Student Attendance Management System

Software Requirements Specification

Version 1.0
Status: Approved

Prepared by

Amritpal Singh, 2017csb1068
Divyanshu, 2017csb1074
Jai Garg, 2017csb1081
Mehakjot Singh, 2017csb1090

Group 1 SRS clients

Indian Institute of Technology Ropar

25 January, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version	Date of Approval
Group 1 SRS clients	25 January, 2020	First SRS	1.0	29 January, 2020

1. Introduction

1.1 Purpose

The main purpose of this specification is to provide a detailed description of the Student Attendance Management System and help people who will work on this system to maintain the objectives and get started working on this project. It will direct people who will work on this project step by step through the process so that they can finish it successfully. It will also describe specific details and the purpose of doing every step of this project. This specification includes details for the whole system and is not limited to its web application counterpart. This specification is for the first release (version 1.0) of the software.

1.2 Document Conventions

The main headings are written in "Heading 1" style, while their subheadings are written in "Heading 2" style, where the former is a little bigger in font size than the latter and both are bold text. Below each subheading is the description for that subheading written in "Normal text" style which is not bold text. To represent distinctive remarks, text is often highlighted with attractive colors. *Italicized* or underlined text also marks important points and whose actual importance seems obvious from the context. For example, a variable name, say *itr*, will be written in italicized text and it seems obvious from the context that the variable name is exactly the same as the italicized word.

1.3 Intended Audience and Reading Suggestions

This document is written for the project managers; programmers; designers; testers; marketing staff; developers; documentation writers and end users such as – *a. the admin* who will manage all the data of the system such as the data of the courses, the data of the students, and the data of the teachers; *b. the students* who will use the system to view their attendance data; and *c. the teachers* who will use this system to mark the attendance of the students enrolled in the concerned course.

This document consists of the various steps and procedures for the construction of the Student Attendance Management System and particularly the website and the mobile application. Following sections describe the *product scope*, *overall description*, *external interface requirements*, *domain model*, *use cases*, *other non-functional requirements* and *other requirements*. Not all sections might be of interest to all the audiences. Please refer to the table below to find which sections might be of most interest to you.

ID	Stakeholder	Description	Suggested sections
S1	Project Manager	Estimating the quote of the project, planning resources and the timeline of work.	Whole document

S2	Programmers	Review the overall descriptions to have a clear vision on what to build and how to proceed in a systematic manner.	Overall Description, Domain Model, Use Cases
S3	Designers	Identify the design aspects that are expected to be followed and use these to match with the practices to be followed to accomplish them.	Overall Description, External Interface Requirements
S4	Testers	A base for the testing strategy so that testing becomes more methodically organized. Making test-plans and test-cases.	Use Cases
S5	Marketing Staff	To verify that what they might claim in an advertisement is actually present in the system or not.	Use Cases, Other non-functional Requirements, Other Requirements
S6	Developers	Forming the accurate vision of the project, detailed functional and non-functional requirements, review project's capabilities and understand where their efforts should be targeted to improve or add more features to it for future development.	Domain Model
S7	Documentation writers	Review user interfaces and overall description, which are written in this document, to understand what needs to be focused upon while writing a documentation for the system.	Overall Description, External Interface Requirements, Use Cases
S8	End users	To get the detailed description of the system and how to use it and what it can and cannot do. Based on this document the Terms of Service and Privacy Policy are created.	Overall Description (User Documentation), Use Cases

1.4 Product Scope

The final product will enable the registered institute to manage their attendance system online which will not only increase the accuracy and credibility of the attendance system, but will also save a lot of time which would otherwise have been spent on taking conventional attendance. It would also decrease the teachers' efforts by managing the attendance percentages of every student and highlighting the students with low attendance. The teacher need not calculate every student's attendance manually. It would also help the students to

have a constant check on their attendance in various courses which will motivate them to attend more classes. Aim is to design a foolproof system so that there is no scope for errors and hence the system cannot be blamed for low or high attendance of any student.

1.5 References

More information about the system will be made available in the future documentation and also on the website where it will be hosted, whose link will also be provided in future documents.

2. Overall Description

2.1 Product Perspective

This document is the initiation of a new system called the Student Attendance Management System for the overall management of attendance in an academic institute, transforming the way attendance is taken, handled, and shown to students. It also aims to increase the transparency of the attendance system, and also warn the students right from the start if their attendance falls below a particular barrier. This system will eliminate the manual attendance system in which a lot of time and energy is wasted, which will be replaced by the event of clicking photographs of the students present in the class so as to cover all of them.

The main components of the system includes a website and a mobile application. The students, teaching assistants and teachers will interact with the mobile application for the tasks of *1. taking the attendance by clicking photographs of the students and selecting the event type (lecture/tutorial/lab), which will be sent to an API which returns the vector of students who were in the photographs*, *2. viewing the attendance* and *3. viewing stats about the attendance*. The website will be used by the admin to add courses', students' and teachers' information; relate courses to teachers, teaching assistants and students; and to view all information of the system.

Product Functions

We will have a web application for the **admin** (although other users can also log into the website and check various data and stats, but the teacher or the teaching assistant cannot take attendance on the website) and mobile application for the students, teachers and teaching assistants. The administrator will have the option to add students, teachers and teaching assistants to the database. He will be responsible to assign teachers and teaching assistants to various courses, and to add the enrolled students to the courses during the start of the semester (when add/drop is complete), and will also be responsible to end the course when the semester ends (so that the data of the courses can be moved to another storage for future references).

We will have a mobile application. For the **teachers**, they will first select a course for which they want to view the stats or take the attendance. After selecting the course, they will have

the option to take attendance (by mentioning whether it is a lecture, tutorial or lab session) by clicking the photographs, or to view attendance stats for that course. Similarly **students** will have to select the course for which they want to view their attendance stats on the mobile application. The **teaching assistants** will be able to behave either as a teaching assistant for a course (where they will be able to take the attendance for the lecture/tutorial/lab for that course) or a student.

The data flow diagrams and sequence diagrams are in Appendix B.

2.2 User Classes and Characteristics

ID	User classes	Description
U1	Admin	Admin will have the right to include, erase or update the data present in the database through the web application.
U2	Instructor	Instructor will only be able to view the data uploaded by the admin. He can click the pic in the app and only Instructor will be the one who can mark the attendance.
U3	Students	Students can only view the information uploaded by the admin and the teacher

2.3 Operating Environment

For the website, the below mentioned minimum requirements are suggested.

