

Software Requirement Specification

ECE 651 Team #6

Introduction

This document is to provide a detailed overview of the requirements for the Attendance Management System (AMS). This system is designed to automate the process of recording and managing student attendance for university professors, ensuring efficiency, accuracy, and security.

System requirements and functional requirements

1. Attendance Recording

- a. The system should allow professors to record student attendance by displaying one student's name at a time and marking them as present, absent, or tardy with a simple keypress.

2. Late Arrival Management

- a. Professors should be able to modify a student's attendance status (e.g., from absent to tardy) at the end of a session.

3. Search Functionality

- a. The system should offer search capabilities to locate students by legal name, display name, or a unique identifier (e.g., netID) for updating attendance records.

4. Data Import and Export

- a. The system should support importing student rosters from CSV files, accommodating various formats and data fields (legal name, unique identifier/netid, and email address).
- b. The system should allow exporting attendance records to plain text, JSON, XML, or custom formats as specified by the user.

5. Enrollment Management

- a. The system should enable the addition of late-enrolled students and the removal of students who drop the course, with appropriate updates to attendance records.
- b. Attendance records for dropped students should be retained but marked to prevent their appearance in future attendance-taking sessions.

6. Reporting and Notifications

- a. Weekly attendance reports should be generated and sent to students via email, covering each class session within the week.

- b. The system should notify students of any changes to their attendance status, detailing the class session and the nature of the change.
- c. In cases where a student does not have an email address, the system should provide the professor with a summary report of anomalies for manual follow-up.

7. Customization

- a. Students may request to have a display name used during the attendance process, which should be kept in sync with their legal name unless otherwise specified.
- b. The system should accommodate future additions of notification methods (e.g., SMS) through a modular design.

8. Security and Authentication

- a. The system should implement encryption for storing sensitive information and manage access through a username and password authentication system, ensuring that only authorized professors can use the system.
- b. Passwords shall not be stored in plaintext.

9. Multi-course and Multi-user Support

- a. The system should support multiple courses managed by a single professor and allow students to be enrolled in multiple courses.

10. Data Preservation

- a. The system should preserve the data after exit, and the data should remain the same when running the system next time.

External interface requirements

1. User Interfaces

- a. Text-based terminal interface for cross-platform compatibility.
- b. Simple, intuitive commands for recording attendance, managing student records, and generating reports.

2. Hardware Interfaces

- a. No specific hardware requirements due to the text-based terminal approach.

3. Software Interfaces

- a. Docker for containerization and multi-platform support.
- b. Email system integration for sending reports and notifications.

4. Communication Interfaces

- a. Email for weekly reports and notifications.
- b. Potential future support for SMS or other communication methods.

Non-functional requirements

1. **Usability:** The system should be user-friendly, allowing professors to efficiently manage attendance with minimal training.
2. **Reliability:** Ensure accurate tracking and reporting of attendance data.
3. **Performance:** System should handle large class sizes efficiently, without significant delays in processing attendance.
4. **Scalability:** Design to accommodate future enhancements, including additional communication methods and integration with university systems.
5. **Security:** Implement robust security measures to protect sensitive student information and system access.