

Attendance Management Software Requirements Specification

Introduction

Purpose

The purpose of this Software Requirements Specification (SRS) document is to present a detailed description of the Attendance Management System (AMS) designed for academic settings. This system aims to streamline the process of attendance taking for professors and enhance the attendance tracking experience for students. Through this document, we intend to provide a comprehensive outline of the software's functionalities, interface design, performance requirements, and the interactions between the users and the system. The SRS will serve as a contractual basis for the development and validation of the AMS, ensuring that the final product aligns with the needs of the users and the objectives of the project. The AMS is envisioned to replace traditional attendance tracking methods with an automated, efficient, and user-friendly solution, thereby reducing the administrative overhead for faculty and improving the accuracy and accessibility of attendance data for students and educational institutions.

This document is intended for use by the software development team, project stakeholders, including academic staff and university IT departments, and future system maintainers. By defining clear and precise requirements, the AMS will be developed to facilitate the attendance process, offering a quick, secure, and reliable means for professors to record attendance, for students to be informed of their attendance status in real-time, and for administrators to oversee and manage these records effectively.

Scope

The Attendance Management System (AMS) is designed as a cross-platform, text-based application to facilitate the attendance recording and management process in educational institutions. This system will enable professors to efficiently mark students present, absent, or tardy with minimal disruption to class time. The scope of the AMS includes:

Attendance Recording

The core functionality of AMS is to provide a swift mechanism for professors to record attendance at the beginning of each class. Students' names will be displayed on the screen, and their attendance status will be updated with a simple keypress.

Late Arrivals Management

AMS will allow professors to amend attendance records at the end of the class session to account for students who arrive late, marking them as tardy.

Search and Browse Capability

Professors will be able to search for and browse through student records to update attendance statuses as necessary throughout the class session.

Data Import and Management

Professors can import student rosters from CSV files, which may vary in format. The system will support ongoing management of the roster as students enroll in or drop the course.

Attendance Data Storage

The AMS will store attendance records in plain text files, with options to export data in multiple formats such as JSON, XML, or custom formats for integration with other systems or for reporting purposes.

Automated Notifications

The system will automatically send attendance reports to students at the end of each week and will notify students via email whenever their attendance status is changed.

Name Management

AMS will allow for the recording of both legal names and preferred display names, ensuring that students are able to be addressed as they wish during the attendance-taking process while keeping records consistent with university systems.

Security

Given the sensitive nature of attendance data, AMS will incorporate security measures to protect against unauthorized access and ensure data integrity.

System Independence

The software will be platform-independent, capable of running on various operating systems including Windows, macOS, and Linux, ensuring wide accessibility.

User Interface

The system will feature a user-friendly, text-based interface that can be navigated from a terminal or command-line interface.

Reporting

AMS will generate attendance reports that can be disseminated to students and used by professors and administrators for record-keeping and analysis.

Extensibility

The system is designed with the future in mind, allowing for the potential integration of additional notification methods, such as SMS, and other enhancements.

The AMS is scoped to improve the academic experience for both professors and students by making attendance management more accurate and less time-consuming, fostering a more focused and engaging learning environment.

Definitions, acronyms and abbreviations

The following table provides definitions for terms and acronyms used within this document.

Term	Definition
Professor	An individual responsible for delivering course content, managing and publishing courses, managing student information, managing classroom activities, and recording student attendance.
Student	An individual enrolled in the course who participates in classroom activities and whose attendance is tracked by the system. A student shall receive reports and notifications regarding their attendance status.

Overall description

Product perspective

The Attendance Management System (AMS) is a self-contained, stand-alone software solution intended to modernize and streamline the process of managing student attendance in educational institutions. It is not a component of a larger system but is designed to operate independently, interfacing directly with users through a text-based terminal. The AMS can integrate with existing university systems to import student roster data and may export attendance records for use in other applications or reporting tools. Its cross-platform capability allows it to function on various operating systems such as Windows, macOS, and Linux, thereby ensuring wide accessibility for users regardless of their hardware preferences.