Operating System - Windows / Mac / Linux (any supported version).

Hardware - Core 2 duo processor.

Database - MySQL or Postgresql.

For the mobile application, the below mentioned minimum requirements are suggested.

Operating System - Android (Version 5.0 or higher) / IOS (Version 8 or higher).

Hardware - Any dual core processor.

2.4 Design and Implementation Constraints

The website will be made on Django (which is a framework in Python). This limits the designers and implementers to restrict their thinking to these frameworks and to design the system accordingly. No user should be able to change the attendance and the system should be foolproof. Once marked, attendance cannot be unmarked for security reasons. The system depends entirely on the facial recognition API to mark the attendance of the students. Therefore the API must be accurate so that no errors arise from its end.

2.5 User Documentation

Q1. How do I create an account?
Ans. You do not need to create an account. The admin will get all the teacher and student details from the institute and will give you your credentials via email.

Q2. How to take attendance?
Ans. Login with your teacher or teaching assistant account on the app. Select the course for which you want to take the attendance and click the camera icon. Choose whether it is a lecture, tutorial or lab session. Click photographs until every student in the class has been captured in at least one photograph. Click the "Mark Attendance" button to mark the attendance of the students automatically.

Q3. What if I accidentally clicked on some other course that I teach on the main page?
Ans. Don't worry! On the course page, you will see a dropdown where you can change your course. Click on the course for which you want to take attendance and you are good to go.

Q4. How do I see the attendance of my students for a selected course?
Ans. Some common stats will be shown right on the course page. In addition to that, there will be two buttons for additional information. One button will be for displaying the day by day attendance of all the students, and other button will display the stats of a particular student which you will select on the next screen after clicking this button. Also, there will be a button to view the students with attendance less than a barrier set up by the instructor at the start of the semester.

Q5. How can I see my attendance?
Ans. Some common stats will be shown right on the course page. In addition to that, there will be buttons to check your day wise attendance for the selected course.

Q6. What to do if my attendance is low?
Ans. Try to attend more classes so that your attendance goes up. Every other decision lies in the hands of the institute and the teacher.

2.6 Assumptions and Dependencies

Following are some assumptions that are made for the system designers' and implementers' help.

1. In case of data corruption at the admin side, there is a backup on the cloud (or somewhere else) from where data can be restored.
2. The data referenced by the Student Attendance Management System is correct.
3. The end user must have a basic level of computer knowledge to use the website and the mobile app.
4. The admin must not delete files unknowingly which can harm the working of the database.
5. Admin has all the data of the courses, teachers, students and teaching assistants (and their relations).

Following are some dependencies that are made for the system designers' and implementers' help.

1. Student information being utilized for student identification is subject to data in a database which is outside of the capacities of the Attendance Management System.
2. Attendance record depends upon the teacher's use of the app for each class period.
3. The hardware and software requirements must be met before installing the mobile app.

3. External Interface Requirements

3.1 User Interfaces

1. *Mobile App/Web Interface:*

    a. *Login page*

    When mobile application is opened for the first time, the app window will appear with the app's name and an input box asking for the username and password of the user. (Refer Appendix C for prototype screen)

    b. *Teacher login*

    When teacher logs into the system, a page containing general header information such as instructor's name, logout button and the application name will appear. The main content will be a list of courses that the instructor registered to teach that semester in the form of cards. (Refer Appendix C for prototype screen)

    c. *Particular course by instructor*

    After login, when the instructor clicks on a particular course, a page containing details of that course, like course id, course name, list of students enrolled in that course, attendance in the last lecture/tutorial/lab, overall attendance stats and other such information will be displayed. It will also

contain buttons which will help to see day by day attendance of all the students, see statistics of a particular student, see students with attendance below set criteria and click picture for today's attendance. (Refer Appendix C for prototype screen)

d. *Student login*

When a student login into the app, page containing header having student name, student ID and list of courses in which he is enrolled is displayed on the screen. (Refer Appendix C for prototype screen)

e. *Particular course by student*

When a student clicks on a particular course after logging into the app, a page containing course name, id, overall attendance and day by attendance of the student for that course is displayed on the screen. (Refer Appendix C for prototype screen)

f. *Upload photo (not for web interface)*

When instructor clicks the button to take attendance, a window will appear with *option to add new photographs* which the instructor should select until all the students are covered in at least one of the photographs, and the *option to select the event type* (lecture/tutorial/lab). When satisfactory number of photographs are clicked, the instructor should click the 'Mark attendance' button.

g. *TA login*

When TA logs into the app, he/she will have the option to navigate between instructor page (list of courses allotted to that TA) and student page (list of courses that TA has enrolled in as a student).

h. *See day by day attendance by instructor*

A button which navigates to a window containing a list of dates along with the overall attendance of that day will be displayed on the course page. When the instructor clicks on a particular day on that page, a list of students with their attendance status will be shown on a new screen.

i. *See student attendance by instructor*

A button which navigates to a window containing a list of dates when lecture/tutorial/lab were conducted with attendance status for each day for a particular student will be displayed on the course page. (Refer Appendix C for prototype screen)

2. *Admin Interface (only on website)*

a. *Login page*

   On the web, home page containing login and password fields will be displayed where admin can log in by entering his credentials.

b. *Admin home page*

   Various options like add student, add course, add TA, add teacher, end course, assign students to course, assign teacher to course, assign TA to course, view all courses and data and logout will be displayed.

The option to take attendance will not be available on the website, it will only be available on the mobile application.

Hardware Interfaces

User Side:

For mobile application: Android or IOS smartphone having a working camera.
For website: A computer with a web browser.

Server Side:

Nginx or Apache server capable of hosting Django website.

