TEAM - 8 Requirements for Final Proj EVOL 1 :

1. Functional Requirements

1.1 Managing Parsed Courses and Parsing Enrollment List

1. When the program starts, the system will ask the professor to choose a task to do. Choices are managing existing course, parsing enrollment list for new course, and “logging out of the program”/ stopping the program by entering “q” from the terminal.
2. If the professor chooses to parse an enrollment list file, the system will ask the professor to give a path to the file as a string. The program will then parse the file and make the necessary objects before the first day of the class.
3. If the professor chooses to manage courses, the system will show the list of the courses the professor teaches so that the teacher can select from the available options.
4. Once the professor chooses a course, the program will ask the professor to choose what tasks they want to complete for the course. Options available are take attendance, add student, drop student, change student display name, and view attendance reports.
5. The professor can not only take attendance at the beginning of a lecture but can also change the attendance record during the class period and/or after class ends.
6. If the professor chooses to take attendance for everyone at the beginning of the lecture, the program will show the names of the students along with attendance status options one by one until all have been marked.
7. If the professor chooses to take attendance by netID, the program will let the professor search for an enrolled student by their netID and let the professor take attendance.
8. The program will allow professors to change old lecture attendance record.
9. Once a professor has completed the current task, the system will ask the professor again what task to do.

1.3 Enrollment List File

1. The system will support parsing Enrollment List from csv file only for evol 1.
2. The system will support csv file with and without headers
3. The system will support csv file where the column orders are switched

1.4 Dynamic Enrollment

1. The system will accommodate late enrollees and dropped students throughout the semester.
2. Teacher will have option to add students to the enrollment list. Teacher will have local copy of the enrollment data because the 1. system is not networked. 2. teacher doesn’t receive late enrollee via another csv file. 3. it is assumed that teacher has received email

on their own email server. This email is assumed to be valid confirmation that student is a valid fully registered student with the school and the course.

3. Teacher will have the option to drop students. Same reasonings as 2.3.2 applies this.

1.5 Data Export

1. An export option for attendance record shall be provided for other formats such as JSON and XML formats only for evol 1.

1.6 Student Enrollment Change Notification

1. Whenever a student's attendance status is changed, the teacher will send a notification to the student from their computer. In evol 1, notifications are delivered via email only.
2. Whenever a student's attendance status is changed, SMS will not be sent. Only email is sent.

1.7 Daeomonizing the system & Attendance Report Notification

1. Even when the teacher is logged out, the system should be able to run and send weekly reports on its own.
2. Two completely separate programs will run. These programs are 1. Professor's attendance taking system 2. Email sender program that sends out sends out attendance records once every week to every student.

1.9 Display Name Management

1. The student's preferred name and last name will be displayed. If a student doesn't have a preferred name, the student's legal first and last name will be displayed.
2. Students can request a change in the name displayed during the attendance process. The teacher will change the student's preferred name.

2. Non-Functional Requirements

2.1 Security

1. No authentication is required for the teacher. Teachers can just run the program without logging in.
2. Attendance records are encrypted and file name is visible to other users but the content itself is not meaningful without key. The standard is if someone else other than the teacher ssh into the server, he can see the attendance record file and but will not be able to modify/see file's valuable contents without the key.

2.3 Reliability

1. Don’t crash the system when the teacher runs the program.
2. If the teacher “logs out”/ stops the program in the middle of taking attendance or after parsing courses or after any activity that involves student record changes, attendance record data will be saved in a text file for persistence.
3. When the teacher “logs back in” to the program again by reruning the program, the teacher should be able to start from where they left off. In other words, the activities the teacher did before restarting the program will be present in the current running program after restart.
4. This persistency data storage file will be encrypted and will require a key to be useful.

2.4 Portability

1. The client doesn’t care if the program is containerized.

2.5. Constraints

1. The system's functionality and scalability will be limited by the capabilities of the text-based terminal interface and the hardware (4 cores, 4 GB RAM) that the program is running.