

SHIVAJI UNIVERSITY

Attendance Monitoring System

by

Bachelor of Engineering

A thesis submitted in partial fulfillment for the
degree of

in the

Department of Computer Science and Engineering
Sharad Institute of Technology College of Engineering, Yadrav-Ichalkaranji

April 2015

SHIVAJI UNIVERSITY

Abstract

Department of Computer Science and Engineering

Sharad Institute of Technology College of Engineering, Yadrav-Ichalkaranji

Bachelor of Engineering

Attendance Monitoring System is a simple smart phone based attendance system that specifically developed for school, colleges and companies. The software application can manage the recordings, controlling and monitoring of student absences. The purpose is to make sure that the students are punctual and do their present on time. Currently, there is no proper system to monitor the student attendance at some school or college. Besides, the colleges still use the paper-based system to store the records of the student. With the implementation of this system, paper-based system will be eliminated. This system can save time and minimize the manpower for manual management. The administrators can easily trace the attendance of the student compare to manual paper recording and file keeping system. Besides, the students records are more secure which are saved into the database. This system is also helps to reduce clerical cost such as papers, files and stationery. As for the Attendance Monitoring System, will be used as the project methodology. Besides that, requirement analysis tends to be more through and better documented in the model-driven approach. The system can be used by the system's administrator such as teacher and the student of the company. Each of users has their own interface through the system login. There are some of the modules included in the developed system such as admin login, student login, registration of student, make attendance or take attendance and get report. As a conclusion, the proposed system is able to help the administrator to manage recordings, monitoring and tracking the attendance of the student. It is also providing an accurate time management for the student in order to marking their attendance.

Acknowledgements

We take this opportunity to express our profound gratitude and deep regards to our guide Miss.P.A.Bagane for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to our Principal, Dr. S. A. Khot and our H.O.D., Prof. O.D.Joshi for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are obliged to all faculty and staff members of all departments of our Sharad Institute of Technology College of Engineering, Yadav for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment.

Lastly, we thank almighty, our parents, brothers, sisters, friends and our colleagues for their constant encouragement without which this assignment would not be possible.

Miss Desai Priyanka Suresh

Miss Patil Arati Jagonda

Miss Gaikwad Supriya Ganapati

Miss Rajput Pritidevi Vijaysingh

Miss Chougule Manjusha Sukumar

Contents

Abstract	i
Acknowledgements	ii
List of Figures	vi
List of Tables	vii
1 INTRODUCTION	1
1.1 Introduction of project	1
1.1.1 Student requirements:	2
1.1.2 Teaching Staff Requirements:	2
1.1.3 Administrator requirements:	3
1.2 In our system the steps are:-	3
1.3 Advantages:	4
1.4 Application:	5
1.4.1 Residential Use-	5
1.4.2 Voter verifications-	5
1.4.3 Banking using ATM-	5
1.4.4 Airport Security-	5
1.4.5 Financial Use-	5
2 LITERATURE REVIEW	6
2.1 Working of Existing System:	6
2.1.1 Disadvantages of Present Working System:	7
2.2 Proposed Plan of Work:	7
2.2.1 Characteristics of Proposed System:	8
3 OBJECTIVE AND SCOPE	9
3.1 Problem Statement:	9
3.2 Objective:	9
3.3 Scope:	10
3.4 Out of Scope:	10
4 REQUIREMENT ANALYSIS	11
4.1 Software Requirements:	11
4.1.1 Software requirement while developing system:	11

4.1.2	Software requirement while using system:	11
4.2	Hardware Requirements:	11
4.2.1	Hardware requirement while developing system:	11
4.2.2	Hardware requirement while using system:	11
4.3	Required softwares:	12
4.3.1	Android Operating System:	12
4.3.2	Eclipse:	12
4.3.3	JDK :	13
4.3.4	SDK:	13
4.3.5	PHP :	13
5	SYSTEM DESIGN	15
5.1	Module:	15
5.1.1	Register Student:	15
5.1.2	Login Student:	16
5.1.3	Authentication:	16
5.1.4	Mark Attendance:	16
5.1.5	Get Report:	16
5.2	An Overview of UML:	16
5.3	Goals of UML:	17
5.4	A Conceptual Model of UML:	17
5.4.1	Building Blocks of UML:	18
5.5	Diagrams in UML:	18
5.6	Use Case Diagram:	19
5.6.1	Description:	19
5.6.1.1	Contents	19
5.6.2	Use-Case Scenario:	21
5.7	Sequence Diagram:	22
5.7.1	Contents:	22
5.7.2	Description:	22
5.8	Class Diagram:	27
5.8.1	Contents:	27
5.8.2	Definition and Common Uses:	27
5.8.3	Description:	28
5.9	Deployment Diagram:	29
5.9.1	Definition:	29
6	CODING	31
6.1	College Form:	31
6.2	MultiOption Form:	33
6.3	Register form:	34
6.4	Login:	36
6.5	Face Detection Activity:	38
6.6	Get Report:	40
7	TESTING	41
7.1	What is Software Testing:	41

7.2	Test methods:	42
7.2.1	Black box testing:	42
7.2.2	White Box Testing:	42
7.3	Test cases and Test data:	43
8	DEPLOYMENT	49
8.1	Welcome form:	49
8.2	College ID Form	50
8.3	Authentication Form	50
8.4	Teacher Login-1	51
8.5	Teacher Login-2	51
8.6	Student Module	52
8.7	Student Registration Form	52
8.8	Student Login Form	53
8.9	Student Login Form-1	53
8.10	Face Detection	54
8.11	Face Recognition-1	54
8.12	Face Recognition-2	55
8.13	Attendance Sheet	55
8.14	Get Report	56
9	CONCLUSION	57
9.1	Conclusion:	57
	Bibliography	58

List of Figures

5.1	Use Cases	19
5.2	Dependencies	20
5.3	Use Case Diagram	20
5.4	Student Registration Diagram	23
5.5	Student Login	24
5.6	Mark Attendance Sheet	25
5.7	Get Report	26
5.8	Class Diagram	28
5.9	Deployment Diagram	30
8.1	Welcome Form	49
8.2	College ID Form	50
8.3	Authentication Form	50
8.4	Teacher Login-1	51
8.5	Teacher Login-2	51
8.6	Student Module	52
8.7	Student Registration Form	52
8.8	Student Login Form.	53
8.9	Student Login Form-1	53
8.10	Face Detection	54
8.11	Face Recognition-1	54
8.12	Face Recognition-2	55
8.13	Attendance Sheet	55
8.14	Get Report	56

List of Tables

5.1 Use Case Scenario	21
7.1 Test Cases	44

Chapter 1

INTRODUCTION

1.1 Introduction of project

Attendance systems of old practises are not quite efficient today, for keeping track on students attendance. Due to the availability of large resources over the internet today, it is very hard to motivate the students to attend lectures without fail have become more challenging. In order to drag the attention of students and make them interactive in observing technologies, we move on to latest upcoming trends on developing attendance systems.[1]This is the strong reason for college attendance monitoring system, has to come up with an approach that ensures strong contribution of students in classrooms.

Person identification is one of the most crucial building blocks for smart interactions. Among the person identification methods, face recognition is known to be the most natural ones, since the face modality is the modality that uses to identify people in everyday lives. Basically face recognition research is aimed for implementing a system that is capable of identifying the employees in an organization and the students in schools and colleges, marking their attendance. Therefore face recognition is used to mark the attendance of the student. [2]

Face recognition as it is often referred to as, analyses characteristics of a person's face image input through a camera. The images are retained in a database and used as a comparison when a user stands before the camera. One of the strongest positive aspects of facial recognition is that it is non-intrusive. Verification or identification can be Accomplished from two feet away or more, without requiring the user to wait for long periods of time.[1]

We are developing the Attendance monitoring system using face recognition technique. The camera image is used to analyze face. The images of the students in that class

are stored in the database. The stored image has the Students information such as the students roll number, Student name, class, branch, year etc. these are used for comparison. The face images are detected from the captured image and recognized. Every teacher gets an username and password to login into the system and then they can take the attendance for that subjects which are allocated to them.[3]

There are two important stages. First web camera takes image, detect the face and then compare with the stored or reference images. And second stage is if image will matched the attendance of student is marked.

The face detection methods focus on detecting frontal faces with good lighting conditions. Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies. [4]

Face Recognition includes feature extraction, where important information for discrimination is saved, and the matching, where the recognition result is given with the aid of a face database. the facial recognition may has several advantages : it is natural, easy to use and does not require aid from the test subject. Because the face detection and recognition database is a collection of images and automatic face recognition system should work with these images. [3]

Analyzing user requirements and needs is a vital task in any system development process. End users must be the main concern of the system designer in order to produce a valid, useful and user-satisfying system. This section examines and analyzes the requirements and needs of the possible different system end users. [2]

1.1.1 Student requirements:

The student needs to keep track of his attendance. The student must be registered into the system. This would require him to login using his ID and password to the system. The system will accept him if his ID and password are the same as the ones saved in the database.

1.1.2 Teaching Staff Requirements:

The teaching staff needs an efficient and reliable automated system for recording the students attendance during lectures, sections, labs. The teaching staff needs to keep track of their courses and the students attendance in these courses.

1.1.3 Administrator requirements:

The administrator should be able to enter the all the users (students, lecturer and teaching assistants) information and creates IDs and passwords for them to access the system.

1.2 In our system the steps are:-

1) REGISTRATION

The First step of the system, actually deals with registering the information of the student in a particular classroom. The information includes-

- Name of the student
- Roll number of the student
- Branch of the student
- Year of studying
- Semester
- Password

These details are placed in the database from which the actual comparison will be done.

2) IMAGE CAPTURING

The camera is used for capturing the images of the student which will be in active mode during the hours of college. The use of camera is that it is capable of capture the image of high quality and also at different angles view. [4]

3) IDENTIFICATION

To identify the student image, tablet which holds the image database of the student, checks for the match using face recognition software technique. [4] Steps followed in face recognition technique are: 1. Obtained image is cropped. 2. To the cropped image a Face algorithm is applied to get different face reactions of the particular image.

4) VERIFICATION

By the time of verification, dual process is done. One, the images of the students that are captured recently is compared for the match in student database. In two of the

probabilities the images are checked. If the captured image matches with the image that has been registered before are processed for attendance management. Second, if it is observed to be unmatched with student database then the image of the person will be consider as new and saved in the separate database called stranger database. The separation of the database will provide some information about the stranger who is new to the environment and gives the information about the person who has been entered. It not only ensures security but also make some fear to the people who needed to be entered without any authority.[1]

5) MARKING ATTENDANCE

The image of the student which is obtained, matched with student database and the attendance will be marked and the information is sent to the server which controls the overall database of the student. The software is installed in the tablet that would have much additional functionality that would improve the AMS features and helps in finding the report of each student. [2]

1.3 Advantages:

- Automated and web-based for easy accessibility
- No compatibility issues - all you need is an internet connection
- Eliminates paperwork and the risk of making errors while tracking attendance on paper
- Simple and easy to use
- Hugely reduces time spent managing staff attendance
- Records are kept safe and confidential
- Current and previous years' records are available in an instant
- Available via the internet at all times and from any location
- Configurable, multi-level, management approval system
- Installation, Maintenance and Data Security are taken care of by our technical team

1.4 Application:

1.4.1 Residential Use-

Alter Homeowner of approaching personal.

