ProSeed

The Only Smart Calendar/Task Manager you’ll need!

Group 32

How our idea was formed

- How we choose this project
  - Must be Local first
  - Complex but not difficult
  - Our team members shown interests in productivity app

- Our goal:
  - User have more control
  - User have more customization
  - Easier to work in a team environment than other preexisting app
  - Clean & easy to understand UI

All Features

Steven Khaw

- CRUD
  - Create events/tasks, activities, collaborations, to-do lists
  - Read calendar, other people’s activities, plans, schedules
  - Update existing calendars, events, to-do lists
  - Delete no-longer-relevant events, cancelled plans
- Local First
  - A lightweight web client that can solely be used alone
    - Simple html file that creates and stores data, events, etc
  - Whole directory can easily be emailed to other user and used and updated
    - No wifi required, wifi only needed if you would like to share the files with someone else
- Extras
  - Ability to categorize events via urgency level
  - To-do list per day - easy planner!
  - Quick ‘n easy collaboration with others via sharing of files

Dentist

Add Location

all-day: □

starts: 05/02/2020 09:00

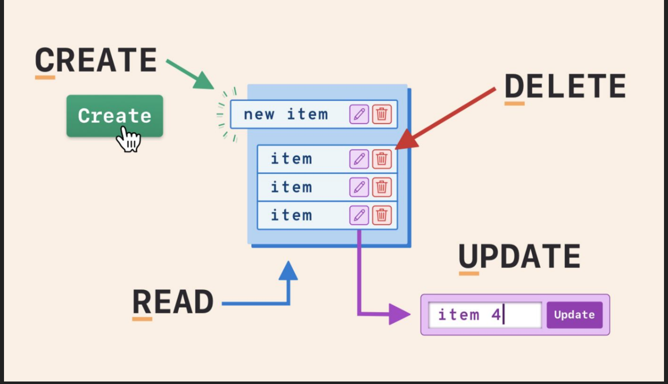
ends: 05/02/2020 10:00

repeat: None

travel time: None

alert: None

Add Notes, URL or Attachments



Features #1

Our basic functionality will include

- A calendar where users can add events, classes, etc. to organize their schedule
- Clicking on a day will show a more detailed schedule as well as a to-do list of tasks for that day
- All their data is stored locally in a database

Feature #2 - Collaboration/Shareability

- When users create tasks/events the calendar and todo list, it will automatically show up on each other
- Marking a task as complete marks the calendar event complete as well.
- Users can share their calendar with others if they are connected to wifi/data

Feature #3 - Extra Functionalities

Haoyi Wang

- Have urgency feature where you can categorize the importance of your tasks and manage to do them in a reliable way.
- Have To-do list feature where you can plan your tasks in a more intuitive way.
- Have collaboration feature where you can manage your group’s progress with your teammates.
- Have shop feature where you can use points to buy new animations/ customizations etc.
- Have Daily challenges feature for different area/fields of people. For gym people for example we can give a daily task that offers 100 points for whom can do 100 push up with videos

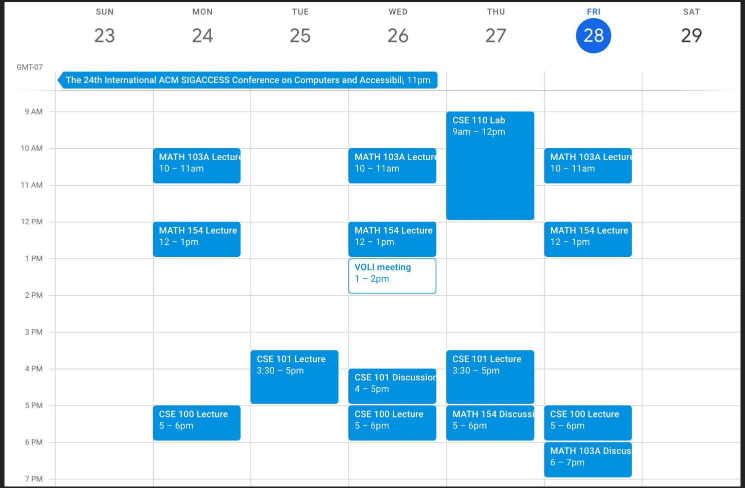
Use Cases

Use Cases #1 - Combines the best the both worlds

Christopher Han

Google Calendar:

- Easy to look at tasks/schedule for each day



SUN 23 · MON 24 · TUE 25 · WED 26 · THU 27 · FRI 28 · SAT 29

GMT-07

The 24th International ACM SIGACCESS Conference on Computers and Accessibil, 11pm

