Project Overview

Our project will be the development of a well-featured planning/task management app, allowing an individual or groups of people to coordinate by placing events onto a calendar. Basic features would include a calendar display, the ability to place events onto it, create recurring events, save, load, and share sets of events, categorize events, and allow multiple people to interact with a calendar and its events, with differing permission levels.

The calendar itself is essentially a particular set of events. The calendar can be displayed as a single day, month, or year at a time. Events consist of a title, an optional description, a timeframe (what date and time they start and end), an optional category for the event (which may be user-defined), and a set of more general properties which can be optionally added to an event, such as which users are involved and where it is happening. Recurring events can be made from an existing event, allowing you to make it repeat automatically at a particular interval (such as daily, weekly, on a particular day of each month, yearly on the same date, etc.). Sharing and saving events is accomplished by having them stored in a file format which can then be loaded back into the calendar app at a later time. Event categories are tags that can be applied to events so they can be filtered and grouped. User permission levels are applied to particular users viewing a calendar to limit what events or categories of events they are allowed to create, delete, edit, etc.

Our group is planning to use a plan-driven approach to creating this app for simplicity. We do not anticipate that the requirements for the project will change in any significant way over the course of development, so an agile approach would be excessive and unhelpful.

Stakeholders

- End Users: Individuals/Students
    - Role: People who want to use the application for personal planning and task management
    - Wants:
        - Easy to use Calendar UI
        - Ability to add, edit, delete, and group events
        - Event/task reminders
        - Option to filter and group tasks
        - Relevant and unique features
- End Users: Teams/Groups
    - Role: Teams using the app such as project groups, study groups, or small organizations using the app to coordinate and collaborate on events/tasks

    - Wants:
        - Shared calendars for collaboration purposes
        - Permission hierarchy to view, edit, and delete
        - Ability to track team progress
        - Notification for group events and deadlines
        - Seamless cross platform compatibility
- Development Team
    - Role: Designing, developing, and testing the application (project group)
    - Wants:
        - Clear requirements to follow in the plan driven approach
        - Functional application that fulfills all requirements
        - Positive feedback from users to validate design choices
- Test Users
    - Role: Users who will test and provide feedback on the application
    - Wants:
        - User friendly interface
        - Features that are practical and realistic to use on a daily basis
        - Stable app that runs without any crashes, glitches, or performance issues

Requirements Elicitation

Gathering Requirements

- We will use a combination of interviewing and observation in order to better understand the needs of our users. The interviews will be closed and composed of predetermined questions so that we can accurately assess their answers. The development team will also brainstorm together alongside some potential users in order to generate a large amount of creative ideas quickly.

Questionnaire Draft

1. Where do you currently organize your daily tasks?
    - A: Mobile
    - B: Web
    - C: Paper
    - D: Other
2. What challenges have you faced when using other task management applications?
3. Of these task management features, which is the most important to you?
    - A: Progress tracking
    - B: Reminders
    - C: Collaboration
    - D: Custom to-do lists

4. Would you prefer team members to have designated access levels that limit their ability to edit a given task?
5. Would you like two-factor authentication to be optional?