1.4.2 Voter verifications-

Where eligible politicians are required to verify their identity during a voting process this is intended to stop voting where the vote may not go as expected.

1.4.3 Banking using ATM-

The software is able to quickly verify a customers face.

1.4.4 Airport Security-

Airport and other transportation terminal security is not a new thing. People have long had to pass through metal detectors before they boarded a plane, been subject to questioning by security personnel, and restricted from entering "secure" areas. What has changed is the vigilance in which these security efforts are being applied.

1.4.5 Financial Use-

It can improve the security of the financial services industry, saving the institution time and money both through a reduction of fraud cases and the administration expenses of dealing with forgotten passwords.

Chapter 2

LITERATURE REVIEW

2.1 Working of Existing System:

In the present system all work is done on paper. The whole session attendance is stored in register and at the end of the session the reports are generated. We are not interested in generating report in the middle of the session or as per the requirement because it takes more time in calculation. At the end of session the students who dont have 75 % attendance get a notice. The second method is finger print recognition. But for some people it is intrusive, because it is still related to criminal identification. Another disadvantage of finger print recognition is that it can make mistakes with the dryness or dirt of the fingers skin. There are various methods for facial recognition like eigenface method. Various extensions have been made to the eigenface method such eigenfeatures.[1] This method combines facial metrics (measuring distance between facial features) with the eigenface representation. Another method similar to the eigenface technique is 'fisher faces' which uses Linear discriminate analysis This method for facial recognition is less sensitive to variation in lighting and pose of the face than using eigenfaces. Fisher face utilizes labeled data to retain more of the class specific information during the dimension reduction stage. A further alternative to eigenfaces and fisherfaces is the active appearance model. This approach uses an Active Shape Model to describe the outline of a face. By collecting many face outlines, Principal Component Analysis can be used to form a basis set of models which, encapsulate the variation of different faces. Many modern approaches still use Principal Component Analysis as a means of dimension reduction or to form basis images for different modes of variation.[4]

2.1.1 Disadvantages of Present Working System:

- Not User Friendly
- Difficulty in report generation
- Manual control
- Lots of paper work
- Time consuming

2.2 Proposed Plan of Work:

This Application is built for automating the processing of attendance. It also enhances the speed of the performing attendance task easily. It also generates periodic reports to keep a check on the students who are regular & who are not. So that creates this application In that Different modules are created. The first module is registration part enter the information of student in that roll number, name, year, semester, password & store in to database. Then second module is login part in that enter the roll number & password then click on login button it open automatically camera through opencv manager. Third module is marking attendance in that enter the roll number, subject name, date & password then click on make attendance button it automatically open camera & take face it detect face, recognition the face mark the attendance then give the remark of the student and store in to database. The last module is get report in that enter the subject name and date & click on the get report button it gives the report of particular report of the subject. A Faculty has to login to the system & then in the attendance option they have to select appropriate class, semester and subject. So this will display the list of the students who are eligible to appear in this session. So now the faculty has to just select the students name from the manual attendance sheet according to their roll number and then submit the sheet. This will add the selected students as present student in that particular session. This system is very useful to the office staff also because, they can generate various types of reports and submit them to respective faculties also or also can be submitted to the College Coordinator. [2]

2.2.1 Characteristics of Proposed System:

- 1) User Friendly: The proposed system is user friendly because the retrieval and storing of data is fast and data is maintained efficiently. Moreover the graphical user interface is provided in the proposed system, which provides user to deal with the system very easily.
- 2) Reports are easily generated: reports can be easily generated in the proposed system so user can generate the report as per the requirement (monthly) or in the middle of the session. User can give the notice to the students so he/she become regular.
- 3) Very less paper work: The proposed system requires very less paper work. All the data is feted into the computer immediately and reports can be generated through computers. Moreover work becomes very easy because there is no need to keep data on papers.
- 4) Computer operator control: Computer operator control will be there so no chance of errors. Moreover storing and retrieving of information is easy. So work can be done speedily and in time.

Chapter 3

OBJECTIVE AND SCOPE

3.1 Problem Statement:

Every time a lecture, section or laboratory starts the lecturer or teaching Assistant delays the lecture to record students attendance. This is a lengthy process and takes lot of time and effort, especially if it is a lecture with huge number of students. It also causes a lot of disturbance and interruption when an exam is held. Moreover the attendance sheet is subjected to damage and loss while being passed on between different students of teaching staff. And when the number of students enrolled in a certain course is huge, the lecturers tend to call a couple of students name at random which is not fair student evaluation process either. Finally, these attendance records are used by the staff to monitor the students attendance rates. This process could be easy and effective with a small number of students but on the other hand dealing with the records of a large number of students often leads to human error. [3]

3.2 Objective:

Attendance Monitoring System (AMS) can be made into smarter way by using face recognition technique, to captures the image of the person and checks the observed image with the face database using android enhanced smart phone & to store in canter database. It is typically used for two purposes. Firstly, marking attendance for student by comparing the face images produced recently and secondly, recognition of human who are strange to the environment i.e. an unauthorized person. For matching the captured images with the database, opencv libraries methodology was used.[4]

- Simplified attendance with respect to traditional method
- Gallery images must be of same size.
- Requires full frontal face to be presented for each time.
- To provide paperless environment by using computerize system that can save lot of time.
- No need to maintain several separate records and mannual calculation.

3.3 Scope:

This project is basically an android application which means self contained software runs on the system on which it has been installed under the user control and it will work for a particular institute or college only.

3.4 Out of Scope:

- It only take attendance and make the report.
- We not providing the automatic SMS facility for defaulters.

Chapter 4

REQUIREMENT ANALYSIS

4.1 Software Requirements:

4.1.1 Software requirement while developing system:

- Operating System - Windows XP or Windows 7
- Database- MySQL
- Technologies- JAVA ,Android,PHP
- IDE- Eclipse kepler,SDK,JDK 7,OpenCV SDK

4.1.2 Software requirement while using system:

- Android OS

4.2 Hardware Requirements:

4.2.1 Hardware requirement while developing system:

- PC/Laptop

4.2.2 Hardware requirement while using system:

- Tablet/Mobile of android platform

4.3 Required softwares:

4.3.1 Android Operating System:

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google OHA. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard.

Android is the most popular mobile OS. As of 2013, Android devices sell more than Windows and Mac OS devices combined, with sales in 2012, 2013 and 2014 close to the installed base of all PCs.. A developer survey conducted in AprilMay 2013 found that 71% of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over 1 billion active monthly Android users (that have been active for 30 days), up from 538 million in June 2013.

Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007. The Androids Initial release is in September 23, 2008 and the latest release is in the 4.4.4 Kit Kat / June 19, 2014. The first release of android is Alpha (1.0).

4.3.2 Eclipse:

In computer programming, Eclipse is an integrated development environment (IDE). It is developed by Eclipse Foundation. The stable release is Stable release 4.4.1 (Luna) / 26 September 2014. It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-INS, eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Natural, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematical. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others. The initial codebase originated from

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

4.3.3 JDK :

The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. The JDK includes a private JVM and a few other resources to finish the recipe to a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK).[citation needed] On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun contributed the source code to the OpenJDK.

4.3.4 SDK:

The Android Software Development Kit (Android SDK) contains the necessary tools to create, compile and package Android applications. Most of these tools are command line based. The primary way to develop Android applications is based on the Java programming language. A software development kit (SDK) is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform. It may be something as simple as the implementation of one or more application programming interfaces (APIs) in the form of some libraries to interface to a particular programming language or to include sophisticated hardware that can communicate with a particular embedded system. Common tools include debugging facilities and other utilities, often presented in an integrated development environment (IDE). SDKs also frequently include sample code and supporting technical notes or other supporting documentation to help clarify points made by the primary reference material.

4.3.5 PHP :

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code can be simply mixed with

HTML code, or it can be used in combination with various templates engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications.

We are using a central database means that use the database online anytime or anywhere. For this accessing database use the free web hosting area website. In that use the php language for database design. PHP can generate dynamic page content and it can create, open, read, write, delete, and close files on the server. Also it can collect form data, send and receive cookies and add, delete, modify data in your database and can be used to control user-access PHP and encrypt data with PHP, you are not limited to output HTML. You can output images, PDF files, and even flash movies. You can also output any text, such as XHTML and XML. PHP runs on various platforms (Windows, Linux, UNIX, Mac OS X, etc.)PHP is compatible with almost all servers used today (Apache, IIS, etc.).It supports a wide range of database PHP is free.

Chapter 5

SYSTEM DESIGN

5.1 Module:

- Register Student
- Login Student
- Authentication
- Mark Attendance
- Get Report

5.1.1 Register Student:

First phase of the system, actually deals with registering the information of the student in a particular classroom. The information includes:

- (1) Name of the student
- (2) Roll number
- (3)Branch
- (4) Semester
- (5) Password

These details are placed in the student database from which the actual comparision will be done.

5.1.2 Login Student:

When registration is finished then enters into login phase. In login phase student enter the roll no & password this is matching with registration phase then save the face or register the face to the particular roll no and saves into database.

5.1.3 Authentication:

In authentication phase, we provide the facility for the teacher and student to login into the system separately. Student do not interfere between attendance sheet and get report.

5.1.4 Mark Attendance:

When login is completed then attendance sheet is open. In that form teacher enter subject name, date and circulate smart phone or tablet to student. Students enter the roll no and password gives the face to mark attendance and to store in database. This face is compare with training faces which is already registering in database.

5.1.5 Get Report:

Display the report for already registered student.

5.2 An Overview of UML:

The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting

THE UML LANGUAGE :

A language provides a vocabulary and the rules for combining words in that vocabulary for the purpose of the communication. A modeling language is a language whose vocabulary and rules focus on conceptual and physical representation of a system. A

modeling language such as UML is thus a standard language for software blueprints. In this context, specifying means building models that are precise, unambiguous, and complete. In particular, UML addresses the specification of all the important analysis, design and implementation decision that must be made in developing and deploying a software intensive system. The UML is not a visual programming language, but its model can be directly connected to a verity of programming languages. This means that its possible to map from a model in UML to a programming language such as java, cpp, or visual basic or even to tables in a relational database. Things that are best expressed graphically are done so graphically in UML, whereas things that best expressed textually are done so in the programming language. A healthy software organization produces all sorts of artifacts in addition to raw executable code. These artifacts include requirements, architecture, design, source code, project plans, tests, prototypes, releases. UML addresses the documentation of a systems architectures and all of its details. UML also provides for expressing requirements and for tests. Finally, UML provides a language for modeling the activities of project planning and release management.