MON 24:
- MATH 103A Lecture 10 – 11am
- MATH 154 Lecture 12 – 1pm
- CSE 100 Lecture 5 – 6pm

TUE 25:
- CSE 101 Lecture 3:30 – 5pm

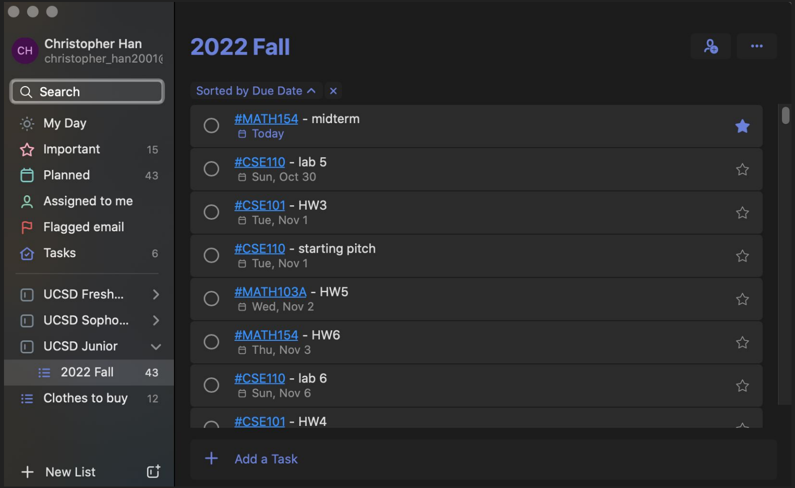
WED 26:
- MATH 103A Lecture 10 – 11am
- MATH 154 Lecture 12 – 1pm
- VOLI meeting 1 – 2pm
- CSE 101 Discussion 4 – 5pm
- CSE 100 Lecture 5 – 6pm

THU 27:
- CSE 110 Lab 9am – 12pm
- CSE 101 Lecture 3:30 – 5pm
- MATH 154 Discussi 5 – 6pm

FRI 28:
- MATH 103A Lecture 10 – 11am
- MATH 154 Lecture 12 – 1pm
- CSE 100 Lecture 5 – 6pm
- MATH 103A Discus 6 – 7pm

Microsoft To Do

- Compiles all the tasks together into one list



2022 Fall

Sorted by Due Date ^ ×

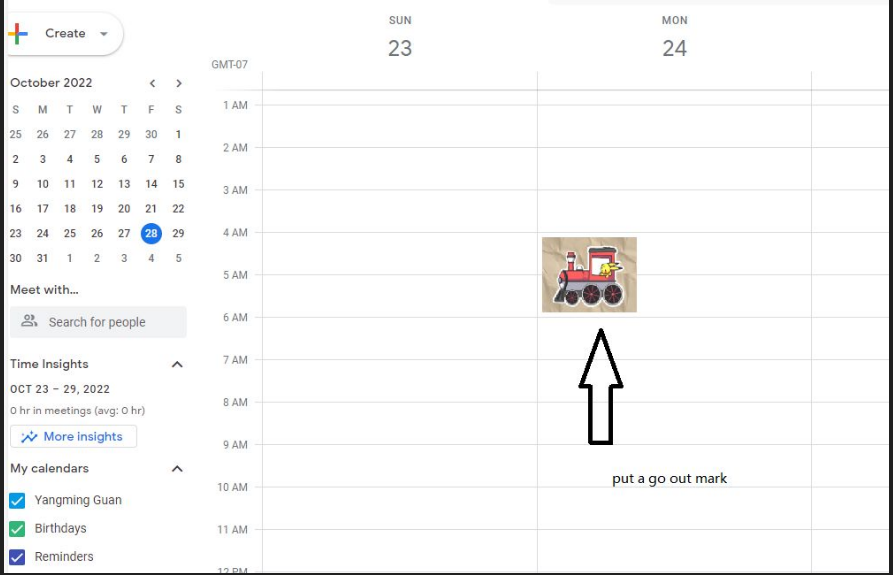
- ○ #MATH154 - midterm — Today ★
- ○ #CSE110 - lab 5 — Sun, Oct 30 ☆
- ○ #CSE101 - HW3 — Tue, Nov 1 ☆
- ○ #CSE110 - starting pitch — Tue, Nov 1 ☆
- ○ #MATH103A - HW5 — Wed, Nov 2 ☆
- ○ #MATH154 - HW6 — Thu, Nov 3 ☆
- ○ #CSE110 - lab 6 — Sun, Nov 6 ☆
- ○ #CSE101 - HW4

+ Add a Task

Use Cases #2 Where to go mark.