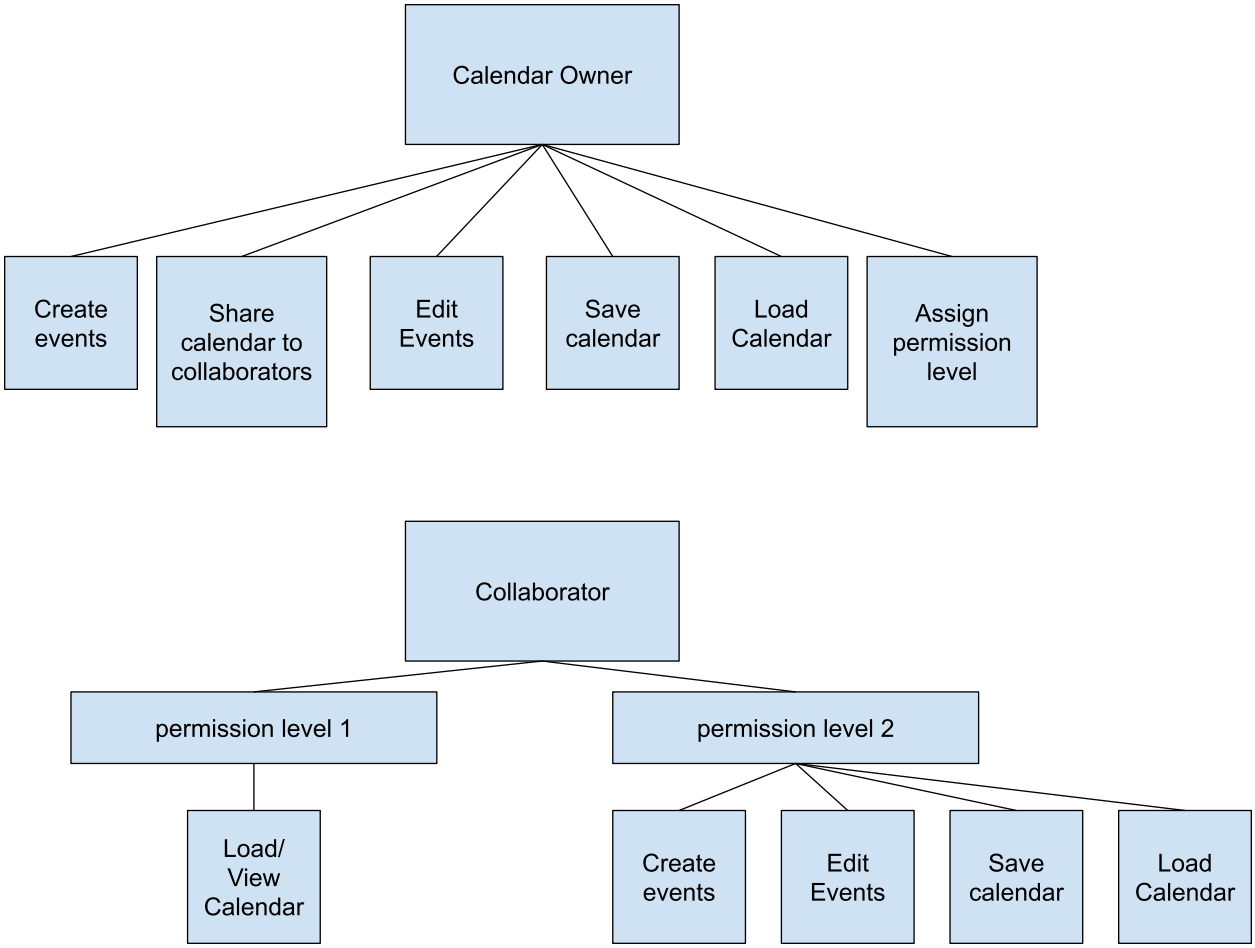
Organize Requirements

- We will organize the requirements into distinct categories such as:
    - Interface. This is for requirements that detail how the system looks as well as the way users can customize their visual experience.
    - Functionality. This is for the requirements that detail the specific features of the system, what tools the user is provided with.
    - Security. This is for the requirements that detail how data is handled and stored as well as who is given permissions to access that data.
    - Performance. This is for the requirements that detail how responsive and efficient the system must be across all platforms.
    -

Requirements Specification

ID	Requirement type	Requirement	Supports
FR-1	Functional	The system shall allow the user to create, edit, and delete events.	Event Management
FR-2	Functional	The system shall allow the user to establish recurring events.	Event Management
FR-3	Functional	The systems shall allow the user to filter the events by category.	Collaboration
FR-4	Functional	The systems shall allow the user to assign permission levels.	Collaboration
FR-5	Functional	The systems shall allow two-factor authentication.	Security
FR-6	Functional	The systems shall	Data Management

		allow users to load and save calendars.	
FR-7	Functional	The system shall let the user see the calendar in either a daily, weekly, or monthly format.	User Interface
NFR-1	Non-functional	The system shall store user data securely.	FR-4 FR-5 FR-6
NFR-2	Non-functional	The system shall provide an intuitive experience such that a new user can be able to create and edit an event within the first five minutes.	FR-1 FR-2 FR-4 FR-6 FR-7
NFR-3	Non-functional	The system shall respond to users' interaction within 3 seconds.	FR-1 FR-2 FR-3


Calendar Owner
Create events
Share calendar to collaborators
Edit Events
Save calendar
Load Calendar
Assign permission level
Collaborator
permission level 1
permission level 2
Load/ View Calendar
Create events
Edit Events
Save calendar
Load Calendar

Requirements Validation

We will conduct a variety of checks to ensure that each of our requirements is correct. In-depth reviews will be conducted with different stakeholders, testing, and providing feedback for each of the requirements. Prototypes will be used to ensure that the requirements are working as intended and any problems can be addressed early. Feedback received from the prototypes will be used to improve future iterations of the requirement. Test cases will be another crucial tool we will use to determine if our requirements are correct. With specific and well-defined test cases, we will identify a set of criteria that our requirements must meet. This criterion will then be tested during the development stage to ensure our requirements are working as intended.

Requirement: The system shall allow the user to create, edit, and delete events.

