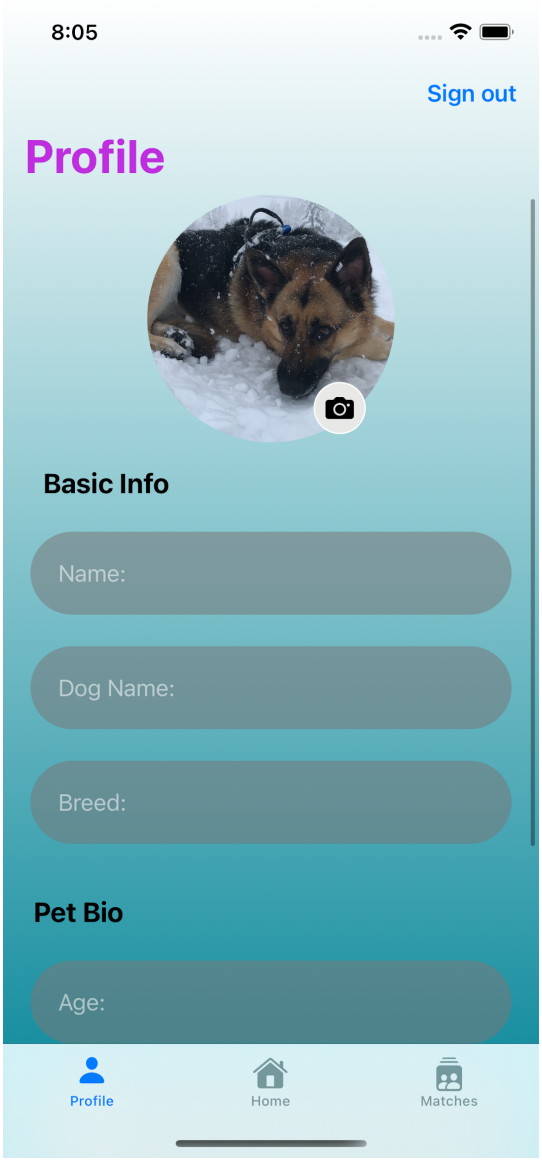
Fetch: The Tinder for Doggy Playdates

Project Plan

Ronny Fuentes, Kyra Novitzky, Jack Sanders, Stephanie Schofield, Callista West
Fetch Team
https://github.com/JackSanders1998/CIS422Proj2
Professor Juan Flores
March 12, 2021

Fetch Application

COVID-19 has brought about many new lifestyle changes--especially when it comes to social interaction and exercise. One way people have managed their cabin fever during quarantine is to get outside and walk their pet. This allows for much needed fresh air and movement, but the issue of social isolation still remains. What if there were a way to get together with friends in a safe way, while also giving your dog the opportunity to meet other dogs? Our team set out to develop a solution that would address all of these needs. Fetch is an iOS application built for dog owners to match with other owners to set up playdates with their dogs. Owners can describe their pet's attributes like name, breed, birthday, sex, whether they are spayed or neutered, and size. A mockup of the profile page is below.

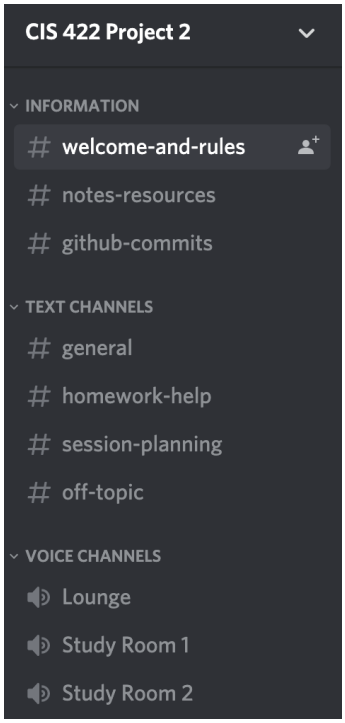


Management Plan

Our team organized the project into two components: the client side and the server side. We divided up the team by component, and then created modules inside each component where we assigned sections to each team member. Our architecture is as follows:

- Client side (What the user sees)
- Log-In/Sign-Up
    - Allows users to log-in to an existing account or create a new one. This will typically be the initial page a user sees when opening the application.
- Profile

