

Party Finder

016-05

| Name | Github | Email |
|--------------------|----------------|--|
| Zoie Nuño | wyntersmith | zonu9746@colorado.edu |
| Aidan St. Cyr | aist9379 | aist9379@colorado.edu |
| Dan Medvedev | DanielMed1620 | dame0706@colorado.edu |
| Hayden Schlichting | Haydebug | Hasc5772@colorado.edu |
| Ali Almutawa Jr. | PiCake314 | alal5051@colorado.edu |
| Nolan Lee | NolanLee100101 | nole5800@colorado.edu |

Project Description

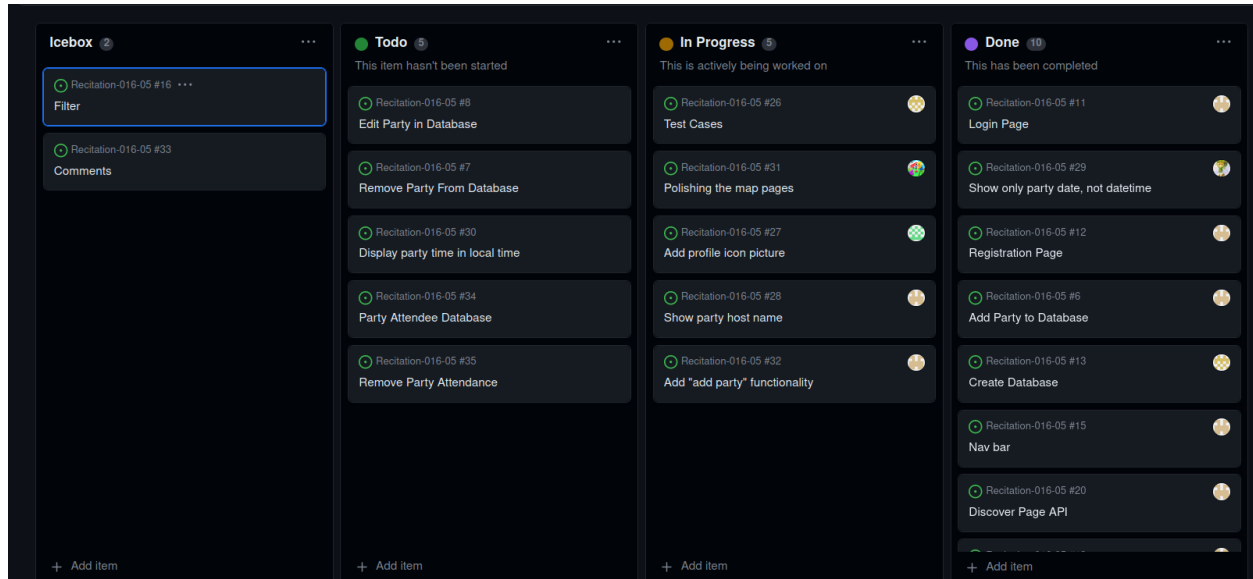
The goal of our application is to bring random individuals together for shared parties. We are going to have a login and register pages that are connected to a database that stores the user's first and last name, email, username, and hashed password. With this we will also have a profile page which displays user information, such as their name, email, the number of parties they have attended, and the number of parties hosted.

On the party page, we will display a map that represents all of the parties going on based on geographic location as pinpoints. When clicked on, these pinpoints display the party information, such as the date of the party, the party start time, the party location, the name of the party, the party image, as well as a description of the party. There will be a separate pop up to display the party host and their information.

In addition, users will be able to create, delete, and edit their own parties via the `add_party` page. All of the parties and their relevant information will be stored in the database to be accessed by the map display on the party page.