5.3 Goals of UML:

The primary goals in the design of the UML were:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models. Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes. Provide a formal basis for understanding the modeling language
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices

5.4 A Conceptual Model of UML:

To understand UML, you need to form a conceptual model of the language, and this requires learning three major elements: the UMLs basic building blocks, the rules that dictate how those building blocks may be put together, and some mechanisms that apply

throughout the UML. Once you have grasped these ideas, you will be able to read UML models and create some basic ones. As you gain more experience in applying UML, you can build on this conceptual model, using more advanced features of the language.

5.4.1 Building Blocks of UML:

The vocabulary of the UML encompasses three kinds of building blocks:

- Things
- Relationships
- Diagrams

These are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams groups interesting collections of things.

5.5 Diagrams in UML:

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). You draw diagrams to visualizing a system from different perspectives, so a diagram is a projection into a system. For all but the most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams. In theory, a diagram may contain any combination of things and relationships. The views that comprise the architecture of software intensive system. For this reason, the UML includes following diagrams:

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Deployment Diagram

5.6 Use Case Diagram:

5.6.1 Description:

A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is a just special kind of diagram and shares the same common properties as do all other diagram-a name and graphical contents. A use case diagram is the simplest representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can define the different types of users of a system and the various ways that they interact with the system. They provide the simplified and graphical representation of what the system must actually do.[1] The purpose of the use case diagrams is simply to provide the high level view of the system and convey the requirements. You can use cases for the following purposes:

- Determine the requirements of the system.
- Describe what the system should do.
- Provide a basis for testing to ensure that the system works as intended.

5.6.1.1 Contents

Use case diagrams commonly contain

- **Use Case**

A use case defines the interactions between external actors and the system under consideration to accomplish a goal. The use cases that a system or component supports appear inside its rectangle. It can be useful to show some use cases outside the rectangle, to clarify the scope of your system. A subsystem in a use case diagram has basically the same type as a component in a component diagram.

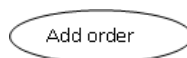


FIGURE 5.1: Use Cases

- **Actors**

An actor represents a role that an outsider takes on when interacting with the business system. For instance, an actor can be a customer, a business partner, a supplier, or another business system and every actor has a name.

- **Dependency, generalization, and association relationships.**

A dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing.



FIGURE 5.2: Dependencies

A *generalization* is a relationship in which objects of specialized elements (the child) are substitutable for objects of the generalized element.

An *association* is a structural relationship that describes a set of links, a link being connection among objects

Like all other diagrams, use case diagram may contain notes and constraints.

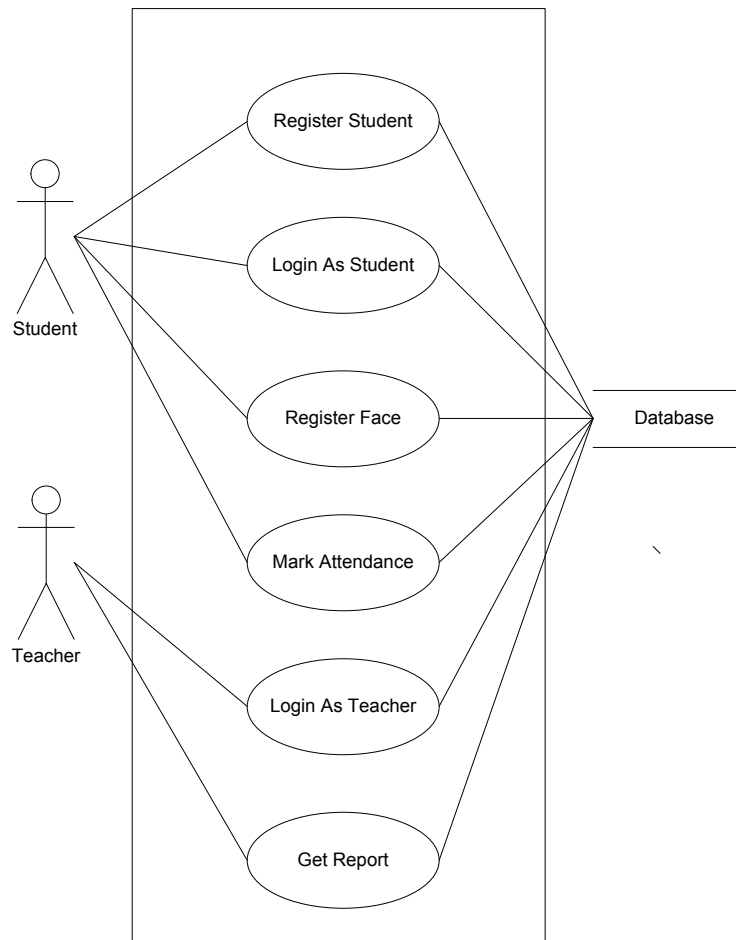


FIGURE 5.3: Use Case Diagram

5.6.2 Use-Case Scenario:

USE CASES	USE CASE SCENARIO
Register Student	<ol style="list-style-type: none"> 1. Open the student registration form. 2. Fill the own record of student like Roll No, Name, Year, Branch, Semester and password. 3. Click on register button. 4. Confirm the student registration. 5. Store into database.
Login	<ol style="list-style-type: none"> 1. Click on login button. 2. Open login form. 3. Login student with Roll number and Password. 4. Student get login.
Register Face	<ol style="list-style-type: none"> 1. Open the OpenCV manager. 2. Enter the face name. 3. Capture the face and compare this face from database. 4. If current captured face match with previous stored face into database then directly open the authentication form.
Mark Attendance	<ol style="list-style-type: none"> 1. Click on Attendance Sheet form. 2. Fill the Subject name, Date and Roll no. 3. Click on Done button. 4. Click on face recognition. 5. Open the OpenCV manager and searching the face. 6. If captured face is matched with previous stored face into database then mark attendance. 7. Otherwise the student is absent.
Get Report	<ol style="list-style-type: none"> 1. Click on Get report button. 2. Open Get report form. 3. Enter the subject name. 4. Enter the date. 5. Click on get report button 6. It give the report of particular subject.

TABLE 5.1: Use Case Scenario

5.7 Sequence Diagram:

5.7.1 Contents:

Sequence diagram commonly contains:

- Objects
- Links
- Messages

5.7.2 Description:

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.[3] UML sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom

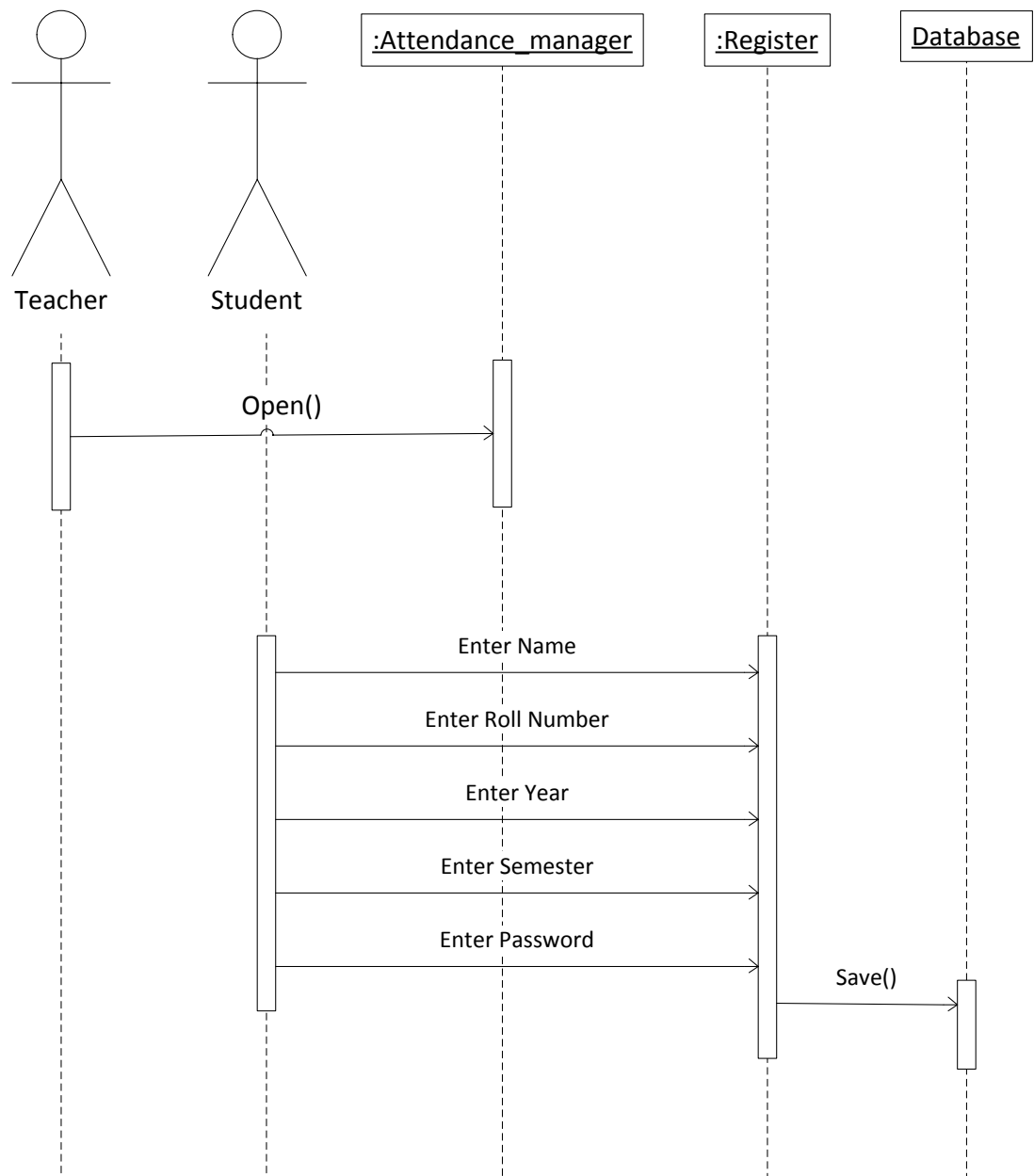


FIGURE 5.4: Student Registration Diagram

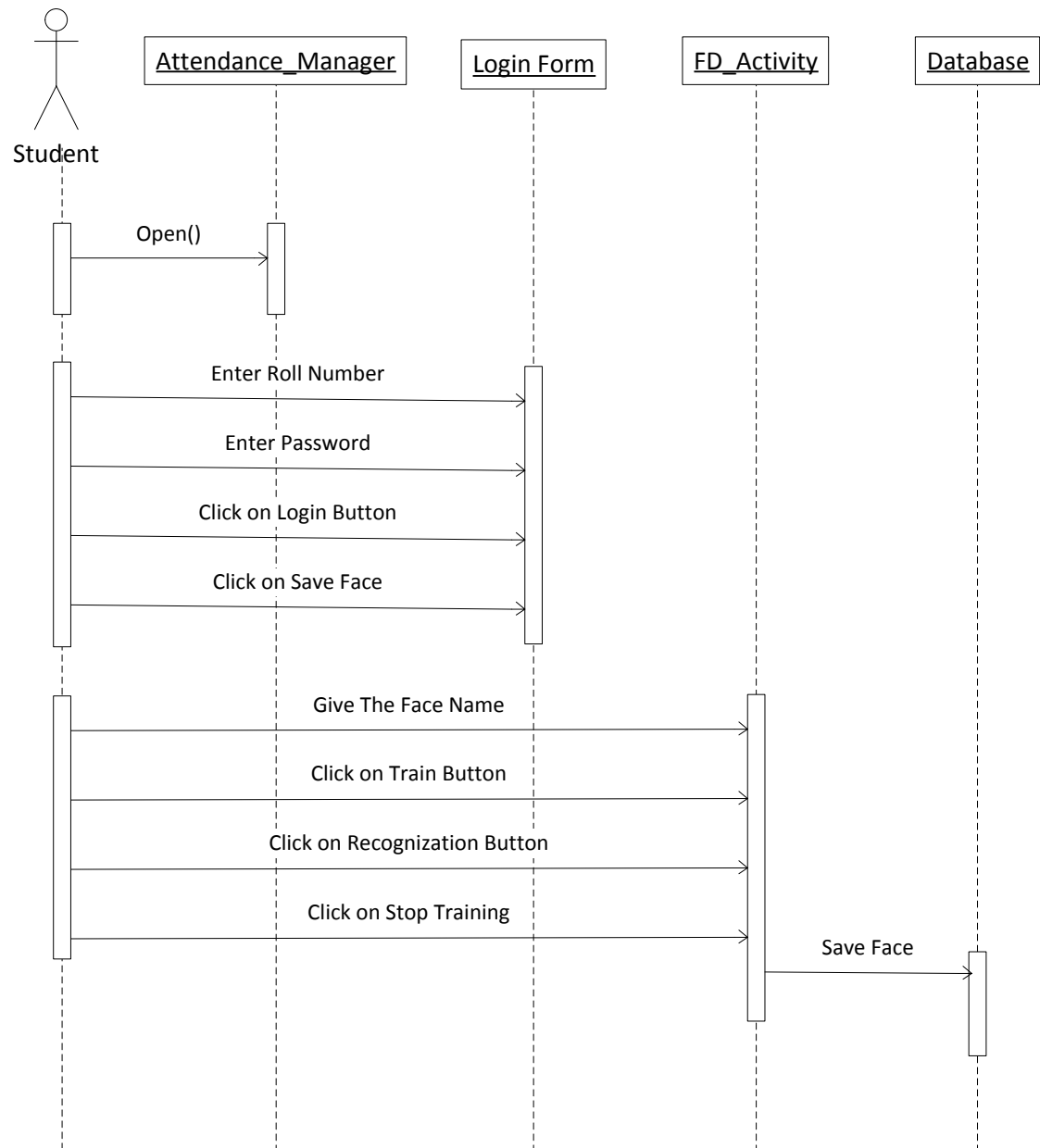


FIGURE 5.5: Student Login

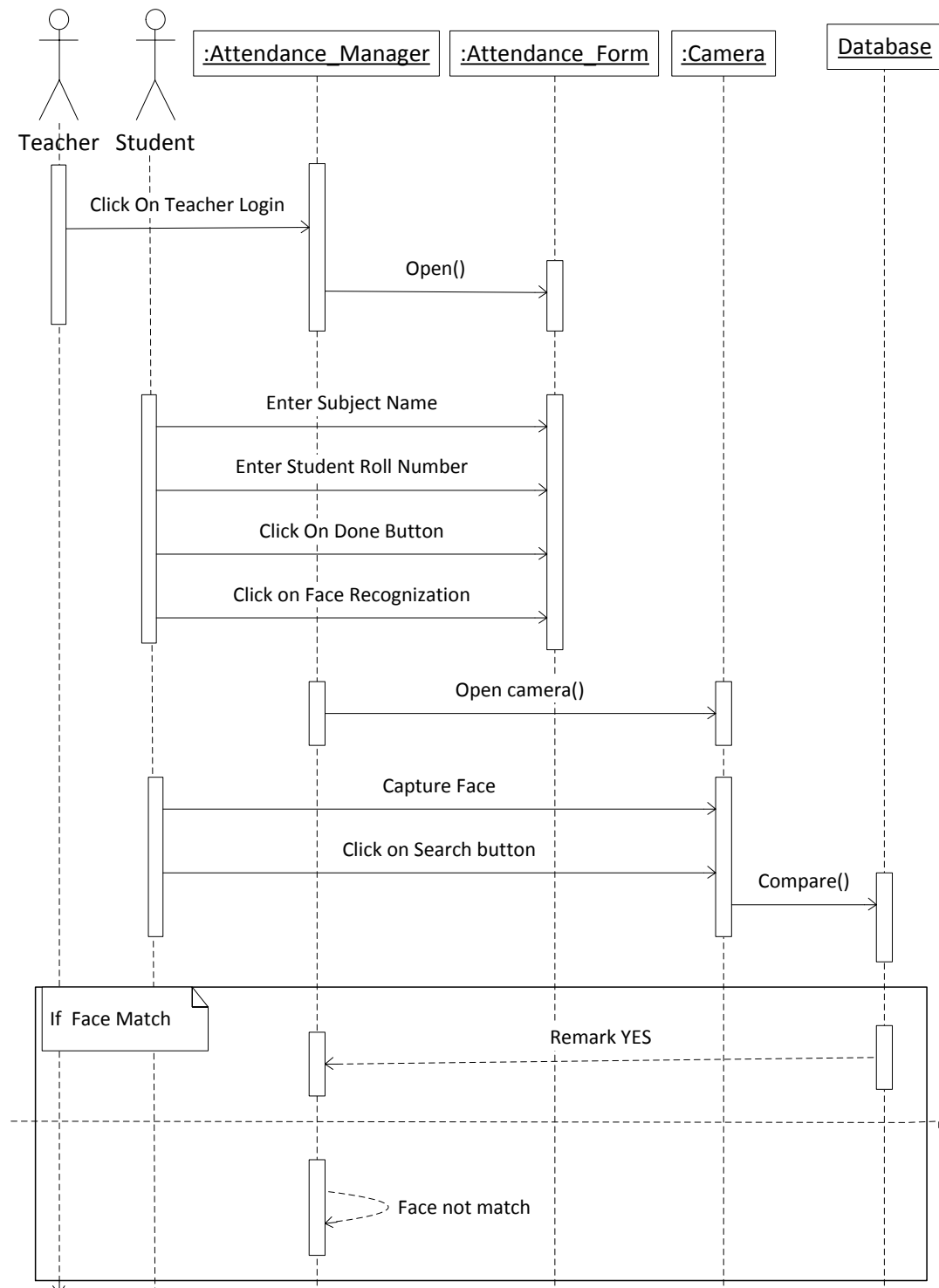


FIGURE 5.6: Mark Attendance Sheet

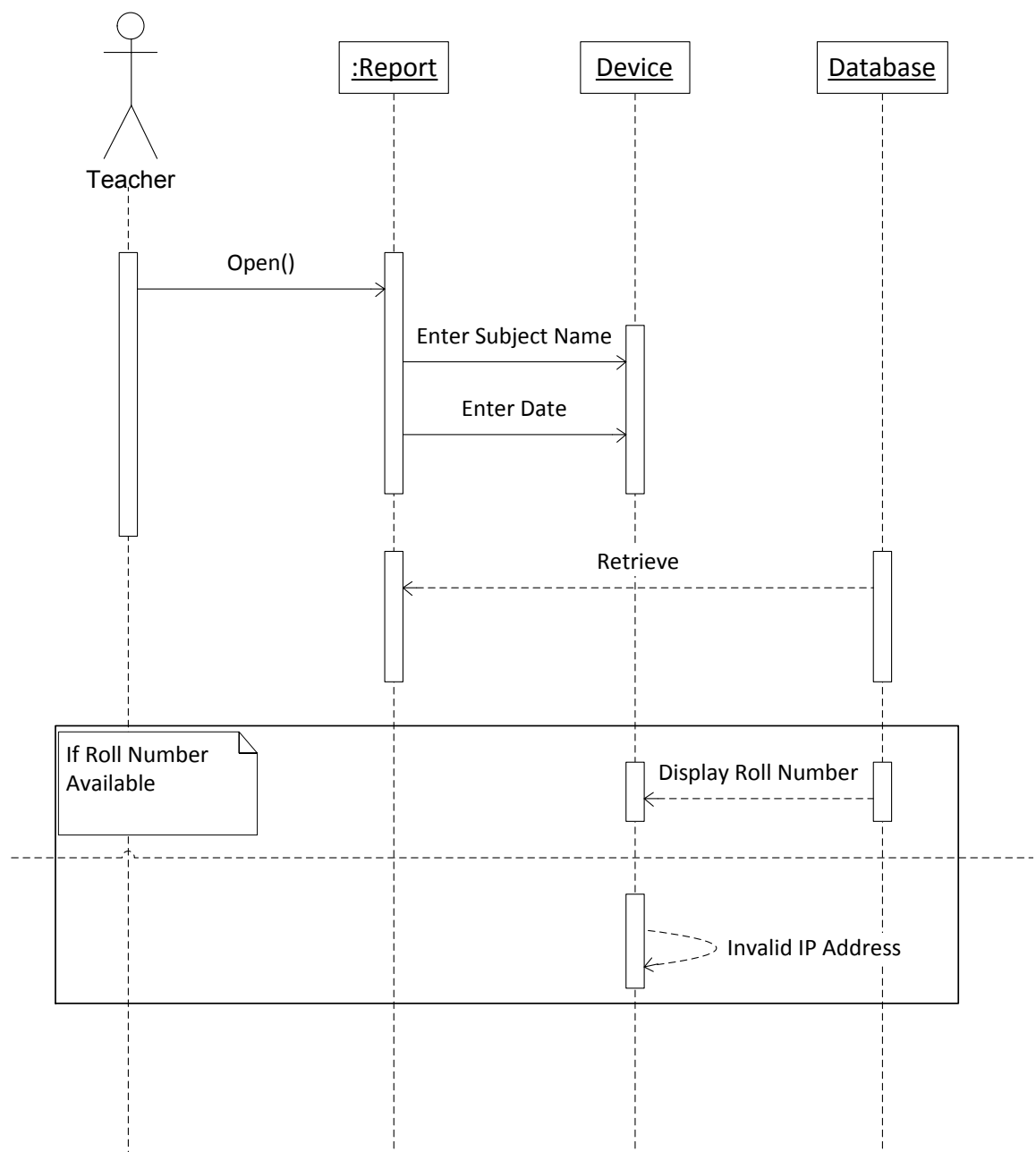


FIGURE 5.7: Get Report

5.8 Class Diagram:

5.8.1 Contents:

Class diagram commonly contain the following things:

- Classes
- Interfaces
- Collaborations
- Dependency, generalization, and association relationships.

