Software Requirements Specification

For

Attendance Application

Version 1.0 approved

Prepared by:

The Lost Phoenix (Group 31)

Utsav Basu (20CS30057)

Anamitra Mukhopadhyay (20CS30064)

Spandan Halder (20CS10063)

IIT KHARAGPUR

Date Created: 13 March 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This software is an application that can be used to take attendance in online/offline lectures. It is proposed to build a software to save time in taking attendance. The purpose of this document is to present a detailed description of the Attendance Application. It will explain the objective of this project and its scopes of application, its features, the interface, and the details about interactions through the interface with database, data management strategies, constraints under which it will operate. This document is intended for understanding requirements and specifications in detail.

1.2 Document Conventions

1) An asterisk after a word indicates more information about it in the glossary

2) To denote sub headers dot separated numbers are used, for ex 1.2.1 is sub header of 1.2

1.3 Intended Audience and Reading Suggestions

This software is intended mostly towards institutions (schools, universities, etc.) and/or independent teachers, who want an easy way to quickly take the attendance of their students, over the conventional roll calling mechanism.

This document is intended for developers contributing towards the project, the client, users, and testers and anyone who is reviewing the project to easily view descriptions about functional and no-functional requirements and understand product scope.

This document contains six (6) sections:

a. An overall description, which includes a summary of the scope, assumptions and limitations of the project, types and characteristics of users, plan for implementation.

b. A brief description of external interfaces requirements.

c. A description of the system, which includes an explanation of the system's purpose, features and benefits, characteristics, and technical and operational background; and,

d. A list of nonfunctional requirements, which contains information regarding performance, safety, security, and business rules.

e. A description of other uncategorised requirements.

f. A list of the appendices. (Definition of terms used in the document, analysis model and future extension possibilities)

1.4 Product Scope

This software provides an easy-to-use medium to take and/or give attendance in an online/offline class.

1.5 References

The material content of the following sites would be helpful in the development process:

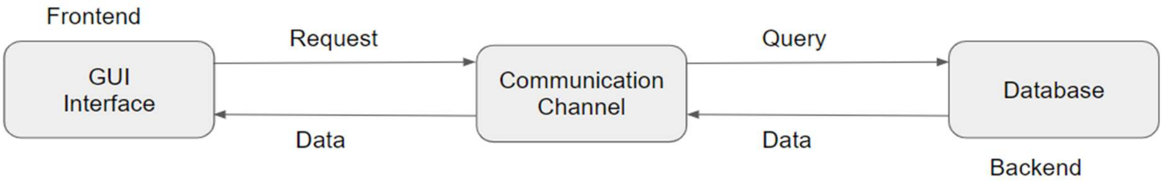
1) Dear ImGUI: https://github.com/ocornut/imgui

2) SQLite: https://www.sqlite.org/docs.html

2. Overall Description

2.1 Product Perspective

This is a self-contained project to take and give attendance. The basic workflow is shown below:



2.2 Product Functions

Our software provides the following functionalities:

Functionalities for **Student**:

1) Students should get all the statistics of the classes he attended.
2) Students can mark the attendance in the time span allotted (only if code matches with the unique code generated by Teacher)

Functionalities for **Teacher**:
1) Get the statistics of all the classes he teaches.
2) Start the attendance.
3) Get possible proxies (Detect multiple attendances from the same IP)

Functionalities for **Admin**:
1) Can create/delete/modify a course from the database.

2.3 User Classes and Characteristics

Basically 3 types of users are expected to use our software:

- Admin
- Student
- Teacher (Professor/Faculty of Institution)

2.4 Operating Environment

This software is built to operate on laptops and PCs with Windows Operating System (version 8 and above). Since this is a web-based project, browsers must be updated and proper internet connectivity should be there.

2.5 Design and Implementation Constraints

This software is dependent on the database of Students, Teachers, and Admins provided by the Institution. The Institution must ensure that the product database is up-to-date with the Institute database, otherwise some security issues might crop up.

In addition to that, since a lot of students might try to give attendance for a certain lecture at the same time, the backend server and/or the communication channel might be overloaded. It is advisable for the teachers and the Institute* heads to keep a note of the above constraints.

2.6 User Documentation

The software UI is highly self-sufficient and obvious. We firmly believe that no ambiguity would be faced by our users.

Extra details are to be provided later (TBD*)

2.7 Assumptions and Dependencies

This software depends upon the following libraries to achieve its functionalities:
a. Dear ImGUI: https://github.com/ocornut/imgui
b. SQLite: https://www.sqlite.org/docs.html

3. External Interface Requirements

3.1 User Interfaces

The UI* will be GUI* displayed in a browser will have the following key functions:

a. A basic Login Screen
b. A dashboard from where the user can navigate and choose his/her actions, like generating new "Magic Code" *, giving attendance, viewing statistics, etc.

