
Software Requirements Specification

for

Attendance Management System

Version 1.0 approved

Prepared by:

Kuldeep Singh Jangir and Kavish Gambhir

Indian Institute Of Technology , Jodhpur

13th Feb , 2018

Software Engineering Course
CS-223

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 User Documentation	2
2.6 Assumptions and Dependencies	3
3. System Features	4
3.1 Requirement Specifications	
3.1.1 Functional Requirements	
3.1.2 Non-Functional requirements	
4. Other Nonfunctional Requirements	4
4.1 Performance Requirements	4
4.2 Safety Requirements	5
4.3 Security Requirements	5
4.4 Software Quality Attributes	5
5. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the development and operation of the Attendance management system. It will illustrate the purpose of software as well as its limitations/constraints. It will elaborate the response of system towards user interface and any external factor. This document is primarily meant for the client but will also be used by other developers as a scale measure to gauge their progress ,extend the features and maintain the software.

1.2 Document Conventions

This document follows MLA Format. Bold-faced text has been used to emphasize section and subsection headings. Italicized text is to point out words in the glossary.

1.3 Intended Audience and Reading Suggestions

This document is meant for *users*, developers, project managers, testers, and documentation writers. Our Stakeholders, company associated to hardware manufacturing, company providing embedded operating system, students , faculty and teacher assistant may review the document to learn about the project and understand the requirements.

1.4 Product Scope

The Attendance Management System will be used to increase the efficiency of the authority or individual using this software for proper management of attendance in all types of Institutes / organizations / singular *user* intending to manage the records of its subject's attendance in an effortless way.

This Product is specifically designed for the Faculty for Attendance record keeping of the students registered in their respective course.

The faculty can use this software to take a record of the attendance of the batch that may contain any number of students. This software will keep a record for each student and check for its *minimum Attendance criteria* which would be provided by the faculty. The attendance record as a whole can be viewed and modified by the faculty and this software gives access to the students to view the attendance.

If required the faculty can register a new student into the database ; also it provides functionality to remove a student as desired. Subjects falling the *minimum attendance criteria* are segregated in a different list that can be viewed and managed by the client.

1.5 References

This document builds on the following references:

- Software Engineering : A practitioner's approach , Roger S Pressman
- Ian Sommerville, Software Engineering, Ninth Edition, Addison-Wesley, 2011
- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Overall Description

2.1 Product Perspective

The Attendance Management System is intended to replace the manual model of attendance record keeping using paper records. The roll call and paper records are replaced with a single interaction between the Faculty and the Attendance Management System. Faculties will be able to view details regarding attendance of individual students on their device and quickly maintain attendance records. The features expressed in this Software Requirements Specification document are intended to be fully implemented in version 1.0. The system will be developed in such a way to provide easy addition of enhanced features, which may be desired in subsequent versions.

2.2 Product Functions

This product provides the following functions to the *user*:

- keep record of attendance for the subject for the dates provided by the faculty.
- The dates for which the Attendance records are to be kept can be modified by the Faculty and Admin.
- view and modify the Attendance as per the needs for any given Student.
- register a new student in the database and delete an existing student.
- A *minimum attendance criteria* (85 %) and segregate students which lack the criteria.
- give a view only access to the students for to attendance.
- gives access to View a list of *defaulters* to all *users*.

2.3 User Classes and Characteristics

This software gives access to two kinds of *user* classes:

- **Faculty**- They will keep track of the subject's attendance and have the administrative access to modify the attendance , register / deregister , the subjects and set the *minimum attendance criteria* .They have the highest frequency of use for the software and have specific security features designed for it. This class basically has access to all the functionality of the software.
- **user(students)** - All *users* will have the access to view their attendance in a simple interface for all dates which are entered by the administrator.They can also view the list containing data of students failing the *minimum attendance criteria*.

2.4 Operating Environment

This software will operate on all existing operating systems with support to c++ programs and all default hardware components provided as of date with any Computer is sufficient to run this software.

A minimal amount of memory (<10mb) may be required to efficiently run this program and store the Data.

2.5 User Documentation

A presentation and user manual describing the features of this software will be provided along the delivery of the software.

On request help for the given software will be available between 12 pm - 12 am on the following addresses:

gambhir.2@iitj.ac.in

jangir.3@iitj.ac.in

Typical response time is ~2 hours.

2.6 Assumptions and Dependencies

1. Only one faculty is allowed to use the system (course instructor)
2. For the students added after the commencement of course, the classes for which he/she was not registered will be taken as Absent.
3. All students can see attendance of all students.
4. In case of any discrepancy of attendance, student need to contact faculty personally.

3. System Features

3.1 Requirements Specifications

3.1.1 Functional Requirements

Use Case Diagrams are given in Appendix B

3.1.1.1 Faculty : Login

Use Case Name :	Log In
Trigger:	<i>Faculty</i> choose to Log in from Home
Precondition:	Faculty must be logged out (isAlreadyLoggedIn = false) password entered = password stored
Basic Path:	1. Faculty choose to Log in 2. Faculty enters password 3. Login credentials verified 4. Faculty sees Dashboard
Alternative Paths:	NO
Post Conditions:	1. Faculty is logged in (isAlreadyLoggedIn = true) 2. <i>Faculty Dashboard</i> is shown
Exceptions Paths:	1. Faculty enters wrong password and is asked to

	enter password again
Others	None

3.1.1.2 User : View All Attendance

Use Case Name :	View All Attendance
Trigger:	<i>User(Student or Faculty)</i> chooses to view all attendance
Precondition:	None
Basic Path:	1. Click on View All Attendance in Homepage or <i>Faculty Dashboard</i> .
Alternative Paths:	1. <i>Faculty</i> chooses to modify attendance.
Post Conditions:	Attendance Records are displayed on the screen
Exceptions Paths:	<i>None</i>
Others	All attendance will show student details and his date wise record as saved by the faculty

