

# Voltron // p01 // “Planner Project”

Connor Oh, Jun Tao Lei, and Leia Park / pd9

**Project Manager:** Connor Oh

## **Necessary:**

Flask will serve as your web microframework.

Multiple supporting Python3 files will be used as necessary.

Bootstrap or Foundation will be used as your front-end framework.

You will provide your own customized CSS where appropriate/necessary.

You will make *meaningful* use (sum > parts) of at least three (3) REST APIs, chosen from Ye Olde SoftDev API KB.

## **Required APIs:**

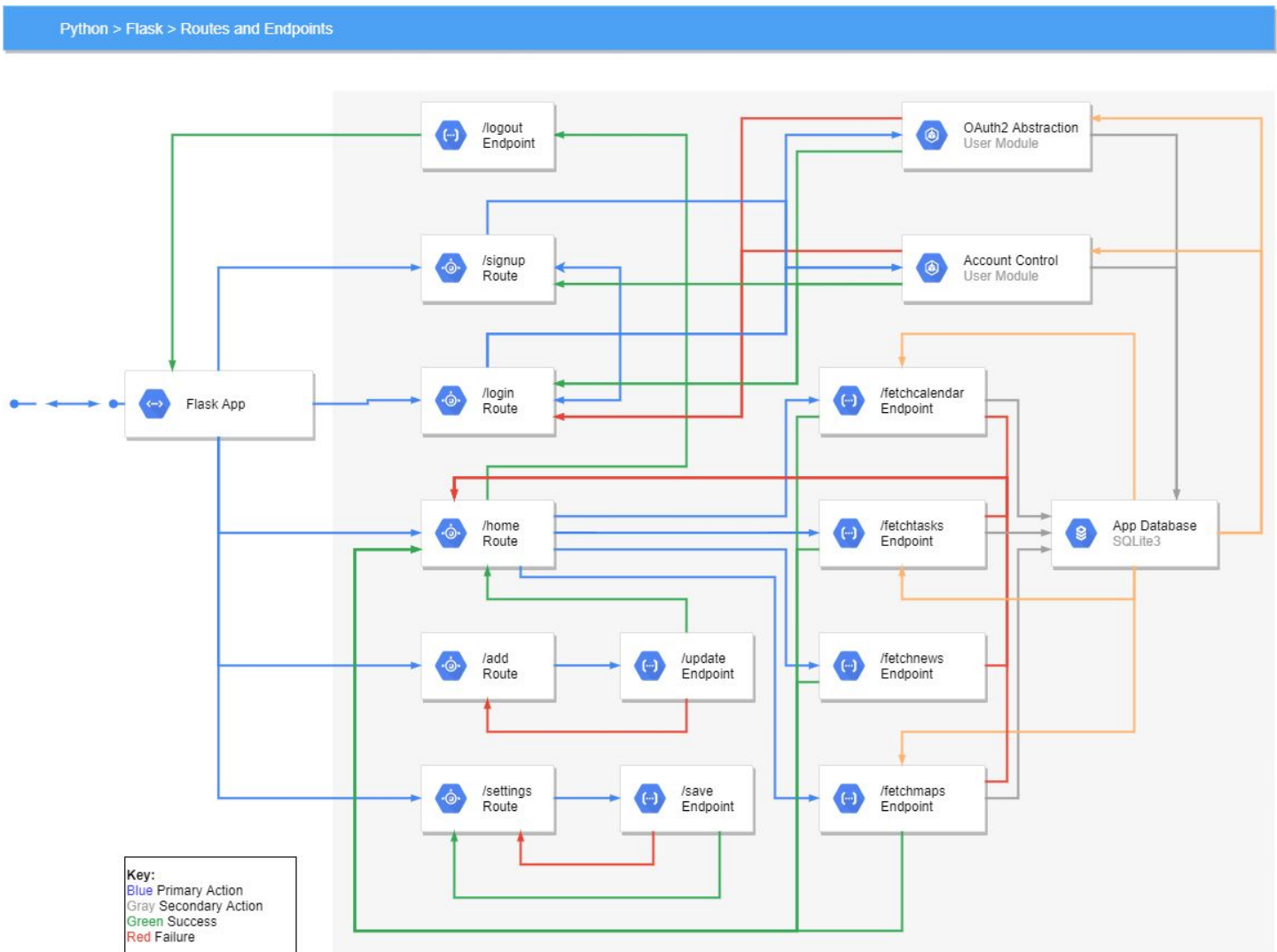
- Google Calendar API
  - See <https://developers.google.com/calendar/v3/reference>
- Google Tasks API
  - See <https://developers.google.com/tasks/v1/reference>
- Google Maps Embedded API
  - See <https://developers.google.com/maps/documentation/embed/guide>
- Reddit API
  - We are only using a very specific endpoint
  - Example:
    - <https://www.reddit.com/r/worldnews/top.json?limit=5>
    - Adding .json to the end of a url returns a subreddit's post(s) in JSON format
    - Adding ?limit=n returns n posts from the subreddit

