Event Horizon - High Level Design Document

Jonathan Lavoie, Mason Cacheino, Nooh Alavi, Rahif Haffeez, Shawn Xiao

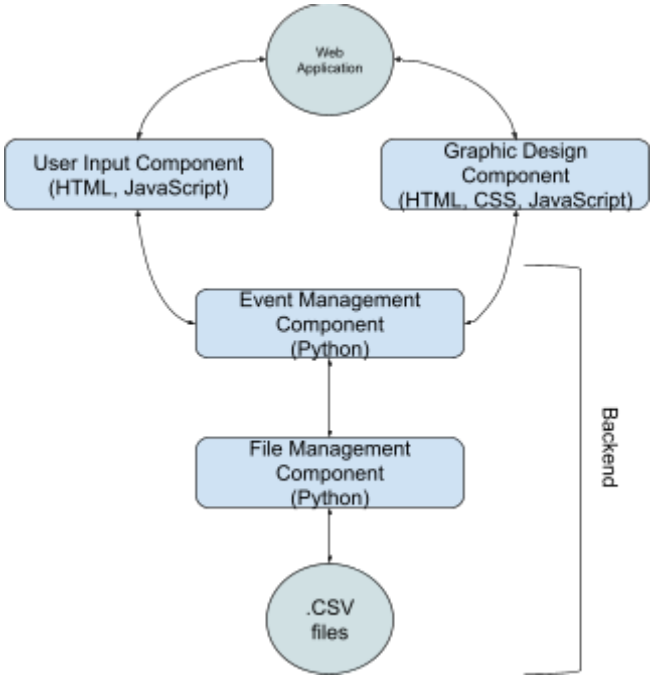
1. Architectural Overview

★ User Input Component
- ➢ Front-end will be done using HTML/CSS/JavaScript
- ➢ This will communicate with the Calendar Management Component and the Graphic Display Component.
- ➢ Forms/buttons for user input and interaction.
- ➢ Validation checks to ensure that users are not entering invalid information and/or injecting malicious code.

★ Graphic Display Component
- ➢ Front-end will be done using HTML/CSS/JavaScript
- ➢ This will communicate with the Calendar Management Component and the User Input Component.
- ➢ CSS styling for a clean GUI.
- ➢ Handle display filtering (by month, week, day).

★ Event Management Component
- ➢ Back-end will be written in Python
- ➢ Receives user events and performs the corresponding action.
- ➢ We will store the events in a dictionary (the key will be each event's unique ID).
- ➢ Dynamically add, remove, and modify events.

★ File Management (Save and Load) Component
- ➢ Back-end will be written in Python
- ➢ Dealing with `.csv` files — maybe will integrate `.ics` later
- ➢ Will also deal with loading in the files, and will communicate with the Calendar Management Component, which will then take over.



2. Technology Stack and Design Choices

★ Front End - HTML, CSS, JavaScript
- ➢ HTML provides the structure for the calendar interface
- ➢ CSS is used to style the user interface. Can ensure the app can work on different screen sizes.
- ➢ All interfaces / GUI can be rendered in the browser.
- ➢ JavaScript allows for interactivity on the front end, which allows for dynamic updates to the calendar, such as event creation and editing without having to refresh the page.
- ➢ More API friendly.
- ➢ HTML and CSS are super easy to work with and can be easy to pick up for those who don't have much web development experience.

★ Backend - Python
- ➢ Python has good readability and simplicity. Most team members already know how to use Python, but it also is really easy to learn for members who don't have experience with it.
- ➢ Python has a lot of different libraries and frameworks. e.g. Flask

★ Database Technology - CSV / ICS
- ➢ CSV is simple and easy to use especially in Python
- ➢ CSV allows for easy reading and writing of data.
- ➢ ICS is a standard calendar format that would be able to be used with other commercial calendar apps like Google Calendar, and Apple's Calendar app

3. Data Design

★ Year:

Will contain 12 Months

★ Month:

Will contain 5 weeks

★ Week:

Will contain 7 days

★ Days:

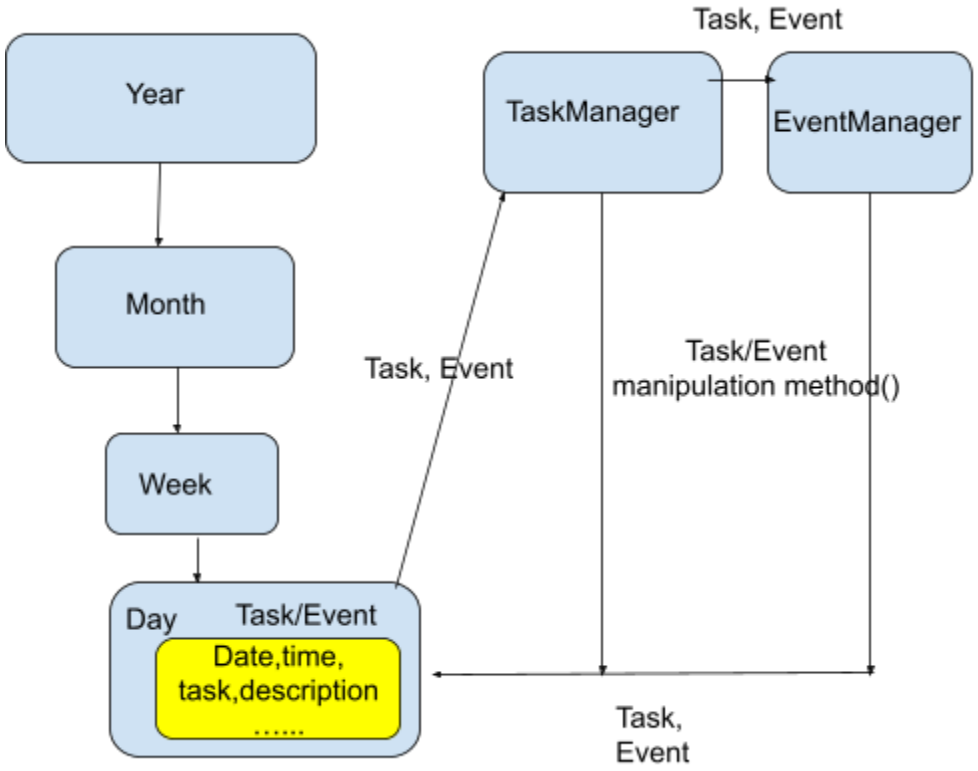
Will contain several tasks and events

★ Event:
- ➢ title (string, name of the event)
- ➢ notes (string, an optional description)