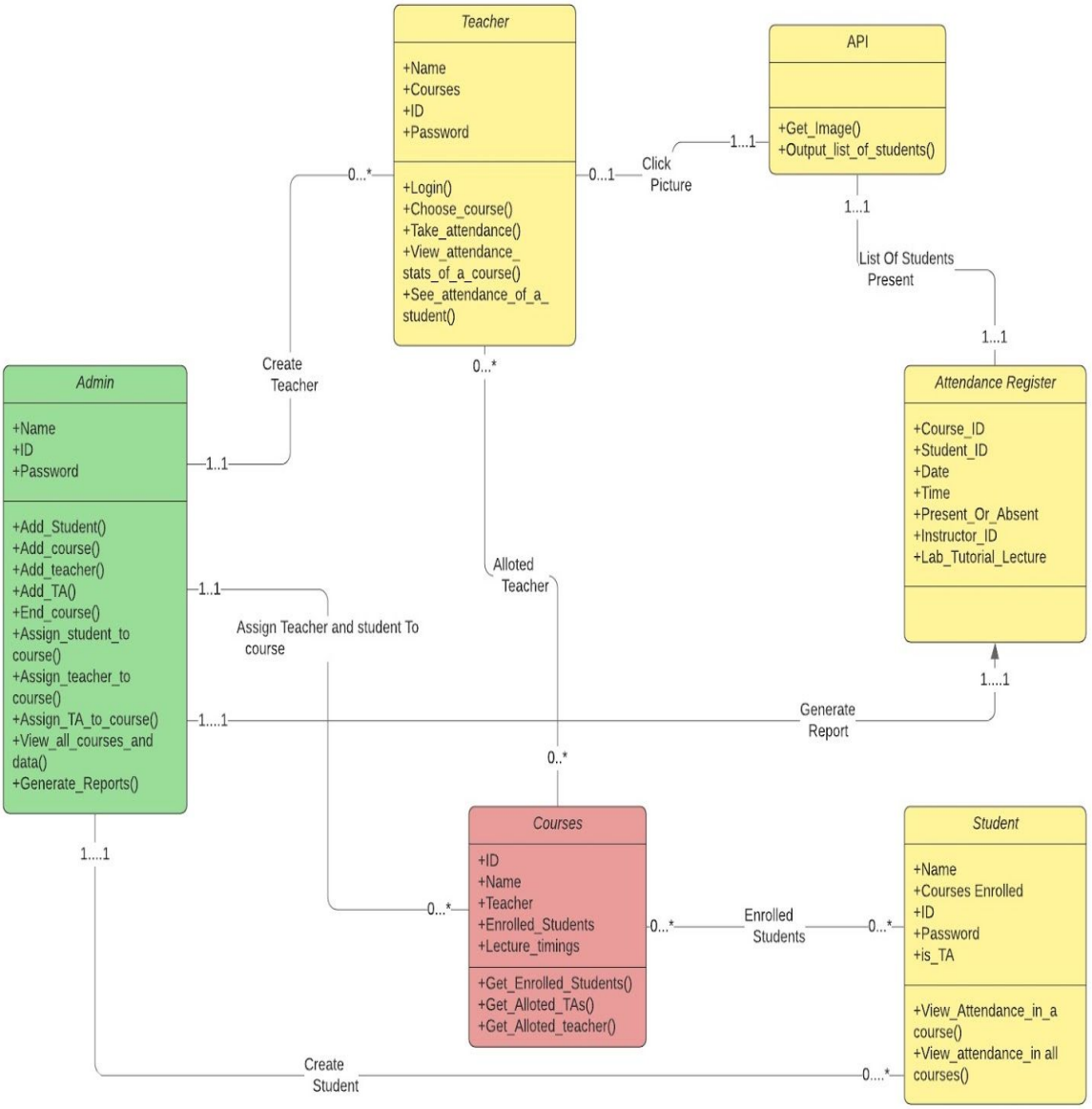
3.2 Software Interface

Face Recognition API: The system will interact with Face-Recognition system for collecting list of students in the photograph. Initially the admin will register each student through the api to link each student's face with their entry/registration number for identification. Then the system will send a ".jpeg" format image to the API (using Django REST Framework) and receive the list of the students by their entry/registration numbers.

Database: The system will allow the admin to enter the course and student information into the database. The system will also allow the users to view relevant attendance reports from the database.

Communications Interfaces: The system shall use HTTPS protocol for communication over the internet.

4. Domain Model



5. System Features (Use Cases)

5.1 Logging In

Brief Description:	1.0 Logging Into the system
Business Trigger:	Either of the 3 follow: 1. The user opens the web portal. 2. The user launches the mobile app for the first time. 3. After logging out from the portal or the app.
Preconditions:	The user has been registered on the system and has the login credentials.

Basic Flow: The user will come across the login page and provide correct credentials, to reach the respective welcome page in accordance with their post.

Assumptions:

1. The credentials to the admin will be provided directly by the developers and then admin will register the teachers and students into the system and provide them the credentials.
2. Admin can only login through the web portal.

Line	System Actor Action	System Response
1	As the user launches the app or the web portal is opened, the user will come across login page.	The system will ask for username and password. The system will provide an option “Forgot Password” if the user that would help the user to reset his/her password.
2	Puts in credentials and press the login button.	The system takes the user to welcome page according to their post as(student/teacher/TA/admin).

Post Condition:	Teacher will land on teacher welcome page, student on student welcome page and TA will be on TA welcome page and admin will land on admin welcome page.

Alternate Flow (AFx):
1.1 Incorrect username/password.

If at line 2 in Logging In Use Case, the user enters incorrect password/username.

Line	System Actor Action	System Response
1	After the user presses the login button	The system will display an error message mentioning wrong credentials.

The use case Logging in restarts at line 1.

Post Condition:	Use Case Logging In restarts.

Prototype Screen:
Refer Appendix C.

5.2 Take Attendance

Brief Description:	2.0 Take attendance by the Teacher/TA.
Business Trigger:	Teacher or TA wants to mark attendance of the class.
Preconditions:	The user has opened the app and logged into the system by Use Case 1.0

Basic Flow:

- The user will choose the course for which attendance needs to be taken.
- The user will click one or more photographs of the class.
- Submit them for marking attendance.

Assumptions: The mobile supports proper functioning camera and permission for using the

Line	System Actor Action	System Response
1	The user on the welcome page will receive the list of courses which they teach, they will choose the one for which attendance needs to be taken	The system open the course id page of the chosen course. The page will provide various options including one to take attendance(photograph). Refer Appendix C.
2	The user chooses to take attendance.	The app opens up upload photo interface. Refer Appendix C.
3	The user clicks one or more photographs and selects send for marking.	The app asks for confirmation for sending the photographs.
4	The user confirms for sending the photographs.	The app provides message for successful marking of the attendance.
Post Condition:	The app returns to course page after displaying success message.	

Alternate Flow (AFx):
2.1 Unsuccessful marking of Attendance

In use case 2.0 line 3 after the user has sent the photographs for marking, if the api is sends to the system some error message.

Line	System Actor Action	System Response
2	User sends the photograph for marking	The system displays an error message and return to upload photo interface for attendance.
The system restarts at the upload photo interface at line 2 of the use case 2.0.		
Post Condition:	The user is expected to retake the photographs for marking the attendance.	

