Marissa Stephens
Nick Mohr
Danielle Man
6.170 Software Studio: Project 3.1

Group Finder

Overview

Motivation
Make it easy for students to form teams for group projects

Key purposes

1. Allow MIT students to find people to work with for team projects.

MIT students often have difficulty forming groups for team projects. As the compatibility of the team often affects the grade a team receives, students want teammates they can rely on and therefore want to find acquaintances that are also in the class. In large classes, it's hard to know how many of your acquaintances are in the class and which of those are actively looking for partners and how to contact them. By providing the names of people looking for partners for a particular project, this app attempts to rectify that issue.

2. Filter potential students by external factors such as location, workstyle, and schedule

One of the first issues students have in forming projects is that they want to have something in common with the people they are working with. Oftentimes teams are made out of locational convenience with teammates all living in the same dorm or FSILG. Students also have very different styles of working. Some are procrastinators, some like to get things done early. Some like to work in a group, some like to work alone. Some want to commit 20 hours a week, some only want to do 5. If students can find partners with similar workstyles, then the team as a whole will have a better understanding of how it can be most efficient.

3. Help students form a team with complementary skill sets.

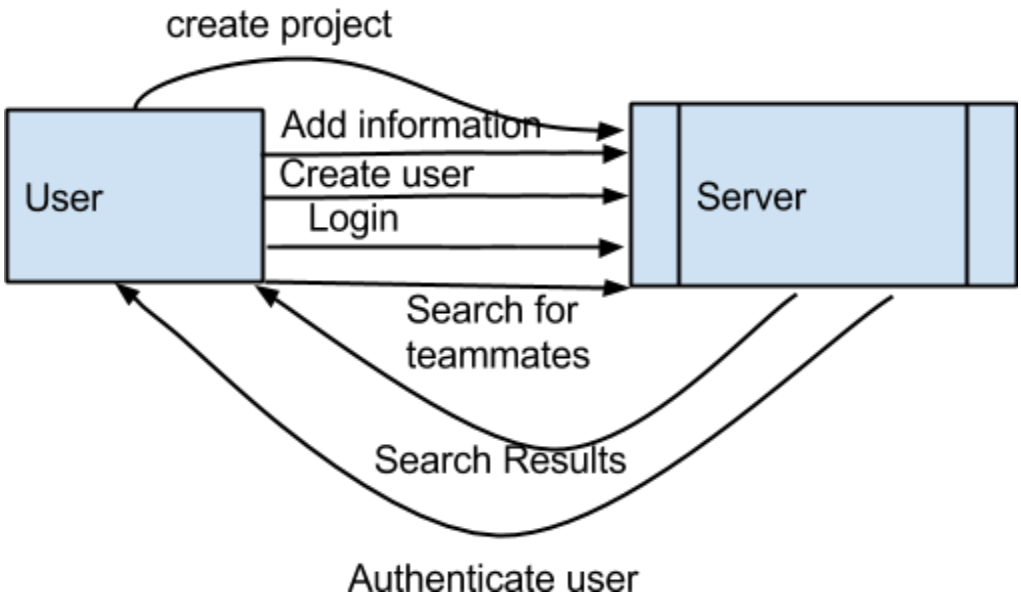
Each student is a unique person with a unique set of skills. Some are natural leaders and organizers whereas others can work hard as long as they have some direction first. Some students prefer frontend, others backend, still others testing (although very rare). A good team has a balanced set of skills, and allowing users to know their teammates' strengths can makes dividing the workload a lot easier.

Deficiencies of existing solutions

Currently there is no commonly used system for creating teams other than speaking to students who are in the class. For the most part, students prefer to work with people they have some previous interactions with outside of the class rather than complete strangers. In large

classes,however, it's often hard to know which of your acquaintances are also taking the class. Previously, there has been an app called Setup that allowed users to find pset partners that have the similar study styles. In a group project, you need a variety of skillsets to be optimally productive. Classes also allow students to fill out a google form (if they were not able to find a team on their own) and allowing the class staff to assign them to other teams. I witnessed this in 6.005: people were writing Kerberos names on the board and yelling to find other people who they had never met before but were now about to work on a big project with. I have described the deficiencies of this system above, but it mainly boils down to leaving something extremely important, which is group dynamics for a big class project, to chance. For this project I also saw many sponge-talk emails to my dorm, where people were asking for one or two more group members for this project and just hoping that someone would respond. We would like for this process to not have to be left to chance.