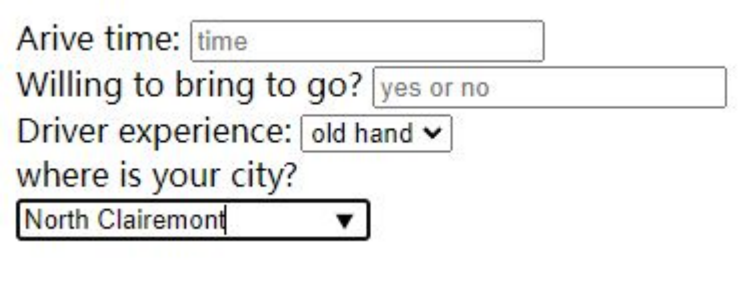
Yangming Guan

When you plan to go to school or have a traveling, you team member can take advantage of it to get a free ride or just help them take a to go selflessly.



put a go out mark

Click it to open a small dialog, let other people know the details.



Arive time: time

Willing to bring to go? yes or no

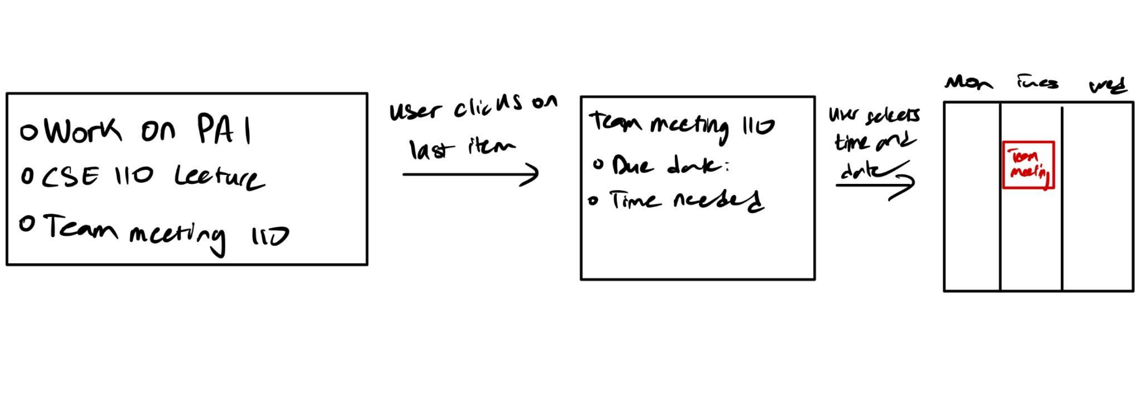
Driver experience: old hand ▾

where is your city?

North Clairemont ▼

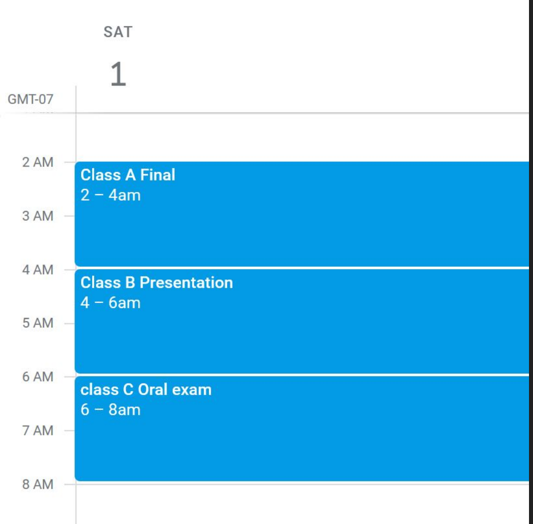
Use Cases #3

- Visually keep track of progress between to-do list and events on the calendar by having to-do list update live with tasks on calendar
- User can view pressing due dates at the top of their to-do list, and can allot time in calendar to work on task



