

Fusion Five Studio's Requirements Draft

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

Step 1 — Kickoff: Meeting the Customer

- Calendar app.
- Add, delete, edit events.
- Load and export files (database) for calendar information.
- Three display views — events by day, week, and month.
- Challenge
 - Add to-do list
 - Cross out tasks already done
 - Focused study times as well as the company events
 - Time block
 - Organize tasks which need to be prioritized — priority list (flag, organize by due date, etc.)
- This app targets both general users and businesses.
 - For businesses:
 - Professionals: to keep track of work-related meetings, appointments, and deadlines + prioritize their tasks.
 - Teams/organizations: may use the app for managing schedules, tracking shared tasks, and organizing company events.
 - For general users:
 - Students: can manage class schedules, assignments, exams, and personal tasks.
 - Other individuals: can manage their daily, weekly, and monthly events, to-do lists, and personal goals.
 - Other kinds of people who may find this useful include freelancers, entrepreneurs, and simply family members trying to make sure they synchronize their schedules and do not forget anything important.

Step 2 — Initial Brainstorming and Follow-Up Questions

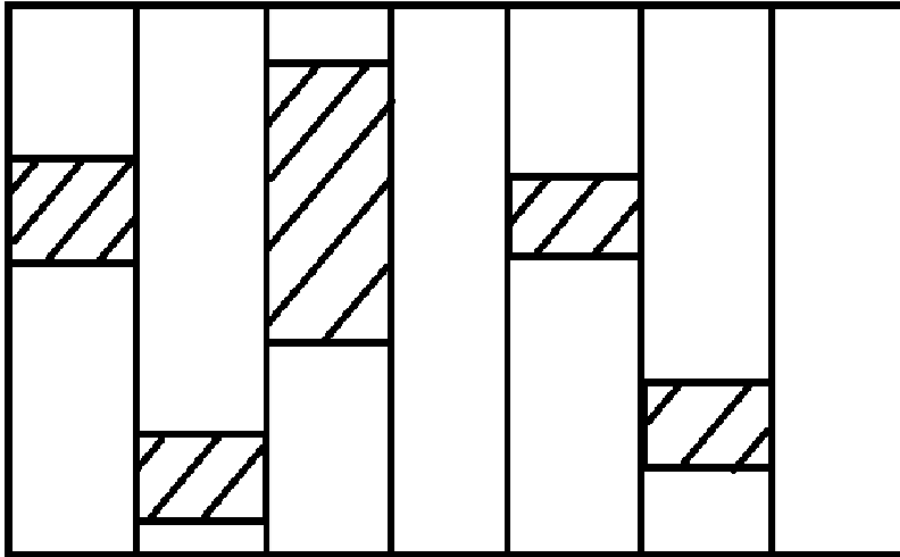
- We need to make a calendar application / event manager, where the user will be able to add, delete, and modify events.
 - A calendar template can determine leap year/common year
 - Each event will have a start time, end time, priority level (?)
 - These events will be saved into and loaded from a database (.json/.ics files ???)
 - Support for different accounts/users
 - Option to download it as a database and upload to the program.
- We need a GUI — calendar in the middle, to-do list on the side.
 - When displaying to the user, they can choose to view the events based on (1) day, (2) week, or (3) month.
 - User can configure when the week will start.
- Priority control: Auto-set higher priority to the events named with specific keywords. (due, deadline, appointment, etc..)

