

Attendance Management Software Requirements Specification

Introduction

Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive outline of the evolved Attendance Management System (AMS), now redesigned for academic settings with a client/server architecture. This new iteration of AMS focuses on enhancing the user experience for a diverse range of users, including faculty, students, and administrative staff, across various classes and sections. It brings forth a more efficient and streamlined process for managing attendance in educational environments.

This document details the expanded functionalities of the AMS, encompassing aspects such as user registration, class and section management, and tailored interfaces for different user types. It also outlines the system's interface design, performance requirements, and user interactions, aiming to align with the project's goals and the stakeholders' expectations. The updated AMS is envisioned as a modern solution to replace conventional attendance methods, offering an automated, efficient, and user-friendly approach to reduce administrative overhead and enhance the accuracy and accessibility of attendance data.

Targeted at the software development team, project stakeholders, academic staff, university IT departments, and future system maintainers, this SRS is a guiding framework for the system's development and validation. It sets clear and precise requirements to develop the AMS in a manner that facilitates an effortless attendance recording process for professors, provides real-time attendance updates for students, and enables administrators to manage these records effectively. In doing so, the AMS adapts to a technology strategy that moves away from specific frameworks like Spring Boot, focusing instead on bespoke data management solutions using Java and JDBC within the client/server model.

Scope

The evolved Attendance Management System (AMS) is re-envisioned as a comprehensive client/server application, aimed at enhancing attendance recording and management processes in educational institutions. This iteration introduces significant enhancements to accommodate a broader range of functionalities, tailored to both faculty and students in a multi-class environment. The system will be developed using Java, with JDBC for database interactions, and is planned to adopt an MVC framework for better class structure management in the future. The scope of the AMS now includes:

Client/Server Architecture

The AMS is architected with a centralized server hosting the repository of student attendance, accessed by lightweight client applications. This design ensures efficient data management and processing, primarily handled by the server operating on a Linux VM.

Multiple User Roles and Authentication

The system supports various user roles, including faculty and students, each with distinct authentication and authorization capabilities. The user administration, including the addition, removal, and updating of users, is handled through a standalone application, not integrated into the client/server model.

Class and Section Management

Faculty can manage multiple classes and sections, with functionalities to add, remove, update, and enroll students. Enrollment can be conducted both manually and through batch processes, such as CSV file uploads.

Attendance Recording and Management

The core functionality of AMS remains the swift and efficient recording of attendance by faculty at the beginning of each class, with an enhanced interface for selecting specific classes and sections.

Late Arrivals and Attendance Amendments

Faculty can amend attendance records post-class to account for late arrivals or other changes, marking students as tardy or modifying their attendance status.

Data Storage in RDBMS

Replacing plain text files, the AMS now utilizes a Relational Database Management System (RDBMS), preferably MySQL, for robust and secure data storage. This shift ensures scalable data management and compliance with institutional standards.

Data Management in Program Using the DAO Pattern

The AMS employs the Data Access Object (DAO) pattern to effectively manage interactions with the underlying RDBMS. This architectural pattern provides a flexible framework for accessing data sources and manipulating persisted data, ensuring that the data access mechanisms are isolated from the business logic layer.

Notifications and Communication Preferences

Students will by default receive automatic notifications on a weekday. Notifications include email alerts about attendance status and weekly reports. Students have the flexibility to modify their preference regarding whether to receive email notifications. If students disable the notification for a class, they will not receive the attendance notification or the weekly report.

Detailed Reporting

The AMS provides detailed attendance reports for both faculty and students. Faculty reports include attendance summaries with particulars like tardiness, while students can access class-specific attendance records.

Security and Data Integrity

Considering the sensitivity of attendance data, enhanced security measures are implemented to protect against unauthorized access and ensure data integrity, in line with the system's client/server architecture.

System and Platform Flexibility

The AMS maintains its platform independence, adaptable to various operating systems, but now prioritizes a user-friendly client interface with minimal processing demands on the client side.

Extensibility and Future-Proofing

The system is designed to accommodate future enhancements, such as alternative notification methods or integration with different RDBMS or non-SQL storage systems, and the future adoption of an MVC framework for structured class management.

The revamped AMS is scoped to significantly elevate the academic experience, streamlining attendance management for faculty, providing students with real-time access to their attendance records, and ensuring administrators can effectively oversee and manage the entire process.