3.1.1.3 User : View Defaulters

Use Case Name :	<i>View Defaulters</i>
Trigger:	<i>Student or Faculty</i> asks to view <i>Defaulters</i>
Precondition:	<i>None</i>
Basic Path:	1. <i>User</i> asks for <i>Defaulters</i> 2. for each <i>student</i> attendance is checked 3. <i>Student</i> details are displayed if low attendance found
Alternative Paths:	1. <i>Faculty</i> logs in and then asks to view <i>defaulters</i>
Post Conditions:	<i>Data containing list of students with low attendance is displayed on the screen</i>
Exceptions Paths:	There are No <i>defaulters</i> .
Others	The records displayed will show the percentage attendance of each <i>defaulter</i> on the screen

3.1.1.4 Faculty : Add Attendance

Use Case Name :	Add Attendance
Trigger:	<i>Faculty</i> choose to add attendance
Precondition:	1. <i>Faculty</i> must be logged in (isLoggedIn = true) 2. <i>Faculty</i> must be on dashboard
Basic Path:	1. <i>Faculty</i> choose to add attendance 2. <i>Faculty</i> enters date 3. Date is validated 4. <i>Faculty</i> enters roll number of <i>students</i> absent 5. Each roll number is verified for existence 6. <i>Faculty</i> chooses to save changes
Alternative Paths:	No
Post Conditions:	1. <i>Faculty</i> is on dashboard 2. Attendance was updated in Database
Exceptions Paths:	1. <i>Faculty</i> choose to stop and discard changes while entering students Absent.
Others	None

3.1.1.5 Faculty : Modify Attendance

Use Case Name :	Modify Attendance
Trigger:	<i>Faculty</i> choose to modify attendance
Precondition:	1. <i>Faculty</i> must be logged in (isLoggedIn = true) 2. <i>Faculty</i> must be on dashboard
Basic Path:	1. <i>Faculty</i> choose to modify attendance 2. <i>Faculty</i> enters roll number of student and date 3. Date and roll number is validated 4. <i>Faculty</i> chooses to mark Present / Absent / Leave 5. <i>Faculty</i> choose to save changes
Alternative Paths:	NO
Post Conditions:	1. <i>Faculty</i> is on dashboard 2. Modification of Attendance was updated in Database

Exceptions Paths:	1. Faculty choose to stop and discard changes while modifying student attendance.
--------------------------	---

3.1.1.6 Faculty : Add Student

Use Case Name :	Add Student
Trigger:	Faculty choose to add Student
Precondition:	1. Faculty must be logged in (isLoggedIn = true) 2. Faculty must be on dashboard 3. Same roll number must not exist in database
Basic Path:	1. Faculty choose to add student 2. Faculty enters Student name and Roll number 3. student roll number is verified in database for repetition 4. Faculty chooses to save changes
Alternative Paths:	None
Post Conditions:	1. Database is updated with new student name and roll number. 2. Faculty is redirected back to it's Dashboard
Exceptions Paths:	1. Student roll number already exists in Database 2. Appropriate message is displayed and control is redirected to the <i>faculty dashboard</i>
Others	None

3.1.1.7 Faculty : Remove Student

Use Case Name :	Remove Student
Trigger:	Faculty choose to remove student
Precondition:	1. Faculty must be logged in (isLoggedIn = true) 2. Faculty must be on dashboard
Basic Path:	1. Faculty choose to remove student 2. Faculty enters roll number of student 3. Roll number is validated from database 4. Faculty choose to remove
Alternative Paths:	None
Post Conditions:	1. Faculty is on dashboard 2. Database is updated.
Exceptions Paths:	1. Student roll number does not exists in

	Database 2.Appropriate message is displayed and control is redirected to the faculty dashboard
--	---

3.1.1.8 Faculty : Remove Student

Use Case Name :	Change Password
Trigger:	Faculty chooses to change Password
Precondition:	1. Faculty must be logged in (isLoggedIn = true) 2. Faculty must be on dashboard
Basic Path:	1. Faculty choose to change password 2. Faculty enters the existing password 3. Existing password is validated from database
Alternative Paths:	No
Post Conditions:	1. Faculty is on dashboard 2. new password is updated in database
Exceptions Paths:	1. Entered password is not correct 2. Appropriate message is displayed and control is redirected to the <i>faculty dashboard</i>
Others	None

3.1.1.9 Faculty : Change Name

Use Case Name :	Change Name
Trigger:	Faculty choose to Change the name
Precondition:	1. Faculty must be logged in (isLoggedIn = true) 2. Faculty must be on dashboard
Basic Path:	1. Faculty choose to change name 2. Faculty enters the new name and existing password 3. Password is validated from database.
Alternative Paths:	No
Post Conditions:	1. Faculty is on dashboard 2. New name is updated in Database
Exceptions Paths:	1. Existing password is incorrect. 2. Appropriate message is displayed and control is redirected to the faculty dashboard

Others	None
--------	------

3.1.2 Non - Functional Requirements

The Attendance management system will run on the computer by using the local hard disk of the machine to support the reading and writing of data from and to a database.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

1. Software should be easy to use and handle.
2. System should require less space to process smoothly.
3. All functional requirements should be fulfilled.
4. System should have at least 128 mb of RAM for smooth functioning.