Target Audience

Yuelin Dai



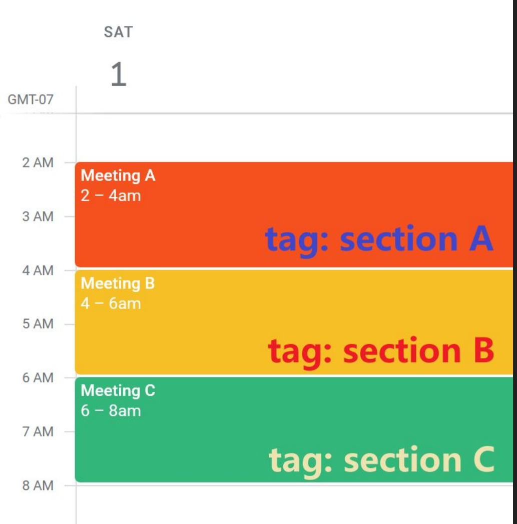
Class A Final
2 – 4am

Class B Presentation
4 – 6am

class C Oral exam
6 – 8am

Students

plan ahead for upcoming classes and meetings



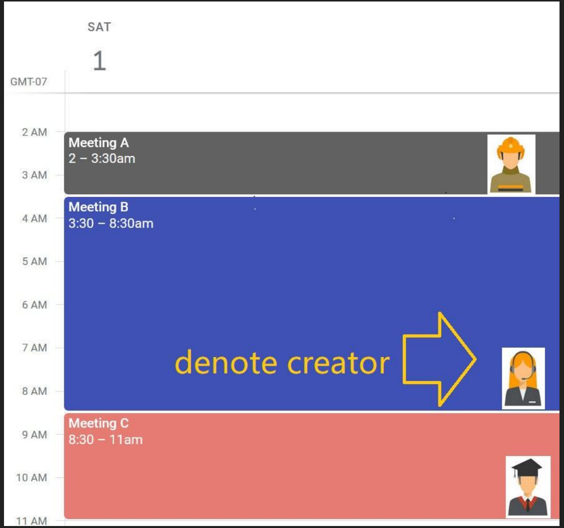
Meeting A
2 – 4am
tag: section A

Meeting B
4 – 6am
tag: section B

Meeting C
6 – 8am
tag: section C

Teachers

use tags to manage time for different classes and student bodies



Meeting A
2 – 3:30am

Meeting B
3:30 – 8:30am

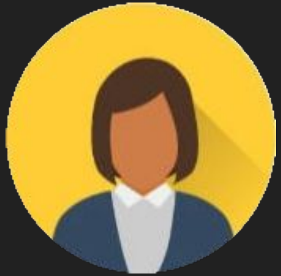
denote creator

Meeting C
8:30 – 11am

Team Developers

ease of sharing allows whole team share plans on same calendar

Meet Our Users (Personas)



Claire J.

Claire is a high school teacher. She wants to share lecture topics with her students at the start of every week. She also wants to send reminders to her class before assignments deadline and exams.

Ben T.

Ben is a third-year college student. Everyday he needs to plan his schedule according to upcoming events and classes. He also wants to have a brief compact overview of tasks that are close to deadline.

Adam G.

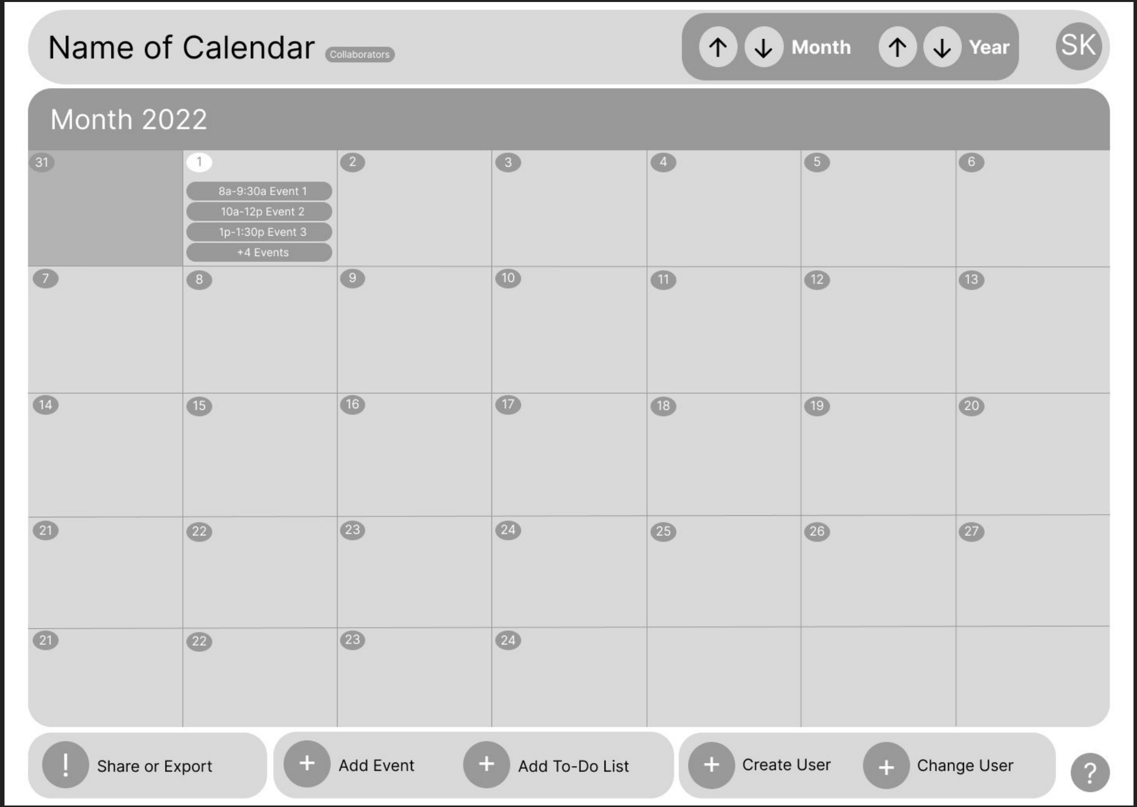
Adam is the lead of a developer team. He needs to make sure his team members share a team schedule that's viable for everyone. He also wants to get notified as soon as any scheduled team tasks are finished.

