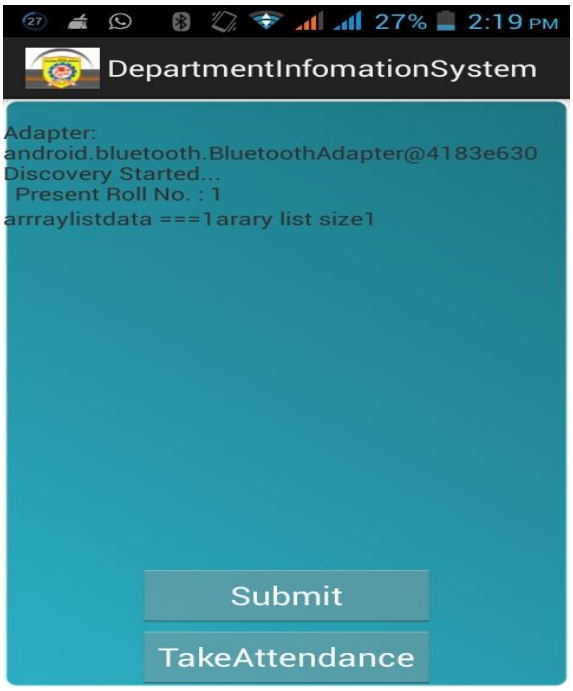
(a) Registration **(b)Authentication**

Fig. 5.3 Registration Details

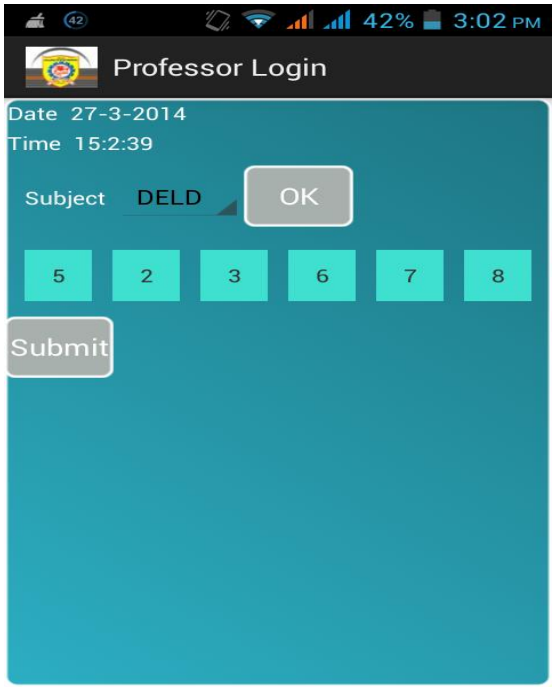
Same like admin login, when professor will log in, professor can view number of buttons as shown in Fig 5.4(a) and how Bluetooth attendance will be taken is shown in Fig 5.4(b) and also manual attendance is shown in Fig 5.4(c).