- Each event can have: name, date, priority, notes, (location? person?).
- Maybe we can add some system for repetitive events (weekly, monthly, etc.)
- Follow Up Questions:
 - Q. Are tasks and events two different things? Should there be any connections between the two? (should a task create events, etc.)
 - We will have tasks, events, and focus time (to do the tasks)
 - Q. An alarm before important events? Or it is just working as a schedule?(User may set an alarm in advance instead of)
 - A. Not a strict requirement but it is fine to add that.
 - Q. confirm regarding GUI layout

Month View

Notes

Week View

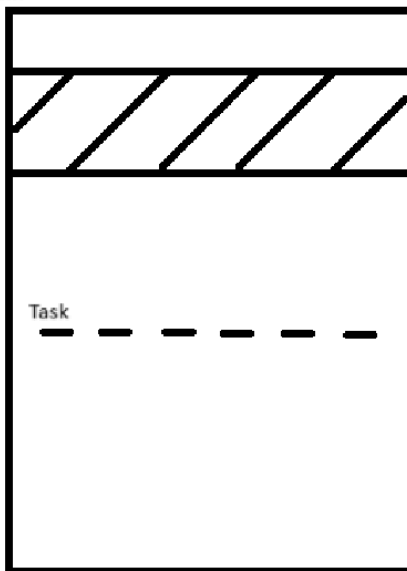


To-Do List

Day View

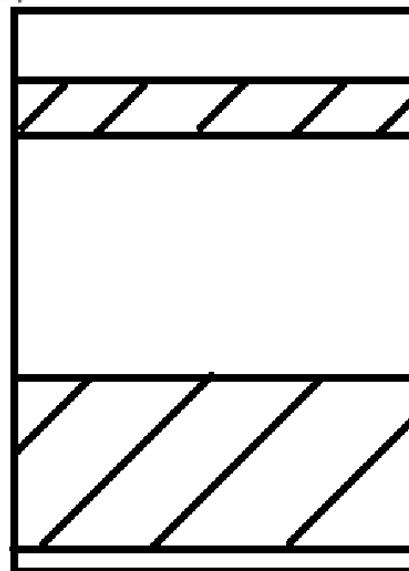
12 am

12 pm



Task

12 pm



12 am

Notes

Step 3 — Independent Research

- ICS files
 - ICS files are plain text files that contain information about a calendar event, including the event's:
 - Title
 - Summary and description
 - Start and end times (Format of time is Timestamp instead of regular time format)
 - Can probably convert w/ Python

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VJOURNAL
DTSTAMP:19970324T120000Z
UID:uid5@example.com
ORGANIZER:MAILTO:jsmith@example.com
STATUS:DRAFT
CLASS:DRAFT
```
 - Location
 - Alert trigger
 - All popular calendar apps use this format including Microsoft Outlook, Apple Calendar, Google Calendar, and Mozilla Thunderbird Lightning Calendar.
 - Can convert to and from .csv
- Similar programs
 - A function that can switch between the month, day, year.
 - A list of the time by hour on the side.
 - A function that allows to colour code tasks/events
 - A function that can toggle Housekeeper(assistant) mode' which fills up all the gaps with randomly generated meaningful activities (dining,housework, homework, reading, etc)
 - A search function that allows the user to search for a certain task/event based on the date, time, and name of the task/name.
- How to generate profit(commercial based)
- Languages/frameworks
 - Python calendar module — can write frontend in HTML/CSS/JS and then backend in Python.
 - Learn how to Web Development

Step 4 — Requirements and Prioritization

- Must-Have Requirements
 - The ability to close the application
 - Calendar application that the user can add, remove, and modify events and tasks.
 - Events are defined with a start and end time and have a priority level.
 - Tasks don't have a start or end time and act as a deadline with a priority level.

- A to-do list on the side that contains events/tasks.
 - Events/tasks can be loaded up from files.
- Should-Have Requirements
 - Multiple accounts/calendars
 - Colour-coding
- Could-Have Requirements
 - Themes
 - Option to change week's starting date
 - Nice features
 - Notification system
- Won't-Have Requirements
 - On-screen keyboard.
 - AI 😞
 - Self-destruction protocol (wiping all events and tasks)
 - A cheater mode which can flip the calendar to one with completely different tasks and events.

Key stakeholders:

- General users, who use it for personal time management like students who use it to manage their class schedule.
- Business users, who would use it for work-related purposes.