Business Rules:

1. After the user sends the attendance for marking, the system sends the photographs one by one to the FACE RECOGNITION API and receive the list of students from it.
2. The system records the list of the students and marks the attendance of each student enrolled into the course and provided by the API also noting the date and time for the attendance.
3. Even if a student appears in multiple photographs in a single deck of submission, the attendance of the student will be marked only once.
4. For all the students enrolled in the course and not sent by the API will be marked absent.

5.3 Add Instructor (or student) to the System

Brief Description:	3.0 Add instructor (or student) by admin to the system.
Business Trigger:	A new instructor (or student) has to be added into the system.
Preconditions:	The database and the system must be appropriately available and the credentials provided by the instructor (or student) are correct.

Basic Flow: Instructor (or student) provides his details to the admin, then the admin creates a profile of the instructor (or student) using these details in the database.

Assumptions: The credentials provided to the admin by the instructor (or student) are verified by the admin and then only admin creates the profile of the respective instructor (or student).

Line	System Actor Action	System Response
1	Admin click on create instructor (or student).	The system takes admin to screen asking for instructor (or student) credentials.
2	Admin enter the instructor (or student) details.	All entered credentials by the admin are displayed on the screen.
3	Admin clicks the create instructor (or student) button.	The screen displays user is created successfully.
Post Condition:	Instructor (or student) will be successfully registered into the database.	

5.4 Add Courses

Brief Description:	4.0 Add courses by the admin
Business Trigger:	A new course has to be added into the system.
Preconditions:	Admin has already logging into the system.

Basic Flow: Admin enters the instructors ids, students ids, course id, semester no., academic year, course structure, attendance policy (can be changed by the instructor), TA's in the course.

Assumptions: The instructor and the list of the students to be registered are available with the admin.		
Line	**System Actor Action**	**System Response**
1	Admin click on add courses.	The system takes admin to screen above mentioned credentials.
2	Admin enter all the credentials asked.	All credentials entered by the admin are displayed on the screen.
3	Admin clicks the add button.	The screen displays course is added successfully.
Post Condition:	Course will successfully registered into the database and will be successfully allotted to the respective instructors and students.	

5.5 View Attendance Reports (Teacher/TA)

Brief Description:	5.0 View Attendance Reports for specific course of class as a whole.
Business Trigger:	If the teacher/TA wants to view the attendance of the course in the system.
Preconditions:	The teacher/TA has logged into the system and the TA is on TA interface rather than student interface.

Basic Flow: The user chooses the course for which they want to view reports, and chooses to view day by day attendance for the whole class.		
Assumptions:		
Line	**System Actor Action**	**System Response**
1	The user chooses the courses for which they want to view the reports.	The system open the particular course page id.
2	The user chooses the option to view day by day attendance of the complete class.	The system will display the view class attendance page (refer Appendix C)
3	The user chooses to view for particular date.	The attendance for particular date will appear.
Post Condition:	The user can view the reports of the class/lab/tutorial for each date the attendance has been marked through the system.	

Alternate Flow (AFx): 5.1 View Lists of Students		
If at line 3 in 5.0, the user clicks view list option to see the lists of the students for particular event of the date.		
Line	**System Actor Action**	**System Response**
1	The user clicks view list	List of students who attended the event will appear on the screen (dialogue box). Refer Appendix C.
Post Condition:	The user sees the lists of the students.	

Alternate Flow (AFx): 5.2 View Attendance of Particular Student		
If at line 2 in 5.0 the user wants to check the attendance of a particular student in the course.		
Line	**System Actor Action**	**System Response**
1	The user clicks button of view for particular student	The system show interface of attendance of particular student in the course. Refer Appendix C.
Post Condition:	The user can see the student's day by day attendance report for each event of the course independently and overall attendance. Also low attendances will be highlighted.	

Alternate Flow (AFx): 5.3 See students with low attendance.		
If at line 2 in 5.0 the user wants to check the students with the low attendance according to the criteria set by admin/teacher.		
Line	**System Actor Action**	**System Response**
1	The user clicks the button of students with low attendance.	List of students who have low attendance in any of the events will appear along with their attendance percentage.
Post Condition:	The user sees the lists of the students	

Sub-Flow: 5.4 Choose a student from the list of students		
Line	**System Actor Action**	**System Response**
1	In the use casese 5.1,5.3 if the user chooses a student from the list of students.	The system will go to the use case 5.2, and show the attendance of the particular student.
Post Condition:		The use case 5.2 will start.

5.6 View Attendance Reports (Student)

Brief Description:	6.0 Attendance Report for Student
Business Trigger:	If a student wants to check their attendance performance in all/any courses they have been enrolled.
Preconditions:	The user has logged into the system and is waiting on the student page.

Basic Flow: The user chooses from the list of the courses to see their attendance for that course.		
Assumptions: NIL		
Line	**System Actor Action**	**System Response**
1	The user chooses the particular course from the list of courses which they have enrolled.	The day by day attendance of the student for that particular course page will be displayed. Refer Appendix C.

Post Condition:	The student can see their day by day attendance report for each event of the course independently and overall attendance. Also low attendances will be highlighted.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

a. The application must be able to operate concurrently on multiple systems especially during college hours (9:00 am to 6:00 pm).
b. Initially, the application must be able to support a maximum of 1000 faculties (or courses) and 20000 students (Further this capacity can be changed according to requirements).
c. Once the image is uploaded, the application must be able to mark the attendance of all the students within 5 seconds.
d. Email must be sent to all students confirming their attendance ie. present or absent within half an hour once attendance is marked by instructor.
e. Queries upon the database should be performed within 5 seconds.
f. We will have a master table which contains data of each and every attendance at the institute (in the current semester) because the design of the system demands this. It is estimated that the entries in this table can be as high as one billion in the worst case. To handle such enormous data, either distributed systems can be used, or some smaller tables which contain regularly updated cached data can be made (such as the number of students in a course can be kept in a table called course_strength).

6.2 Safety Requirements

a. System must be protected by at least 2 layers of passwords.
b. As the application is connected via LAN appropriate antivirus must be installed on the system.