## **About/Features:**

- PERSONAL PLANNER WEBSITE
  - Calendar Events (Google Calendar API)
  - Tasks (Google Tasks API)
  - Displays the Current Traffic Information for Commute (Maps Embedded API)
  - News Feed (Reddit API)
- REGISTRATION
  - User should be able to provide a unique username, email, and password to create an account on the site
  - User should be able to login using his/her Google account in place of registering an account
- AFTER LOGIN
  - User should be able to access their account, provided they have already created one, using the same unique username and password
  - Edit/delete/add reminders
  - Personalize their “to do” list of tasks & scale them according to importance

- Display the tasks in the positions returned by the Task API
  - This is usually the importance of the task.
- View through news feed
- Set their home address and other locations that they commute to
- NOT LOGGED IN
  - Users can do nothing but stare at login/sign-up screen

## Components Map

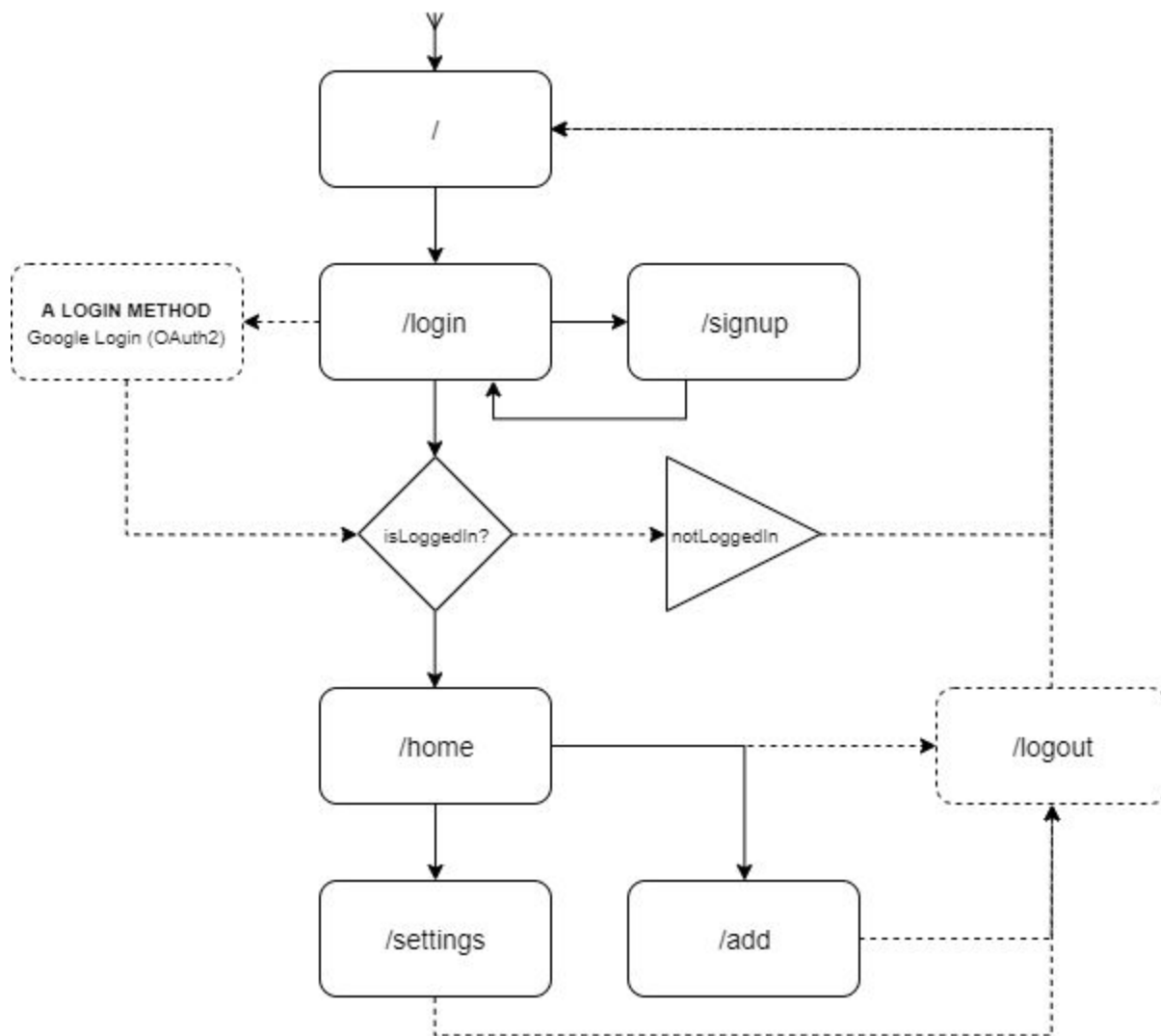


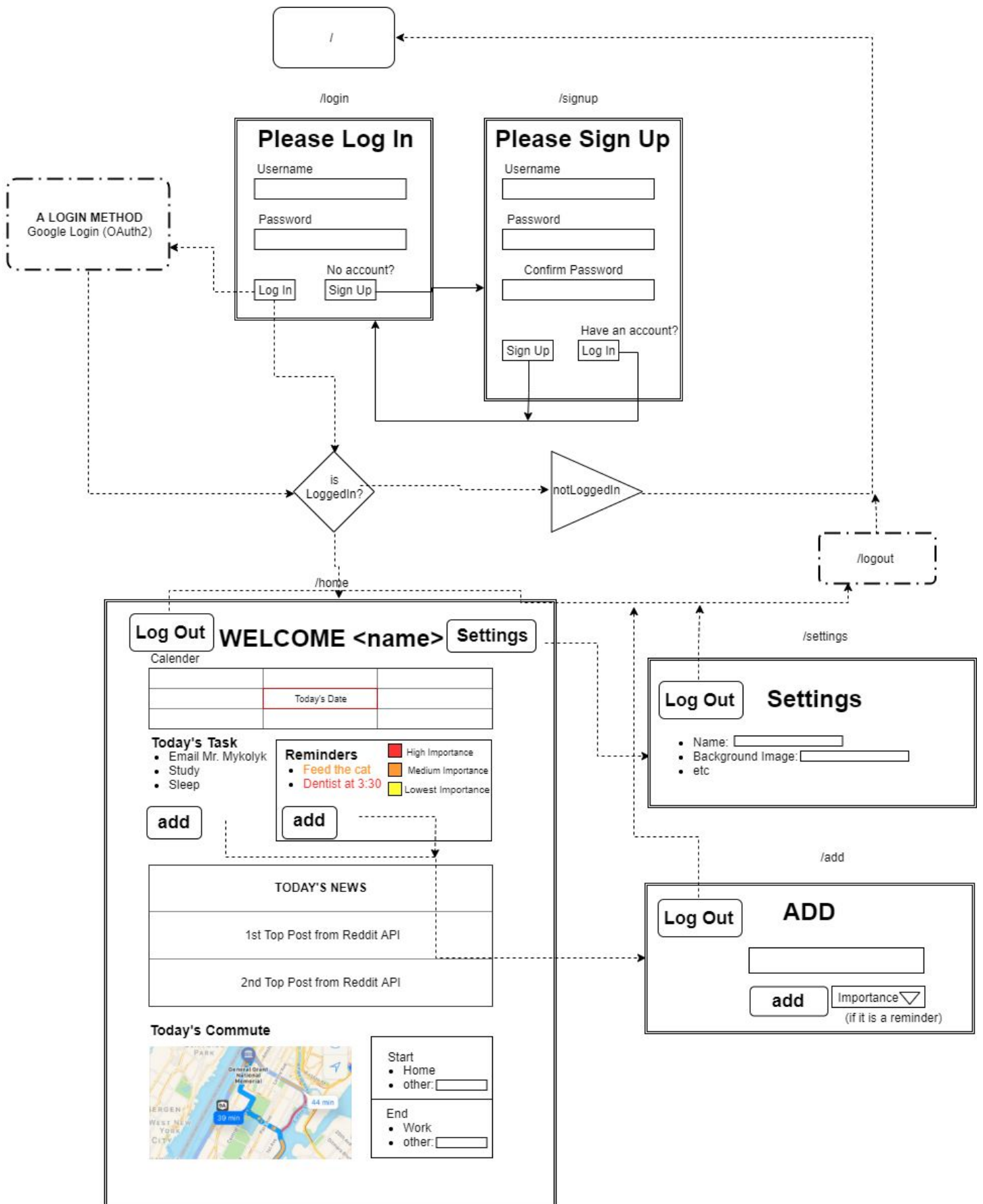
## Website Pages (foundational bootstrap)

- LOGIN page (<https://getbootstrap.com/docs/4.3/examples/floating-labels/>)
  - Enter username and password
    - IF new user logs in THEN “this user does not exist”
    - IF existent user logs in THEN routes to own blog
    - Routes to homepage
  - Register Button -> routes to register page
- REGISTER page (<https://getbootstrap.com/docs/4.3/examples/floating-labels/>)
  - Enter username, email, and matching passwords

- IF new user registers THEN routes to own blog
- IF existent user registers THEN “this username is taken”
- IF email is already used THEN “this email is used already”
- IF passwords do not match THEN “passwords do not match”
- Routes to login page
- HOME page (<https://getbootstrap.com/docs/4.3/examples/blog/>)
  - Buttons
    - Logout button -> routes to login page
    - Settings button -> routes to settings page
  - Shows list of reminders & “to do” tasks
  - Shows a news feed
  - Shows current traffic for commute

## Site Map





## Database Layout Diagram

users
PK uniqueId INT PRIMARY KEY AUTOINCREMENT
hash TEXT
token TEXT
name TEXT NOT NULL
email TEXT UNIQUE NOT NULL
homeAddress TEXT
workAddress TEXT
newsPreference TEXT

## Tasks & Roles

- Front-end Appearance and UI (*Leia Park & Connor Oh*)
  - Bootstrap
    - Login, Register, Home pages
- Back-end
  - Google (*Jun Tao Lei*)
    - OAuth2
    - Google API
      - Google Calendar API
      - Google Tasks API
        - Filter & personalization system
      - Google Maps Embedded API
  - Reddit API (*Connor Oh*)
    - No key necessary for pulling posts from subreddits
  - Users database (*Leia Park*)
    - Creation of database
    - Input to database from Login page
    - Accessing information from database (in order to login and register)