Design Document for *asap*

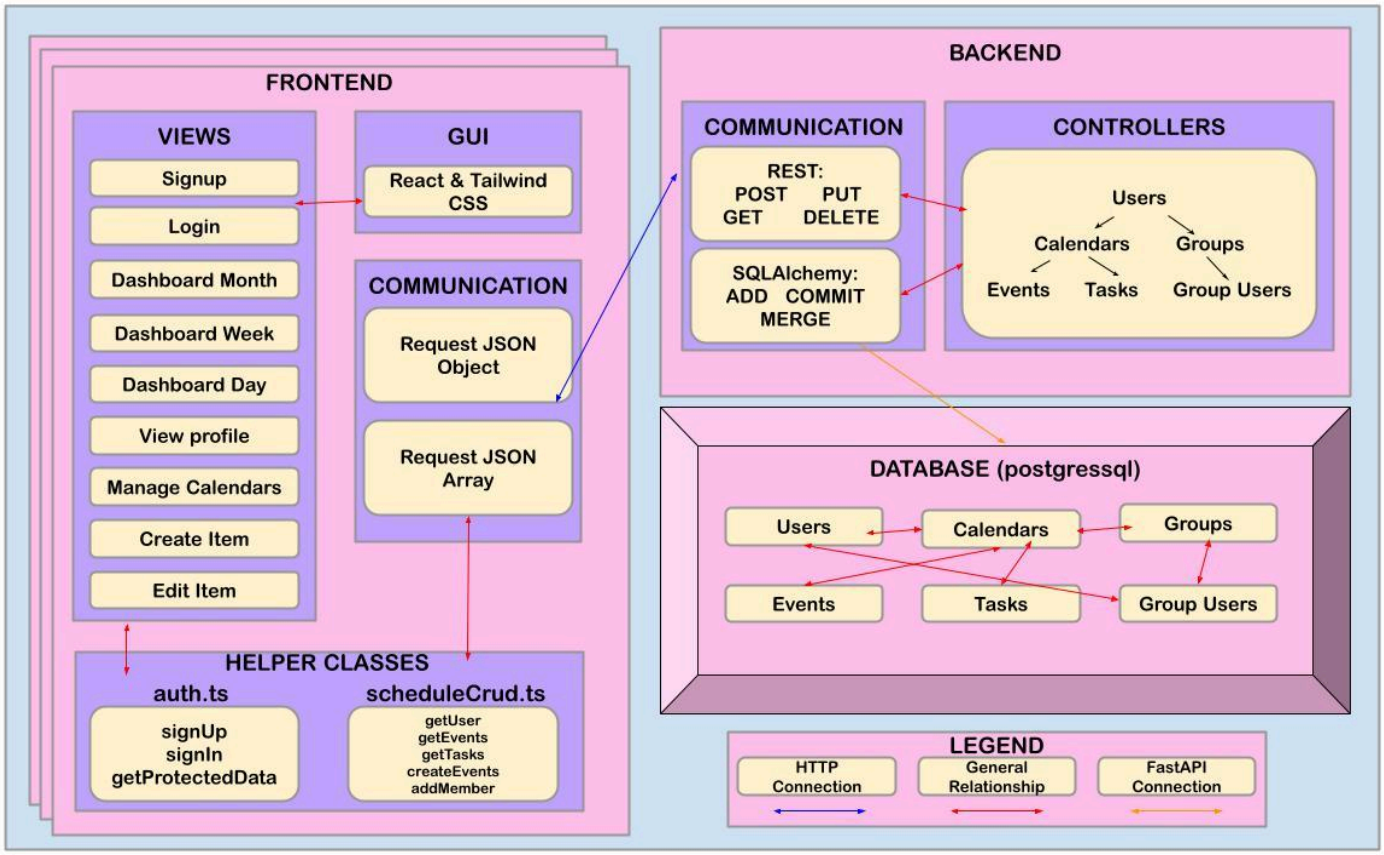
Group 2_mstafa_8

Member 1: Caiti Harts 25% contribution

Member 2: Kaylee Maczek 25% contribution

Member 3: Alex Young 25% contribution

Member 4: Divya Lidder 25% contribution


FRONTEND
VIEWS
Signup
Login
Dashboard Month
Dashboard Week
Dashboard Day
View profile
Manage Calendars
Create Item
Edit Item
GUI
React & Tailwind CSS
COMMUNICATION
Request JSON Object
Request JSON Array
HELPER CLASSES
auth.ts
signUp
signIn
getProtectedData
scheduleCrud.ts
getUser
getEvents
getTasks
createEvents
addMember
BACKEND
COMMUNICATION
REST:
POST PUT
GET DELETE
SQLAlchemy:
ADD COMMIT
MERGE
CONTROLLERS
Users
Calendars
Groups
Events
Tasks
Group Users
DATABASE (postgressql)
Users
Calendars
Groups
Events
Tasks
Group Users
LEGEND
HTTP Connection
General Relationship
FastAPI Connection

Frontend (currently implemented)

SignUp(User)

- Sign up generates a page with the following elements:
    - Form Input: First and Last Name
    - Form Input: Password
    - Form Input: Email
    - Button: SignUp
    - Upon clicking the button ‘SignUp’ the values of the First and Last Name, Password, and Email are sent as a POST request to the server.

Login (Admin, User)

- Login generates a page with the following elements:
    - Form Input: Email
    - Form Input: Password
    - Button: Login
    - Login Screen takes user input of their email and password, and if that email and password is valid, it will take the user to the dashboard, where they are logged in as that user. An authentication token is provided for the user to allow them access to features under their account.

Create / Edit Item

- Create and edit item generates two separate pages, both including the following elements:
    - Form Input: Title
    - Form Input: Start and End Date and time
    - Form Input: Description
    - Select: Calendar
    - Button: Create item / save item
    - These two pages take the user input of the item information, and send a post request or a put request to the backend to create or edit the item.

Backend

Communication

The backend uses mappings to update the database based on information sent to the given

mappings' URLs. These include:

- Post: send information on an item to be added to the database.
- Get: request information, often with an identifier for the specific item requested from the database
- Put: send information to update a specific item in the database
- Delete: send an identifier to delete a specific item from the database

Controllers

The controllers contain the mappings for communication between frontend and the database. These include:

- User: Contains the above mappings to create users, which contain one-to-many relationships with the below calendars, groups, and events, and group type users.
- Calendars: This allows users to have multiple calendars with different events, and has a one-to-many relationship with events. Can be a group or not, can have group users or just the single admin user.
- Groups: Groups multiple users together to share a calendar. Has a one-to-many relationship with the groups groupUsers, which have different permissions.
- Events: Actual calendar events, bound to a specific calendar to be toggled only when that calendar is selected.
- GroupUsers: The users in a group. Contains the user information as well as permissions information.

Made using Postgres because that is what we are using (the table names are by the blue table icon):