User Stories

- As a student, I want all kinds of things in different aspect show in different priorities, so that I can organize my schedule.
- As a teacher, I want to easily manipulate students' calendars at the same time, so that I can remind them not to forget to do their homework and hand in on time.
- As a team developer, I want the team's progress on different aspect to be displayed at the same time, so that I can control the team's work and decide which task to do next.

Wireframe: Full View

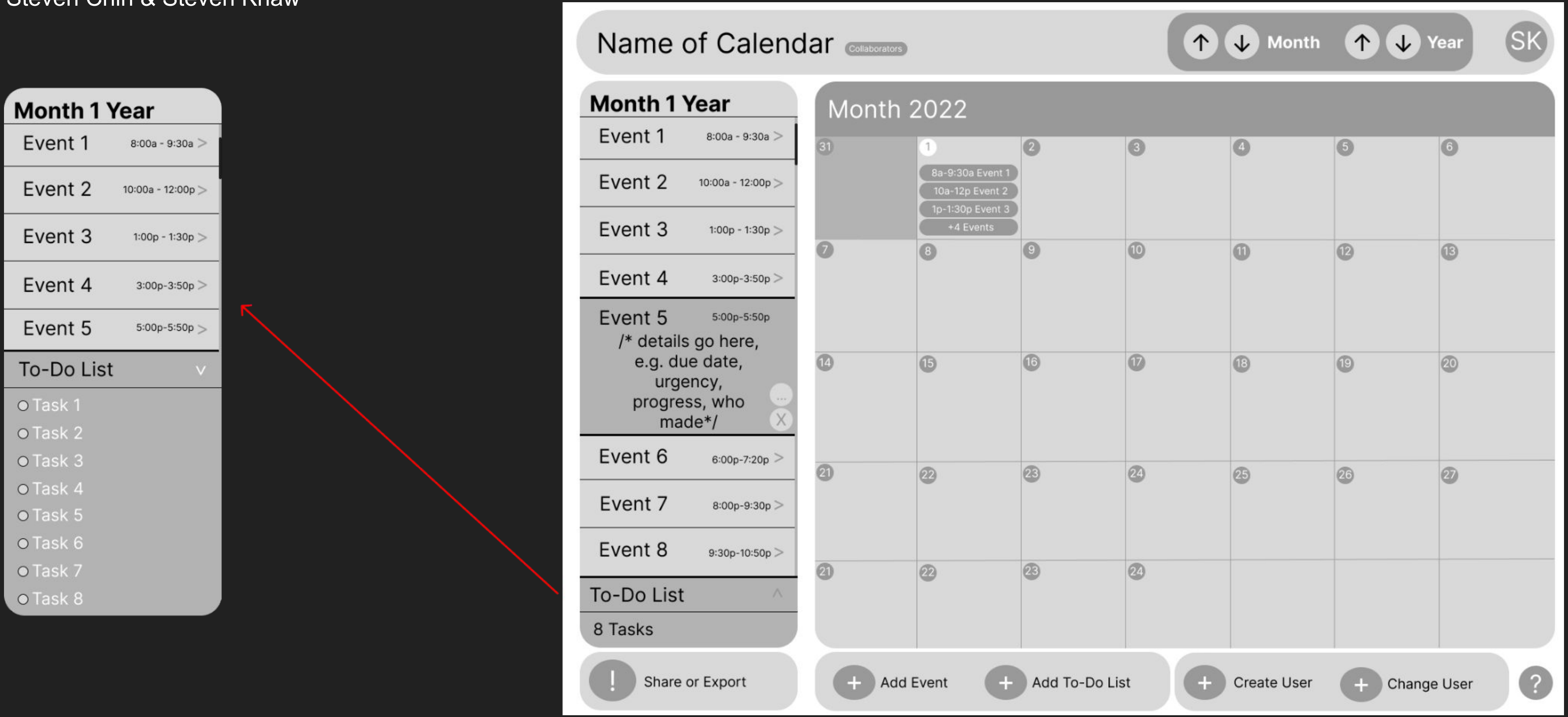
Steven Chin & Steven Khaw



31	1 8a-9:30a Event 1 10a-12p Event 2 1p-1:30p Event 3 +4 Events	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
