CSE 110 Team 35

Starting Pitch: Project Specification for Journal App

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide a starting outline specifying the requirements, features, and functionality of the Journal App. This document serves as an expected plan for Teaching Assistants, mentors, and Professor Powell. It guides the development team so they know what to expect and accomplish.

1.2 Product Scope

The Journal App is an integrated platform that assists users to outline, visualize, and manage their tasks and journal entries. It attempts to address the issue of not being able to have a singular platform for closely associated task management related functionalities.

In Scope:
- Task Management
- Aggregated task completion statistics
- Calendar
- Kanban task view
- Cloud synchronization
- Partial accessibility support

Out-Of-Scope:
- Collaboration
- Social media integration
- Translation services
- Support for languages other than English
- Complete accessibility support

2. Statement of Purpose

2.1 Product Perspective

A lot of people tend to lose track on what they have done, currently doing, and what they are doing in the future. It is a huge hassle for them to switch back and forth between different tools like Google Calendar, Trello, etc. This product provides a centralized platform for regular users to keep track of their workflow.

2.2 Product Functions

The major functions of this product include:

- Task Manager
    - Displays today's tasks with at least the following information:
        - Name
        - Deadline
    - Displays an aggregated statistic for today's task information
        - For example, a pie chart displaying % for completed tasks vs incomplete
    - Special indicator for completed tasks
    - Displays upcoming tasks for up to 1 week from today with a more condensed view
- Calendar
    - Displays a calendar view for current month with task of the day in each cell
    - Able to perform simple operations (e.g. click-to-complete task) when clicking on an individual task item in each cell
    - Able to expand this view for future and past dates as well
- Kanban Task Viewer
    - Provide a Kanban view for selected tasks that are categorized by their status
        - For example, one column for planned tasks, one for working tasks, one for completed tasks
    - Able to drag-and-drop tasks around to quickly change their status

    - Able to add tasks directly under each column
    - Able to create as many columns as user needs
- Journal
    - Provides a explorer-based view to intuitively manage the journals
    - Able to link journal entries to task
    - Supports LaTex & Markdown
    - Toggle between seeing a split screen (Markdown/LaTex view & overview) and normal screen (editing directly on overview) for editing journal entries
    - Entries from moodlet widget from Portal show up in selected Journal entry with timestamp of when the entry was made

The optional / bonus functions of this product include:

- Cloud Synchronization
    - Able to make users synchronize their tasks to different devices though an account system

2.3 Personas

2.3.1 Thomas

Background: Thomas is an university student at University of California, Riverside majoring in Computer Science. He is a hardcore competitive programming enjoyer and likes to type in Markdown and LaTex on his laptop in NeoVim. He likes stuff being "raw" like using LaTex instead of equations on Google Docs since that means he will have the most control and he believes it is elegant.

Wants: Be able to type and render entries that have Markdown and LaTex. Keyboard shortcuts to quickly interact with the apps.

Pain Points: Websites have so many unnecessary components and features he doesn't even use but will make his device slow down.

2.3.2 Jessie

Background: Jessie is a young Instagram influencer. She loves to show off her active lifestyle and her personality traits include being a

"clean girl" and the "it girl". Every product she uses goes viral, and her followers will do anything that she does. She has lots of sponsorships waiting to be filmed.

Wants: Aesthetically pleasing app to show off how busy she is on her IG story.

Pain Points: Some apps are too complicated to use.

