SOFTWARE REQUIREMENTS SPECIFICATION

for

Attendance Application

Version 1.0 approved

Prepared by:	**Sanskar Mittal**	**21CS10057**
	Yash Sirvi	**21CS10083**
	Ashwin Prasanth	**21CS30009**

Indian Institute of Technology Kharagpur

March 27, 2023

Contents

Revision History

Name	Date	Reason For Changes	Version
21	22	23	24
31	32	33	34

1 Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the development of an Attendance Application. This software is intended to be used by educational institutions, businesses, and organizations to track the attendance of students, employees, or members. The application will be designed to provide a user-friendly interface, allowing easy and efficient management of attendance records.

The Attendance Application will be developed to provide an automated system that simplifies the attendance-taking process. The software will be designed to reduce the workload of teachers, administrators, and human resource personnel who are responsible for tracking attendance.

This document will outline the functional and non-functional requirements of the software, including the user interface.

1.2 Intended Audience and Reading Suggestions

This document is intended for the following audience:

- The development team responsible for implementing the Attendance Application software
- System administrators and users who will be responsible for installing and using the Attendance Application software
- Any other parties involved in the development, testing, and deployment of the Attendance Application software

To fully understand the requirements and constraints outlined in this document, it is recommended that readers have a working knowledge of the following:

- Object-oriented programming concepts and practices
- Android application development technologies and tools such as Android Studio and React Native
- Relational database management systems and SQL

To read the document, readers should start with the Overview sections, including the Introduction and Overall Description, to gain a high-level understanding of the

Attendance Application software and its purpose. From there, readers can move on to the sections that are most pertinent to their specific role or area of responsibility, such as the Other Nonfunctional Requirements section for developers, the External Interface Requirements section for project managers and testers. Finally, readers should review any relevant appendices for additional information.

1.3 Project Scope

The Attendance Application software is designed to provide an efficient and reliable way for educational institutions to track student attendance. The system is intended to be used by both students and faculty, with different roles and permissions assigned to each user type.

The Attendance Application software will be designed as a smartphone-based application that can be accessed by both students and faculty using their mobile devices. The system will be developed using industry-standard technologies and will be scalable to accommodate additional users and institutions as needed.

2 Overall Description

2.1 Product Perspective

The Attendance Application software is a new, self-contained product that is being developed to provide an efficient and reliable way for educational institutions to track student attendance. The system will be designed as a smartphone-based application that can be accessed by both students and faculty using their mobile devices.

2.2 Product Functions

The Attendance Application software will include the following features and functionality:

- Student attendance tracking
- Attendance reports
- Notification system
- User authentication and security
- Proxy Detection
- Importing and Exporting data

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

The Attendance Application software will be designed to operate on mobile devices running Android 7.0 and above, as well as iOS 11 and above.

The software components and applications that need to peacefully coexist with the Attendance Application software are: Web server, Database system, and Network infrastructure Overall, the Attendance Application software will operate in a networked environment that requires the peaceful coexistence of multiple software components and applications.

2.5 Design and Implementation Constraints

The development of the Attendance Application software is subject to the following constraints:

- Limited time: The development team has a fixed short timeline for completing the project.
- Learning new frameworks: The development team will need to learn and use new frameworks to develop the app. This requires additional time and resources to ensure the team has the necessary skills.
- Security considerations: The Attendance Application software will handle sensitive data, including student and course information. Security considerations will need to be considered during the design and development phases.

2.6 User Documentation

The Attendance Application software will be accompanied by a user documentation package that will include the following materials:

- User manual: User manual including step-by-step instructions will be provided to guide users through installing and using the software.
- Quick reference guide: A quick reference guide will be provided to help users perform common tasks, such as taking attendance or generating and exporting reports.
- Relevant technology documentation: The user documentation package will also include links to relevant technical documentation, such as React Native documentation.

3 External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4 System Features

4.1 Student Attendance Tracking

4.1.1 Description and Priority

Priority: **High**

This feature allows teachers and instructors to keep track of student attendance in real-time. The feature enables teachers to create and manage courses, and edit attendance for each student. Students can also view their attendance record, including their current attendance percentage, the number of days present, absent, and late.

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

- REQ-1: The system shall allow teachers to create and manage courses.
- REQ-2: The system shall allow teachers to edit attendance for each student.
- REQ-3: The system shall allow students to enroll in courses set up by teachers.
- REQ-4: The system shall allow students to mark their attendance within a specified time frame for each course.
- REQ-5: The system shall allow students to view their attendance records.

4.2 User Authentication

4.2.1 Description and Priority

Priority: **High**

This feature allows users to create an account and log in to the app. The allowed user types are students, teachers, and administrators. Only administrators can accept new user account creation requests.

4.2.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.2.3 Functional Requirements

- REQ-1: The system shall allow users to create an account in categories of student, teacher, or administrator.
- REQ-2: The system shall allow users to log in to the app using correct credentials.
- REQ-3: The system shall allow only administrators to accept new user account creation requests.

4.3 Attendance Reports

4.3.1 Description and Priority

Priority: **Medium**

This feature allows teachers and students to view attendance reports for each course. Teachers can view attendance reports for all students in a course, while students can view their own attendance reports.

4.3.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.3.3 Functional Requirements

- REQ-1: The system shall create attendance reports for each student in each course.
- REQ-2: The system shall allow teachers to view attendance reports for all students in a course.
- REQ-3: The system shall allow students to view their own attendance reports.

4.4 Proxy Detection

4.4.1 Description and Priority

Priority: **Medium**

This feature allows teachers to detect students who are using a proxy to mark their attendance.

4.4.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.4.3 Functional Requirements

- REQ-1: The system should detect students who are using a proxy to mark their attendance.

4.5 Notification System

4.5.1 Description and Priority

Priority: **Low**

This feature allows teachers to send notifications to students regarding low or irregular attendance to any student in a course.

4.5.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.5.3 Functional Requirements

- REQ-1: System should be able to notify students.
- REQ-2: System should allow teachers to modify and send notifications to students.

4.6 Importing and Exporting Data

4.6.1 Description and Priority

Priority: **Low**

This feature should allow teachers and administrators to import and export data from the system. This will allow already existing data to be imported into the system, and data from the system to be exported to other systems.

4.6.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.6.3 Functional Requirements

- REQ-1: System should work with CSV files to store data.
- REQ-2: System should allow teachers and administrators to import and export data from the system.

5 Other Nonfunctional Requirements

5.1 Safety Requirements

- The system will store sensitive information essential to the functioning of Institutions which will be using it. It is therefore essential to keep backups of everything and to ensure that the data is not lost.
- The system will be used by students and teachers who are not necessarily tech-savvy. It is therefore essential to ensure that the system is easy to use and does not require any special training.
- The system should also be able to handle any errors that may occur.
- The system should expect the proxy detection to not be perfect and should be able to handle false positives.

5.2 Security Requirements

- The system will handle sensitive information such as student and teacher details. It is therefore essential to ensure that the system is secure and that the data is not leaked.
- The login feature should not be vulnerable to brute force attacks.
- The import and export feature should not be vulnerable to SQL injection attacks.

5.3 Software Quality Attributes

The following are the quality attributes that the system should have from highest to lowest priority:

- The application should not be platform dependent and should be able to run on both Android and IOS devices.
- The system should be easy to use as it will be used by students and teachers who are not necessarily tech-savvy.
- The codebase should be well documented and easy to understand. This will allow for easier maintenance and addition of new features.

- The application should be able to handle any errors that may occur.
- The application should be lightweight and fast to use.