Verifiability - Yes, specific test cases can be used to verify that the system meets the specified requirement. Test cases will verify that events are properly added, edited, and deleted.
Comprehensibility - Requirement is clear and easily understood.
Traceability- This requirement supports the system's event management goal. It provides crucial features needed for event management. This requirement is traceable to the stakeholder's need to create, edit, and delete events.
Adaptability- Highly adaptable, can be updated later to accommodate different needs.

Requirement: The system shall allow the user to establish recurring events.

Verifiability - Yes, specific test cases can be used to verify that the system meets the specified requirement. Test cases will verify that a repeated event is properly added.
Comprehensibility - Requirement is clear and easily understood.
Traceability- This requirement supports the system's event management goal. This requirement is traceable to the stakeholder’s need to have an efficient feature that handles repeating events.
Adaptability- Moderately adaptable; more complex patterns may be difficult to implement. Basic patterns such as daily, weekly, or monthly can be modified with ease.

Requirement: The systems shall allow the user to filter the events by category.

Verifiability - Yes, specific test cases can be used to verify that the system meets the specified requirement. Test cases will verify if the filter is correctly sorting and displaying the correct information.
Comprehensibility - Requirement is clear and easily understood.

Traceability- This requirement supports the system's collaboration and usability goal. This can be traced back to the stakeholders' need to filter through events.
Adaptability- Highly adaptable, can be updated later to accommodate more advanced selections.

Requirement: The systems shall allow the user to assign permission levels.

Verifiability - Yes, specific test cases can be used to verify that the system meets the specified requirement. Test cases will verify that the users' viewing, editing, and other abilities are in line with their permission levels.
Comprehensibility - Requirement is somewhat clear, as it does not specify the different types of permission levels.
Traceability- This requirement supports the system's collaboration goal. This can be traced back to the stakeholders' need to have a collaboration system that allows teams to have a hierarchy system.
Adaptability- Can be changed so long as the changes aren't too drastic.

Requirement: The systems shall allow two-factor authentication.

Verifiability - Yes, specific test cases can be used to verify that the system meets the specified requirement. Test cases will verify if two-factor authentication is correctly sending and receiving the authentication data before gaining access to the system.
Comprehensibility - Requirement is clear, could be improved by specifying which type of two-factor authentication system will be used.
Traceability- This requirement supports the security needs of the system. This can be traced back to the stakeholders' need for a secure system where they can safely input their information.
Adaptability- Can be changed to support different types of two-factor authentication methods.

Requirement: The systems shall allow users to load and save calendars.

Verifiability - Yes, specific test cases can be used to verify that the system meets the specified requirement. Test cases will verify that a user's calendars are correctly loaded and saved.
Comprehensibility - The requirement is clear and simple to understand.
Traceability- This requirement supports the data management goal of the system. This can be traced back to the stakeholders' need to store calendar information between devices or sessions.

Adaptability- Can be easily changed to accommodate different saving and loading methods.

Requirement: The system shall let the user see the calendar in either a daily, weekly, or monthly format.

Verifiability - Yes, specific test cases can be used to verify that the system meets the specified requirement. Test cases will verify that each view (daily, weekly, or monthly) can be accessed and properly shown.
Comprehensibility - The requirement is clear and simple to understand. The specific time periods (daily, weekly, or monthly) allow for no confusion on this requirement.
Traceability- This requirement supports the user interface goal of the system. This can be traced back to the stakeholders' need for an easy-to-use calendar UI.
Adaptability- Can be easily changed to include a greater variety of calendar views.

Requirement: The system shall store user data securely.

Verifiability - Can be somewhat verified in the form of tests that ensure data is being properly encrypted.
Comprehensibility - Requirement is mostly clear, could be improved by providing a standard that must be met to ensure the data is secured.
Traceability- This requirement is traceable to the stakeholders' need for a secure system. This supports the security goal of the system and other requirements such as FR-4, FR-5, and FR-6.
Adaptability- Can be changed to include more advanced security practices to match the needs of users.

Requirement: The system shall provide an intuitive experience such that a new user can create and edit an event within the first five minutes.

Verifiability - Yes, this can be easily verified through usability testing, where testers will check if they are able to create and edit an event within the first five minutes.
Comprehensibility - The requirement is clear and simple to understand.
Traceability- This requirement supports the user interface goal of the system. This can be traced back to the stakeholders' need for an easy-to-use calendar UI.
Adaptability- Changes to the user interface can easily be added to provide improvements to the user's experience.

Requirement: The system shall respond to users' interaction within 3 seconds.

Verifiability - Yes, this requirement can be verified through performance testing to ensure that the system responds to users' interaction within 3 seconds.
Comprehensibility - The requirement is clear and simple to understand.
Traceability- This requirement supports the performance goal of the system. This can be traced to the stakeholders' expectations of a responsive system. Support other requirements by providing a quick response time.
Adaptability- Can be changed to provide further optimization for the system.