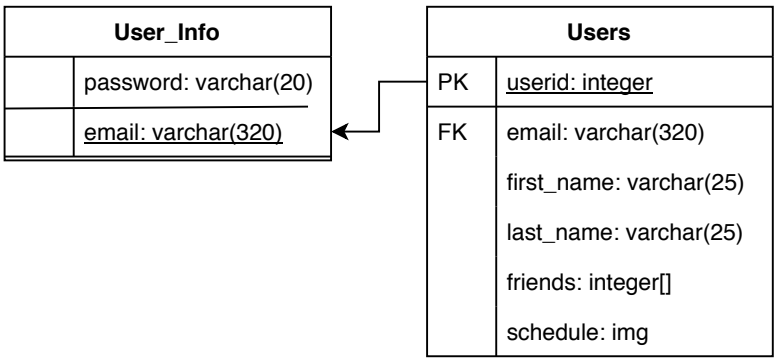
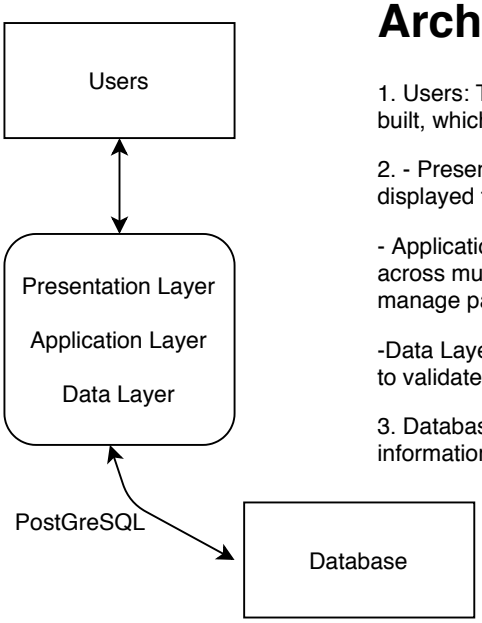


Database Design

We are using PostGreSQL to connect and manage our database. We have a user_auth database which tracks all the credentials that users can use to log in, and then we also have a user_info table which has all of the user's information which includes all of the fields that are included in the diagram.



Architecture Diagram

1. Users: The users will interact with the webpage that we have built, which is the Presentation Layer

2. - Presentation Layer: This includes the webpage that is displayed to the user.

- Application Layer: This includes the logic to direct the user across multiple webpages, display accurate information and manage pages that need to access the database or backend

-Data Layer: This part handles the backend queries, for example to validate users and passwords.

3. Database: We are accessing the backend to store user information

We are planning on using Google's calendar API for our project to simplify displaying our users' calendars on our webpage. We are currently focused on other aspects of our project and have not looked into the API in depth yet, but as of now, we are planning on using the interface part of the API to avoid having to make our own calendar view from scratch.

Wireframes for Group Project
Group 102-2:

Wireframe 1: This shows a login feature and a sign up feature with all associated modals, which will allow users to either sign into an existing account, or create a new one. The login modal will verify that the username entered exists, and that the password matches the one in the user database. The sign up modal will verify that no user currently has the selected username, and if it is it will prompt the user to select a different username. Additionally, the modal will enforce certain security guidelines on the password/confirm password fields: they must be exactly the same, at least a certain number of characters, etc.

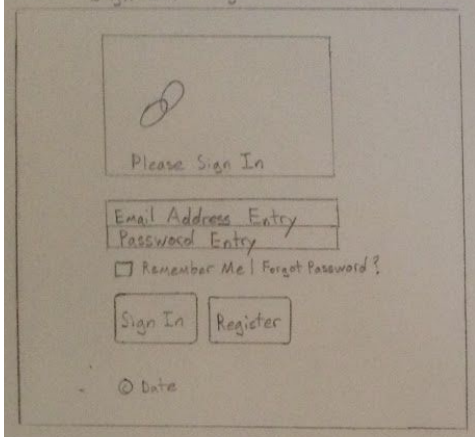
Wireframe 2: This shows the home page and all associated forms/modals, which the user will see upon logging into their account. The buttons here represent three functions. The user may upload their schedule, prompting a modal to open. This modal allows the user to upload a pdf document with their schedule, or an image of their schedule, and each event in the document will be added to their schedule. The user may also choose to manually enter a schedule event, which will open a modal to enter the events start time/day and end time/day. Lastly, the user will be given the option to add a friend, which opens a modal to search current users by first name, last name, or username. If the user is found, a friend request will be sent for confirmation, otherwise the user will be prompted to double-check their entry.

Page 3: This page contains the Friends page and the Schedule page. The friends page populates a list with the first name, last name, and a link that will navigate to that friend's schedule page (with the buttons disabled). The add friend option will open the same modal, with the same functionality as the corresponding function from the home page. The remove friend button will open a modal similar to the add friend modal, however this will search the friend array, rather than the active users in the database. If the friend is found, they are removed from the users friend array, otherwise the user is prompted to double-check their entry.