6.3 Security Requirements

a. Faculty shall only be permitted to view the attendance of all the students. He is not permitted to delete or update any individual's attendance or in other words he shall not be able to edit the database. He is allowed to upload the image and attendance is marked automatically by the application.
b. Student is also allowed to only view his attendance in various courses. He shall not be given access to the database.
c. In order to upload the image instructor must enter his password.
d. Both student and instructor must enter the login credentials in order to access his profile.

e. While signing up, both student and instructor email id must be verified by sending a confirmation email to the entered email. Signup process is completed only after verification of email.
f. Changing password by instructor or student is allowed only after entering the existing password. On forgetting the password, the reset must be sent to registered email id.
g. Once marked the attendance cannot be changed.

6.4 Software Quality Attributes

Availability:- The application shall be available to various instructors and must load quickly and work well on various terminals.

Reusability:- The application must support to add or delete faculties and students so that it can be reused for the new semester.

Usability:- Usability of application shall be introduced to instructors and students through introductory training session upon installation of the application.

Reliability:- As application is used on wireless network reliability cannot be guaranteed.

Robustness:- If there is any problem in network establishment then instructor must be able to enter the attendance later.

7. Appendix A: Glossary

Term	Description
Faculty	Faculty, teacher and instructor are interchangeably used throughout the document.