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.
User	An individual with access to create, read, update, and delete (CRUD) all data within the system. This role typically encapsulates administrative privileges or specialized access rights.

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 evolved Attendance Management System (AMS) encompasses a range of enhanced functionalities to address the complexities of attendance management in an academic setting, particularly in a client/server environment:

Client/Server-Based Attendance Recording and Management

- **Centralized Attendance Tracking:** Allows faculty to record attendance via client applications interfacing with the centralized server, ensuring real-time data accuracy.
- **Dynamic Updating and Status Modification:** Faculty can mark students present, absent, or tardy through the client interface, with updates immediately reflected on the server.

Robust Roster Management

- **Database-Driven Roster Import and Management:** Supports importing student rosters from various formats like CSV into the RDBMS, facilitating seamless enrollment updates.
- **Enrollment Adjustments:** Automatically updates rosters as students enroll or drop out, maintaining current class composition.

Comprehensive Late Arrival Handling

- **Late Arrival Marking:** Provides a straightforward method for faculty to mark students tardy, reflecting in the server database.
- **Adjustable Attendance Records:** Allows post-class amendments to attendance records, accommodating late arrivals and other changes.

Advanced Data Storage and Export Capabilities

- **RDBMS Storage:** Transition from plain text to RDBMS like MySQL for robust, scalable, and secure storage of attendance data.
- **Flexible Data Export:** Supports exporting data in formats like JSON, XML, and others, catering to diverse reporting and integration needs.

Enhanced Automated Communication

- **Configurable Notifications:** Facilitates automated, customizable attendance notifications to students based on their preferences and specific class requirements.
- **Detailed Weekly and Real-Time Reports:** Generates comprehensive attendance reports for students, offering insights into their attendance patterns.

User-Centric Interaction Design

- **Intuitive Client Applications:** Develops user-friendly interfaces for both faculty and student clients, streamlining the attendance recording and checking process.
- **Advanced Search and Navigation:** Enhances the capability to search and navigate through class lists for efficient and accurate attendance management.

Reinforced Security

- **Enhanced Data Protection:** Incorporates advanced security measures to safeguard sensitive attendance information, in line with client/server architecture.
- **Strict Access Controls:** Implements robust authentication and authorization mechanisms to ensure system access is restricted to authorized users only.

These enhanced functions aim to significantly improve the attendance management process, reducing administrative efforts and elevating the accuracy and efficiency of attendance tracking in educational institutions.

Constraints

The development and implementation of the Attendance Management System (AMS) must adhere to several constraints that define its operational parameters. These constraints are essential to ensure the system's compatibility, usability, security, and scalability:

- **Client/Server Architecture Compatibility:** The AMS, transitioning to a client/server model, must ensure seamless interaction between client applications and the central server. This includes compatibility with various client operating systems, including Windows, macOS, and Linux, and server stability on a Linux VM.
- **Interface and Interaction Limitations:** Considering the user-friendly, light-client design, the interface is constrained to be straightforward yet functional, focusing on accessibility and ease of use without complex graphical elements.
- **Data Privacy and Protection:** With the system handling sensitive student information, it must comply with stringent data privacy regulations like GDPR, FERPA, or equivalent frameworks, necessitating robust data handling and storage practices.
- **Enhanced Security Measures:** The AMS must implement advanced security protocols, particularly in the client/server communication, authentication processes, and data encryption, to safeguard sensitive information against unauthorized access.
- **RDBMS-Based Data Storage:** Shifting from file-based to database storage, the system must efficiently manage data within a Relational Database Management System, accommodating large datasets and ensuring data integrity.
- **Data Export and Conversion:** The system will provide data export functionality in multiple formats (JSON, XML, etc.) from the RDBMS, requiring reliable conversion and validation mechanisms.
- **Scalability and Performance:** The AMS should be optimized for high performance, particularly in handling concurrent client requests and database interactions, maintaining efficiency during peak usage.
- **Network Dependability:** The effectiveness of automated notifications and report distribution is contingent on consistent network connectivity and the reliability of email servers.
- **Future-Proofing and Extensibility:** The system's architecture must be designed for easy extensibility, allowing for future enhancements, such as different notification methods or database system changes, without necessitating significant architecture modifications.

These constraints guide the AMS development, ensuring it meets the required standards of functionality, security, and flexibility, while accommodating the evolving needs of educational institutions.

Assumptions:

- **Dynamic Enrollment Management:** The system anticipates enrollment changes to occur mainly at predictable intervals, such as semester beginnings and endings, with the capability to handle unexpected mid-term adjustments.
- **Consistent User Engagement:** It is assumed that both professors and students will engage consistently with the AMS, utilizing its functionalities as per the established processes.
- **Technological Adaptability:** Users, including professors and students, are expected to possess the necessary technological skills to interact with the system's client interfaces.
- **Robust Infrastructure Availability:** The AMS assumes the availability of reliable infrastructure, encompassing stable power and network services, and hardware compatible with the system's requirements.
- **Data Integrity:** There is an expectation of accurate and correctly formatted data input, such as student information in CSV files for roster imports.

Dependencies:

- **Java and JDBC:** The system's development and functionality are heavily reliant on Java and JDBC for backend operations, database interactions, and client-server communication.
- **Operating Systems Compatibility:** Cross-platform compatibility is essential, requiring the system to function seamlessly across diverse operating systems like Windows, macOS, and Linux, which should support the Java environment.
- **Email Service Reliability:** The effectiveness of the notification system depends on the integration with a reliable email service, crucial for sending automated attendance notifications and reports.
- **Stable Network Connectivity:** Continuous access to a reliable network is vital for system operations, especially for functionalities that involve client-server communication and external integrations.
- **Data Management Libraries:** The system's ability to manage and export data in various formats (JSON, XML, etc.) hinges on the availability of Java-compatible libraries for data processing, serialization, and deserialization.

Requirements

Specific Requirements

The Specific Requirements of the Attendance Management System (AMS) reflect the updated client/server model and expanded functionalities for different user roles. These requirements aim to streamline attendance management for instructors, enhance student communication, and facilitate comprehensive database management:

- **Requirement 1 (Importing Rosters):** Enable instructors to import student rosters from various CSV file formats into the server's database, handling different headers and column orders.
- **Requirement 2 (Attendance Recording):** Allow instructors to mark students as present, absent, or tardy using the client interface, with real-time server updates.
- **Requirement 3 (Late Arrivals):** Provide functionality for instructors to mark late-arriving students as tardy during or after class.
- **Requirement 4 (Attendance Adjustment):** Offer instructors the option to adjust attendance records post-class for accuracy.
- **Requirement 5 (Data Storage):** Store attendance data in an RDBMS, with export capabilities in formats like JSON and XML.
- **Requirement 6 (Automated Notifications):** Send automated email notifications to students with weekly summaries and real-time status updates.
- **Requirement 7 (Alias Flexibility):** Support legal and preferred student display names, enabling easy updates while maintaining university record synchronization.
- **Requirement 8 (Security):** Implement robust security measures, including data encryption and secure authentication.
- **Requirement 9 (System Accessibility):** Ensure system accessibility across various operating systems through lightweight client applications.

Functional Requirements

User Class 1: Professors

- **FR1.1 Attendance Input:** Enable professors to input attendance during or after classes via the client application.
- **FR1.2 Roster Management:** Allow professors to manage student rosters through the client, with server database updates.
- **FR1.3 Attendance Modification:** Provide functionality for professors to modify attendance statuses for late arrivals or corrections.
- **FR1.4 Export Functionality:** Facilitate professors exporting attendance data from the server in multiple formats.

User Class 2: Students

- **FR2.1 Attendance Visibility:** Allow students to access their attendance records via automated emails.
- **FR2.2 Personal Information Update:** Enable students to request preferred name updates, which professors can modify in the system.
- **FR2.3 Real-time Notifications:** Send immediate notifications to students for attendance status updates and weekly reports.

User Class 3: Power User/Database Manager

- **FR3.1 Comprehensive Data Management:** Enable Power Users to manage all aspects of database content, including adding/removing professors, students, classes, and sections.
- **FR3.2 User Access Control:** Grant Power Users the ability to control access rights and permissions for professors and students.
- **FR3.3 System Configuration and Maintenance:** Allow Power Users to configure system settings, perform regular maintenance, and ensure data integrity and system performance.
- **FR3.4 Advanced Reporting:** Provide advanced data reporting tools and capabilities for comprehensive insights and analysis.