Hinder

Tinder for Hackathons

CS 130: Software Engineering
Professor: Miryung Kim

Discussion 1B
TA: Twinkle Gupta

Team Name: TBD
Kimberly Svatos, 604425426
Marshall Briggs, 304417630
Kyle Haacker, 904467146
George Archbold, 604407413
Daniel Berestov, 404441309
Apurva Panse, 504488023

Team Repository: https://github.com/marshdevs/Hinder

1 Motivation

The process of forming teams for hackathons, class projects or other events, can be difficult and awkward. You never know where to start looking. It usually devolves into some form of weird speed dating where you are trying to talk to the most people and hear the most ideas but quickly form a team in order to avoid being left out. Our app aims to mitigate the awkward aspects of this process and provide an all inclusive platform which facilitates team forming while also allowing the most exposure and interaction between projects and possible team members.

2 Feature Description and Requirements

Overview

Hinder consists of events, projects, and users. Events may include, for example, hackathons and classes (e.g., LA Hacks, CS 130, etc.). “Event Planners” (e.g., hackathon coordinators, professors) are those users who create and manage events that are advertised on the app. Users who browse and join the events listed on the app are known as “Event Attendees”. For events of interest, Attendees may want to either join an existing group project for that event, or create and manage a new group project. The goal of the app is for Attendees and projects to be seamlessly matched together based on skillsets and areas of interest.

To facilitate these processes, Hinder will consist of the following features: an ***application dashboard*** where users log in and then proceed to browse or manage events; a ***user profile*** where users manage their personal information, skillset, and areas of interest; ***project creation and management*** that enables users to learn more about existing projects, or create and manage a new project; ***project and user interaction*** similar to Tinder that will ensure users are matched to projects based on area of interest and desired skillset; and a ***mediator*** that helps handle the matching between users and project. Further details for each feature and requirement are given below.

2.1 Application Dashboard

When a user opens the app, the app will launch to a screen with the option to login or create an account. A user can create two types of accounts: first, an “Event Attendee” account, or second, a “Event Planner” account. If the user is an Attendee, once logged in, the home screen is a simple scrollview showing events, with a settings tab on the top left, and a menu button of the top right. If the user is already marked as attending some events, these events will show up at the top of the home screen scroll view, followed by a section of “Events You May Be Interested In” below, suggesting nearby or similar events to that user. The menu in the top right will allow a user to choose to go either to a page where all the user’s projects/groups are stored, or a page showing all events that user is enrolled in. The settings tab in the top left will allow a user to view and edit their profile.

If the user is an “Event Planner,” their homepage will show a scroll view of all events they have created and are the “owner” of. They will have the option to create an event, which will need a picture, location, date, and a description of what the event is. They will have the option to edit events they’ve already made, and see information about it such as how many people have marked themselves as attending.

2.2 User Profile

For the best possible matching experience, each user will want to ensure that their profile information is up-to-date. Hinder users will be able to access their personal profile from the settings tab on the application dashboard (as discussed in 2.1). From the user profile, users will be able to manage personal information such as their profile picture, name, occupation (e.g., school or company), and general experience level. Each user will also have a skillset. The app will define an extensive list of skills from which users can select skills to add to their skillset. A user may also remove skills from their skillset at any time. The skillset will be an especially important aspect of the matching process, as it allows groups to determine whether an event attendee will be able to fill a required role on the project team. Event planners are also encouraged to keep their profile up-to-date, so attendees can learn more about the person coordinating the event. The user interface mockups in 2.6.3 illustrate how users may update their profile information and also view the profiles of other users.

2.3 Project Creation and Description

Having created and fine-tuned their profile, users will enter the Event Selection page, where they can choose to enroll in one of several available events. Users can come back and enroll in as many events as they like, though they will be blocked from joining more than one group per event. A typical user will, upon reaching the My Events page, select their desired event, choose to Search projects, and start looking for a group they like. The majority of users are expected to sign up for Hinder without a group/project idea in mind. Fortunately for this majority, some users will have one in mind: Hinder’s Project Creation interface allows users with bright ideas, or just a penchant for organization, to create teams instead of joining. The Project Creation page lets organizers provide a name, add a cover photo, set the desired size for their group, add a short description, specify desired skills for their project, and finally associate that group with an event.