The AMS is designed to function without the need for specialized hardware, relying instead on standard computing devices available to most educational professionals. From the product perspective, the AMS aims to occupy a unique position in the market by combining ease of use with robust functionality, prioritizing data security and user accessibility.

Product Functions

The Attendance Management System (AMS) provides several key functions to facilitate the attendance management process:

Attendance Recording and Management

- **Real-time Attendance Tracking:** Enables professors to record attendance in real-time as students arrive in class.
- **Dynamic Updating:** Professors can mark students present, absent, or tardy with a simple keypress.
- **Attendance Status Modification:** Allows professors to update attendance status at any point during or after the class.

Roster Import and Update

- **CSV Roster Import:** Professors can upload student rosters via CSV files, accommodating various formats and structures.
- **Enrollment Management:** The system can add newly enrolled students and remove dropped students, updating the roster accordingly.

Late Arrival Handling

- **Late Check-in:** The system provides a mechanism to mark late-arriving students as tardy.
- **Post-Class Attendance Adjustment:** Professors can amend attendance records post-session to accommodate late arrivals.

Data Storage and Export

- **Plain Text Storage:** Attendance records are saved in plain text files for ease of access and simplicity.
- **Data Export Options:** The system supports exporting attendance data in JSON, XML, and custom formats for flexibility and reporting purposes.

Automated Communication

- **Weekly Attendance Reports:** Sends weekly attendance summaries to students via email.
- **Real-time Attendance Notifications:** Notifies students of any changes to their attendance status immediately.

User-friendly Interaction

- **Name Flexibility:** Accommodates preferred display names alongside legal names for student comfort and inclusivity.
- **Search and Browse:** Users can search and navigate through student lists for efficient attendance management.

Security

- **Sensitive Data Protection:** Implements security protocols to ensure the confidentiality and integrity of attendance data.
- **Access Control:** Restricts system access to authorized personnel only, with secure authentication mechanisms.

These functions are designed to streamline the attendance process, reduce administrative workload, and improve the overall accuracy and efficiency of attendance tracking within academic settings.

Constraints

The development and operation of the Attendance Management System (AMS) are subject to several constraints that define the boundaries within which it must operate. These constraints are necessary to ensure compatibility, usability, and security of the system:

- **Cross-Platform Compatibility:** The AMS must function consistently across different operating systems, including Windows, macOS, and Linux. This requires careful consideration of OS-specific features and limitations.
- **Interface Limitations:** Given that the AMS is a text-based terminal application, its user interface is limited to text input and output. This constraint prioritizes function and accessibility over graphical design elements.
- **Data Privacy Regulations:** The AMS will handle sensitive student information, requiring compliance with data protection laws such as GDPR, FERPA, or other regional privacy frameworks.
- **Security:** The system must employ robust security measures to protect sensitive data from unauthorized access, including secure login processes and data encryption.
- **File-Based Data Storage:** The requirement for attendance records to be stored in plain text files may limit the complexity and structure of data that can be handled and could affect performance with very large datasets.
- **Data Export Formats:** While the system will support multiple export formats (plain text, JSON, XML, etc.), each format will need to be generated from the same source data, which may require additional conversion and validation logic.
- **Performance:** The system must perform well under the expected load of concurrent users, especially during peak times like class commencement and dismissal.
- **Network Reliability:** Automated notifications and report sending will depend on reliable network connectivity and email server uptime.
- **Extensibility:** Future expansion for additional features, like SMS notifications, must be planned for without requiring major overhauls to the existing system architecture.

Assumptions and dependencies

The effectiveness and efficient operation of the Attendance Management System (AMS) rely on several key assumptions and dependencies, which are expected to remain consistent throughout its

lifecycle:

Assumptions:

- **Stable Enrollment:** The system assumes that fluctuations in student enrollment will mainly occur at predictable times, such as the beginning and end of semesters.
- **Consistent User Behavior:** Professors are expected to consistently utilize the system for attendance recording, adhering to established workflows.
- **Technological Proficiency:** Both professors and students are presumed to have sufficient proficiency with technology to interact with a text-based terminal application.
- **Reliable Infrastructure:** The AMS presupposes the existence of dependable infrastructure, including consistent power and network services, along with adequate hardware that supports the software requirements.
- **Data Accuracy:** There is an assumption that the data inputted, such as student rosters imported via CSV files, is accurate and formatted correctly.

Dependencies:

- **Spring Boot:** The system is developed using the Spring Boot framework, which provides the necessary backend infrastructure, including server and application context management. The AMS's performance and scalability are thus dependent on the continued support and stability of the Spring Boot framework.
- **Operating Systems:** Cross-platform functionality relies on the compatibility of Spring Boot with various operating systems, including Windows, macOS, and Linux, and their support for the Java Runtime Environment.
- **Email Service Integration:** The notification system's ability to send emails is dependent on the integration with a dependable email service, which must provide consistent API support for Spring Boot applications.
- **Network Connectivity:** Ongoing access to a reliable network is critical, especially for features like email notifications and potential cloud-based features or integrations.
- **Data Export Libraries:** The system's capacity to export data in different formats (JSON, XML, etc.) depends on the availability of Spring Boot compatible libraries that support data serialization and deserialization.

Requirements

Specific requirements

The Specific Requirements of the Attendance Management System (AMS) are derived from the operational needs and user interactions identified in the user story. These requirements are focused on delivering a system that streamlines attendance management for instructors and enhances communication with students. Here are the key specific requirements:

- **Requirement 1 (Importing Rosters):** The system shall allow instructors to import student rosters from varying CSV file formats, handling different header arrangements and column orders efficiently.
- **Requirement 2 (Attendance Recording):** The system shall enable instructors to quickly mark students as present, absent, or tardy using a simple keypress for each student name presented on the terminal interface.
- **Requirement 3 (Late Arrivals):** The system shall provide functionality for instructors to update attendance statuses to accommodate students who arrive late, with an intuitive method for marking these students as tardy.
- **Requirement 4 (Attendance Adjustment):** The system shall offer an option for instructors to adjust attendance records after the class session has ended, reflecting changes for students who arrived late or incorrectly inserted data
- **Requirement 5 (Data Storage):** The system shall record attendance data in servers with the capability to export this data into other formats such as JSON and XML.
- **Requirement 6 (Automated Notifications):** The system shall send automated email notifications to students, including weekly attendance reports and real-time updates when their attendance status changes.
- **Requirement 7 (Alias flexibility):** The system shall support both legal names and preferred display names for students, allowing instructors to easily update these upon request while maintaining synchronization with university records.
- **Requirement 8 (Security):** The system shall implement security measures to ensure the confidentiality and integrity of attendance data, employing encryption and secure authentication protocols.
- **Requirement 9 (System Accessibility):** The system shall be operable on various operating systems such as Windows, macOS, and Linux without requiring additional modifications or configurations.

Functional requirements

User Class 1: Professors

- **FR1.1 Attendance Input:** Professors must be able to mark students as present, absent, or tardy within the terminal interface during the lecture.
- **FR1.2 Roster Management:** Professors should be able to import, update, and manage student rosters through a CSV file import functionality.
- **FR1.3 Attendance Modification:** The system must allow Professors to modify the attendance status for students who arrive late or need corrections post-class.
- **FR1.4 Export Functionality:** Professors need the ability to export attendance records in multiple formats for reporting purposes or to share with other systems.

User Class 2: Students

- **FR2.1 Attendance Visibility:** Students should be able to receive notifications and view their attendance records through email communication.
- **FR2.2 Personal Information Update:** Students must be able to request updates to their preferred display names, which Professors should be able to modify in the system.
- **FR2.3 Real-time Notifications:** The system must send immediate notifications to Students when their attendance status is updated or when weekly reports are generated.