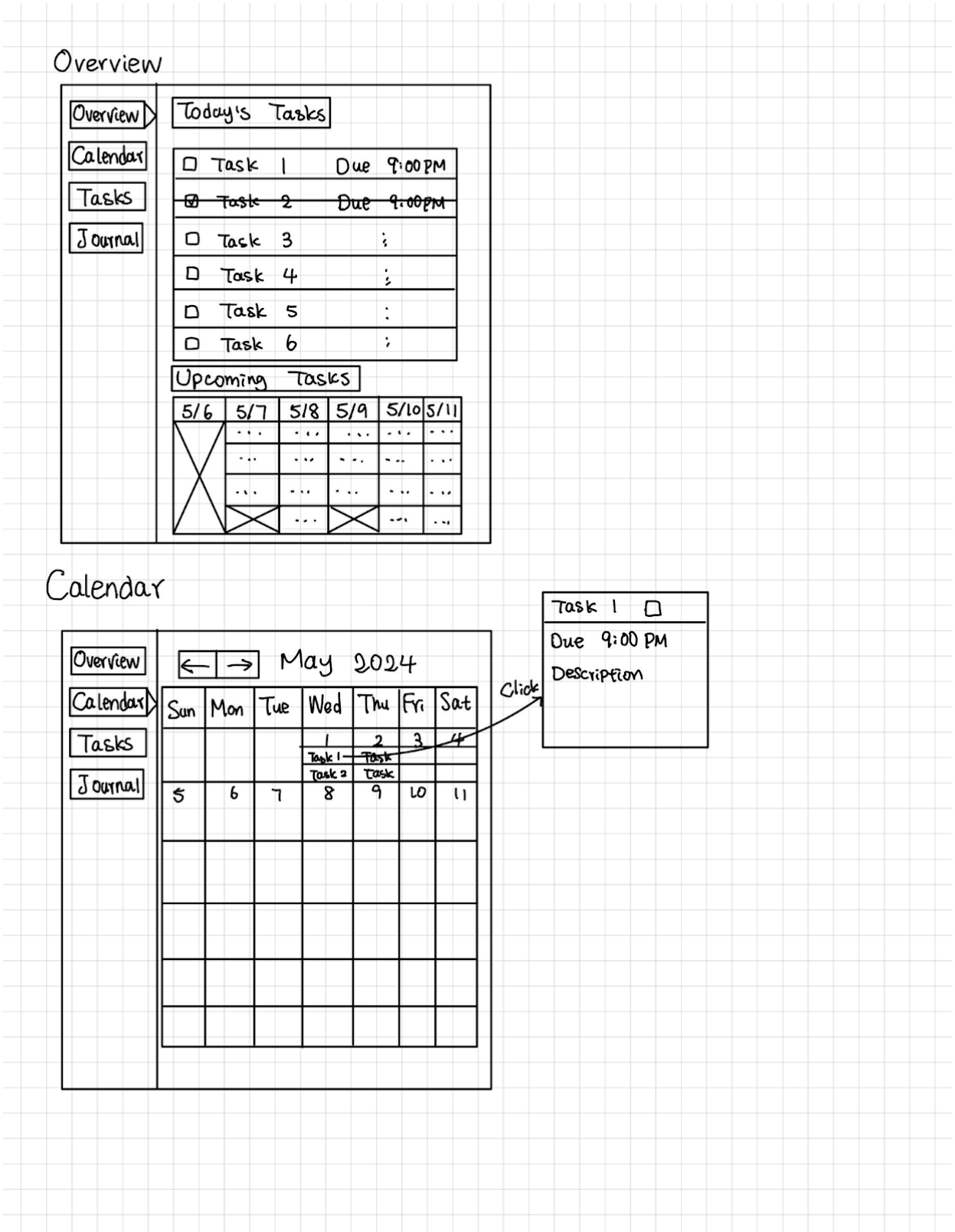
2.3.3 Flora

Background: Flora is the CEO of a very successful lifestyle brand. She's a very busy person with only very little free time since she's always travelling and doesn't have the time to talk to her employees very often.

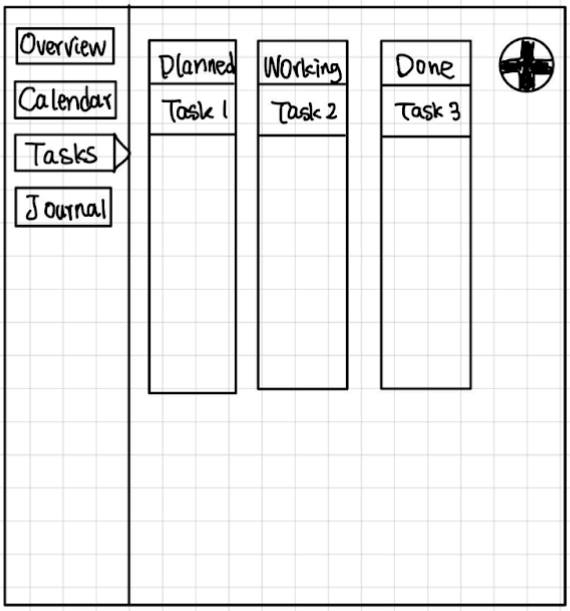
Wants: A way to see all of their employee's schedules and what's currently being worked on. A summary of everything could be nice too.

Pain Points: She has to go to many different tabs to get all the data she wants. Too complicated.

