1 Vision Statement

The goal of this project is to create a personalized scheduler web app for groups of physicians. There are several different ways that physicians can be employed. One such way involves multiple physicians of one specialty coming together to form a group which contracts its services out to hospitals/surgery centers. This method of organization often involves different physicians filling different positions at different hospitals on different days. Scheduling which doctors do what position at which hospital is tricky and time consuming. Current scheduling techniques involve inputting every doctor into a spreadsheet by hand or using general scheduling programs that are difficult to customize for different individual groups. The first method is time consuming for busy physicians and the second method is not optimized to the specific needs of the group.

Moon Jelly scheduler aims to be a solution to this issue by being an easy to use, time efficient way to schedule physicians optimally. Groups will be able to customize the number of doctors they have working each day and the number of positions that need to be filled at different facilities. An algorithm will take into account the stressfulness of each position and create an optimized schedule that spaces out stressful work positions while giving every member of the group roughly the same amount of every position. This results in a better working environment for the group while ensuring equal participation. The algorithm will also account for the needs of specific doctors. For instance, the schedule will consider if certain doctors do not work at cer-
tain hospitals or if certain doctors specialize in certain procedures. These cases can change on a doctor by doctor basis. Through this app doctors will be able to request for certain days to be shorter work days or days off. We will also prioritize clean and intuitive user interface that can be easily used by doctors that are not so tech savvy.

This project will result in a web app that can be used by real physicians. A group of physicians based in Santa Cruz CA has already agreed to test the project. We will work with them to create a product that fulfills their needs in a real world environment. This will result in their group of physicians getting a customized scheduling system while our project group learns about development cycles in a real customer/developer relationship.

We plan to implement this scheduler through a web app hosted by Heroku. We will utilize the built in database that comes with Heroku web apps to store physician data safely. We will built the app primarily with python-flask and access the database with sqlalchemy. Our first goal is to learn the basics of web development and set up a basic app with the ability to access a database (Sprint 1). From there we will move on to creating a basic scheduling algorithm and creating an intuitive user interface (Sprint 2). After this we will optimize the scheduling algorithm and perfect the account system allowing physician group members to create requests (Sprint 3).
2 System Architecture

(a) High Level Diagram

(b) UML Diagram

(a) db and Administrator

(b) db and Doctor

Figure 2: db Interfaces (Continued)
Figure 3: db Interfaces

2.1 User Interaction and Interface Mockups

Figure 4: User Interaction Diagram

See next page for UI mockups.
3 Requirements (Functional and Non-Functional)

3.1 User Stories (With Trello Links)

User Story 1: As a team member, I will be able to use and contribute to making a web app through heroku.

User Story 2: As a developer, I can interact with the data base, (add/ delete, etc).

User Story 3: As a registered user, I am able to log in with my information.

User Story 4: As an administrator, I should be able to add and delete new members.

User Story 5: As an administrator, after I have input the necessary data, I can log in and check a formatted schedule and check added members.

User Story 6: A user can get a schedule that fills variable number of spots every day.
**User Story 7**: A user can get a schedule that includes special cases like a certain specialist always in the top 3 spots, certain people not working in certain spots and the more days between the same shift the better.

**User Story 8**: As a user, the user interface should be intuitive and compelling to me as well as easy to understand.

## 4 Appendix

![Technologies Used](image)

Figure 8: Technologies Used