5.8.2 Definition and Common Uses:

A class diagram is a diagram that shows a set of classes, interfaces and their relationships. Graphically, a class diagram is a collection of vertices and arcs. A class diagram will share the same common properties as do all other diagrams. A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved. In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.

- Class: A definition of objects that share given structural or behavioral characteristics.
- Attribute: A typed value attached to each instance of a classifier.
- Operation: A method or function that can be performed by instances of a classifier

5.8.3 Description:

A UML class diagram describes the object and information structures used by your application, both internally and in communication with its users. It describes the information without reference to any particular implementation. Its classes and relationships can be implemented in many ways, such as database tables, XML nodes, or compositions of software objects.[1]

- Class: A definition of objects that share given structural or behavioral characteristics.
- Attribute: A typed value attached to each instance of a classifier.
- Operation: A method or function that can be performed by instances of a classifier.

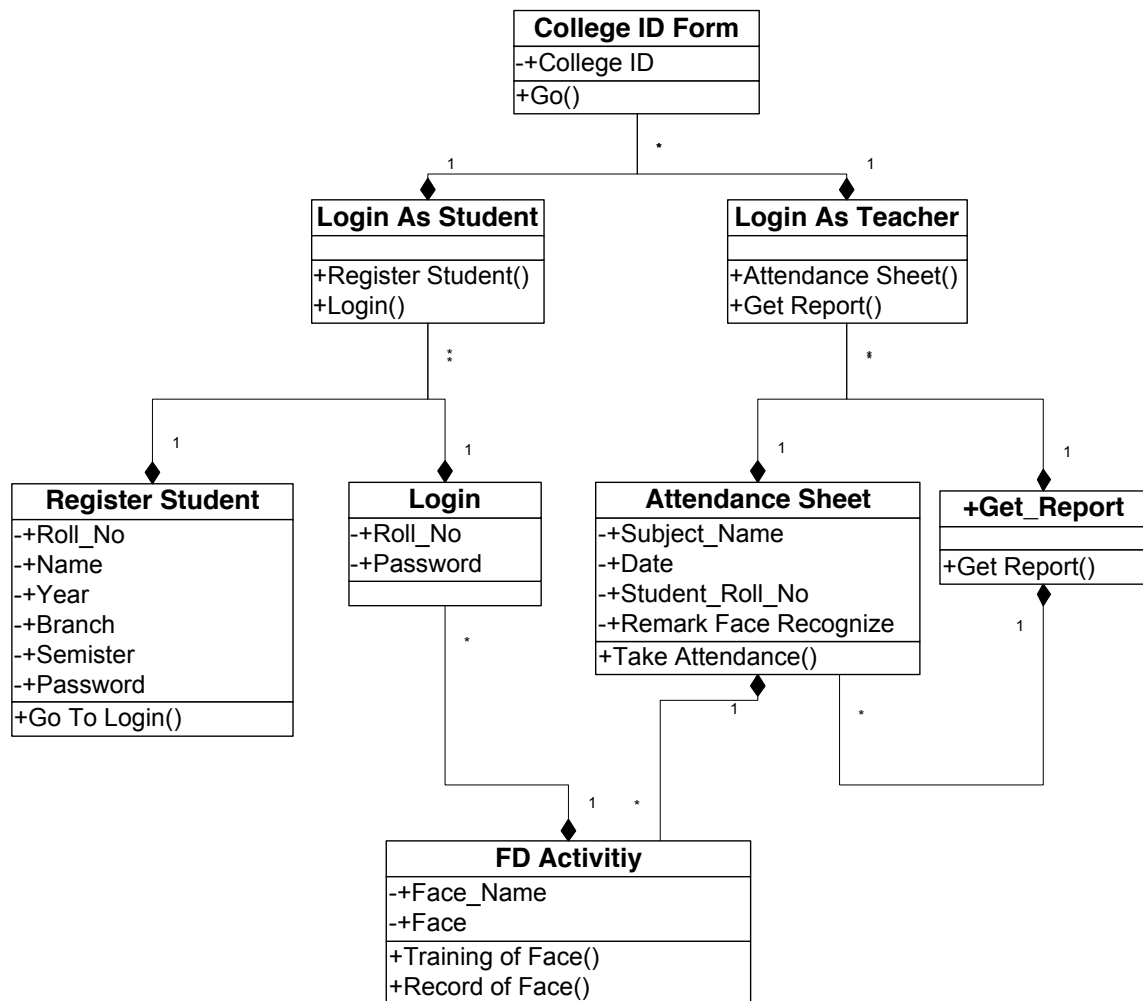


FIGURE 5.8: Class Diagram

5.9 Deployment Diagram:

5.9.1 Definition:

A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagram address the static deployment view of architecture. They are related to component diagrams in that a node typically encloses one or more components. The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes.

Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important because it controls the following parameters:

- Performance
- Scalability
- Maintainability
- Portability

So before drawing a deployment diagram the following artifacts should be identified:

- Nodes
- Relationships among nodes

The following deployment diagram is a sample to give an idea of the deployment view of order management system. Here we have shown nodes as:

- Monitor
- Modem
- Caching server
- Server

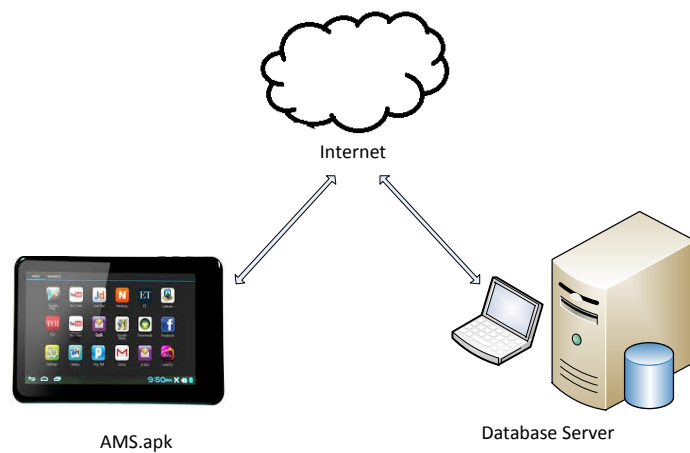


FIGURE 5.9: Deployment Diagram

Chapter 6

CODING

6.1 College Form:

This is First form in go method teacher must be enter college id. After entering college Id it can go to clgverify method it can compare the database if matches then display welcome to respective college. otherwise It can display Welcome to Null college.

```
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.json.JSONObject;
```

```
public class CollegeForm extends Activity
{
    public void goMethod(View v)
    {
        if(clgId.equals(""))
        {
            Toast.makeText(getApplicationContext(), "Must Be Enter
                College ID", Toast.LENGTH_SHORT).show();
        }
        Else
        {
            clgVerify();
            flagOfClgName=compareNm;
            Toast.makeText(CollegeForm.this, "WELCOME TO "+compareNm +
                "College", Toast.LENGTH_SHORT).show();
            Intent i=new Intent(CollegeForm.this, MultiOption
                Form.class);
```

```
startActivity(i); }
}
}
public void clgVerify()
{
    try
    {
        HttpEntity entity = response.getEntity();
        is = entity.getContent();
        Log.e("pass 1", "connection success ");
    }
    catch(Exception e)
    {
        Log.e("Fail 1", e.toString());
        Toast.makeText(getApplicationContext(), "Invalid IP
            Address",
            Toast.LENGTH_LONG).show();
    }
    try
    {
        while ((line = reader.readLine()) != null)
        {
            sb.append(line + "\n");
        }
        is.close();
        result = sb.toString();
        Log.e("pass 2", "connection success ");
    }
    catch(Exception e)
    {
        Log.e("Fail 2", e.toString());
    }
    try
    {
        JSONObject json_data = new JSONObject(result);
        compareNm=(json_data.getString("name"));
    }
    catch(Exception e)
    {

```



```
Log.e("Fail 3", e.toString());
}
}
}
```

6.2 MultiOption Form:

The main class MultiOption Form for the multi-optional form. There are several methods are present.

- registerOptionMethod- This is the method for the student registration form. This method is intended to the Register form. Then activity is started.
- void loginOptionForm- This is the method for the Login form. This method is intended to the Login Form. Then activity is started.
- attendanceMethod- This is the method for the Attendance Sheet. This method is intended to the Attendance Sheet. Then activity is started.
- getReportMethod This is the method for the get report . This method is intended to the Get Report . Then activity is started.

```
public class MultiOption Form extends Activity
public void registerOptionMethod(View v)
{
Intent i=new Intent(MultiOption
    Form.this,Register_form.class);
startActivity(i);
}
public void loginOptionForm(View v1)
{
Intent i=new Intent(MultiOption Form.this,Login_Form.class);
startActivity(i);
}
public void attendanceMethod(View v11)
{
Intent i=new Intent(MultiOption
    Form.this,Attendance_Sheet.class);
startActivity(i);
}
```

```
}  
  
public void getReportMethod(View v111)  
{  
    Intent ii=new Intent(MultiOption  
        Form.this,Get_Report.class);  
    startActivity(ii);  
}
```

6.3 Register form:

The main class Register form for the student registration form. In this form the student fill the basic information like roll no, name, password, year, branch, semester. There are some of the methods are present.They are as follows:

- addListenerOnSpinnerItemSelection2- The first method is for the selection of the semester for the student in which they are studying & it is given in the form of the spinner.
- addListenerOnSpinnerItemSelection1-

The another method is faceDetectMethod. This is intended to the FdActivity for the purpose of the face detection.

```
public class Register form extends Activity  
private void addListenerOnSpinnerItemSelection2()  
{  
    semester.setOnItemSelectedListener(new  
        CustomOnItemSelectedListener());  
}  
  
private void addListenerOnSpinnerItemSelection1() {  
    branch.setOnItemSelectedListener(new  
        CustomOnItemSelectedListener());  
}  
  
private void addListenerOnSpinnerItemSelection()  
{  
    year.setOnItemSelectedListener(new  
        CustomOnItemSelectedListener());  
}
```

```
public void registerMethod(View v)
{
    if((StudRollNo.equals("")) || (StudName.equals("")) ||
        (StudPassword.equals("")) || (StudYear.equals("")) ||
        (StudBranch.equals("")) || (StudSemister.equals("")))
    {
        Toast.makeText(getApplicationContext(), "Field Vacant",
            Toast.LENGTH_SHORT).show();
    }
    else{
        Toast.makeText(Register_form.this, "Successfully",
            Toast.LENGTH_SHORT).show();
        insert();
        Intent i=new Intent(Register_form.this,Login_Form.class);
        startActivity(i); }
    }

public void faceDetectMethod(View v1)
{
    Intent ii=new Intent(Register_form.this,FdActivity.class);
    startActivity(ii);
}

public void insert()
{
    if(code==1)
    {
        Toast.makeText(Register_form.this, "Insert Successfully",
            Toast.LENGTH_SHORT).show();
    }
    else
    {
        Toast.makeText(getBaseContext(), "Sorry, Try
            Again",Toast.LENGTH_LONG).show();
    }
}

public class Login_Form extends Activity {
public void loginMethod(View v)
{
```

```
studRollNo=rollNo.getText().toString();
studPassword=password.getText().toString();
getlogin();
}

public void saveFaceMethod(View v)
{
Intent i=new Intent(Login_Form.this,FdActivity.class);
startActivity(i);
}

public void getlogin()
{
try
{
if(studPassword.equals(pass))
{
Toast.makeText(Login_Form.this, "Login Successfully",
    Toast.LENGTH_SHORT).show();
}
}
catch(Exception e)
{
Log.e("Fail 3", e.toString());
}
}

public void backMethod(View v)
{
Intent i=new Intent(Login_Form.this,MultiOption_Form.class);
startActivity(i);
}
}
```

6.4 Login:

In this method the Login Form is intended to the MultiOption Form. Then activity is performed.