The Schedule page will open a view of the logged in users schedule, with an option to toggle between a calendar view and a week-to-week view. The add event button will open the same modal with the same functionality as the option to manually add schedule events from the home page.

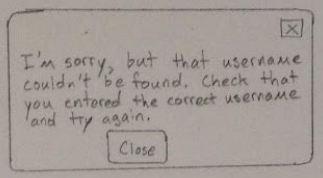
Wireframes

Sign-In Page

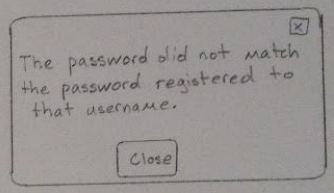


Associated Modals

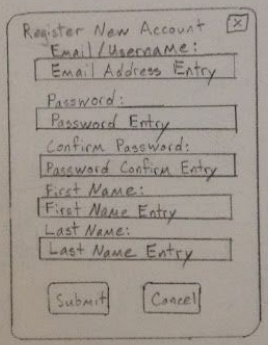
Invalid User



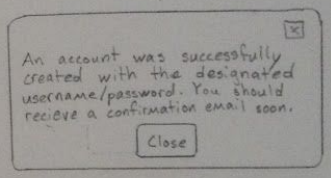
Invalid Password



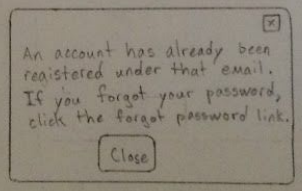
Registration

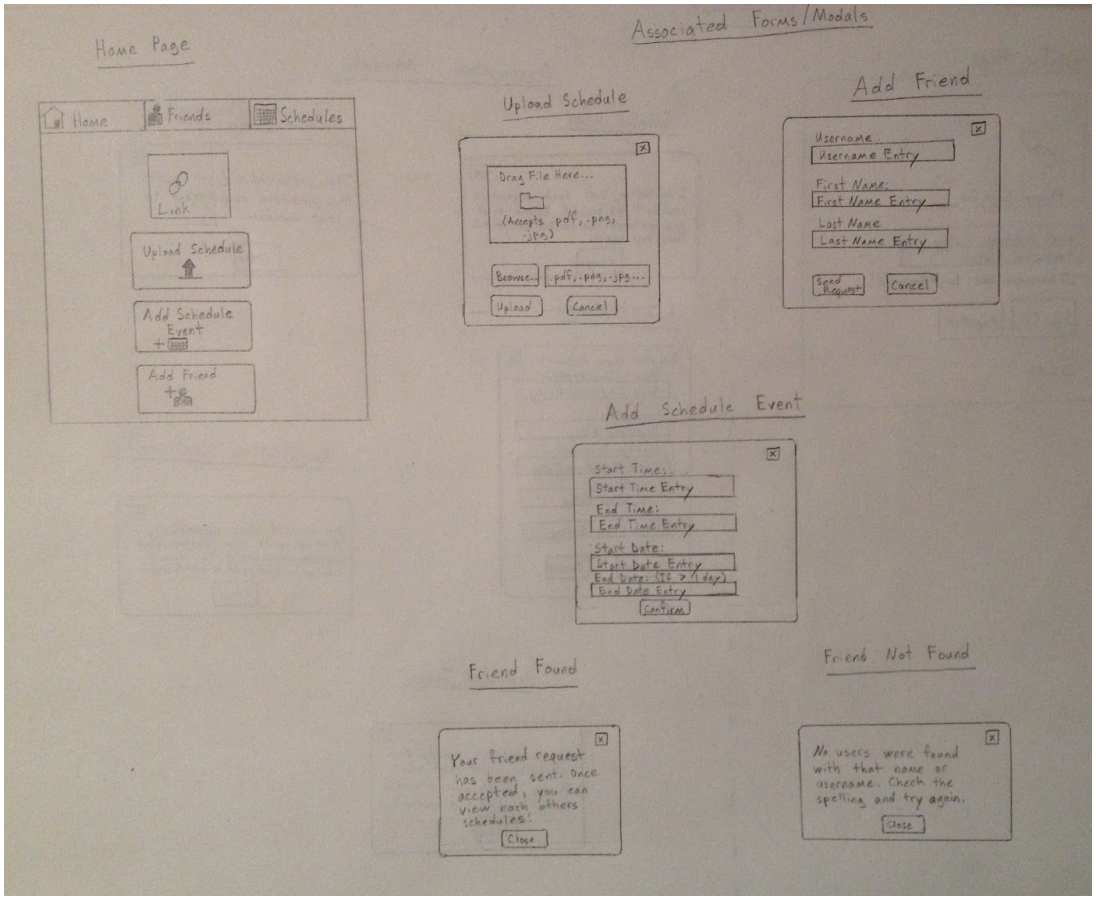


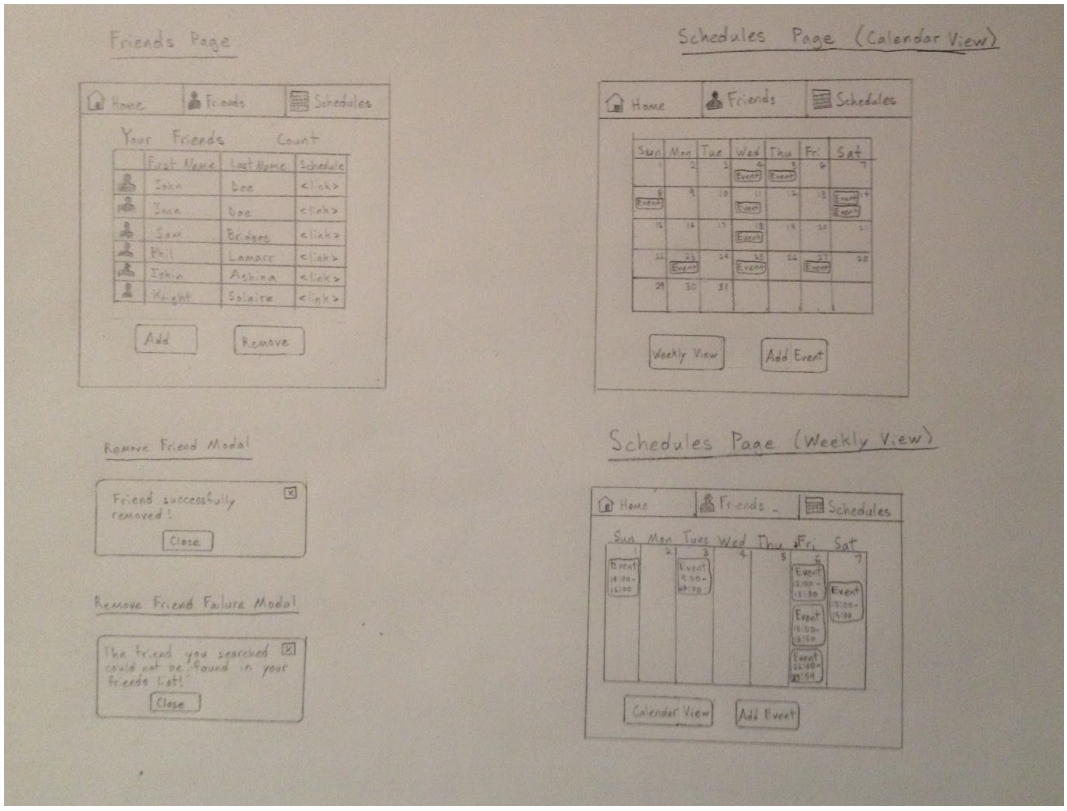
Registration Confirmation



Username Taken




Home Page
Home
Friends
Schedules
Link
Upload Schedule
Add Schedule Event
Add Friend
Associated Forms/Modals
Upload Schedule
Drag File Here...
(Accepts .pdf, .png, .jpg)
Browse...
.pdf, .png, .jpg...
Upload
Cancel
Add Friend
Username:
Username Entry
First Name:
First Name Entry
Last Name
Last Name Entry
Send Request
Cancel
Add Schedule Event
Start Time:
Start Time Entry
End Time:
End Time Entry
Start Date:
Start Date Entry
End Date: (If > 1 day)
End Date Entry
Confirm
Friend Found
Your friend request has been sent. Once accepted, you can view each others schedules!
Close
Friend Not Found
No users were found with that name or username. Check the spelling and try again.
Close


Friends Page
Home
Friends
Schedules