4.2 Safety Requirements

1. Secondary Power Supply should be there while using the system to ensure no data loss in case of Power failure.
2. There should be no data loss or breach if system is abruptly closed
3. The Database should not be located in a secure directory to prevent intrusions.
4. The Faculty managing the attendance should Log out of the program to ensure that the program is not misused by any other *user*.

4.3 Security Requirements

1. The Database must be stored in a Secure directory to prevent external intrusions and breaches in the data.
2. The faculty Password must be unique and must not be shared with any other *user*.

4.4 Software Quality Attributes

The Product architecture is based on the MVC model which ensures loose coupling and indeed helps in extending the product features in the future releases by minimum changes in the current code.

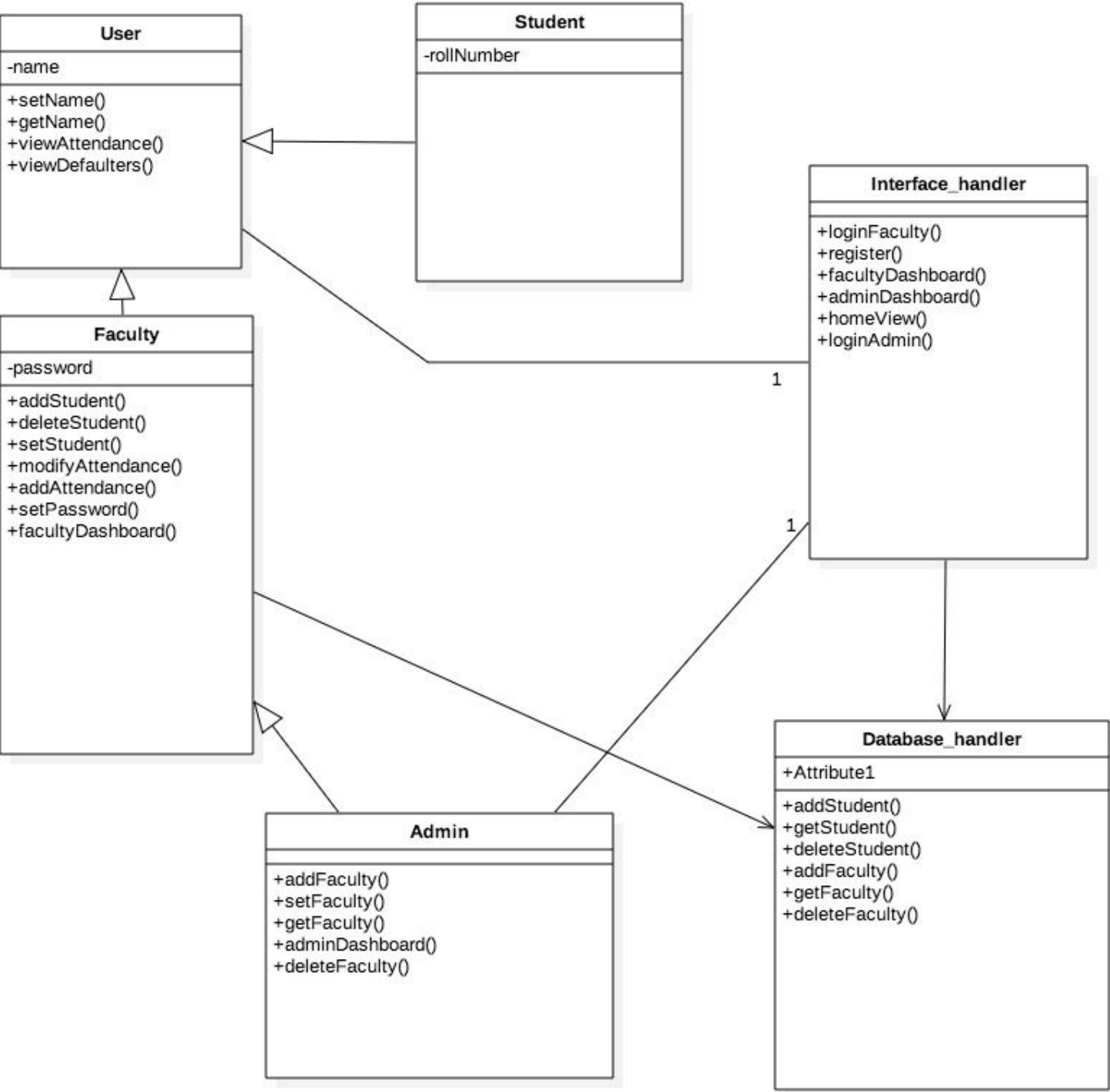
It's reliable to the customers and portable in all ways . The Faculty (main *user*) is the main Model which can be later given extended functionality by modifying the given class with minimum changes and affects to the rest of the code. Faculty remains Logged in unless he/she chooses to logout. This feature enables faculty with OneLogin session. It will eliminate excessive login time at each program run to one time login.

Appendix A: Glossary

1. User : Student and Faculty using the system. Typically anyone who is using the software gets the features of *users* without any authentication
2. Student : The subjects whose attendance has to be managed by the primary *User* i.e. Faculty.
3. Faculty: The primary *user* who is managing the attendance. He has special
4. Faculty Dashboard: The welcome screen presenting all the features available to faculty.
5. Minimum Attendance Criteria: The attendance value set by the faculty to segregate students in the database.
6. *Defaulters*: Those students who lack the *minimum attendance criteria* i.e. whose aggregate attendance is less than the *minimum attendance criteria*.