8. Appendix B: Analysis Models

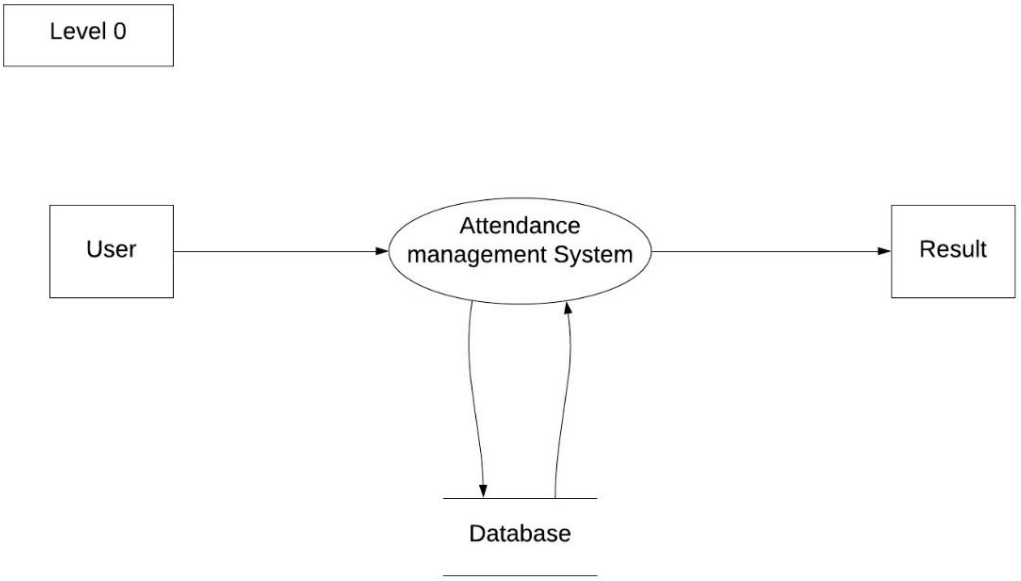


Fig. 1 Data Flow Diagram Level 0

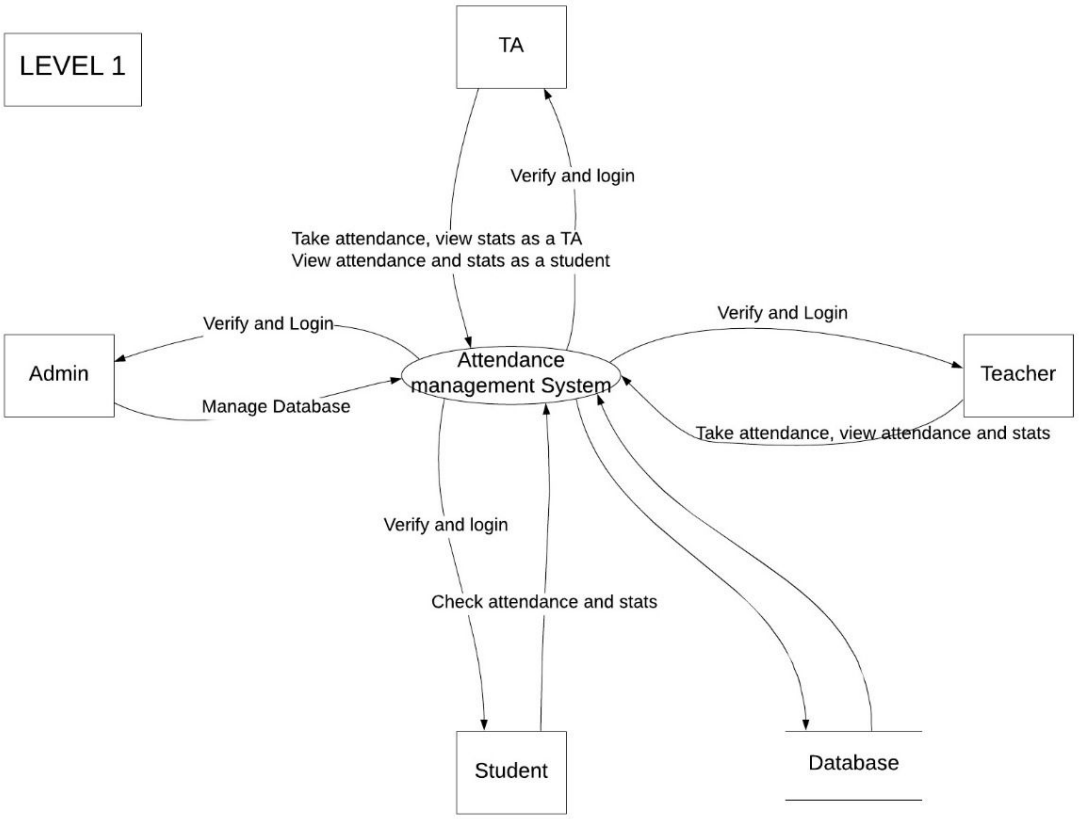


Fig. 2 Data Flow Diagram Level 1

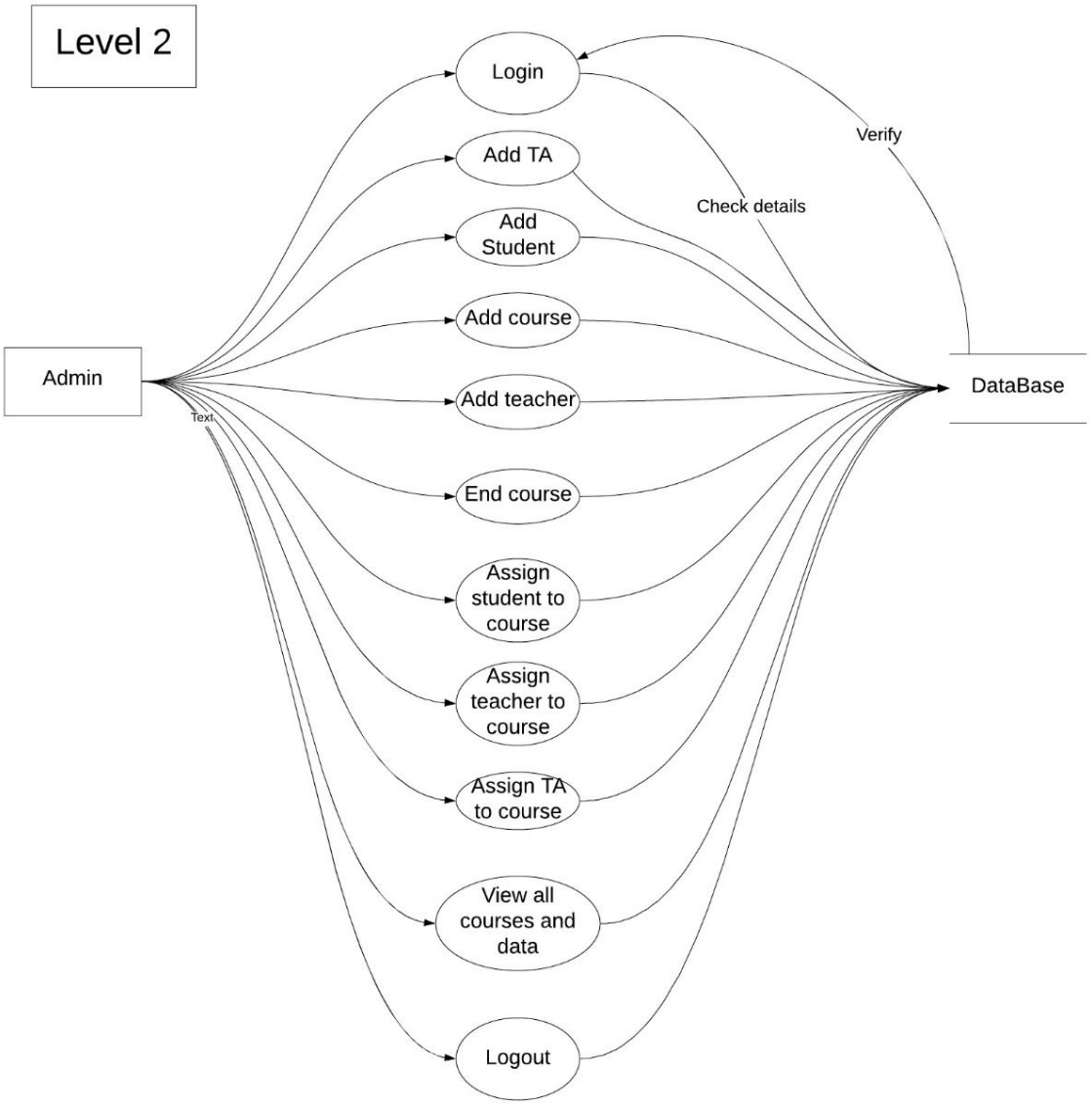


Fig. 3 Data Flow Diagram Level 2 (Admin)

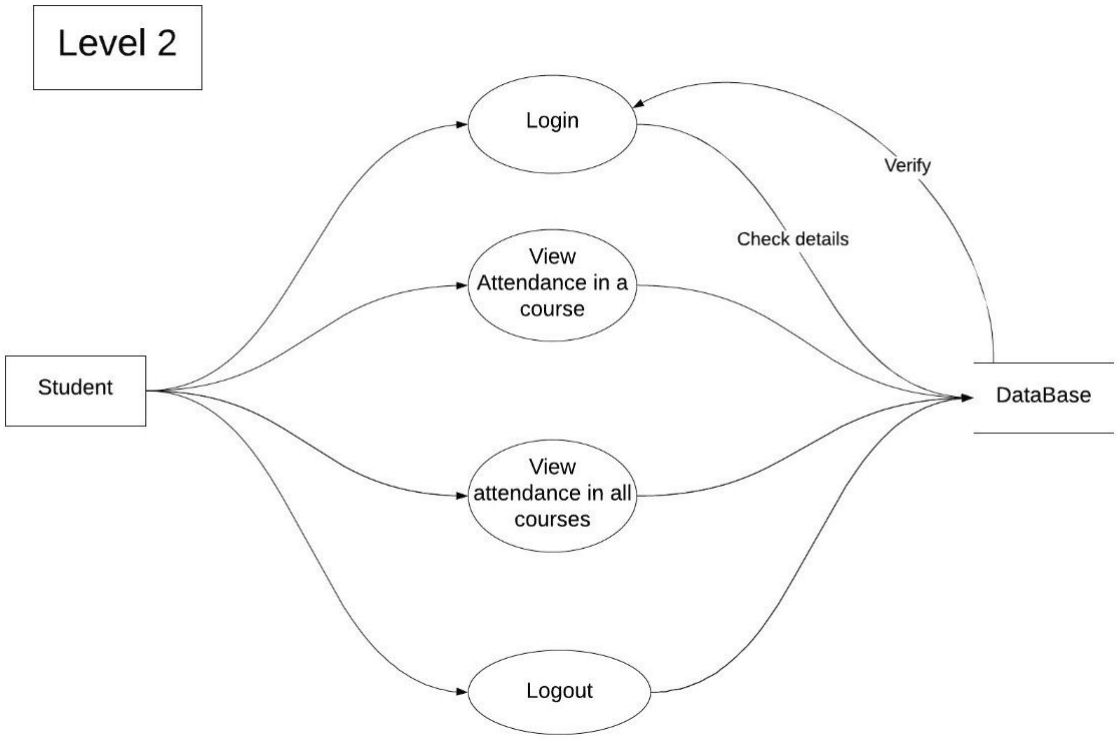


Fig. 4 Data Flow Diagram Level 2 (Student)