2.4 Project and User Interaction

Once a user creates a project they are going to want to fill up their team with qualified team members. Likewise, a hackathon attendee may want to browse teams looking for members and join one if the project seems interesting. The interface of interactions between the attendee and project is going to be based off of the Tinder interface. If an attendee likes a project or vice versa the user will swipe right on the profile of the project or user. If both the user and project swipe right on each other, there is a match! The user gets added to the group and they can

contact the project manager and start working on the project. If they do not like the project or user, then they will swipe left and can keep browsing. When you swipe left on a project or attendee you may see them again as they will be added to the back end of a queue. This feature is necessary so that if a project needs another member that they have previously swiped left on, or vice versa, they have a chance to reconsider their decision.

2.5 Mediator Role

The Mediator Class will serve as the main logic handler for the swiping interaction. Each Event Class contains a Mediator Object which will record every swipe into a database. Swipes are registered via one of two functions: userInitSwipe and projectInitSwipe. Each right swipe will get added the database in the form of a tuple: (projectID, userID) or (userID, projectID). Depending on order, when a swipe tuple is added it will search for a reverse tuple pair already in the database which would signify that two parties both right swiped on each other. At that point the user would be added to a project and the project would add a new user. In this fashion, the Mediator will handle all database calls (SQL queries) and match-making logic.

2.6 Usage Scenarios

As an event organizer, I want to market my event to a large crowd of capable students and hackers. I also want to keep tabs on the status of my event, to make sure my infrastructure can handle enough users and groups. First, the organizer opens the app on their iOS device, and registers as an event organizer. They will immediately be redirected to the Event Creation page, where they provide the name of their event, a brief description, a cover photo, and the time, date and location of their event. As soon as they tap the save button, their event will be created, and made visible to all prospective hackers using the app. Once the event is made, organizers can continue to check on it by signing into the app. The Single Event page, from the organizers point of view, shows the details they provided at creation, as well as the current number of users and teams registered.

As an event attendee, with a specific project in mind, I want to state my concept clearly and attract other hackers interested in joining me. I also want to be able to dictate which skills I am looking for, and match those skills with my prospective teammates. This type of user begins similarly to the previous user, but registers as an event attendee. Next, the Profile Creation page, where they provide details about themselves such as name, occupation, profile picture, and their relevant skills. Once finished with their profile, they are sent to the My Events page. Initially it suggests events for them to join; later, it will show them the events they have already registered for. To create a new project, they can (a) choose an event to register for, and select ‘Create a project’ under their event, or (b) they can open the Menu, visit the My Projects page, and press the plus icon to create a new project. After providing the necessary info (project name, picture, event, desired group size, and desired skills), their project will be created. From its Single Project page, they can view their current members, or browse the event’s current attendees, swiping right on them, to indicate their interest, or left, to indicate disinterest.

As an event attendee, looking to join an existing group, I want to find a project that fits my interest and to display my abilities so managers can find me. These users open the app, register, create their profile, and specify their skills identically to project managers, diverging when selecting an event to attend. Instead of choosing ‘Create a project’, they choose to ‘Search projects’ under their desired event. Projects registered with that event will be presented to them in a deck style interface. They can swipe right or left on a project. When project and user have both indicated interest for each other, the user is added to that group. The user’s email address will then be shared with the manager, and the two can coordinate the structure and expectations of their project. If they decide the match doesn’t fit, either can abort it at will.

As an event attendee, with a specific group in mind, I want to be added to an existing project (I’ve spoken with the manager in person) but I don’t want to have to search through every possible group. To be added, users register and create a profile same as other event attendees. The manager of the group simply needs to know the email address they registered with. That manager will open the app, go to My Projects, open the project in question, select ‘Members’, and invite the member in question by entering their email address. This allows teams to coordinate outside the app, and still have their project’s info stored in a friendly interface.

2.7 Diagrams

2.7.1 Class Diagram

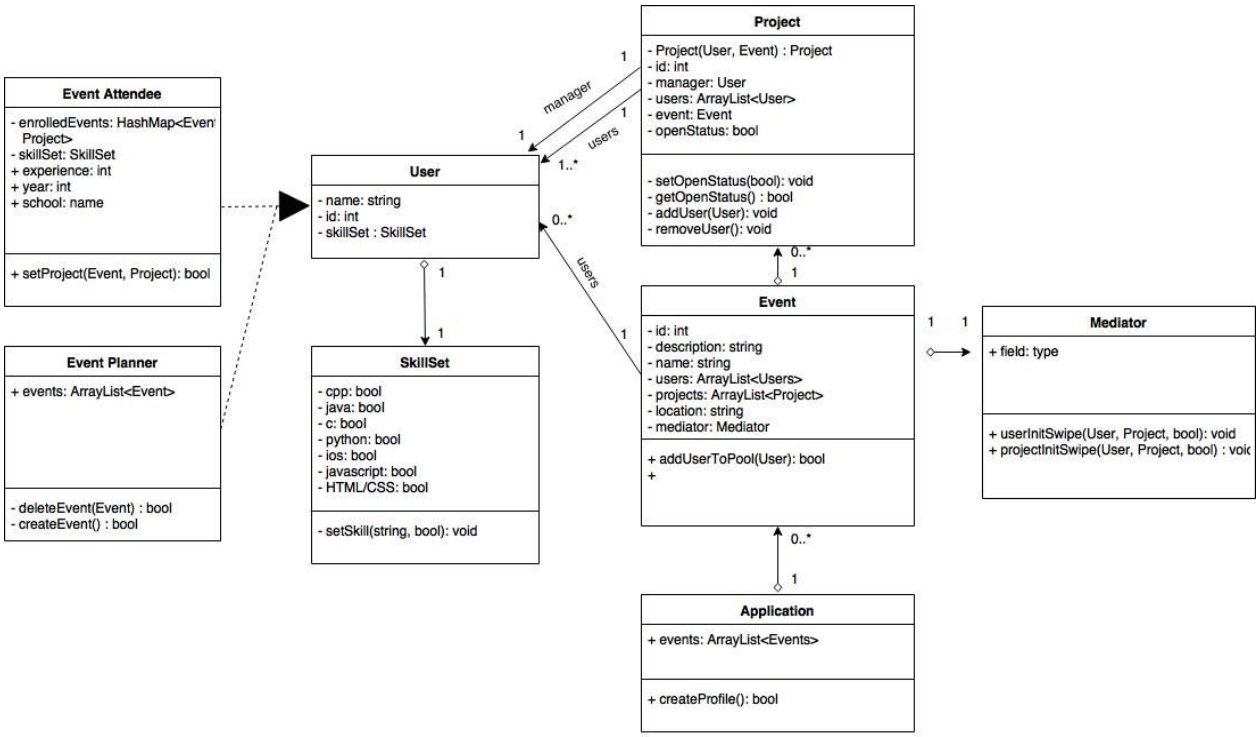

Figure 1: Class diagram for Hinder. The diagram captures the notion that, in Hinder, there are users (attendees and planners), events, and projects. Attendees browse events and projects of interest that meet their skillset. The Mediator performs the Tinder-like matchmaking.

2.7.2 Use Case Diagram

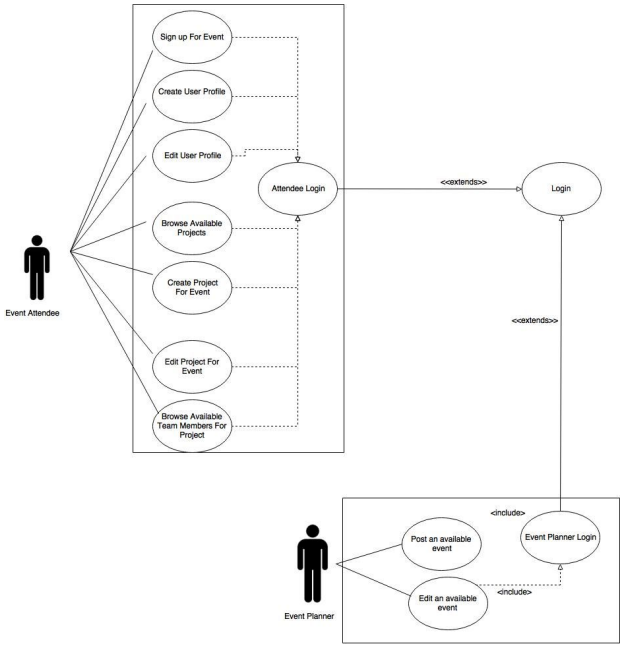

Figure 2: Use case diagram for Hinder. As discussed, there are two types of users: event attendees and event planners. Planners create and manage events. Attendees browse and sign up for events, and search for (or create) and projects that match the user's area of interest and skillset. All users of Hinder must have an account that they must log into.

2.7.3.1 User Interface Outline (Overview)

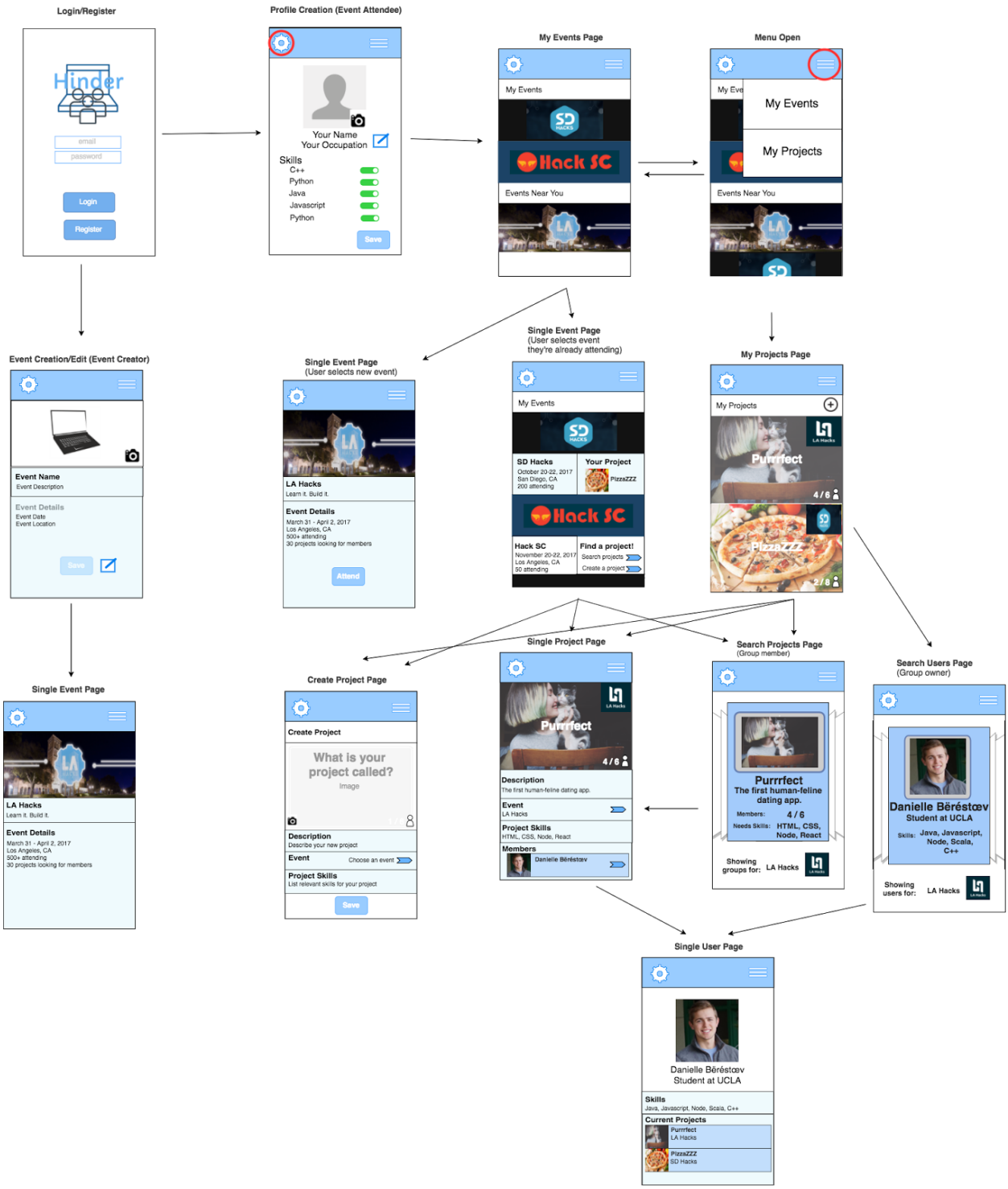

Figure 3: Overview of user interface outline. The mockups follow from the discussion of features and requirements in sections 2.1 - 2.4.

2.7.3.2 User Interface Outline (Page View)

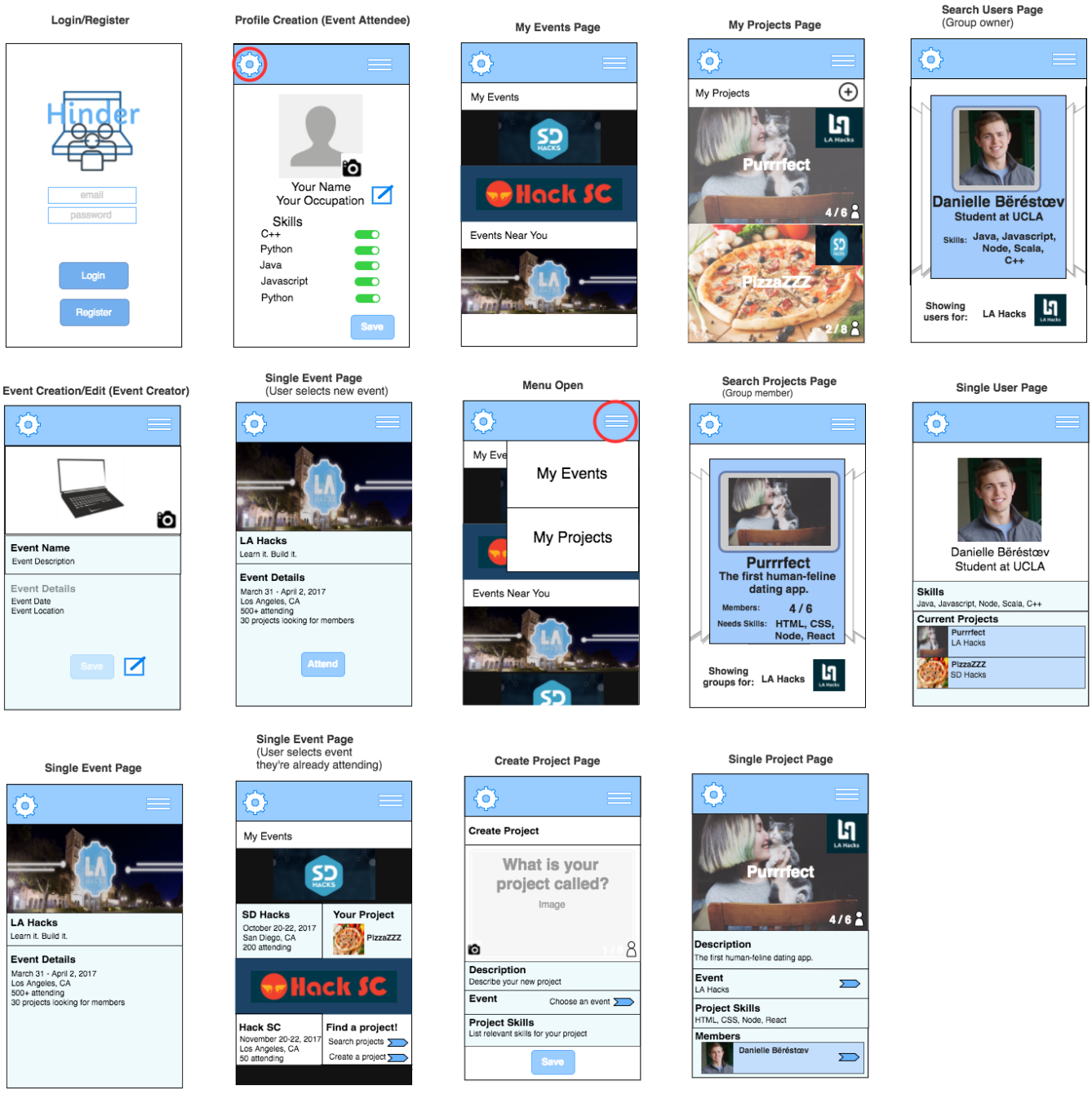

Figure 4: User interface, individual page views.

3 Feasibility

Hinder offers a service that no other applications provide: a small scale, easy way to find partners and teammates around you for hackathons and class projects. The current method is word-of-mouth, attending a matching event or posting in Piazza for partners. None of these methods are particularly effective, because before making a successful match you need information about the person's skills and experiences, and these can take time to learn and find. Hinder will offer users the chance to filter out candidates based on skill and experience, so they can find the most effective partners in the most effective way.

The application is feasible, because we have narrowed down our MVP to key features. Users will be able to create an account, create an event, and then begin searching for candidates, which will be shown to them based on their preferences. The features are all streamlined to deliver the best possible experience, in the most feasible manner. Many members of our team have strong familiarity with Objective-C and Swift, so building this as an iOS application will be doable, because we have passed the learning curve. We can use location data to ensure that only users within a specific geographic radius can be considered or join a project. Most of our work will rely on tools internal to Swift, so we will not have to depend on any unreliable APIs. Apple has a Core Location Data framework that we can use to correctly display candidates that are nearby. The documentation is here.

4 Capability

George worked as a software engineering intern at ITG this summer. Primarily worked in a web environment with node.js and javascript. From school projects, I have some experience with iOS development and primarily swift. This experience will help to build out the Hinder class hierarchy and inner app structure. I hope to gain a more holistic understanding of the iOS development cycle. For Part A, George helped create the the Class architecture and implement the mediator design pattern into the class diagram. He also as fleshed out the swipe logic algorithm and the researched the database that would be used to execute that functionality.

Kyle worked on the back-end development of pricing and invoicing applications while working at USG. He hopes his prior development experience in Oracle and SQL Server will aid in the implementation of Hinder's back-end functionality. This project is a great opportunity for Kyle to build on his limited experience with iOS - he is excited to become more acquainted with Swift and Objective-C. For part A, Kyle created mockups, contributed to several sections of the written report, and assembled the presentation slides.

Marshall interned with Amazon this summer and worked on the Alexa Voice Shopping platform, as well as on key aspects of the retail website. Through his work on the Alexa platform, Marshall has experience inventing and navigating a complicated and abstract user experience, as well

building and optimizing client-server interaction. Marshall hopes to help craft a pleasant and intuitive experience for users of Hinder, and plans to make his OOP and networking experience relevant by solidifying the connection between backend and frontend services. Never having developed for mobile devices before, Marshall is excited to take his current skills and apply them to a new and exciting platform. For Part A, Marshall designed the logo, composed a significant amount of the class diagram and user interface mockups, and transcribed the project creation description and explicit user scenarios.

Daniel worked at Amazon this summer where he worked with database analytics. He will be working on the back end part of the application integrating database connections and other back end features. He hasn't worked with iOS extensively, but looks forward to using Swift and working with the team. For part A of the project Daniel created the use case diagram, helped in designing the class diagram and assisted in writing the feature description and requirements.

Kim worked at Facebook this previous summer, specifically on the Instagram User Interfaces and Guidelines team, which focussed on keeping the UX of the app consistent across all platforms. She wrote the backend for a new feature for Instagram, and was responsible for creating the UX for the feature herself, keeping within the Instagram guidelines. At Instagram, she mastered Objective-C, as well as several open source libraries also available for Swift. Though she doesn't know Swift, she thinks the general skills and APIs learned using Objective-C will carry over so that she is able to create the UX for Hinder. She hopes to become as comfortable with Swift as she is with Objective-C by the end of the project, and become comfortable with larger scale projects, rather than adding a small feature to add already-big app. For part A, Kim helped map out and create the class diagrams, created a sizeable portion of the UX mockups, and wrote part of the feature description of the report.

Apurva: Worked as a software engineering intern at Symantec and Google. At Symantec, worked on an web scraping tool in Python and XML for an internal search engine. At Google, worked on an iOS VR field trip application, specifically on a feature that focused on communication between the Objective-C and C++ layers. I hope to translate my Objective-C experience into an understanding of Swift and work on front-end. For part A, Apurva worked on the feasibility section and researching various APIs and related applications to make sure this was a doable, and unique project.