The class Login Form for the login activity. There are some methods are used. They are:

- loginMethod- Here when student gives the Roll no.& password then it goes to the getlogin() method.
- saveFaceMethod- This method is intended to the FdActivity class for the purpose of the face detection activity. Here the face is saved which is given by the student.
- Getlogin- In this method if the login get successfully then the roll no & password is stored in the database otherwise it gives the message Invalid IP address.
- void backMethod- In this method the Login Form is intended to the MultiOption Form. Then activity is performed.

```
public class Login_Form extends Activity {
public void loginMethod(View v)
{
studRollNo=rollNo.getText().toString();
studPassword=password.getText().toString();
getlogin();
}
public void saveFaceMethod(View v)
{
Intent i=new Intent(Login_Form.this,FdActivity.class);
startActivity(i);
}
public void getlogin()
{
try
{
if(studPassword.equals(pass))
{
Toast.makeText(Login_Form.this, "Login Successfully",
    Toast.LENGTH_SHORT).show();
}
}
catch(Exception e)
{
Log.e("Fail 3", e.toString());
}
}
public void backMethod(View v)
{
```

```
Intent i=new Intent(Login_Form.this,MultiOption_Form.class);
startActivity(i);
}
}
```

6.5 Face Detection Activity:

This package is import help for face detection activity in that the camera is opened and the detecting the face which store the Registration phase.

The next method is for the selection of the branch for the student in which they are studying & it is given in the form of the spinner.

- `addListenerOnSpinnerItemSelection`- The next method is for the selection of the year for the student in which they are studying & it is given in the form of the spinner.
- `registerMethod`- The next method is `registerMethod`. In this method it checks for the field. If the single of the field is remaining to fill it gives the message Field Vacant otherwise Successfully. And also it is intended to the login form.
- `faceDetectMethod`- The another method is `insert`. In this if the all field in form are filled successfully the it gives the "Insert Successfully" message otherwise it gives the Sorry, Try Again message.

```
Import org.opencv.android.BaseLoaderCallback;
```

Default `BaseLoaderCallback` implementation treats application context as `Activity` and calls `Activity.finish()` method to exit in case of initialization failure. To override this behavior you need to override `finish()` method of `BaseLoaderCallback` class and implement your own finalization method. In an android activity this line of code should be in the `onResume()`, but in an android service I don't an `onResume()` method.

```
Import
    org.opencv.android.CameraBridgeViewBase.CvCameraViewFrame;
```

We need to pick an appropriate size for the preview frames, as too small frames will result in a bad result when we do the processing, and too large frames will slow down everything to an unacceptable level. Since smartphone cameras often have a different

set of supported sizes for preview frames, we need to pick a minimum acceptable size. In order to convert a color image to Grayscale image using OpenCV, we read the image into `BufferedImage` and convert it into `Mat` Object. Its syntax is given below:

```
File input = new File("digital_image_processing.jpg");
BufferedImage image = ImageIO.read(input);
```

Then you can transform the image from RGB to Grayscale format by using method `cvtColor()` in the `Imgproc` class. Its syntax is given below:

```
Imgproc.cvtColor(source mat, destination mat1,
    Imgproc.COLOR_RGB2GRAY);
```

In order to change color space of one image to another using OpenCV, we read image into `BufferedImage` and convert it into `Mat` Object. Its syntax is given below:

```
File input = new File("digital_image_processing.jpg");
BufferedImage image = ImageIO.read(input);
//convert BufferedImage to Mat.
public class FdActivity extends Activity implements
    CvCameraViewListener2
{
private static final String TAG =
    "OCVSample::Activity";
private static final Scalar FACE_RECT_COLOR = new
    Scalar(0, 255, 0, 255);
public static final int JAVA_DETECTOR = 0;
public static final int NATIVE_DETECTOR = 1;
public static String flag="No";
public static String flagRollNo=" ";
public static final int TRAINING= 0;
public static final int SEARCHING= 1;
public static final int IDLE= 2;
private static final int frontCam =1;
private static final int backCam =2;
```

6.6 Get Report:

Click on the get report button then entering the subject name and date. This is a client request to go http server in that use the php server page. In that use the post and get method. So that gives the response to the client or user.

It can goto the HttpPost method is used it goto he link dhttp: If connection is successfully display the report. Otherwise display Invalid IP address.

```
public void getreport()
{
    ArrayList<NameValuePair> nameValuePairs = new
        ArrayList<NameValuePair>();
    nameValuePairs.add(new
        BasicNameValuePair("SelectSubject",SelectSubject));
    nameValuePairs.add(new
        BasicNameValuePair("SelectDate",SelectDate));
    try
    {
        HttpClient httpclient = new DefaultHttpClient();
        HttpPost httppost = new
            HttpPost("http://original.orgfree.com/attendance_report.php");
        Httppost.setEntity(new
            UrlEncodedFormEntity(nameValuePairs));
        HttpResponse response = httpclient.execute(httppost);
        HttpEntity entity = response.getEntity();
        is = entity.getContent();
        Log.e("pass 1", "connection success ");
    }
}
```

Chapter 7

TESTING

7.1 What is Software Testing:

Software testing is the process of analyzing or operating software for the purpose of finding bugs. Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attribute. The fundamental objective of testing is to find defects, as early as possible and get them fixed.

Software Testing Process:

- Test Planning: high level plans which list test objectives, test approach, measurement criteria along with test schedule and resources.
- Test Design: create test cases, identify test cases for automation(if applicable), prioritize test cases and finalize test iterations.
- Test Implementation: Create test scripts using automated testing tools.
- Test Execution: Execute the test cases on the test environment and test reports.
- Test analysis: Use test and project metrics to calculate key indicators. The data usually will be obtained from your defect tracking system.
- Postmortem Reviews: Discuss lessons learnt and identify strategies which will prevent such problems in future.

7.2 Test methods:

7.2.1 Black box testing:

It is also called as functional testing, it is the process of giving the input to the system and checking the output of the system. Without bothering about the system that how the system generates the output. It is also called as Behavior testing.

- Approach to testing where the program is considered as a Black Box.
- Testing based solely on analysis of requirements user specification, user documentation etc.
- The test cases are based on the specifications.
- Black box testing techniques apply to all levels of testing.
- Test planning and design can begin early in the software process.
- Tests are done from a users point of view.

As per black box testing rule we are testing the overview of project. we are testing the our implemented whole project working or not. just checking the giving input and getting output or not. just clicking on any android file it browse or not .

7.2.2 White Box Testing:

1) white box testing : White Box Testing method is applicable to the following levels of software testing those applied in your project:

- Unit Testing: For testing paths within a unit.
- Integration Testing: For testing paths between units.
- System Testing: For testing paths between subsystems.

1) Unit Testing : In this testing check the coding of android file those are installed on mobile. also check the php code those are use full for hosting the website. also check the code of interpreter those are interfacing the java compiler.

2) Integration Testing : In Intergation testing we are testing the path of different sub-model in this checking the Android codes whole path those are conneted properly or

not means on the clicking on the video the video index is opened properly or not. doing same thing for every unit.

3) System Testing : In System Testing We are testing the whole system. In this mostly We are testing the paths withing subsystems.and also to checking whole system will be connected propely or not we have check as per IP adderss and android aaps linking the whole system working properly not . in that mostly We are testing the cloud is properly acessible on android. In that testing we are checking the those IASS, PASS, SaaS these concept is implemented properly well or not and its working properly or not .

7.3 Test cases and Test data:

- Test data are inputs that have been devised to test the system.
- Test cases are inputs and output specification plus a statement of the function under test.
- Test data can be generated automatically or real.

TABLE 7.1: Test Cases

Sr.No.	Objectives	Prerequisites	Steps to be followed	Expected Result	Actual Result	Remark
1	Start the application.	Application should be installed.	1.Go to menu. 2.Click on application icon.	Application should Start.	Application get started successfully.	Pass
2	Enter College code id.	Application should be started.	1. Enter college id. 2.Click on Go button.	Accept the college id and then go to next step.	Take the id and go to next step.	Pass
3	Different option forms are open.	It can be open form	It open the different forms.	Different forms are open.	It can open different forms successfully.	Pass
4	Click on student registration button.	click on it and open the form.	Click on the student registration form.	Open the registration form.	It can open form successfully.	Pass
5	Fill the registration form.	Form should be open.	Fill the form.	It show the all the content.	It show all data.	Pass
6	Click on Name field.	It should be open and give the only character or letter.	Enter the name.	It take the name only character.	It taken character letter name.	Pass
7	Click on roll number field.	Should be click on it and give only number.	Enter the roll no.	It take only roll number	It taken the roll number	Pass

Table 7.1 — continued from previous page

Sr.No.	Objectives	Prerequisites	Steps to be followed	Expected Result	Actual Result	Remark
8	Click on the Year field.	It should be open the field.	Enter the Year.	It take the year of studying.	It taken the year of study.	Pass
9	Click on Branch field.	It should be open the field.	Enter the branch of student.	It take the branch of the student.	It taken the branch.	Pass
10	Click on Semester field	It should be open the field	Enter the semester	It take the Semester of the student	It taken the semester	Pass
11	Click on the Password field	It should be open the field	Enter the Password	It take the password from student	it taken the password	Pass
12	Click on registraion botton	All field in this form should be filled	Click on the registration button	Submit all the information which is filled by student	Submitted all information	Pass
13	Open next form	It should be fill all the field	Fill the all the field	It fill all record	It not procede if any field is empty	Pass
14	Click on the login form	Registration form must be filled	Click on the Login form	Open the login form	Login form opened	Pass