Project Tracker

<https://github.com/users/wyntersmith/projects/2/views/1>



Video

VCS

<https://github.com/wyntersmith/Recitation-016-05.git>

Contributions

Zoie Nuño

I focused on the development of the database, profile page, and testing. I created the database in PostgreSQL, using it to store information in three tables: users, party_info, and user_parties. I created the backend API for the profile page by gathering the user information at login, and I created a template frontend page to display all of the information. I created test cases for each of the pages to make sure each was functioning correctly. I also fixed some minor errors that were present, such as preventing a person from registering with no information inputted.

Aidan St. Cyr

I initially launched the login and registration page, as well as the basic layout of the website with blank pages. From there I focused primarily on front-end to back-end linking as new user inputs, databases, and APIs were implemented. This meant I would frequently test for user edge cases and adjust constraints to disallow unwanted user input.

I worked closely with Zoie as any database change influenced most API calls. However since we communicated regularly in the database and API planning phase of this project, we made the APIs flexible to changes we knew were coming down the road.

I also worked closely with Hayden to ensure that all the information he needed for the map API was stored correctly in the databases as needed. As the project matured, we would experiment with easier and more efficient uses of APIs so that we could eliminate any errors induced by complex processes in our code.

Dan Medvedev

I worked on both back end and front end. Dealing with formatting issues, mostly making sure things just looked smooth and symmetrical. In addition, I also was working on the backend for the register page, taking in the users information and making sure that it was correctly matching with the database. I created case diagrams, and aided in the deployment/testing portion of our page with Nolan.

Hayden Schlichting

I was in charge of the map implementation in our project, so I learned all about MapBox API, how to make coordinates show up, how to get the map to show up, how to display party

info when you click a pinpoint, showing pinpoints, and showing host details. I also worked on the design for the login and register page, and helped with converting user addresses entered to an autofill suggestion form of addresses near you using mapbox. I also assisted with test case design.

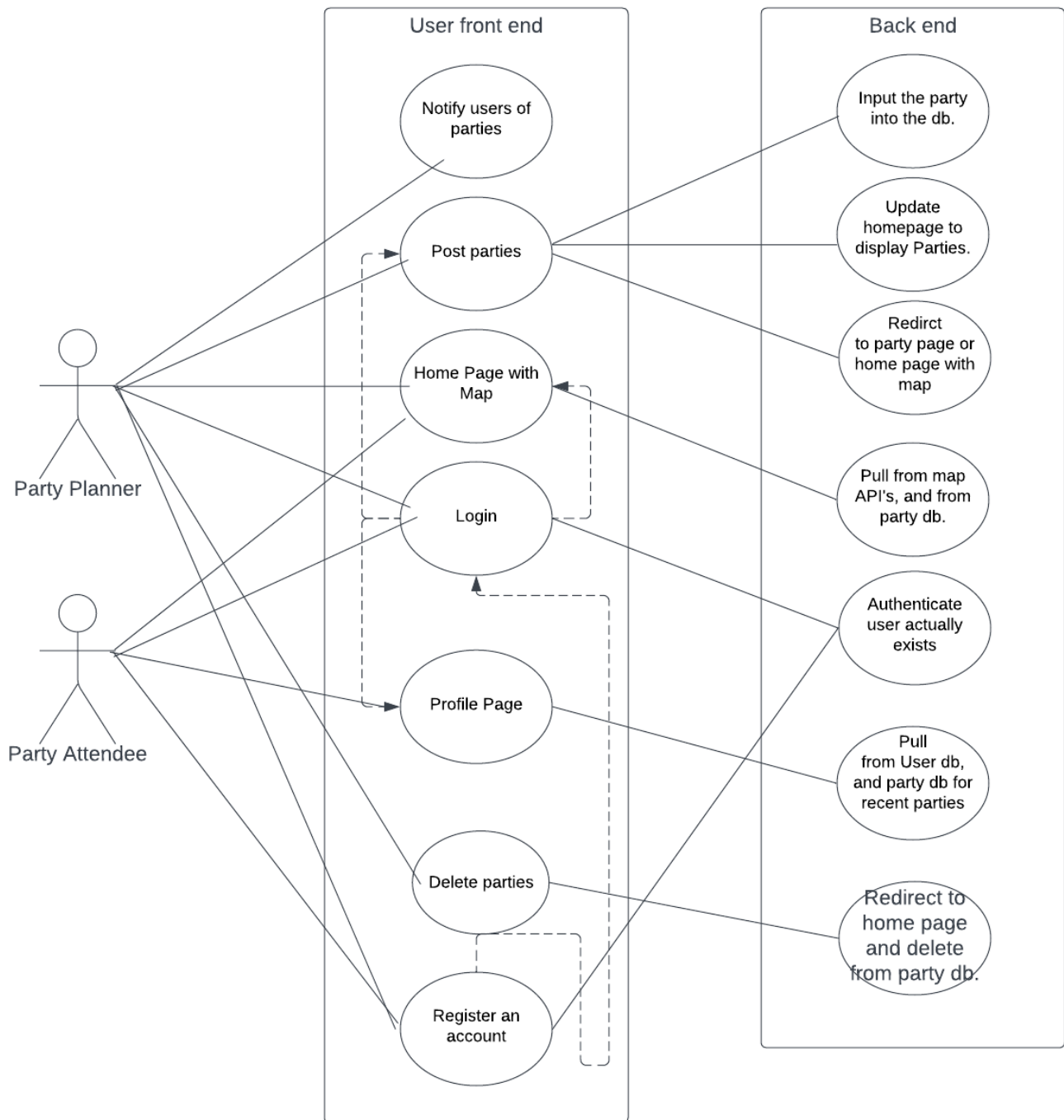
Ali Almutawa Jr.