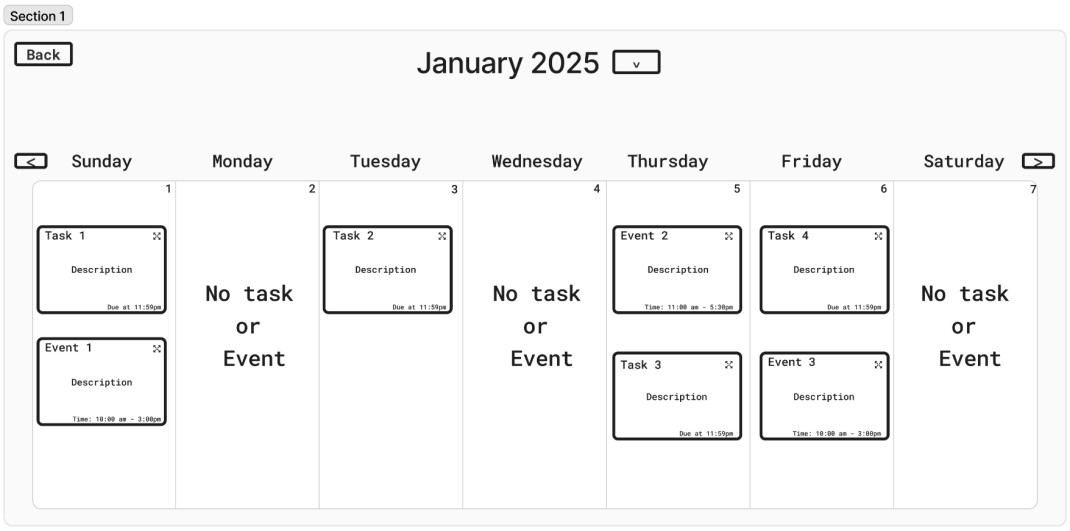
- ➢ start_time (start of the event,(type of data depends on if the team settles on csv or ics))
- ➢ end_time (integer/ics, end of the event)
- ➢ location (string, optional)
- ➢ repeatability (string, e.g. “weekly”, “monthly”, “once (default)”, etc)
- ➢ event_id (integer/ics, unique identifier)
- ➢ color (hex, how it shows up on the calendar)

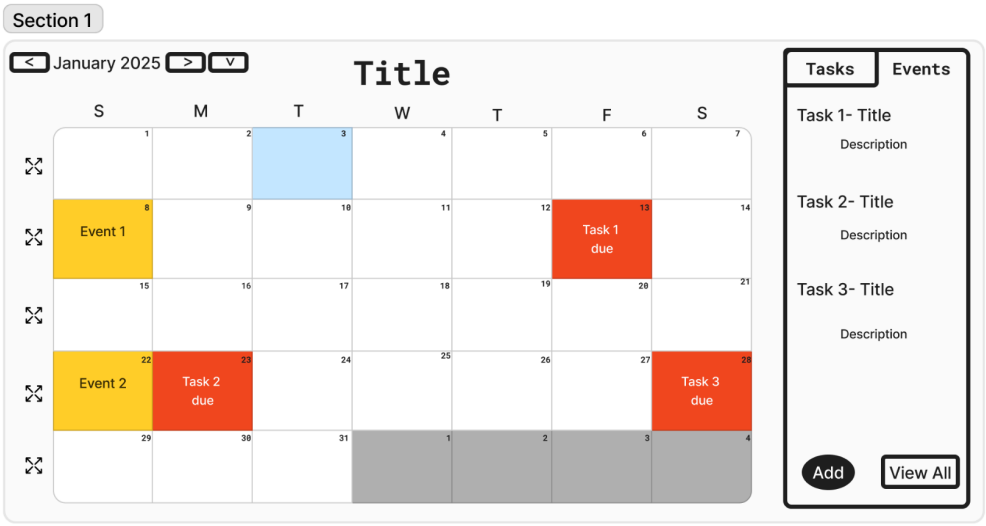
★ Task:

- ➢ title (string, name of task)
- ➢ notes (string, optional description)
- ➢ deadline (integer/ics, when task is due)
- ➢ status (pending, done)
- ➢ priority (low, medium, high)
- ➢ color (hex)



4. User Interface Sketches





5. Risk Analysis

- ★ Problem: Losing track of time and not getting features done
  - ➢ Mitigation: With Trello, the team will list the deliverables and deadline dates for the deliverables. If a deadline passes without the deliverable not being complete, the team should handle the deliverable ASAP to keep track of.
- ★ Problem: The team not knowing their responsibilities and working on the same features which causes redundant work and wastes time.
  - ➢ Mitigation: Through communication with the meetings or the Discord, members will tell others what they'll be working on so other members don't work on it.
- ★ Problem: Local ICS/CSV file may not load correctly when rendering the calendar.
  - ➢ Mitigation: For ICS, using a more robust library to parse the files will help mitigate this. Validating CSV files to make sure they're written correctly.