Table 7.1 — continued from previous page

Sr.No.	Objectives	Prerequisites	Steps to be followed	Expected Result	Actual Result	Remark
15	Click on Roll no	It should be open the field	Enter roll no which is given in the registration form	It takee roll no	Roll Number is taken	pass
16	Click on Password	It should be open the field	Enter password which is given in the registration form	it take password	Password is taken	Pass
17	Click on the Login Button	All field in this form should be filled	Fill the form	Filling successful form	form successfully filled	Pass
18	Click on Save Face Button	Successfully filled registration & Login form	Click on the Save face button	Submit all the information which is filled by student	Submitted all information	Pass
19	Open the camera	Fill login form successfully	It capture the face of student	Capturing the face of student	successfully face is captured	Pass
20	Camera	It should requird opencv manager application	It atomatically open the camera from open cv manager	Open camera	Successfully open camera	Pass
21	Click on train button	i)Fill login form successfully. ii)Give the roll no.	1.Give face number. 2.Give face.	Collect the images	Successfully it generates the set of the images	Pass

Table 7.1 — continued from previous page

Sr.No.	Objectives	Prerequisites	Steps to be followed	Expected Result	Actual Result	Remark
22	Click on the rec button	Roll no should be given	Click on rec button	Face will be recognized	Face successfully recognized	Pass
23	Take one image only	Give face image	Give face image	Face is capture	It give more than one image	Fail
24	Click on the stop train button	Face should be captured	Click on the train button	Stop the trainy set	Trainy set is stopped successfully	Pass
25	Click on the Search button	Trainy set must be given	Click on the serach button	Direct go to attendance sheet form	Attendance sheet form is opened	Pass
26	Open the select optional form	It should be open	Registration is completed	Open the multi optional form	It not open this form dirctly	Fail
27	Click on Subject Name	It should be open the field.	Enter the subject name	Take the subject name	Subject name is taken	Pass
28	Click on Date	It should be open the field	Enter the Date	Take the Date	Date is taken	Pass
29	Click on Student Roll No	It should be open the field	Enter the student roll no	Take the students roll no	Roll number is taken	Pass

Table 7.1 — continued from previous page

Sr.No.	Objectives	Prerequisites	Steps to be followed	Expected Result	Actual Result	Remark
30	Click on done button	Should be click on done button	Click on done button	It gives the remark	Remark is given	pass
31	Click on face recognition button	It should be click on it and open camera	Click on face recognition button	Open the camera and take face image	It open camera and take the face image	Pass
32	Give remark	It should give remark	It give atomatically remark	It give remark	It give remark	Pass
33	Click on back button	Should be Click on back button	Click on back button	It go to previous form	Previous form is opened	Pass
34	Click on get report	It should be open the field	Click on the get report button	Gives the report	Report is given	Pass
35	Click on Subject Name	It should be open the field	Enter the subject name	Enter the subject name	Subject name is taken	Pass
36	Click on Date	It should be open the field	Enter the date	Enter the date	Date is taken	Pass
37	Click on get report button	It should be open the field	Click on the get report button	Report should give	Report is given	Pass
38	It give report of subject	It should give report	Click on get report button	Get report	It give report of subject	Pass

Chapter 8

DEPLOYMENT

8.1 Welcome form:



FIGURE 8.1: Welcome Form

This is the first form of our application based on the Attendance Monitoring System.

8.2 College ID Form

The image shows a web interface for a College ID Form. At the top, there is a text input field labeled "College Code No :" with the number "123" entered. Below the input field is a "Go" button. The background of the form is a photograph of a modern, multi-story building with a yellow and white facade.

FIGURE 8.2: College ID Form

In this form enter the college code number and click on the go button. When we enter the specific college code then it will go to the specific college and then messege will display like WELCOME TO SITCollege.

8.3 Authentication Form

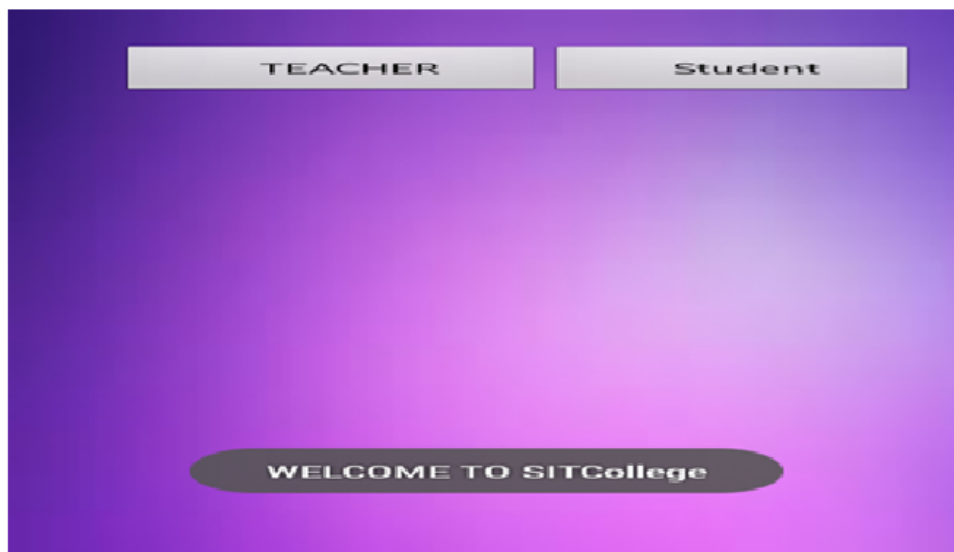
The image shows a web interface for an Authentication Form. The background is a solid purple color. At the top, there are two buttons: "TEACHER" and "Student". Below these buttons, there is a large, dark purple oval button with the text "WELCOME TO SITCollege" in white.

FIGURE 8.3: Authentication Form

This is the authentication form. In this form there are two buttons Teacher and Student.

8.4 Teacher Login-1

A screenshot of a login form titled 'Teacher Login-1'. It features two input fields: the top one contains the text 'Admin' and the bottom one contains five dots, representing a password. The password field is highlighted with an orange border. Below the fields is a grey button labeled 'Login'.

FIGURE 8.4: Teacher Login-1

This form is open after clicking on Teacher button in the Authentication Form. Fill the Name and password and click on the Login button.

8.5 Teacher Login-2

A screenshot of a form titled 'Teacher Login-2'. It contains two buttons: 'Attendance Sheet' and 'Get Report'. At the bottom, there is a dark grey oval button with the text 'Admin Login Successfully'.

FIGURE 8.5: Teacher Login-2

When teacher filling information in the Teacher Login form then it display message Admin Login Successfully and it opens the teacher login-2 form.

8.6 Student Module

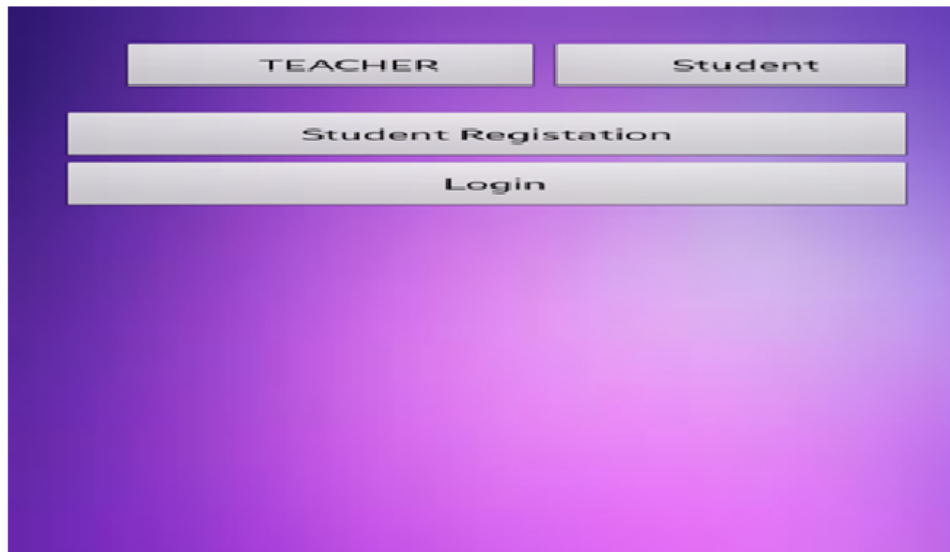


FIGURE 8.6: Student Module

When we click on the Student button then this form is open. There are also two button Student Registration and the Login.

8.7 Student Registration Form

A screenshot of a web application interface for student registration. The form has a purple gradient background. It contains the following fields and controls: 'Roll Number' with the value '69', 'Name' with the value 'priti Rajput', 'Year' with a dropdown menu showing 'B.E', 'Branch' with a dropdown menu showing 'Computer', 'Semester' with a dropdown menu showing 'II', and 'Password' with a masked input field showing '.....'. At the bottom, there is a 'Register' button.

FIGURE 8.7: Student Registration Form

In this form student fill the basic information like Roll Number, Name, Year, Branch, Semester and Password and then click on the Register button.

8.8 Student Login Form

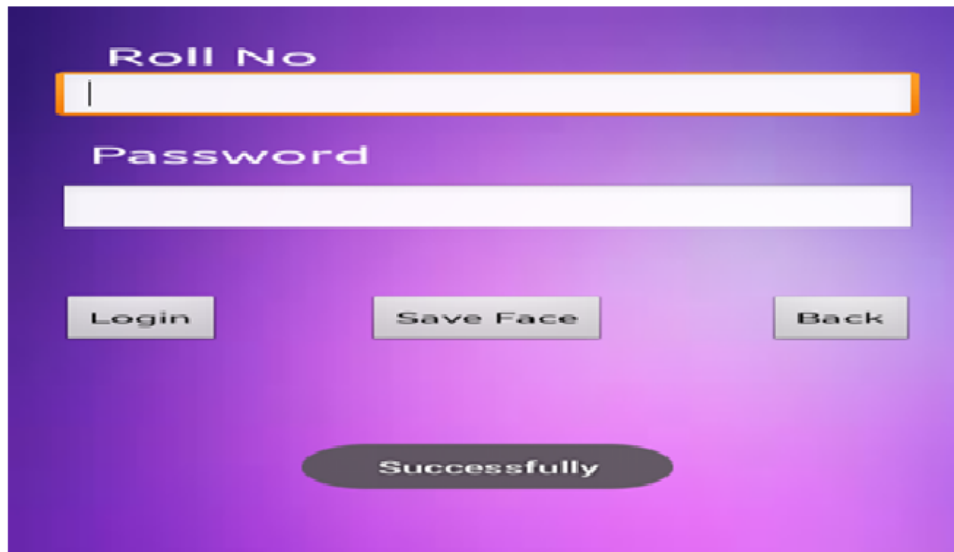
The screenshot shows a login form with a purple gradient background. At the top, there is a label 'Roll No' above a white text input field. Below this is a label 'Password' above another white text input field. Under the password field, there are three buttons: 'Login', 'Save Face', and 'Back'. At the bottom center, there is a dark grey oval button with the text 'Successfully'.

FIGURE 8.8: Student Login Form.

This is the Login form. When Student registration get successfully then Login form is open.