3.2 Hardware Interfaces

- This software will work on any modern PC or laptop, provided that it has internet connection.
- Large database managing memory and resolution on server side

3.3 Software Interfaces

Languages used:
Frontend: C++ (Dear IMGUI library)
Backend: C++ and SQL (via SQLite engine)

Operating System: Windows
IDE: Visual Studio
Database: SQLite

3.4 Communications Interfaces

TBD

4. System Features

4.1 Student Login

4.1.1 Description and Priority (High): Login as student by providing the correct username and password, give attendance and view the list of attended classes.

4.1.2 Stimulus/Response Sequences:

Stimulus: Give attendance using the “magic code” provided by the teacher., within the time-span allotted.

Response: Attendance gets recorded if attendance is given within the given time span.

Stimulus: View the attendance of any course he/she attended.

Response: Display the list of lessons he/she attended on the given course.

4.1.3 Functional Requirements:

- Retrieving the attendance of a particular student
- Recording attendance of every student

4.2 Teacher Login

4.2.1 Description and Priority (High): Login as teacher by providing the correct username and password, take attendance, and view the statistics of his/her course(s).

4.2.2 Stimulus/Response Sequences:

Stimulus: Take attendance by generating a “magic code”.

Response: A “magic code” is generated and attendance is recorded when a student uses that code to give attendance.

Stimulus: View the number of students in any course he/she teaches.

Response: Display the number of students on that course.

Stimulus: View the attendance in a lesson in a course.

Response: Display the percentage attendance in that lesson of that course.

Stimulus: View the possible proxies during attendance.

Response: Display the suspected students in every lesson in a course.

4.2.3 Functional Requirements:

- Generating the "magic code”
- Taking attendance by matching the code entered by students
- Retrieving the statistics of attendance of the students

4.3 Admin Login

4.3.1 Description and Priority (Low): Login as admin by providing the correct username and password, add, modify, or delete any course(s).

4.3.2 Stimulus/Response Sequences:

Stimulus: Create a course and assign the teacher(s).

Response: Create the course as specified.

Stimulus: Modify or delete a course

Response: Modify or delete a course as specified.

4.3.3 Functional Requirements:

- Creating a course
- Modifying a course
- Deleting a course

5. Other Nonfunctional Requirements

5.2 Performance Requirements

- The Database Management system should be public-domain
- The system should be able to handle multiple logins (up to 1000 to ensure negligible down-time)
- The server hosting the webpage should be efficient, and also the net connection needs to be good

5.3 Safety Requirements

- Databases must be updated periodically (preferably each semester) to ensure no erroneous record is present.
- Databases must not be updated within a semester, unless done so by the Institute itself

5.4 Security Requirements

- The password of each user must be strong to prevent misuse.
- Any user account must be authenticated by the Institute to ensure no privacy breach

5.5 Software Quality Attributes

- The software should be user-friendly.
- The software should be reliable, testable, and maintainable.
- The software should be efficient in terms of speed and security

5.6 Business Rules

A database manager may be hired to update and maintain the databases used by our software. This is crucial for the unhindered functioning of our software.

6. Other Requirements

The database details need to be provided by the Institute for seamless functionality.

Appendix A: Glossary

Magic code: The system generated code provided by the teacher used for taking attendance.