(a) Professor Login



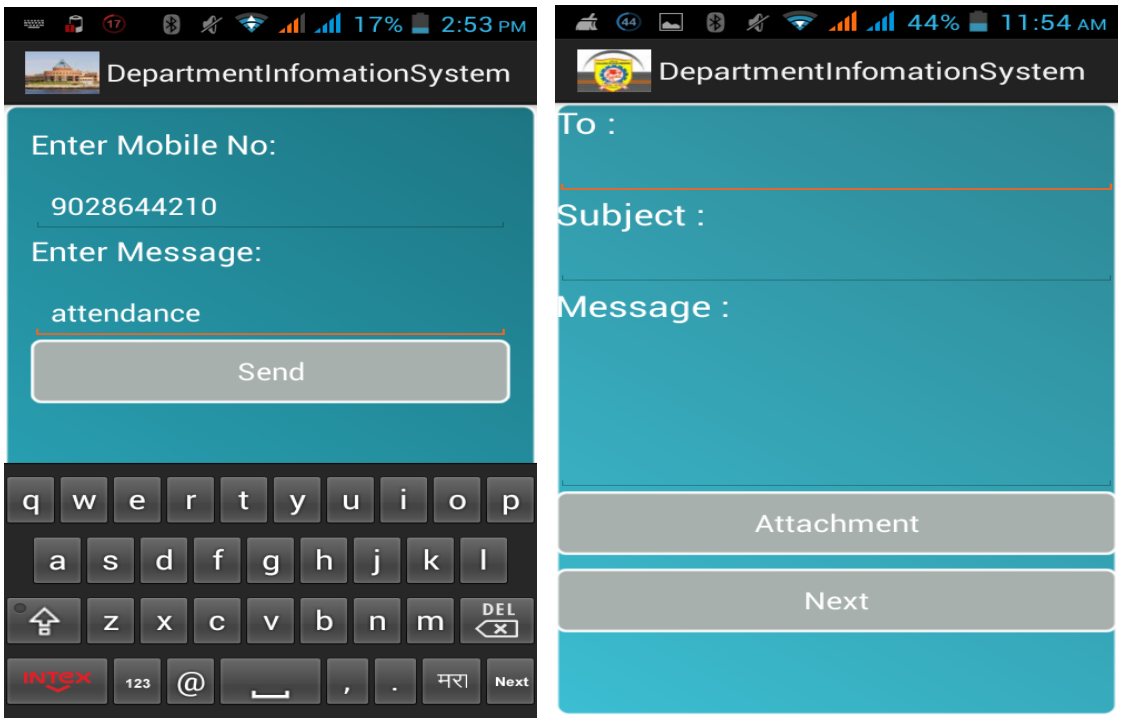
(b) Attendance



(c) Manual Attendance

Fig. 5.4 Professor Task Details

The notifications are sent by professors via SMS and Email are shown in Fig 5.5(a) and Fig 5.5(b) respectively.



(a) Sending SMS **(b) Sending Email**

Fig. 5.5 Notification Details

Each user can view his/her profile using this app and admin can view all profiles as shown in fig.5.12

Fig. 5.6 Profile View

The attendance and profile records are stored at the server Fig 5.7shows how admin login to server using username and Password.

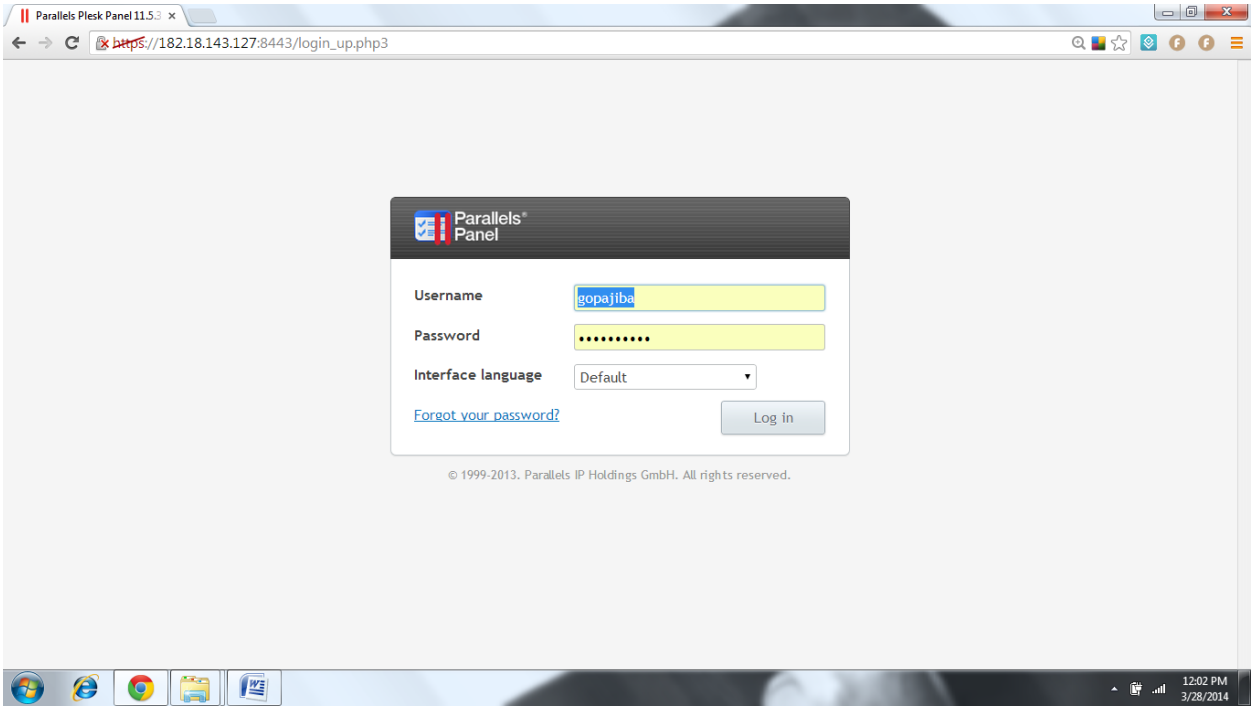


Fig. 5.7 Login to the DB

After authentication, the final report of the attendance is shown at the server and it is shown in Fig 5.8.