8.9 Student Login Form-1

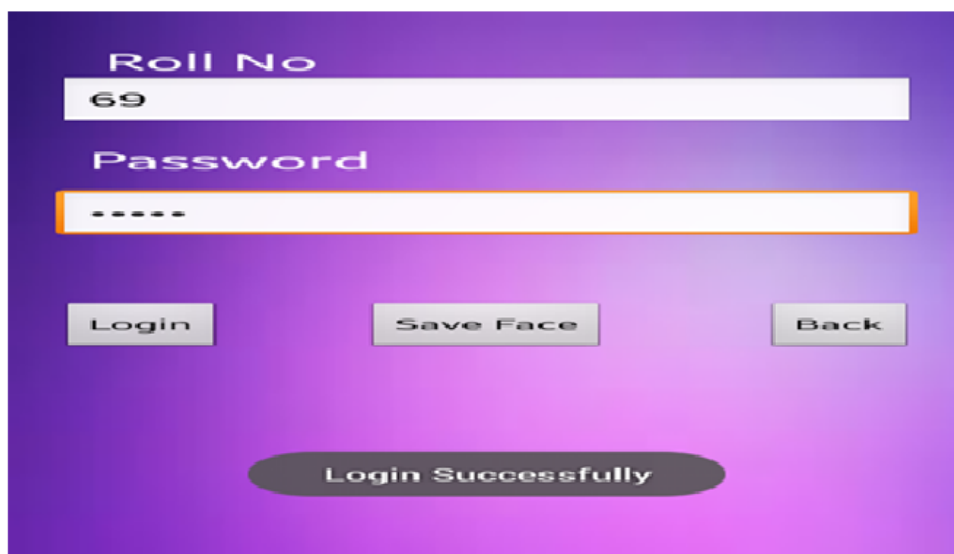
This screenshot shows the same login form as Figure 8.8, but with data entered. The 'Roll No' field contains the number '69'. The 'Password' field contains five dots, indicating a masked password. The 'Login' button is highlighted with an orange border. At the bottom, the 'Successfully' button has been replaced by a dark grey oval button with the text 'Login Successfully'.

FIGURE 8.9: Student Login Form-1

After filling the information in the Login form then it displays the message Login Successfully. Then click on the Save Face button.

8.10 Face Detection

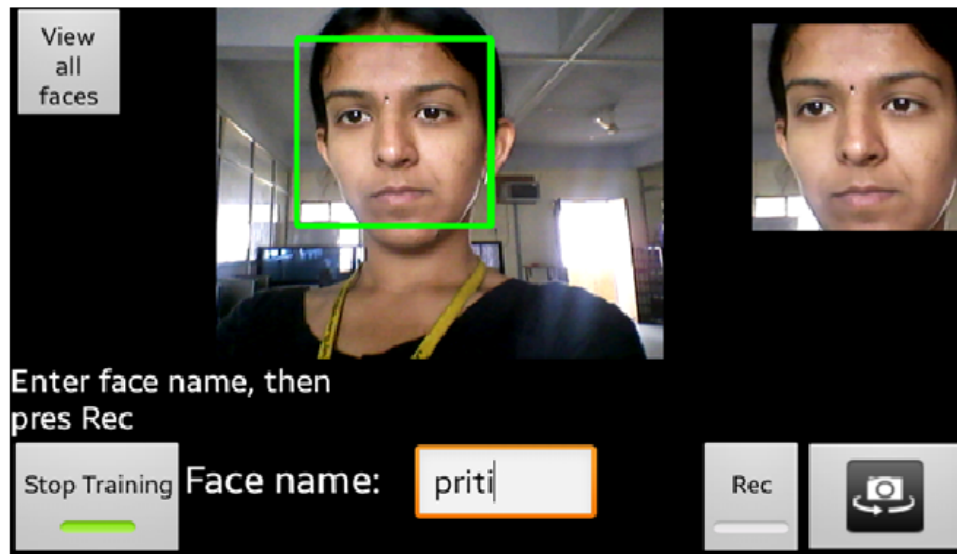


FIGURE 8.10: Face Detection

In this form the student gives his or her face, Name of the face. Then click on the Rec button after that click on Stop Training button.

8.11 Face Recognition-1

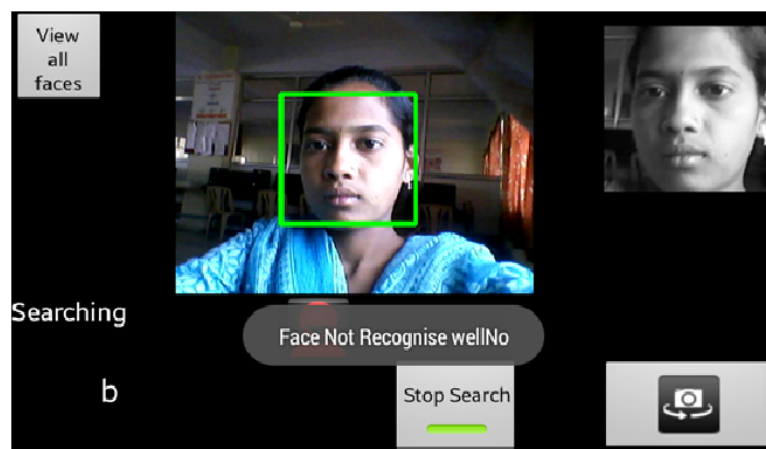


FIGURE 8.11: Face Recognition-1

When the trainee set is completed then it proceeded for the face recognition if the face is not recognized well then it indicate the red and gives the message Face Not Recognize well No.

8.12 Face Recognition-2

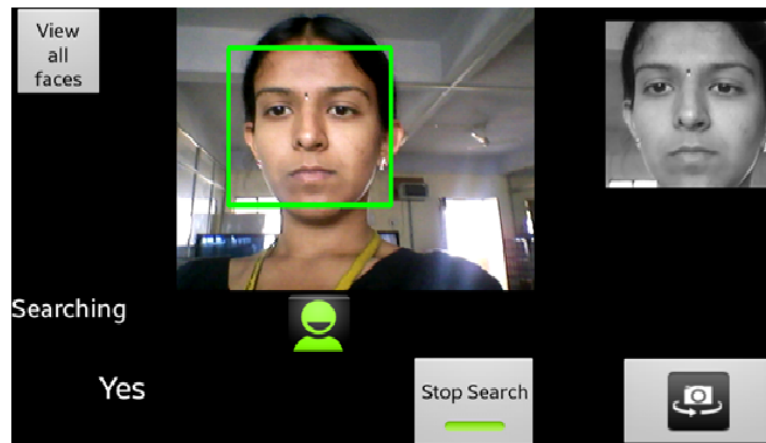


FIGURE 8.12: Face Recognition-2

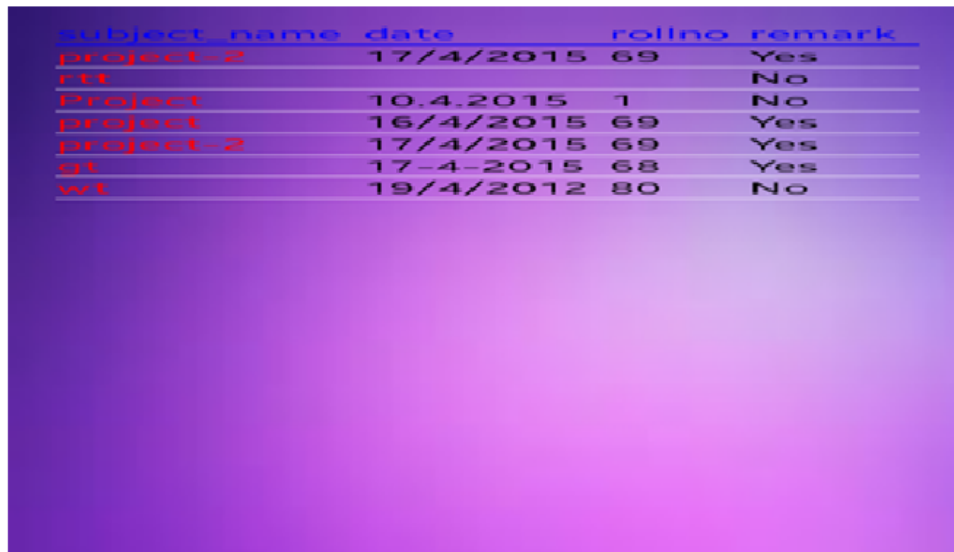
In this form it searching for the face if the face is matched then it indicated green and then it will go to the Attendance Sheet.

8.13 Attendance Sheet

FIGURE 8.13: Attendance Sheet

When the face reorganization is completed then this Attendance Sheet is opened. In this form fill the all field like Subject Name, Date, Student Roll number. After the filling this information click on the Done button it gives the message Complete Attendance Sheet.

8.14 Get Report



subject_name	date	rollno	remark
project-2	17/4/2015	69	Yes
rtt			No
Project	10.4.2015	1	No
project	16/4/2015	69	Yes
project-2	17/4/2015	69	Yes
qt	17-4-2015	68	Yes
wt	19/4/2012	80	No

FIGURE 8.14: Get Report

This is Get Report form. Here display the report of the registered student.

Chapter 9

CONCLUSION

9.1 Conclusion:

The aimed of requirement model for Student Attendance Monitoring System, to help system designer in designing a good attendance system and also conduct them to develop the attendance system in the future. Attendance system are important because, can gives many benefit to schools such as security on attendance, reduce work time on taking attendance and create connection between school staffs and parents. There is no comprehensive and generally accepted manual, on how to design good human factors into computer systems (Shackle, 1984), but there is a lot of guideline on how to design a system.

However, this requirement model can guide any system designers, who want straightly focus to design Student Attendance Monitoring System. Creativity and innovation are required to make a great AMS. The system should be usable. Usability consists of many pieces such as system performance, system functions, and user interfaces organization and so on. In this project has provided a convenient method of attendance marking, compared to the traditional method of attendance system. By using databases, the data is more organized. This system is also a user friendly system as data manipulation and retrieval can be done finally, this attendance system can be improved by adding a feature, where the attendance system indicates, when a student is late for work or classes as the case maybe.

Bibliography

- [1] Seema Rao and Prof.K.J.Satoal. an attendance monitoring system using biometrics authentication. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(4):213–217, April 2013.
- [2] APARNA BEHARA and M.V. RAGHUNADH. face recognition system for time and attendance applications. *International Journal of Electrical, Electronics and Data Communication*, 1(4), 5-May-2013.
- [3] MuthuKalyani.K and VeeraMuthu.A. smart application for ams using face recognition. *Computer Science and Engineering: An International Journal (CSEIJ)*, 3(5), October 2013.
- [4] Deepanshu GOYAL Dr. S. Ramnarayan REDDY and Ankit BANSAL. mobile based attendance management system. *Student Attendance Tracker System in Android, INTERNATIONAL JOURNAL FOR ENGINEERING APPLICATIONS AND TECHNOLOGY (IJFEAT)*, 9:119–124, February 2013.