21	22	23	24			

Wireframe: Day View

Steven Chin & Steven Khaw



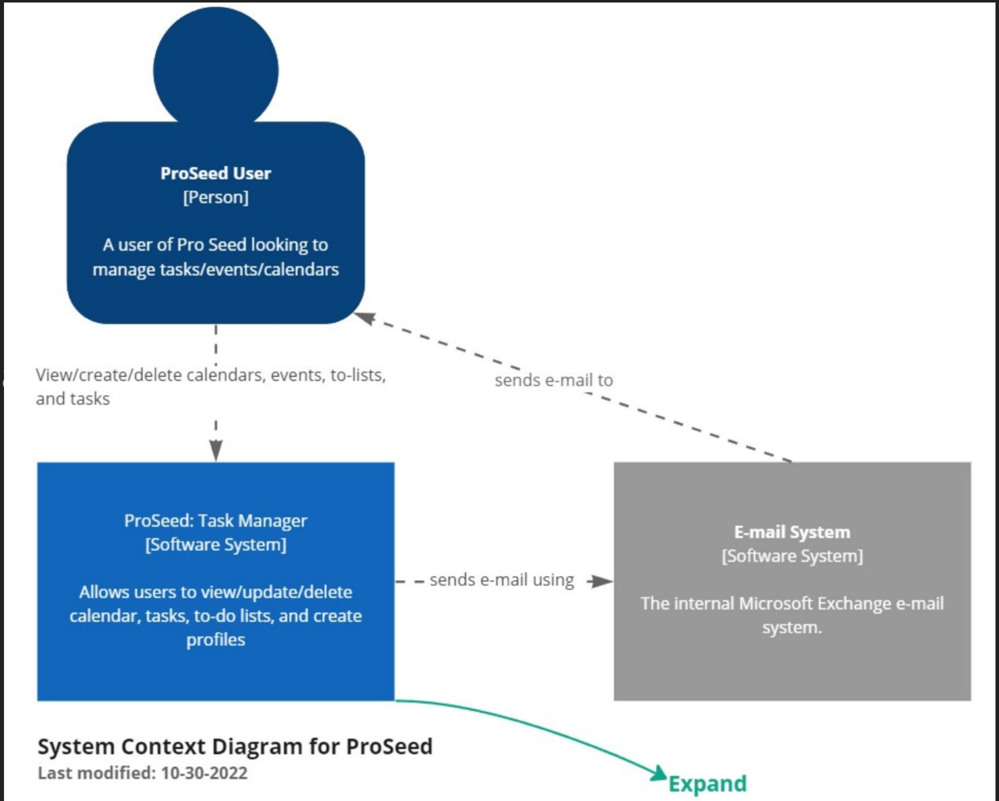
System Diagram (System Context)

User:

- CRUD tasks onto ProSeed

System:

- Allows users to CRUD, sends em back to user



ProSeed User
[Person]

A user of Pro Seed looking to manage tasks/events/calendars

View/create/delete calendars, events, to-lists, and tasks

sends e-mail to

ProSeed: Task Manager
[Software System]

Allows users to view/update/delete calendar, tasks, to-do lists, and create profiles

sends e-mail using

E-mail System
[Software System]

The internal Microsoft Exchange e-mail system.