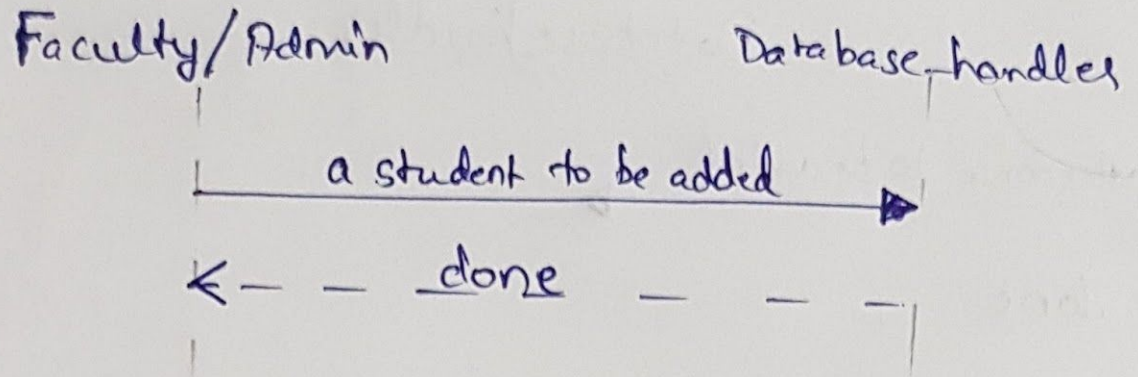
Appendix B: Analysis Models

Class Diagram:

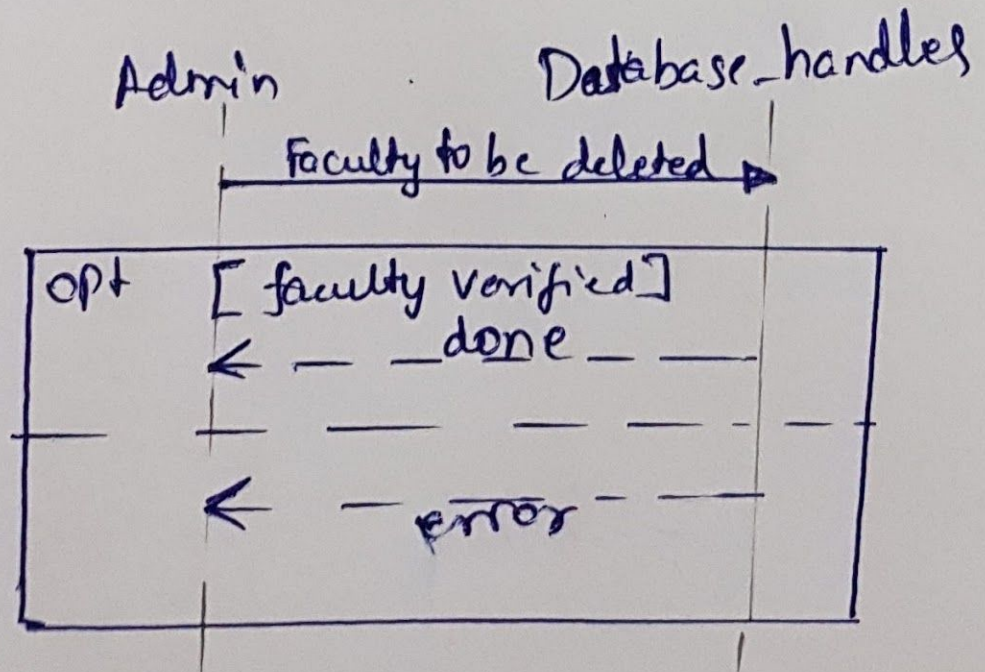


Sequence Diagrams:

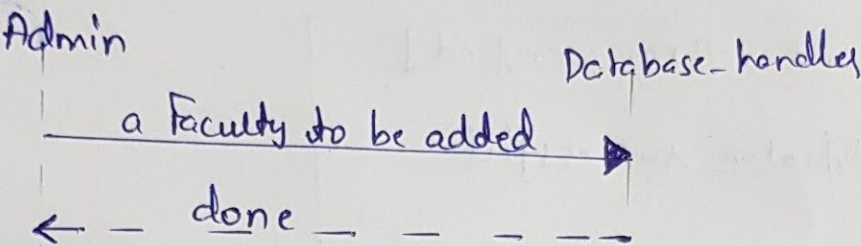
Add Student



Delete Faculty

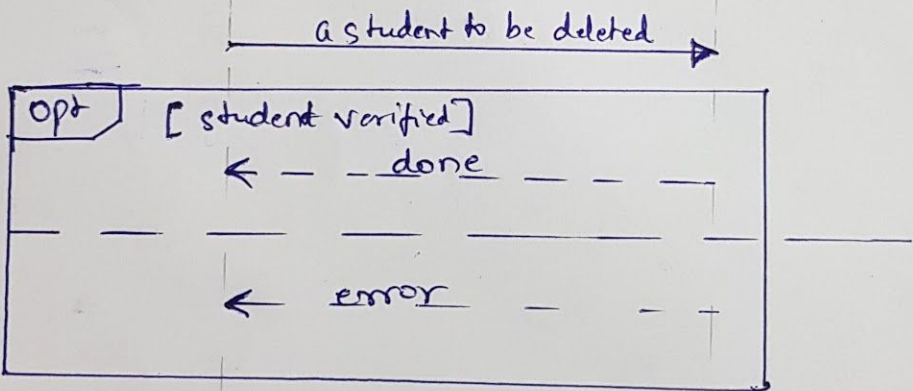


Add Faculty



~~Delete Faculty~~ Delete Student