My main job in the team was creating good design, namely working on the frontend part of the project. While another teammate would write the code for the functionality, I would take the page and transform it from being only functional to something that is pleasing to look at. Doing this however required communication with myself and the person in charge of the functionality, which essentially would lead me to understand how the technical stuff works too. I did also end up helping in the backend part when my teammates would need help writing an api, or doing a sequel query. I also designed the logo for our website!

Nolan Lee

I worked on beautifying the profile page, as well as adding a profile picture where the user can upload an image from their computer. I used a library to import icons and use them to make the page look cleaner. I also worked on the deployment of our page. We used Microsoft Azure to host and run our application. You can access the website, [click here](#), but it was taken down because it was a free trial and we did not want our free time expiring.

Use Case Diagram



Test Results

Test Cases

1. A user should be able to login if they enter the mandatory information: username and password.
 - a. If the user enters the wrong username and password combination, login should fail and return status code 403.
 - b. If the user enters a username that is not present in the database, login should fail and return status code 403.
 - c. If login is successful, then the user should be redirected to the profile.
2. A user should be able to create an account if they fill in the mandatory fields: first name, last name, username, email, and password.
 - a. If the user is missing any of these fields, or if the username or email are already in use, the registration attempt should fail and provide status 400.
 - b. If registration is successful, then the user should be redirected to the login page.

Observations

1. The users are entering their username and password into the login page in order to view the profile, party, and add party pages.
 - a. This behavior is consistent with the use case.
 - b. Any deviation is due to entering incorrect information or by attempting to login without registering.
 - c. This information was used to show errors when incorrect information or information that does not exist within the database is entered into the login form.
2. The users are entering in their first name, last name, username, email, and password in order to create an account that they can use to later login.
 - a. This behavior is consistent with the use case.
 - b. Potential deviations would be leaving any/all of the fields blank, or by entering an invalid email.
 - c. This information was used to make changes that checked whether or not the email was valid and prevented the user from leaving any of the forms blank.

Deployment

<http://recitation-016-team-05.eastus.cloudapp.azure.com:3000/login>

For deployment we had created a VM(Virtual Machine), using Microsoft Azure. To make sure the application is running correctly, make sure docker is running on the VM allowing any users to connect using the link above.