System Context Diagram for ProSeed
Last modified: 10-30-2022

System Design (Container Diagram)

Web app:

- Frontend: static container of html/css
- UI user will interact with

Single-page app:

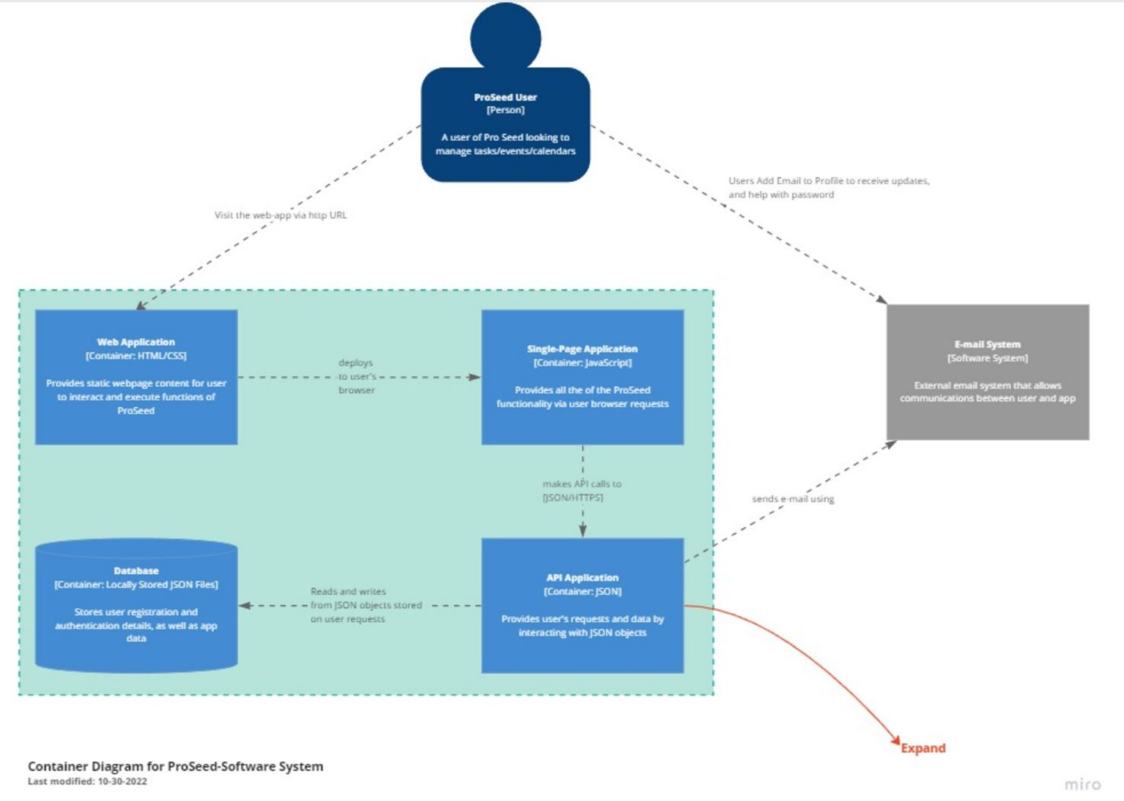
- Contains the Javascript that allows user interaction with static elements

API app:

- Backend of JSON and HTTP requests that interacts with objects stored in database to read in data
- Processes requests and data

Database:

- Stores data locally on user's machine



ProSeed User
[Person]

A user of Pro Seed looking to manage tasks/events/calendars

Visit the web-app via http URL

Users Add Email to Profile to receive updates, and help with password

Web Application
[Container: HTML/CSS]

Provides static webpage content for user to interact and execute functions of ProSeed

deploys to user's browser

Single-Page Application
[Container: JavaScript]

Provides all the of the ProSeed functionality via user browser requests

E-mail System
[Software System]

External email system that allows communications between user and app

makes API calls to [JSON/HTTPS]

sends e-mail using

Database
[Container: Locally Stored JSON Files]

Stores user registration and authentication details, as well as app data

Reads and writes from JSON objects stored on user requests

API Application
[Container: JSON]