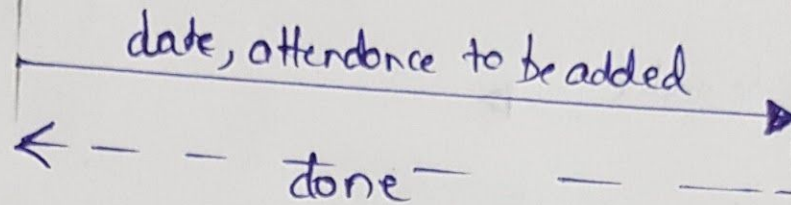
Admin/Faculty Database-handles



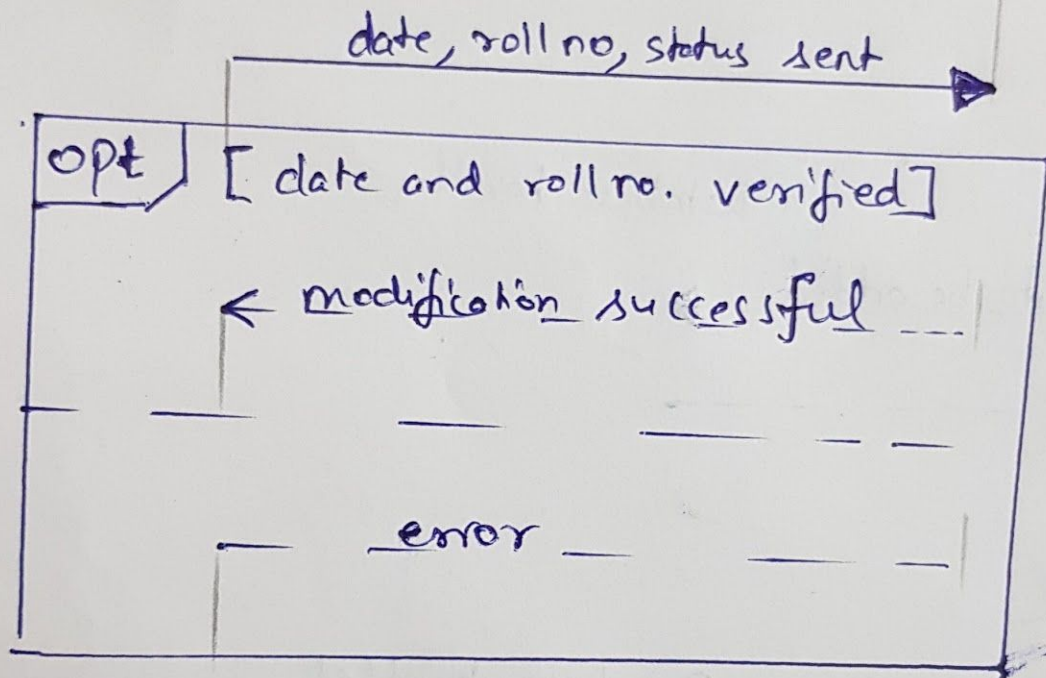
Add attendance,