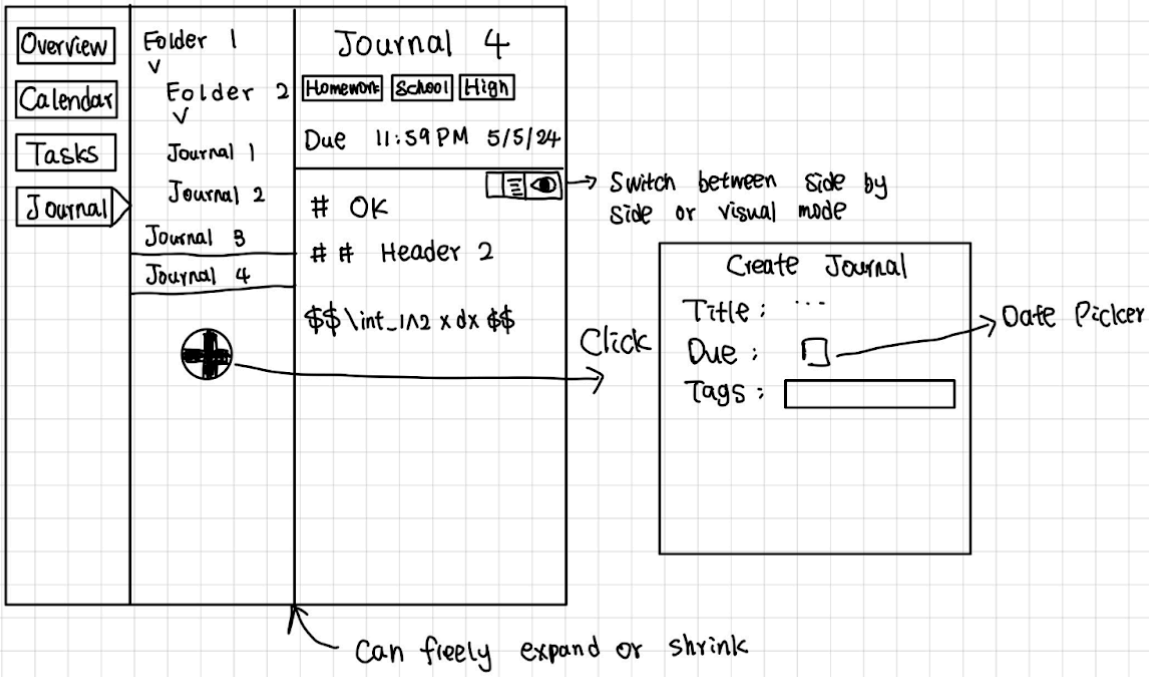
3. Wireframes



Kanban Task View



Journal



4. Risks / Rabbit Holes

4.1 Risks

4.1.1 Time vs Quality

Per the iron triangle in project management discussed in lecture, we need to compromise on the quality due to tight time constraints and deadlines. It is not possible to implement every single feature to the best extent possible.

4.1.2 Difference in Experience

It is crucial to consider the experience gap between team members. Some have a lot of experience in web development, some have a bit, and some have essentially none. It is hard to coordinate everyone together with such a skill gap.

4.2 Rabbit Holes

4.2.1 Journal

Journal is a pivot part of our application. However, since it involves users being able to type and visualize what they want, it is easy to go unnecessarily deep into the feature development (e.g. support Mermaid diagram or even embed Google Slide preview). Due to the time constraint, this needs to be carefully considered to be able to complete the project on time.

4.2.2 Task Manager

Task Manager is also a core feature of our application. However, choosing what to display in the overview page is very important. We need to just fit the right amount of information. If we fit too much then it’ll be too “fat” for an overview page but if we fit too little

then users won’t be able to perform simple task management operations. This needs to be carefully considered to account for users’ needs.

5. Technical Requirements

5.1 Backend

Since this is a website app that is “local first”, strictly speaking there isn’t going to be any formal “backend tech stack usage”. However, when we implement cloud synchronization we expect the usages of

- Node.JS
- Firebase

5.2 Frontend

We are going to “keep things simple”, so we expect the usage of

- JavaScript
- HTML
- CSS
- Markdown renderer & Latex renderer library
- Modal library

5.3 DevOps

We expect the usage of

- GitHub Actions

5.4 Testing

We expect the usage of

- Jest (Unit Testing)
- Cypress (E2E)

6. Timeline

The timelines here are all subject to change based on our work progress.

Week 6
- Get pitch approved
- finish wireframing and system diagrams

Week 7
- Have basic HTML / JavaScript pages set up for all components
- Determine styling

Week 8
- Start implementing functionalities in JavaScript
- Finish styling

Week 9
- Unit and E2E Testing setup

Week 10
- Optimization / code cleanup