Sr.No	Stud_id	Roll_No	First_Name	Last_Name	Branch	Year	total_attendance	out_of_attendance	average
1	1	1	shafaq	soparkar	civil	2nd year	5	30	60
2	2	2	Amruta	Kulkarni	computer	final year	18	30	60
3	3	3	Prachi	Kulkarni	computer	final year	15	30	50
4	4	4	Ajinkya	Rode	computer	final year	20	30	66
5	5	5	Darshana	Kadam	computer	final year	22	30	73
6	6	6	Akshada	Deshmukh	computer	final year	22	30	73
7	7	7	Bhagyashree	Gaikwad	computer	final year	18	30	60
8	8	8	Chaudhari	Shweta	computer	final year	18	30	60
9	9	9	Sagar	pawal	computer	final year	15	30	50
10	10	10	Devendra	pawar	computer	2nd year	18	30	60
11	11	11	Sarang	Patil	computer	2nd year	20	30	66
12	12	12	Payal	Gaikwad	electronics and tele	final year	12	30	40
12	12	12	Payal	Gaikwad	e & tc	final year	18	30	60
13	13	13	Aditi	Wable	e & tc	final year	19	30	63.33

Fig. 5.8 Final Reports at the DB

CHAPTER 6

APPLICATIONS AND FUTURE SCOPE

This system is completely automatic and market ready for use we just need to make changes according to stakeholders need. Every system has its merits and de-merits. So, in the same sense, for the developed system, following advantages and disadvantages are identified.

6.1 Advantages

- Time saving as attendance and all other work can be done just by checking the list of students and making it automated way.
- No needs to maintain several separate records and manual calculations.
- 24x7 availability of information.
- More secured than traditional mobile attendance system.
- A step towards the futuristic e-schools
- Priority levels of login for each user.
- Each user having his own profile.

6.2 Disadvantage

- Bluetooth name of the phone should be the roll number of the student.
- Bluetooth range of any phone is limited about 10 meters.
- Fake attendance by Bluetooth attendance cannot be eliminated.
- For submitting attendance to server we need GPRS.
- For using this app we need high speed internet or 3G network.
- Any android app can get unfortunately stop when it is used so fast.

6.3 Applications

- Small institutes who don't afford the server maintenance and still wants the ERP system. Those small institutes can use this system and can get benefits of traditional ERP systems.
- Teachers can take attendance manually and via Bluetooth devices also so this will reduce their workload and time consumed for this task to be done manually.
- Professors can send notifications using SMS and Email.

- Each & every user of this app has his own profile he or she can view his or her profile.
- Student can be in with their attendance status through this app.
- This is the one step towards the future ERP system.

6.4 Future Scope

- This application is one forward step towards the future ERP systems which are going to be online and on the Android platform.
- There is one application in market which gives takes picture of class student and after applying the image processing algorithm the attendance can be taken automatically.
- Future of ERP system is also the android apps.
- In future, we can add each and every module which includes in ERP system.
- We can have website interface for the same application.

CHAPTER 7

CONCLUSION

This project is automizing the manual work at departmental level and provides paperless environment and also help the user of this system to reduce their workload by reducing the time and calculations required to do their tasks like to take & update the attendance, to calculate the marks of regular assessment, to circulate the notices of events, to maintain profile details of each individual staff, students & associated personnel. Even stakeholders can use this system to get details about department. The system is definitely be effective in today's modern techno-savvy educational system.

REFERENCES

[1] Shraddha S. Chawhan, Mangesh P. Girhale and GunjanMankar, "Mobile Phone Based Attendance System" ,In IOSR Journal of Computer Engineering (IOSR-JCE), Mar. - Apr. 2013, Available:www.iosrjournals.org , Access Date:15/7/2013 .

[2] Benjamin Speckmann, "The Android mobile platform", In A Review Paper Submitted to the Eastern Michigan University Department of Computer Science , April 2008, Access date: 1/7/2013.

[3] Karan Balkar, Reyomi Roy and PreeyankPable, "A Mobile Application to Access Remote Database using Web Services" , In NCNTE-2012,Third Biennial National Conference on Nascent Technologies, Access date 2/7/2013 .

[4] Bernhard J. Berger, Michaela Bunke, and KarstenSohr ,"An Android Bluetooth Case Study", Available :fberberjmbunkejsohrg@tzi.de, 2012, Access Date:31/7/2013.

[5] Jaya Bharathichintalapati and SrinivasaRao T.Y.S "Remote computer access through Android mobiles", In IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012, Access Date:2/8/2013

[6] Dr. Khanna Samrat and Vivekanand Omprakash " Accessing information on mobile client from mobile & web server with internet from remote place", In International Journal of Advanced Engineering Technology E-ISSN 0976-3945, 2012, Access date: 15/7/2013.

[7] J.F. DiMarzio "Android – a programmers guide", book by The McGraw-Hill Companies ,Edition 2008.