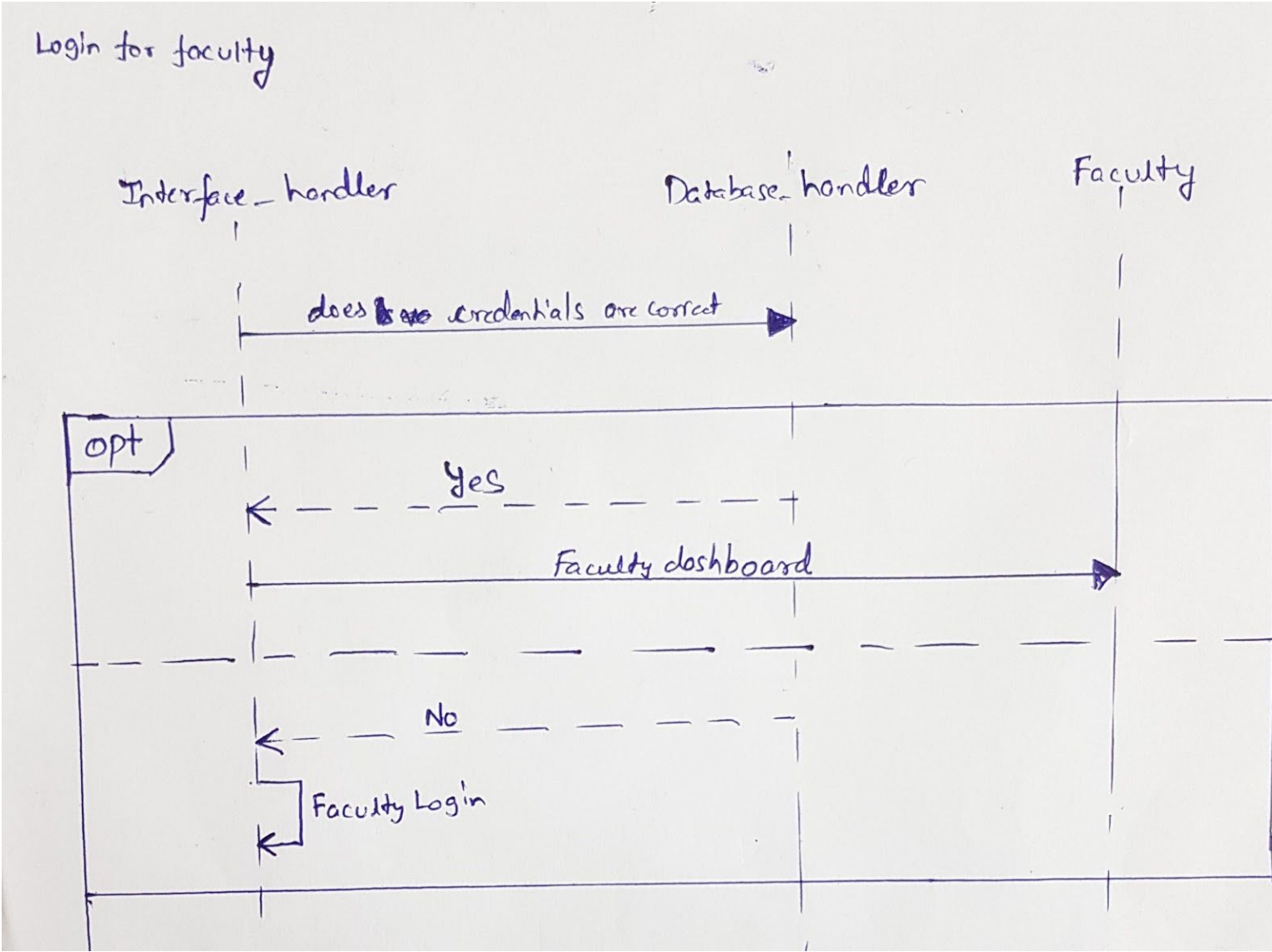
Faculty/ Admin

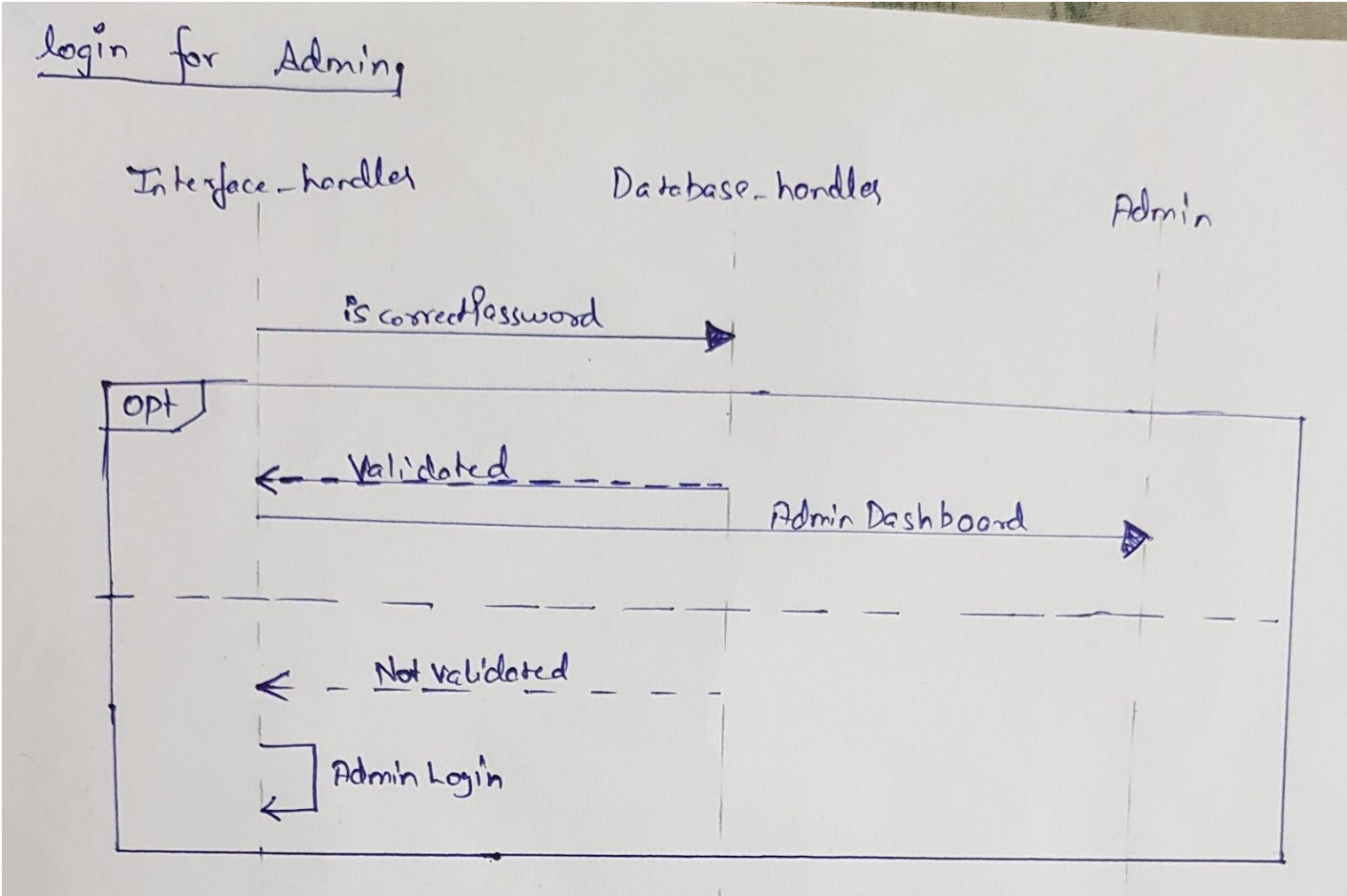
Database_handler



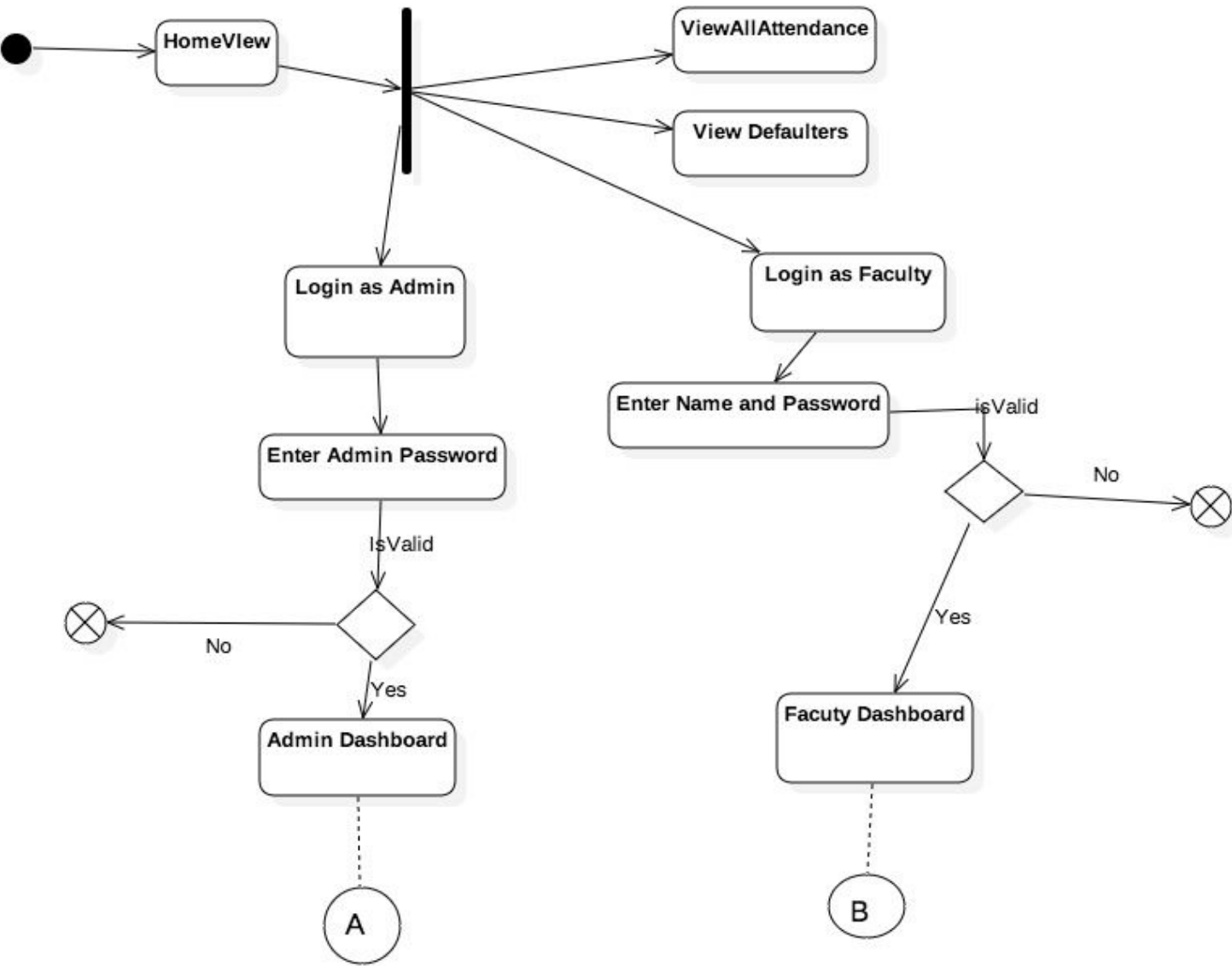
