Calendar Evolution 1: Initial Requirements

This semester’s project is to make a calendar program. You can look to existing calendar programs (e.g., Google Calendar) for inspiration and UI ideas, but you are not constrained to develop something that looks or feels like any specific piece of software—you can be as creative as you want with how it actually works, as long as you meet these basic requirements.

1. Server

(a) Your software must have a server that supports an arbitrary number of users.
(b) Each user must create an account before using the system.
(c) Passwords must be kept in a secure manner (e.g., salted + hashed)
(d) All communication between the clients and server must be encrypted
(e) The server must maintain state in a persistent fashion.

2. Client: Basic Calendar Functionality

(a) A user shall be able to create events on his/her calendar, including the date/time of the event, the name of the event, and other information they may wish to include.
(b) A user shall be able to indicate that an event repeats at regular intervals (weekly, daily, monthly, etc) for a specified period.
(c) The client shall display the user’s calendar with its events in a intuitive manner.
(d) The user shall be able to elect to have the system send him/her an e-mail reminder about an event at a time of his/her choosing.
(e) A user shall be able to control which other user’s can see the events on his/her calendar
   i. By default, events shall be private to the user (not visible to others)
   ii. A user shall be able to create named groups of other users (e.g., “Coworkers”, “Friends”, “Family”).
   iii. A user shall be able to change the level of access for any combination of users and/or groups to any of his/her calendar events. The levels of access shall be Private (no access/no visibility), Busy Only (shows the time as occupied, but no details), See All (show all information), Modify (can see and modify the event).
   iv. The user shall be able to enter multiple rules, which take precedence in the order that they appear. For example, if “Frodo” is in the group “Hobbits”, the user shall be able to enter a rule for Frodo first, then a rule for the Hobbits group after it, with the former applying to Frodo, and the later applying to other members of Hobbits.
   v. The user shall be able to re-order or delete existing rules for an event.
   vi. You may (but do not have to) organize events into multiple logical calendars, and control the access of events at this granularity if you wish.
   vii. Events shared from other users’ calendars shall be displayed in such a way that they can be easily distinguished from the user’s own events.