        - This is where a user updates and stores all of their information. It will be filled out upon initial sign-up, however, if a user decides they would like to change anything they will always have access to it.
- Homepage
    - Deck of unseen dogs
        - This is the card-like deck that consists of all dog profiles that a user has yet to see. In other words, dog profiles that a user must make a decision about.
    - Like/Dislike buttons
        - These buttons send profiles to the discard piles: Accepted, Declined, or Pending.
- Matches
    - All currently existing matches will exist under this page.
    - Chat
        - Every match will have a chat feature to allow profiles to communicate with each other and set up playdates.
    - Unmatch
        - If a user decides that they change their mind on a profile, there will always be an unmatch button next to a profile.
- Server side (What the programmer sees)
    - Database
        - Existing User Profiles
            - Name, Dog's info, profile picture
            - Matches pile
            - Dislikes pile
            - Pending pile

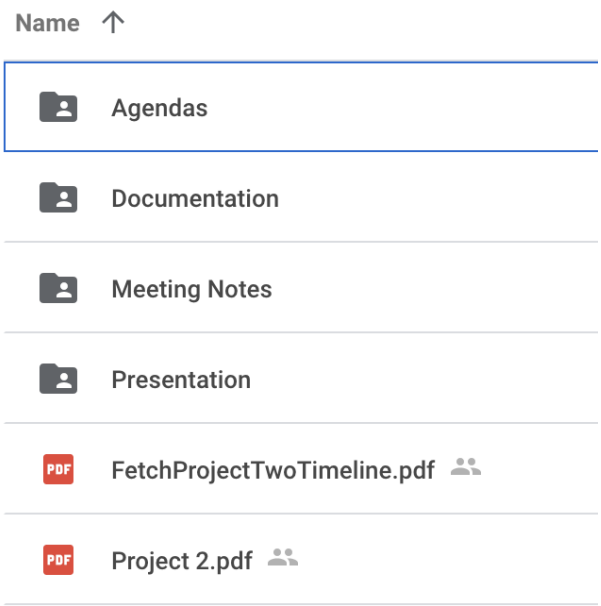


On the client side, we designated the login and sign up section to Kyra and Callista, the profile and account section to Ronny and Stephanie, the homepage section to Jack and Stephanie, the matches to Ronny, and the database management of user profiles to Jack. We agreed that we would all contribute to each section when assistance was needed, but distributed these sections to help organize our workflow.

Our team met frequently to brainstorm, distribute tasks, find clarification, and maintain accountability among the group. We used Discord as our main

form of communication, and utilized the voice channel for meeting. Jack created a webhook for a #github-commits channel that updates every time someone pushes to our git repository, which further motivated us to make progress. A visual of our Discord group is shown to the right:

We also created a shared Google drive to store all of our documentation and visuals, shown below:



Project Timeline

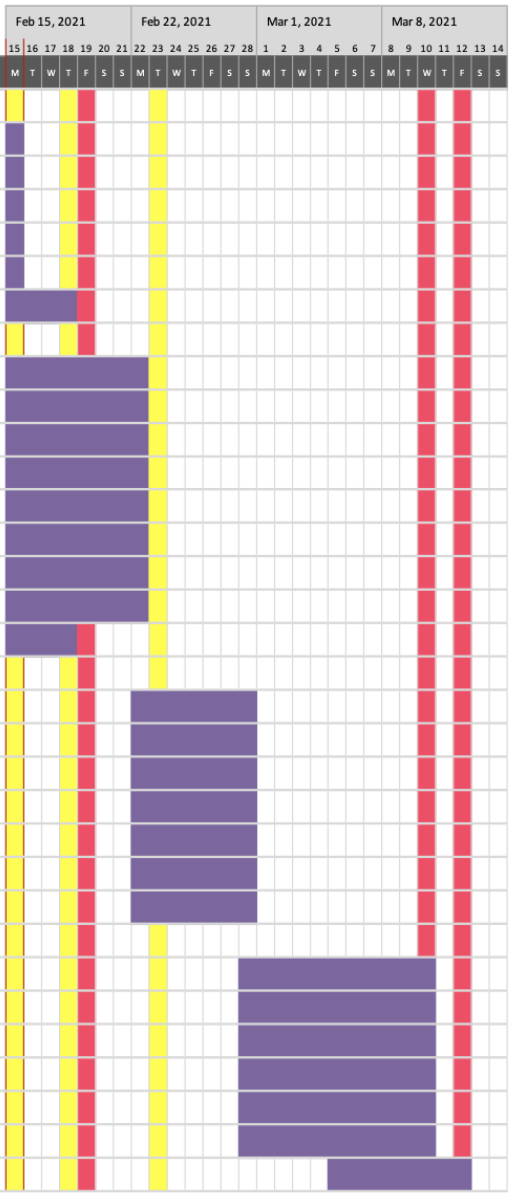
In order to stay accountable and make consistent progress, tasks were divided based on the modules we were given with due dates on each task. Our project timeline is shown below:

CIS 422 Project Two: Fetch

Ronny Fuentes, Kyra Novitzky, Jack Sanders, Stephanie Schofield, Callista West
Fetch Team

Project Start: Mon, 2/15/2021
Display Week: 1

TASK	ASSIGNED TO	START	END
Project Organization			
Team meeting and deciding on project idea	Team	2/15/21	2/15/21
Developing software architecture	Team	2/15/21	2/15/21
Setting up Google Drive	Stephy	2/15/21	2/15/21
Setting up Discord	Kyra	2/15/21	2/15/21
Setting up Git repository	Jack	2/15/21	2/15/21
Distributing tasks and writing SDS + SRS docs	Team	2/15/21	2/18/21
Data Management and Functionality of App			
Discussing functionality and logic of app use	Team	2/15/21	2/22/21
Creating UX/UI designs	Ronny	2/15/21	2/22/21
Creating account information for current users	Stephy	2/15/21	2/22/21
Draft functions for deck and swiping logic	Jack + Stephy	2/15/21	2/22/21
Draft functions for login screen	Kyra + Callista	2/15/21	2/22/21
Draft functions for utilizing new user's account info for matches	Callista	2/15/21	2/22/21
Draft functions for utilizing current users' account info for matches	Kyra	2/15/21	2/22/21
Draft functions for private chat feature with matches	Ronny	2/15/21	2/22/21
Working on SDS and SRS documents	Team	2/15/21	2/18/21
Debugging + UI Design			
Debugging deck and swiping functions	Jack + Stephy + Team	2/22/21	2/28/21
Debugging login screen functions	Kyra + Callista + Team	2/22/21	2/28/21
Debugging new user account info functions	Callista + Team	2/22/21	2/28/21
Debugging current user account info functions	Kyra + Team	2/22/21	2/28/21
Debugging private chat functions	Ronny Team	2/22/21	2/28/21
Updating SRS + SDS functions	Team	2/22/21	2/28/21
Designing user interface	Ronny + Team	2/22/21	2/28/21
Output and preparing presentation			
Finalizing and merging deck and swiping functions	Jack + Stephy + Team	2/28/21	3/10/21
Finalizing and merging login screen functions	Kyra + Callista + Team	2/28/21	3/10/21
Finalizing and merging new user account functions	Callista + Team	2/28/21	3/10/21
Finalizing and merging current user account functions	Kyra + Team	2/28/21	3/10/21
Finalizing and merging private chat functions	Ronny + Team	2/28/21	3/10/21
Preparing for final presentation	Team	2/28/21	3/10/21
Project one due, code and reports delivered	Team	3/5/21	3/12/21



Building Plan

Team members were assigned to modules, and fulfilled tasks related to those modules by a certain date as specified in the timeline. The first stage of our project involved meeting over Discord, getting to know one another, establish our programming environment, and come up with ideas for the project. From there, we decided on these modules and asked all members to contribute to the functionality of their module. In the second stage, we have been meeting frequently to gain clarification for our code, the logic of the app, and attributes of our database.

We are also focusing our efforts on finding an appropriate API or constructing data sets that will be used for account information on the app. Our plan for the third stage of the project is to continue supporting the functionality of these modules by debugging and revising our code. We will also focus on the user interface (UI) design in this stage. Any debugging or missing functionality will be implemented here as part of our Agile approach to tackling this software project. We will also meet with our client often throughout the stages, but especially during the debugging stage to confirm the functionality is up to standards. Finally, we will start wrapping up the project in the fourth stage by merging our branches, and preparing for the final presentation.

Monitoring and Reporting

Our approach to monitoring our work will be clear and consistent communication on Discord, and viewing the project timeline often to keep each other accountable. In the above timeline, the purple bar represented the timespan allotted for each task, and is how we could measure our progress. This document has been altered often throughout our project, but allows us to stay on top of our work.

Rationale

We decided to break our system down in this way to streamline the programming process and establish clear expectations. However, one challenge we will face is testing the functions that rely on other modules. We will need to collaborate often so we can understand our gaps and errors in each program. We also want to communicate often so that we can discuss any problems or get clarification about logic early on in the project. This is so we can avoid merging conflicts, logical errors, and major problems down the road. Our Discord channel, shared Google Drive, and distribution of tasks in the timeline will help us manage this project efficiently.