Modify attendance

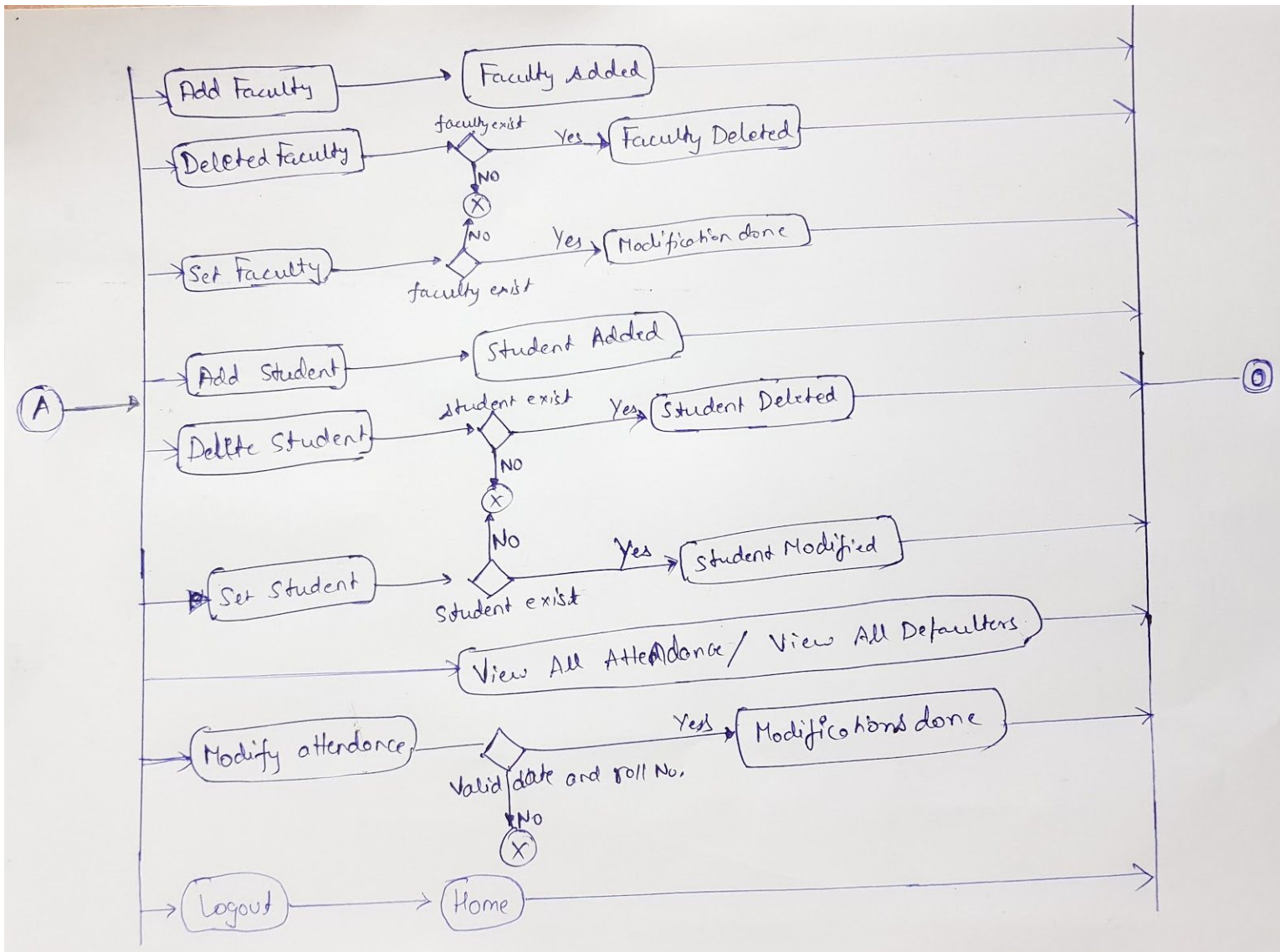


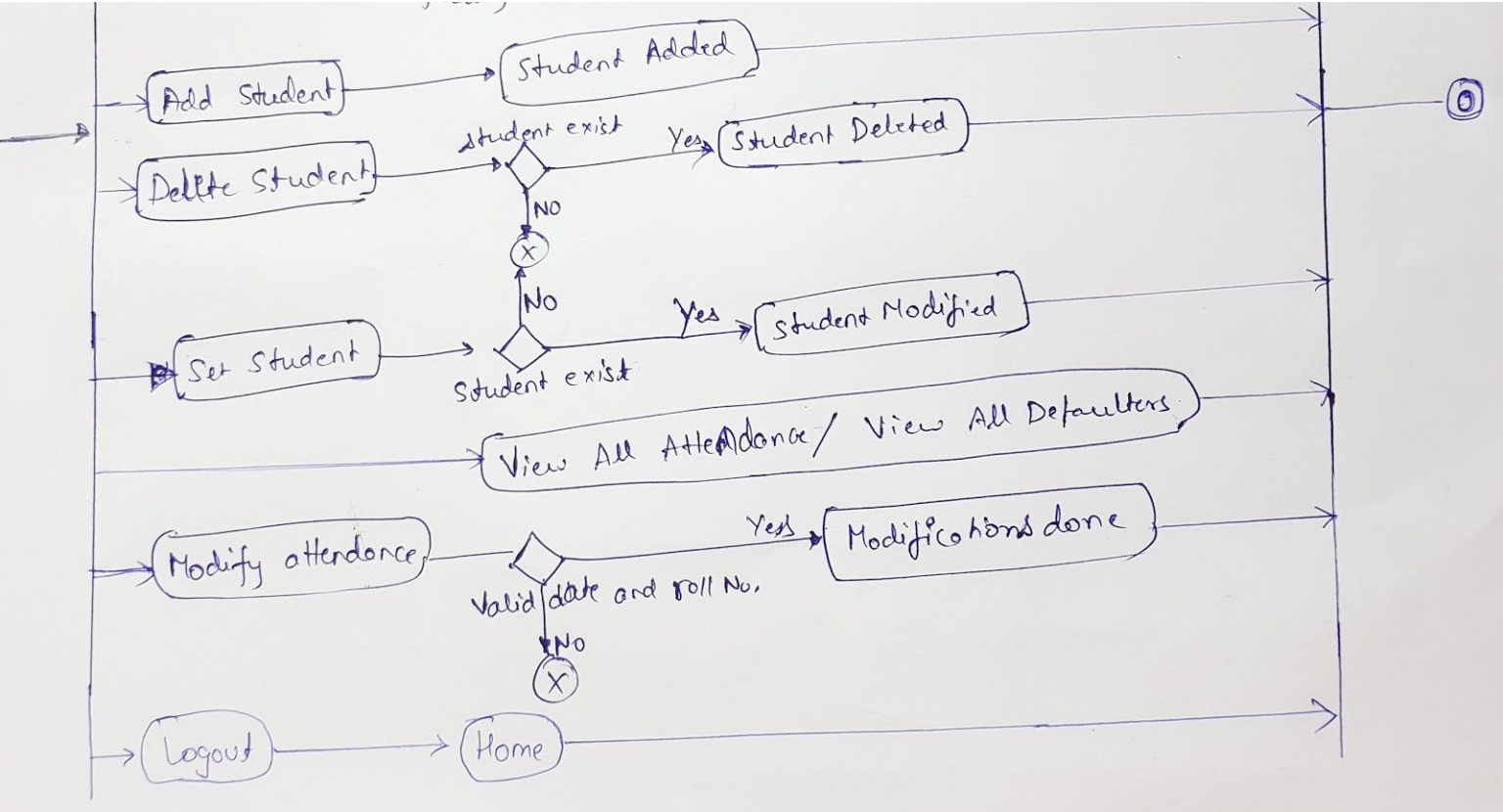




Activity Diagrams:







Use Case Diagram:

