B-ACTIVE

by the Active Bruins

KRISHNA BABU, EDDIE HUANG, ERIC KONG,
RYAN LO, RAMYA SATISH, & NICOLE WONG[1]

PREAMBLE

This project is being developed under Dr. Miryung Kim's COM SCI 130, in Brett Chalabian's Discussion 1C. Its Github repository may be found at

https://github.com/ryanlo7/bActive

The UCLA IDs of each team member are as follows.

Krishna Babu	604460009
Eddie Huang	204607720
Eric Kong	304601223
Ryan Lo	704579791
Ramya Satish	104601436
Nicole Wong	104651105

CONTENTS

1 Motivation 2
2 Feature Description 2
3 Usage Scenarios 4
4 Diagrams 4
5 Mockups 4
6 Feasibility 6
7 Capabilities 9

LIST OF FIGURES

Figure 1 Use-case diagram 5
Figure 2 Class diagram 5
Figure 3 Welcome page 6
Figure 4 Profile page 6
Figure 5 Matching page 7
Figure 6 Events page 7

[1] *Department of Computer Science, University of California, Los Angeles.*

1 MOTIVATION

A common problem faced by UCLA students is finding a time to meet with others for leisure activities, given everybody's busy schedules. For example, if a student wants to participate in an activity such as working out or playing basketball, but none of their friends are available, it is currently difficult to take advantage of the size of UCLA and find new people who are available at the specific time the student would like to meet up for the activity. Furthermore, given the stress associated with attending a university, a way to keep students active while promoting new friendships would be physically, mentally, and socially beneficial.

The proposed application allows users to save their favorite activities along with their preferred locations and times to meet up into a profile. The application then matches each student with others who have similar athletic activity interests and time availabilities. For pair activities, such as weightlifting, the application suggests a partner for the user to do the activity with. For group activities, such as team sports, the application suggests groups of people. These suggestions will allow students to form a group of a desired size for an activity, taking advantage of the university atmosphere to promote new friendships between students with similar recreational interests. Ultimately, the vision for the application is to make it easier for UCLA students to form groups for leisure activities, while encouraging new friendships and promoting an active lifestyle.

2 FEATURE DESCRIPTION

Our product comprises four features: a registration page, an interests and skills list, a user-matching algorithm, and activity event generation. We examine each of these components in more detail in the following sections, from the perspective of a user.

2.1 Main Features

In this section we expound on the core functionality of the application, whose implementation we can guarantee.

2.1.1 *Registration*

To register for B-Active, the user enters her name and e-mail address and selects a password. She receives a registration token by e-mail, which she enters into B-Active's registration page to activate her account. At registration, she can select her times of availability and (optionally) enter auxiliary contact information, such as her cell phone number. This leads her to the Interests and Skills List.

Feature Extension. The user can also register through a Facebook account.

2.1.2 *Interests and Skills List*

The user selects, from a list of activities, the ones in which she seeks to participate (her interests), by rating them on a scale from 1 to 10. For each activity, she also indicates how proficient she fancies herself to be in it, on a friendly scale from 1 to 10. These activities will include pair activities such as lifting, tennis, and racquetball, team activities such as basketball, volleyball, badminton, soccer, and group activities such as running, rock-climbing, and swimming.

This selection and self-assessment happens automatically at registration, but the user can return to the list and edit her preferences. She can also edit her times of availability.

2.1.3 *User Matching*

The application matches users based on similar interests and availabilities, using a "similarity score" algorithm. For activities that require groups of people, the application suggests groups of people. The application also takes into account the user-indicated skill levels to form balanced teams in group activities.

From the user's perspective, other users with similar interests and compatible availabilities will be suggested to her. She can browse the profiles of these other users.

2.1.4 *Activity Event Generation*

When users agree with their matching, the system will generate an activity event for them, and invite each of them to the event. This event will occur at a time which is compatible with all of its users involved, and at a place which is most convenient for the users. The users can adjust the time and place of the event. The system will tactfully broadcast each of the users' contact information to one another, so that they may communicate and coordinate.

On her dashboard, a user can see a list of pending activities displayed in a calendar format.

Feature Extension. The system can use map data to suggest a most convenient venue for each event, instead of relying on user-selected preferences.

Feature Extension. Users can manually add, to these events, other users which the activity's algorithm neglected to suggest.

2.2 Extra features

2.2.1 *Alert Messages*

If the user has entered her cell phone number, then she can opt in to receive text alerts. These alerts will be sent to her cell phone number, an hour before each of her activities is scheduled to happen. (Messaging rates will apply.)

2.2.2 *User Ratings System*

After each activity is complete, the user can review her experience by giving points to participants who showed up, and penalising those who did not. This gives the participants an incentive to attend their events.

Alternatively, these points can be replaced by a five-star ratings system.

2.2.3 *Inter-User Messaging*

A user can message other users to whom she is matched by the application. This would be in the form of a friendly chat system for matched users and groups.

3 USAGE SCENARIOS

Since this app is targeted at the UCLA student demographic, the app will feature various athletic activities usually seen played on campus, as well as favorite locations to do them at (such as Wooden Center and Sunset Recreation Center). As such, all usage scenarios will involve UCLA students using the app.

An example user of our app would be a UCLA student seeking to find motivation to workout. This can be due to laziness, desire to lose weight or live a healthier lifestyle, relieve stress, etc. Having a friend or friends to go to the gym with, is a great motivator because then the user doesn't have to do it alone. After inputting his or her contact info, interests (in this example, let's use weightlifting), and available times, the app will send the user a possible match for a gym buddy. All of the matching work is done by our app and cuts down the user's barrier to get and stay active. Given the ease of the matching process, registered users simply have to confirm or decline activity matches and show up. This system helps keep users accountable for going to the gym at their desired times, whether with new peers or old buddies.

An additional usage scenario is a UCLA student who cannot join a club or an IM team for a variety of reasons, such as time commitment or not being skilled enough. The student wants a low-commitment way to stay active and get the benefits of such activities, or build up their skills in a friendly environment. By joining our app, we can cut down all the work involved in setting up a low-commitment activity for fun with college peers. After meeting up once or twice, the participating students can then decide to continue to play the sport recreationally or keep it casual.

Finally, a UCLA student, active or not, could simply want to look for friends or people to do fun physical activities with. The user can use the matching feature to find a group of 4 people who share an interest in playing basketball at Hitch courts late at night and they can exchange contact info to not only bond over basketball but to also do a hangout elsewhere. Less people using the app will be lonely and can make new friends through our app.

4 DIAGRAMS

In Figure 1 on the following page, we present a UML use-case diagram detailing the functionalities that users can expect.

In Figure 2 on the next page, we present a UML class diagram detailing the skeleton of our application logic.

5 MOCKUPS

In this section, we demonstrate a prototype of our user interface.

A draft of B-Active's welcome page is shown in Figure 3 on page 6.

Once the user has logged in, the profile page in Figure 4 on page 6 is displayed.

The user can see the other users which match her interests in the page shown in Figure 5 on page 7.

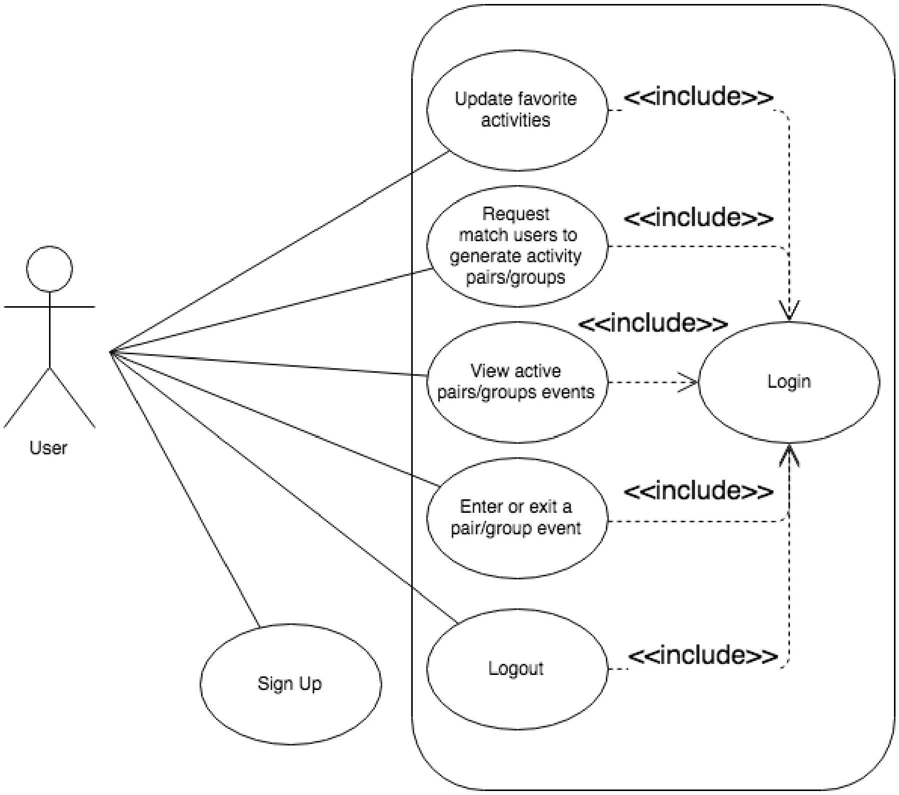

Update favorite activities
<<include>>
Request match users to generate activity pairs/groups
<<include>>
<<include>>
User
View active pairs/groups events
Login
Enter or exit a pair/group event
<<include>>
Sign Up
Logout
<<include>>

Figure 1: Use-case diagram for our B-Active application. We plan to implement only one type of user with many use cases only allowed when logged in.

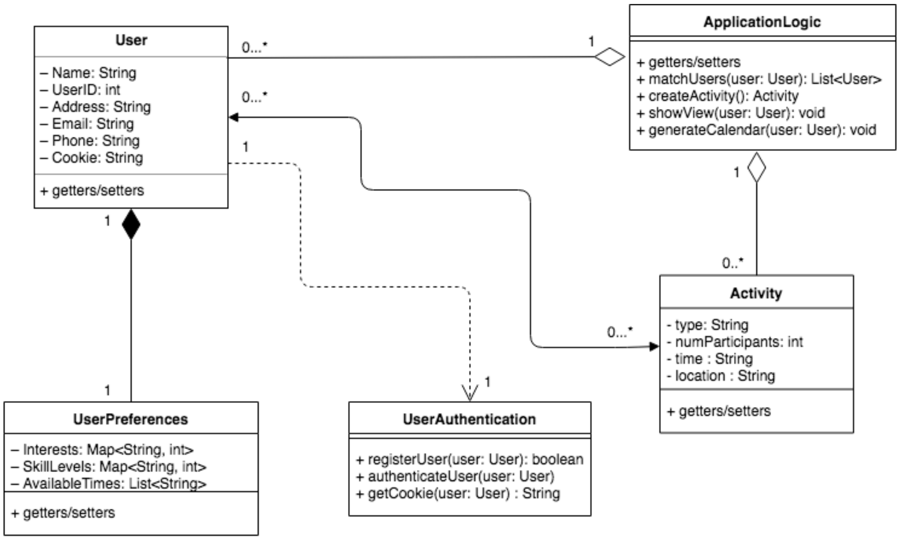

User
– Name: String
– UserID: int
– Address: String
– Email: String
– Phone: String
– Cookie: String
+ getters/setters
0...*
1
0...*
1
ApplicationLogic
+ getters/setters
+ matchUsers(user: User): List<User>
+ createActivity(): Activity
+ showView(user: User): void
+ generateCalendar(user: User): void
1
0..*
Activity
- type: String
- numParticipants: int
- time : String
- location : String
+ getters/setters
0...*
1
1
1
UserPreferences
– Interests: Map<String, int>
– SkillLevels: Map<String, int>
– AvailableTimes: List<String>
+ getters/setters
UserAuthentication
+ registerUser(user: User): boolean
+ authenticateUser(user: User)
+ getCookie(user: User) : String

Figure 2: Class diagram for our B-Active application.

The user can also see the events in which she is participating in Figure 6 on page 7.

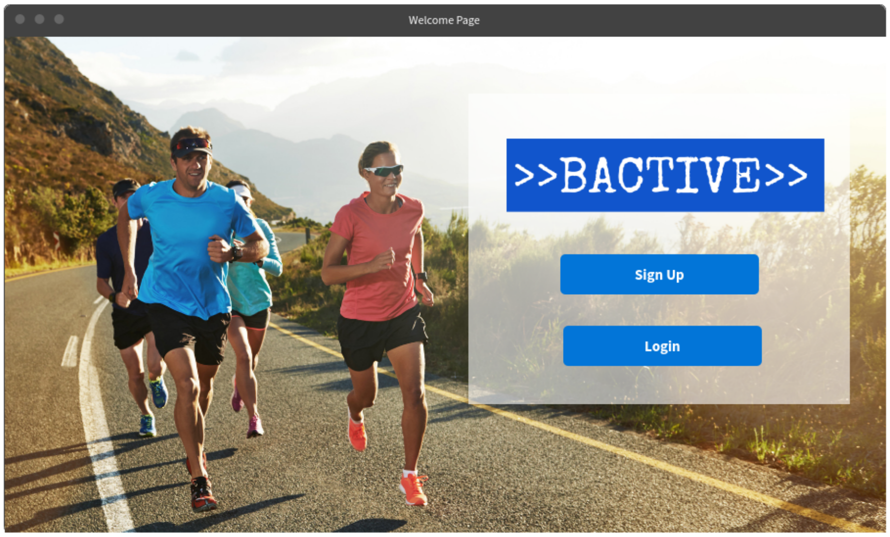
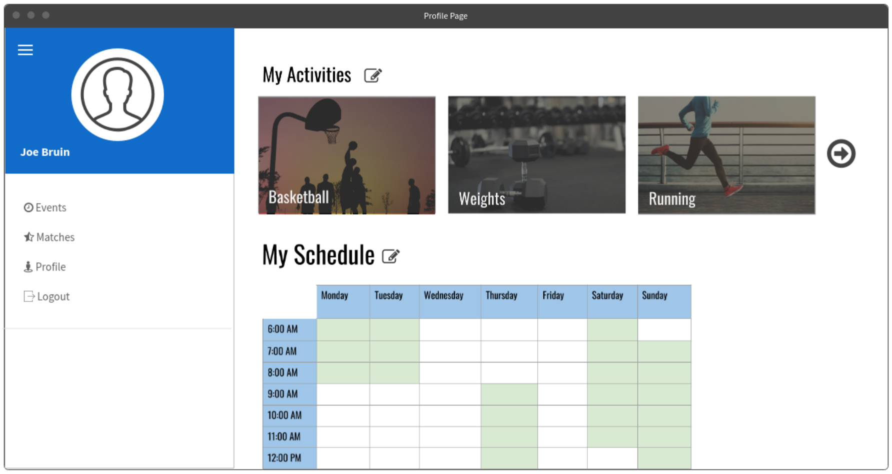


Figure 3: A prototype of B-Active's welcome page



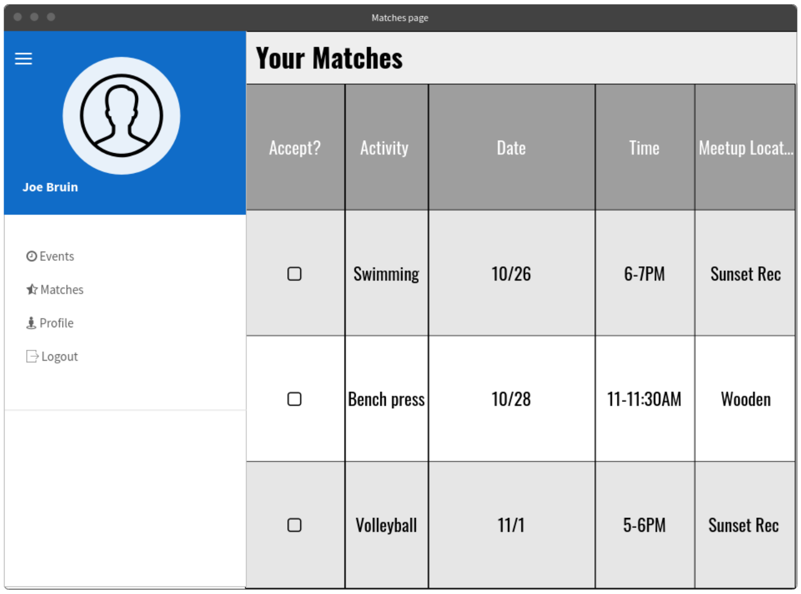
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
6:00 AM	🟩	🟩				🟩	
7:00 AM	🟩	🟩				🟩	🟩
8:00 AM	🟩	🟩				🟩	🟩
9:00 AM				🟩		🟩	🟩
10:00 AM				🟩		🟩	🟩
11:00 AM				🟩		🟩	🟩
12:00 PM				🟩			🟩

Figure 4: A prototype of B-Active's profile page.

6 FEASIBILITY

6.1 Justification of newness

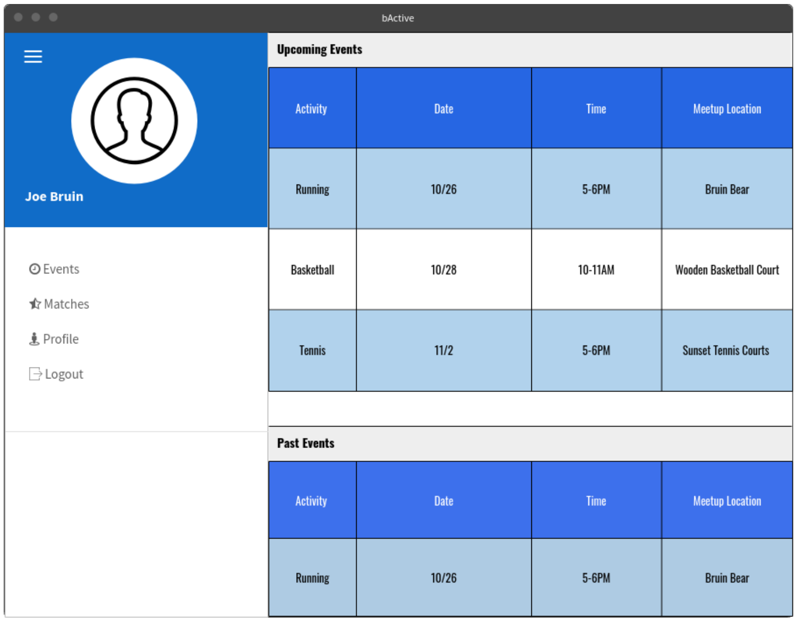
Our B-Active web application offers new features and functionalities that no other existing application currently does. Existing apps for finding workout buddies only search for potential buddies based on the user's location; our app will attempt to match users in pairs or groups for whichever activities they are interested in, based on their availability schedules and interests. In addition, our app is not only restricted to gym workouts, such as weightlifting, usually restricted to teaming up in pairs, as well. We intend this app to be more centered on many physical sports and activities, with our target demographic being UCLA students. This, as well as the unique ability to be able to form both pairs and larger groups for team-oriented activities, will allow more diversity in group sizes and physical activities to spur our efforts in encouraging more students to be more active.



Your Matches

Accept?	Activity	Date	Time	Meetup Locat...
□	Swimming	10/26	6-7PM	Sunset Rec
□	Bench press	10/28	11-11:30AM	Wooden
□	Volleyball	11/1	5-6PM	Sunset Rec

Figure 5: A prototype of B-Active's matching page.



Upcoming Events

Activity	Date	Time	Meetup Location
Running	10/26	5-6PM	Bruin Bear
Basketball	10/28	10-11AM	Wooden Basketball Court
Tennis	11/2	5-6PM	Sunset Tennis Courts

Past Events

Activity	Date	Time	Meetup Location
Running	10/26	5-6PM	Bruin Bear

Figure 6: A prototype of B-Active's events page.

Additionally, existing apps like Bumble allow for meeting new friends and connections. However, the user still has to do the work of selecting matches that they like and messaging them to set up an interaction. Our app instead cuts down the barrier to meeting someone with similar interests, specifically a peer at UCLA, by doing all the matching work from user profile information. Users are matched to desired activities at certain times that they elect, and are notified to accept or decline the event invite. After this sole step the users simply have to show up to the pre-set event location. This significantly reduces the barrier to meeting people with similar interests, as the user simply has to maintain an accurate profile and confirm events while the app does the rest.

The feature of forming groups for team sports on the B-Active application differs significantly from joining a sports team or club, since the application allows each user to choose the dates, times, and frequencies for participating instead of adhering to an organization's schedule; the application provides total time flexibility to the user. Furthermore, B-Active differs from using Facebook groups to find people who have similar interests, since our application automatically matches people with similar time availabilities. Facebook groups are usually rather large, and are not convenient if each user in the group must search for other people whose schedules match. Ultimately, tools like Facebook or large sports clubs are more useful for finding one time that works for most members of a group, rather than performing a match for each user. Therefore, unlike other alternatives, the B-Active application finds buddies based on locations and schedules for a variety of athletic activities, encouraging all students to be more active.

6.2 Justification of feasibility

Given the expected timeframe of this project, we believe this project is feasible. First and foremost, this entire project and its basic functionalities can be simplified down to a standard full-stack web application. The site's frontend will display the user interface and all the information the user needs to see. Meanwhile, the backend will be able to respond to the user's interactions with the user interface, as well as retrieve and update that information (through reads and writes to the database that stores all users' fields) and send it to the user's display to dynamically alter the page. Furthermore, majority of the members of this team have had experience with website applications before (more expounded upon in our "Capabilities" section) and, thus, we will need less ramp up time to learn the required technologies and languages before starting work on the project. Lastly, we have divided the project into several distinct features that will be implemented on separate web pages: registration and login, user activities matching, and activity event viewing. For registration and login, we will be using the Facebook Javascript SDK at `https://developers.facebook.com/docs/facebook-login/web/` to implement secure login and access the user's public profile information, such as name and profile picture.

6.3 Non-functional requirements

Some non-functional requirements that are critical to the success of our application are usability, testability, and modifiability. One caveat to usability is that the number of matches likely depends on the number of people using the application. The more people using the application, the higher the chances that a person would be matched to similar times and activities; there is a network effect present. Testability of the similarity scoring and matching process will be done by creating unit tests to ensure correct behavior for similarity score computation. Regarding modifiability, we ensure that the application is easily enhanceable, in order to provide a better experience for users. We achieve modifiability by using a design that allows for easy addition of new activities in the future.

7 CAPABILITIES

Our team members have a wide array of skills and experiences with the different technologies aforementioned and to be mentioned. Most notably, a majority of the team has dealt with full-stack web applications, both front and backend sides, at some point in their lives and would be able to greatly contribute to the different parts of the project. Consequently, the development of our application should not be too difficult and will be completed on time.

Krishna has experience in back-end development, specifically in Java and Scala. She has developed APIs in her summer internship and worked to establish full test coverage on her code. At UCLA, she has taken CS 143 Databases and has provided data engineering and backend support at UCLA lab. **Given her back-end experience, she will be working on integrating Facebook APIs to facilitate user sign-ups, login, and messaging.** For Part A of the project, she worked on the Feasibility and Usage Scenario Sections, as well create UI mockups for the events and matches pages.

Eddie has experience in back-end web development, especially in Python, having developed server-side algorithms for a summer internship for a news company to handle comment storage and ranking. He is currently taking CS 144: Web Applications, and is currently working on a project that implements the MEAN stack for web development, and **as such, has the experience necessary for developing the back-end of the project. Therefore, he will be writing the logic that handles how activities are stored and their interactions with users.** For Part A of the project, Eddie, along with Eric and Ryan, defined the feature requirements, along with the extra features. In addition, they discussed and designed both the use case and class UML diagrams.

Eric has experience in full-stack web development in Javascript, having taken the Databases and Web Applications courses at UCLA and having served as a mentor for UCLA ACM HackSchool's MEAN stack web development course, and has also implemented Python web-scrapers for personal projects. **He will be working with Ryan on the user interface and frontend application logic, and shall dabble in the backend for a few tasks**. For Part A of this project, he wrote and refined the feature description section with Ryan and Eddie, and provided the outlines for the structure of the UML class diagram in Figure 2 on page 5. He also typeset this project report in LaTeX.

Ryan has created two full-stack web applications for a personal project and internship a few years ago. Both web apps involved a user login and registration page and user interactions with the database. His most recent internships dealt with server-side programming for mobile applications and databases. He is also concurrently taking CS 144: Web Applications along with this class to learn how to implement the MEAN stack for web development and seeks to utilize those skills here. **As a result, he will be working with Eric primarily on the user interface and display on the frontend as well as some server-side tasks.** For Part A of the project, he helped plan out the application's features and draw the UML diagrams with Eric and Eddie, and also wrote up some of the lengthier sections of the report with Nicole, Eric, and Krishna, such as the Feasibility and Usage Scenarios sections.

Ramya has experience in backend development in Java and C++, primarily with development involving statistical analysis. She also has experience creating a Google Sheets add-on using Google Apps Script, which is extremely similar to JavaScript. This project involved developing an object-oriented design for structured data. Due to Ramya's experience with object-oriented design in different programming languages, **she will work on the ApplicationLogic class, the User class, and the UserPreferences class**. For Part A of the project, she worked on the motivation section and the feasibility section of the report.

Nicole has experience working with the backend of web services as well as client-side Javascript through her internships. She has taken CS 143: Databases, and is currently enrolled in CS 144: Web applications learning how to implement the MEAN stack for web development. **Consequently, she will be working on the server side to handle user authentication and help build the application logic.** For Part A of the project, she helped write out the feasibility section as well as create mockups for the login and profile pages.