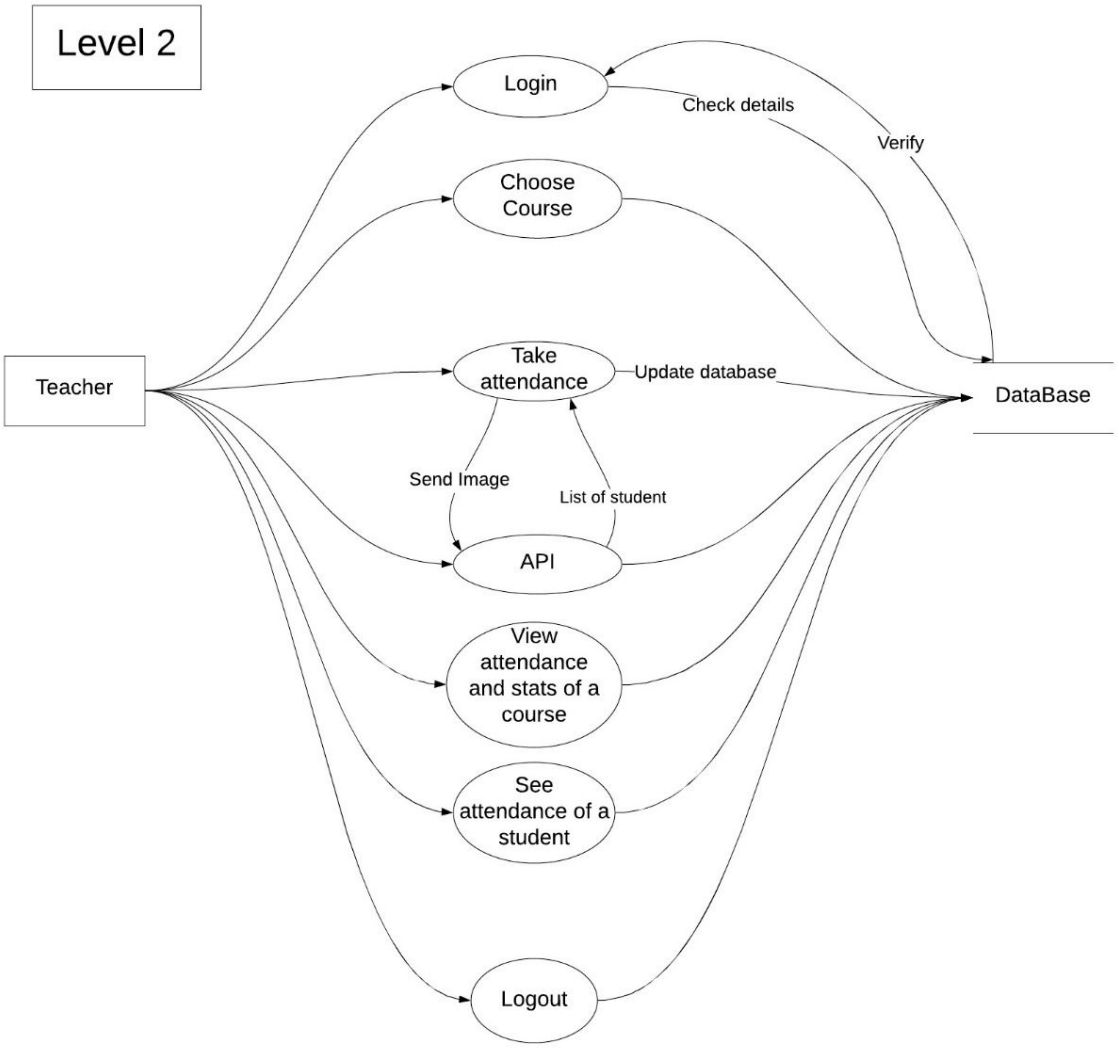


Fig. 5 Data Flow Diagram Level 2 (Teacher)

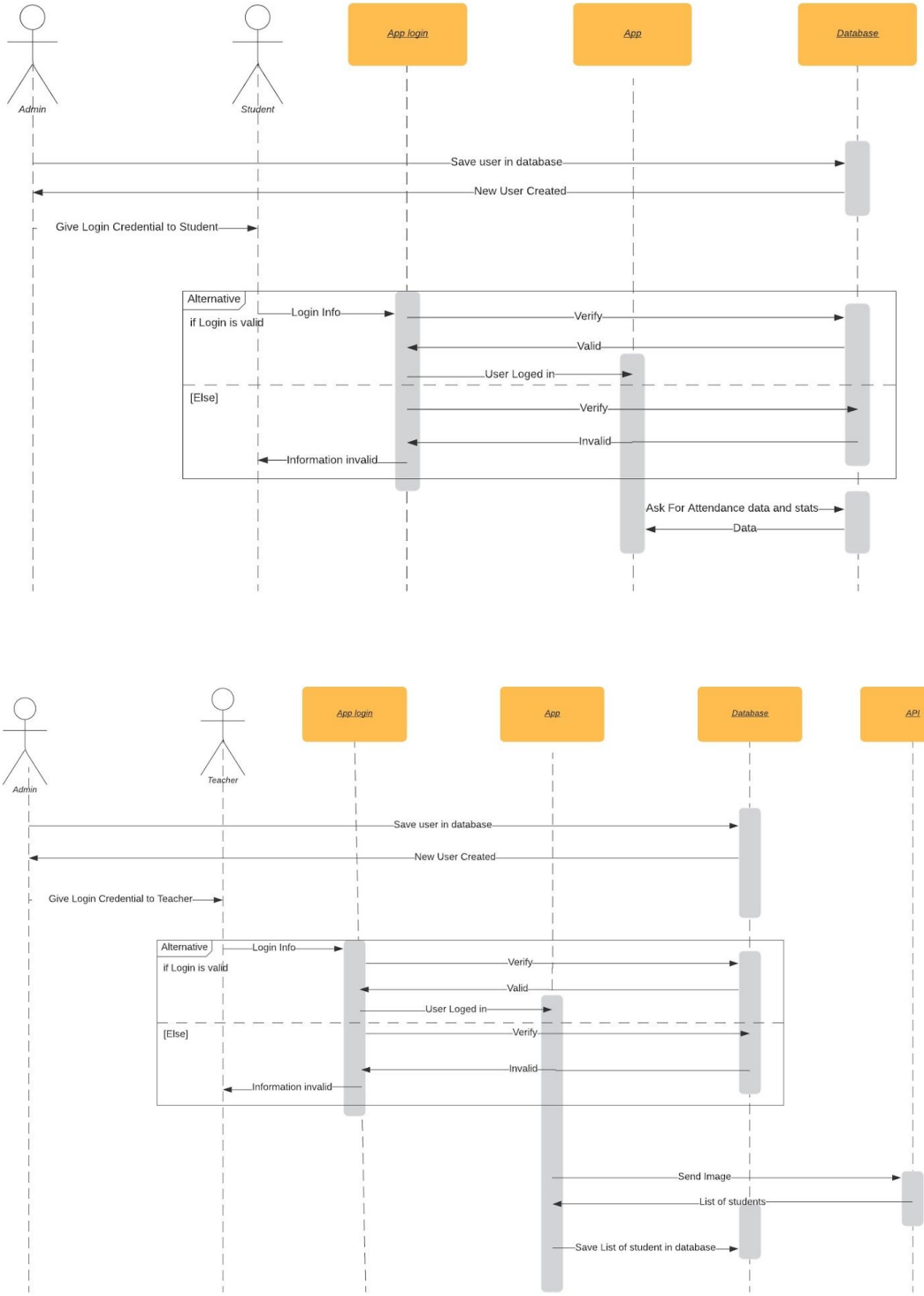
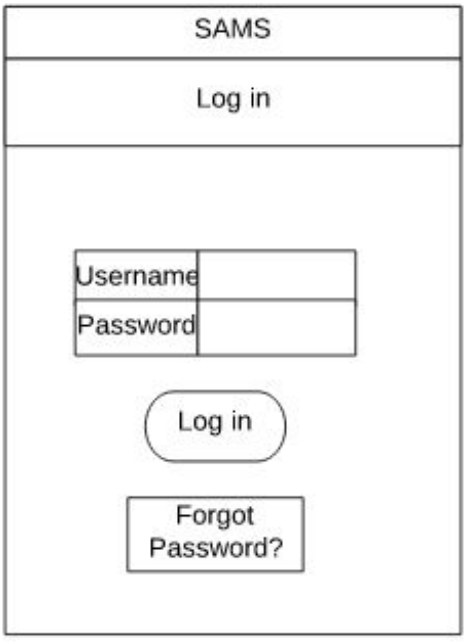


