

Software Requirement Specification (Version 2)

ECE 651 Team #6

Introduction

This document outlines the enhancements and additional requirements for the Attendance Management System (AMS) version 2.0. Following the successful deployment of the initial application, the scope has been expanded to include new functionalities catering to a broader audience of faculty and students. Version 2.0 transitions the system into a client/server architecture, supports multiple user registrations, handles multiple classes with sections, and introduces separate application modules.

System Overview

The AMS 2.0 will operate in a client/server model, supporting multiple clients interacting with a centralized server. This model facilitates the management of student attendance records stored on a server running on a Linux virtual machine. The system will now support multiple user roles, including faculty and students, with differentiated access rights and functionalities.

System Requirements and Functional Requirements

Client/Server Architecture

- The server application will manage the attendance repository, providing services to standalone console applications (clients).
- Clients will perform minimal processing, primarily focused on user input.
- The data will be stored in a relational database management system (RDBMS), preferably MySQL, to avoid additional licensing costs.

Multiple User Registration

- The system must support user authentication for different roles (faculty and students).
- A standalone user administration application will manage user registrations, including Add/Remove/Update functionalities.

Multiple Classes and Sections

- Faculty and students can be associated with multiple classes, with each class having one or more sections.
- The system should allow the Add/Remove/Update of classes and sections.
- Enrollment into classes can be manual or batch (via CSV file upload).

Faculty Access

- Faculty members can select a class to take attendance, record attendance (present, absent, tardy), and modify attendance records.
- The system will provide attendance reports for faculty, indicating student participation in each class.

Student Application

- Students can log in to view and modify notification preferences for each class.
- Students can access summary and detailed attendance reports for their classes.

Non-Functional Requirements

Usability

- The system should maintain a user-friendly interface, minimizing the learning curve for new users.
- The interface for managing class and section information, as well as for taking attendance, should be intuitive and straightforward.

Security and Authentication

- Enhanced security measures for data encryption and user authentication to protect sensitive information.
- The system must ensure that access is restricted to authorized users based on their roles.

Performance and Scalability

- The system should efficiently manage large volumes of data without performance degradation.
- It should be scalable to accommodate future expansions, such as additional functionalities or integrations.

Reliability

- The system must ensure high reliability in attendance tracking and reporting.

External Interface Requirements

User Interfaces

- The system will provide separate text-based terminal interfaces for faculty and students to ensure cross-platform compatibility.

Hardware and Software Interfaces

- No specific hardware requirements. The system will use Docker for containerization to support multi-platform deployment.
- Integration with an email system for sending reports and notifications.

Communication Interfaces

- Email for sending weekly reports and notifications, with potential future support for SMS or other communication methods.

Conclusion

AMS version 2.0 addresses the evolving needs of the university's attendance management process. By implementing these requirements, the system will offer enhanced flexibility, security, and efficiency, catering to both faculty and students' needs.

Use Case Mapping