TBD: To Be Done

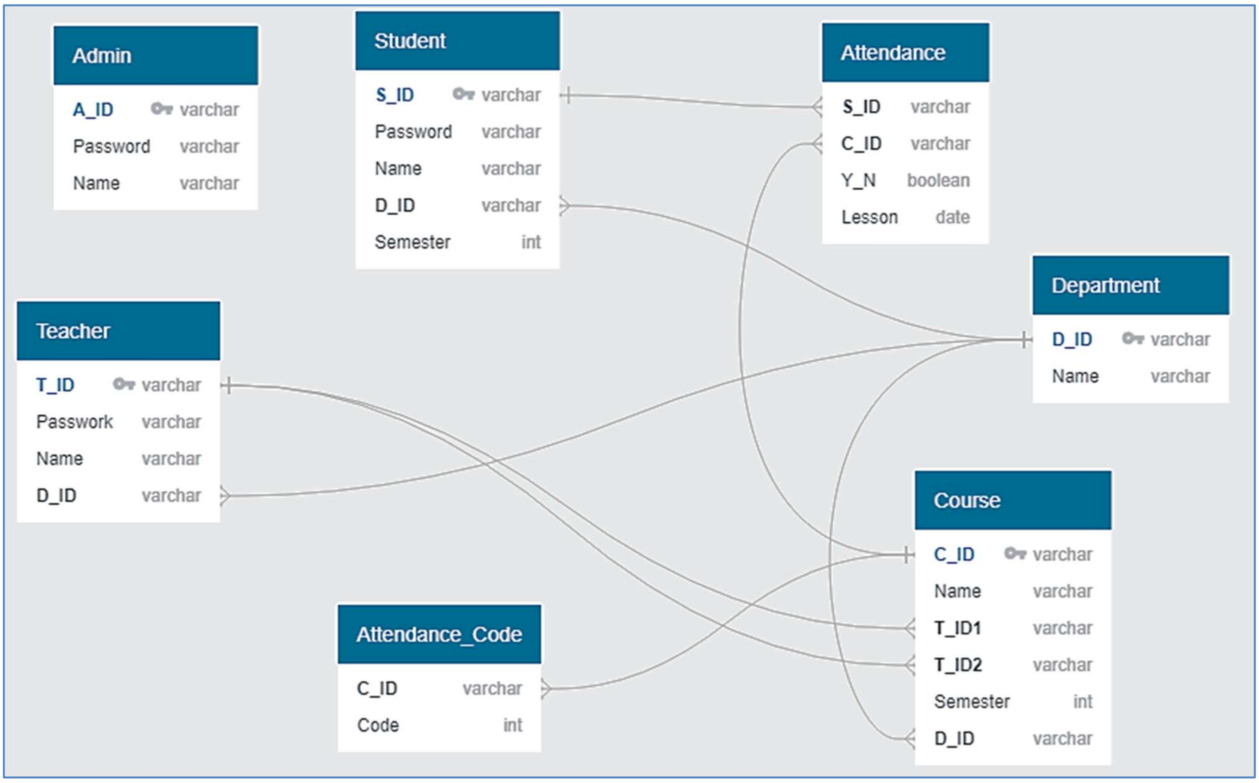
UI: User Interface

GUI: Graphical User Interface

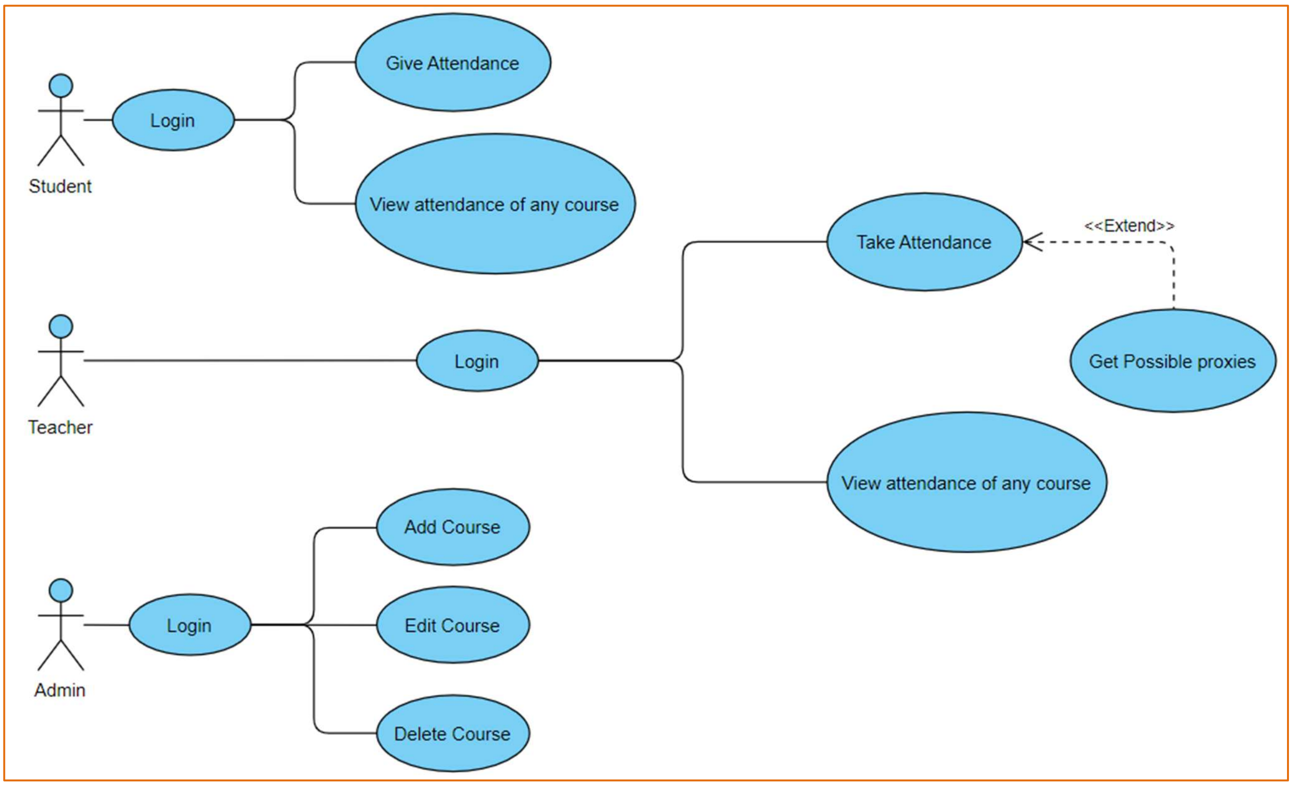
Institute: The institution which is using our software

Appendix B: Analysis Models

Database Diagram



Use Case Diagram



Appendix C: To Be Determined List

- Detailed User Documentation
- Communication Protocols
- Extra functionality may be added later on