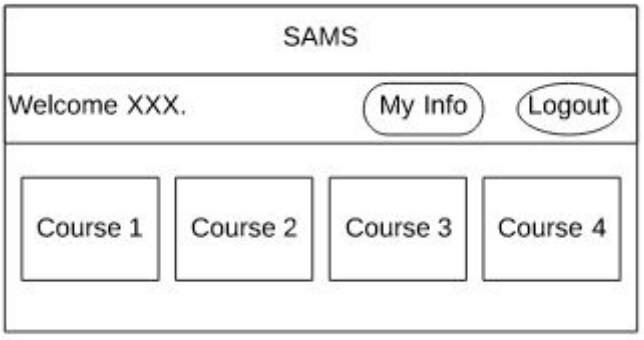
Fig. 6 Sequence diagram for teacher and student

9. Appendix C: User Interfaces

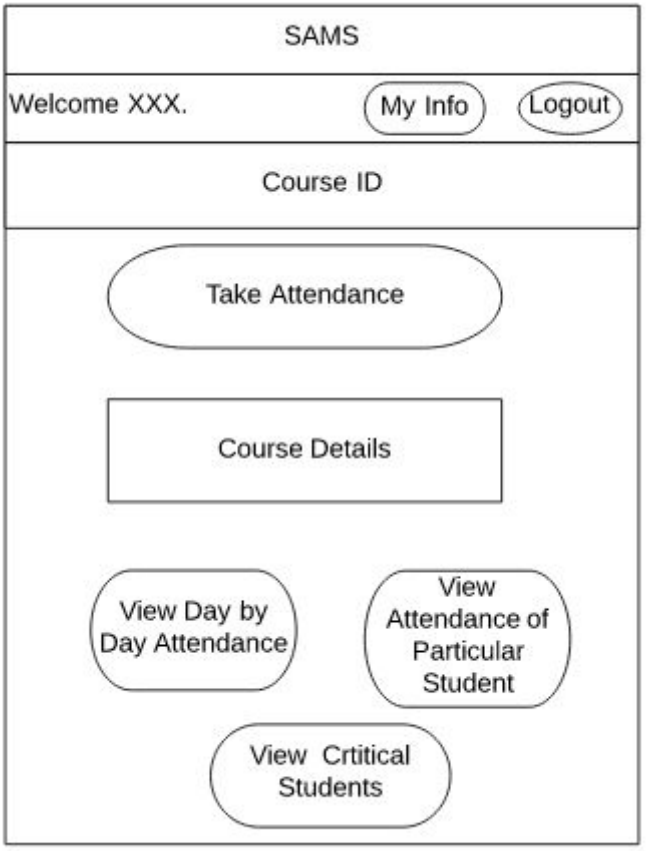
1. Login Page



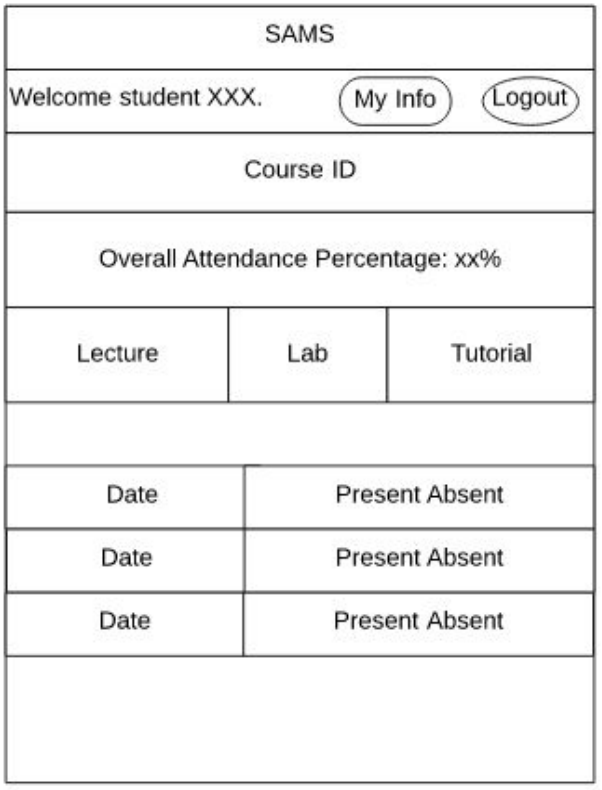
2. Teacher Welcome Page



3. Teacher Course Page



4. Student Attendance For Particular Course



5. Teacher Course Attendance Page