Provides user's requests and data by interacting with JSON objects

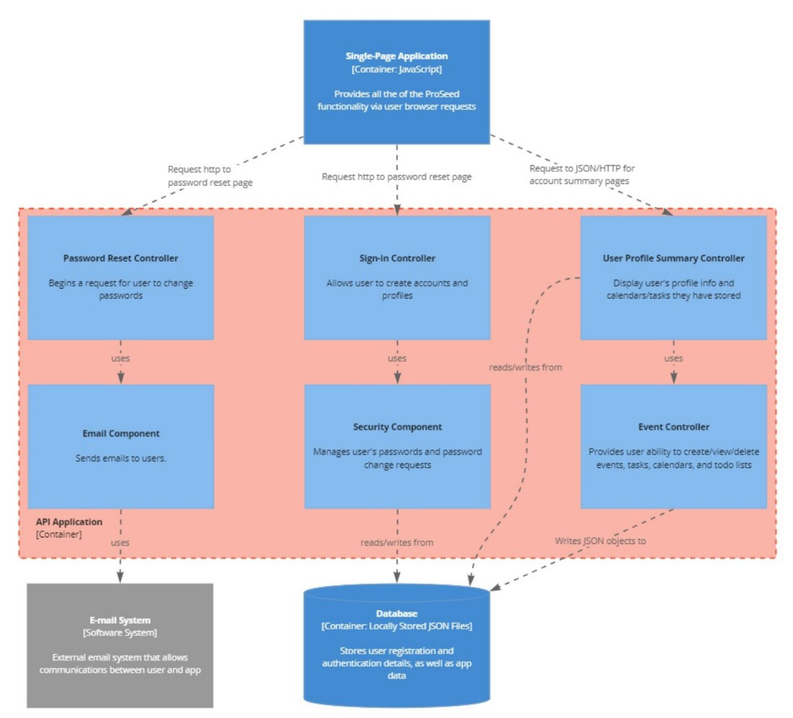
Container Diagram for ProSeed-Software System

Last modified: 10-30-2022

System Design (Component Diagram)

The single-page app has 3 components:

- Password reset
  - Uses email system to send users methods to reset their password
- Sign in
  - Accesses and stores login information into the database
- User profile
  - Accesses user profile and login information from the database
  - Stores users data (i.e. tasks, profile, calendars) into database


Single-Page Application
[Container: JavaScript]
Provides all the of the ProSeed functionality via user browser requests
Request http to password reset page
Request http to password reset page
Request to JSON/HTTP for account summary pages
Password Reset Controller
Begins a request for user to change passwords
Sign-in Controller
Allows user to create accounts and profiles
User Profile Summary Controller
Display user's profile info and calendars/tasks they have stored
uses
uses
reads/writes from
uses
Email Component
Sends emails to users.
Security Component
Manages user's passwords and password change requests
Event Controller
Provides user ability to create/view/delete events, tasks, calendars, and todo lists
API Application
[Container]
uses
reads/writes from
Writes JSON objects to
E-mail System
[Software System]
External email system that allows communications between user and app
Database
[Container: Locally Stored JSON Files]
Stores user registration and authentication details, as well as app data

Possible Risks

- Data safety
  - Even though the project will be local first and that the data would be usually stored locally. If the hacker is capable of hacking into the phone. There will be some danger of data safety.
  - We can fix this issue if we are able to perform some sort of data protection

- Technology use
  - We will try our best to make the calendar project better but depending on the time we spend, we might not use too complex technology to have a small improvement in access speed or system safety.

- Too many features
  - There could be dangers where our features would interfere with each other.
  - For example, There could be issues / delays with trying to implement the to-do list and calendar at the same time when they are referencing through the same database. Which could causes more issues.

Possible Rabbit Holes

- Shop feature:
  - If we correctly implementing our shop feature. We will have to implement more features on cosmetics / rewards for our virtual currency.
  - Could be very messy to distribute part of the team to focus on a code non-related topics and could potentially delay the development process.

- Graph:
  - We will limit our resource usage to avoid making the perfect user interface that case too much time to maintain it when we want to change or develop another sub-interface. color and style might not be fully coordinated.

- Data storage
  - The data storage could be a mess since we are trying to merge two system of productivity together as well as implementing the ability to share your work with other people.
  - We might have to come up with a more clever way to manage the data or else we might spend a lot of time to figure out how it would work.

- Bugs:
  - If a bug causes little damage to our project, but in order to fix it, we need to modify the whole project, we might just leave it there and make a mark or notice for the user to help them bypass this bug.

Project Roadmap Overview

- First couple of week do research to select specific technologies that work best with a productivity app
- Plan out wireframe in detail + layout and
- Build the calendar portion with todo list functionality linked with it (MVP)
- If time permits, add features including:
  - Log to keep track of how much spend on specific activities
  - Reward users with in-game currency based on how many tasks they’ve completed