Your Friends Count
First Name
Last Name
Schedule
John
Doe
<link>
Jane
Doe
<link>
Sam
Bridges
<link>
Phil
Lamarr
<link>
John
Ashma
<link>
Knight
Solaire
<link>
Add
Remove
Schedules Page (Calendar View)
Home
Friends
Schedules
Sun Mon Tue Wed Thu Fri Sat
Weekly View
Add Event
Remove Friend Modal
Friend successfully removed!
Close
Remove Friend Failure Modal
The friend you searched could not be found in your friends list!
Close
Schedules Page (Weekly View)
Home
Friends
Schedules
Sun Mon Tues Wed Thu Fri Sat
Calendar View
Add Event

New Features List:

Switching from app to website changed out plan for the format of pages, so now we will have a handful of pages that link to each other and communicate with a database instead of one app.

Besides that, most of our functions remain the same. Ranked from most to least priority:

Basic calendar
  view own events and also view friends events
Edit calendar
  update above-mentioned calendar with user inputted information
Add friends
  add friends by name search or with an id input
Create time matching system
  create a system that allows you to view what times you and your friends are both available
User creation
  create an account and update the database with login information
Aesthetics
  make the pages look nice and other small stuff like profile pictures, placement of on screen features, and other details
User Authentication
  check to match the user login input with a database