Context diagram

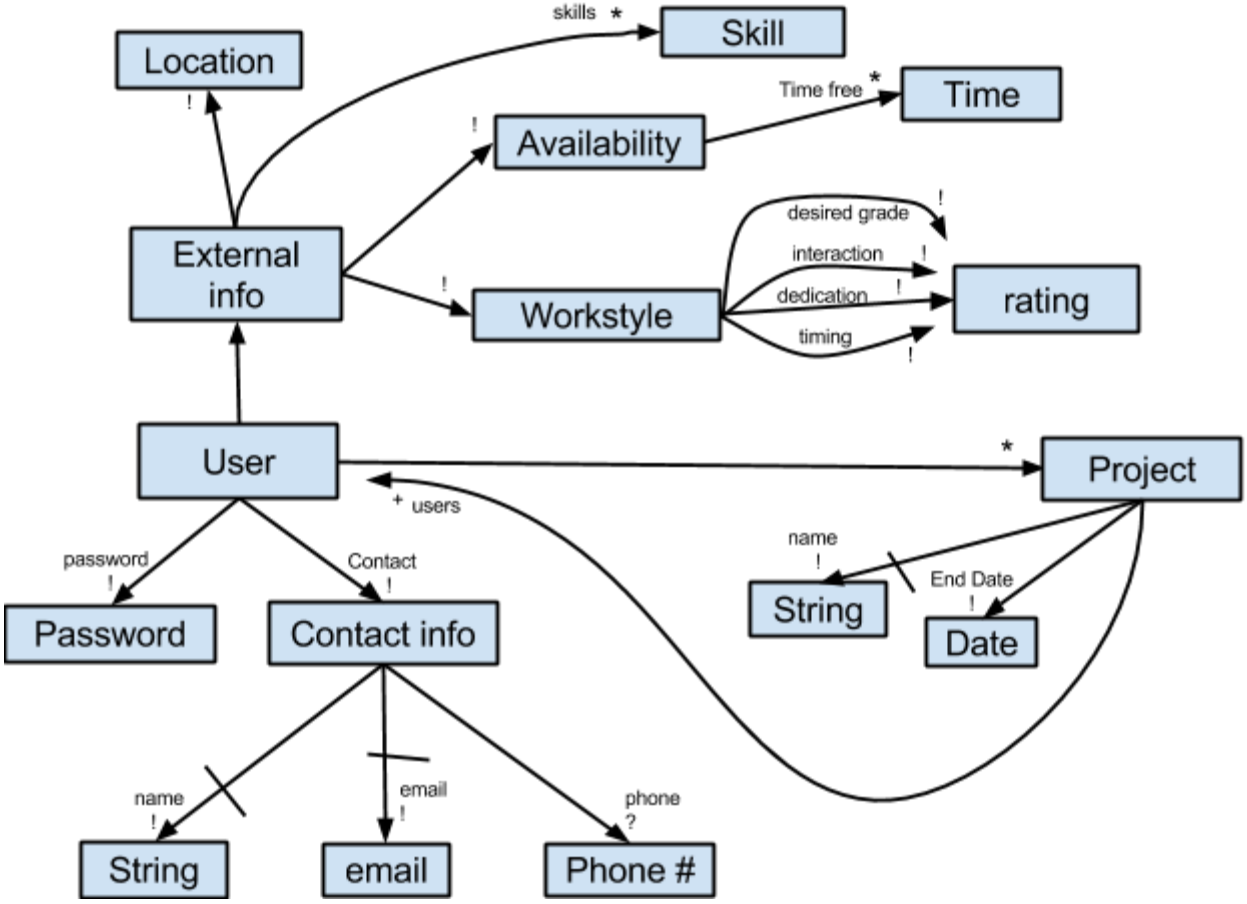


Design Model

Concepts

Project: created by a user such that it can be accessed by other students
Filtering by external factor: Users can prioritize other users that have specific external factors
Matches: Users that have a desired external factor
External Factor: something such as location, workstyle, availability, or skillset that a user might want their teammates to have.
Skill: A student can have some specific skills that will help other people select based on skill sets

Data Model



Challenges

- Should a user contain a list of projects or should a project contain a list of users? A user containing a list of projects makes it easier to sort users based on their projects. However, a project containing a list of users makes it easier for a group of users to be associated with the same project, so if project information changes then it only has to change in one place, which is why we decided to give project a list of users
- How do we get rid of old projects? One solution is to never get rid of them. That was unfavorable since that could lead to a lot of unused, expired projects clogging up the database. Another idea was to have a specific end date, say the date the project is due. This was obviously preferable so that led to the question of who gets to set the end date. Either the first user who created the project gets to set the end date, which could lead to an unfortunate case where the user sets it too early or too late, or we automatically delete a project if it has no users actively searching for a partner. This is problematic since if the users do not all notify the database that they are no longer looking for a partner, then we have the same issue as if we never got rid of any project. Therefore we decided to allow the first user to set a project due date at which time the project is deleted.

- How do we create projects? One solution was to get a list of all MIT courses and then allow users to select the course for the specific project they were trying to build a team for. However this does not take into account the fact that students might have several group projects in a course or the fact that not all group projects will be associated with a course-such as a hackathon team. Therefore, we decided to let the users create projects, or select a pre-existing project. While this does allow users the chance to create several similarly named projects, it was decided that the benefit of extending the definition of group project far surpassed the error that could occur between the mouse and the keyboard.
- How do we authenticate users are from MIT? While it would be nice to use certificates, some students might not want the app to access their certificate if it is not safe, which would reduce the appeal of the app. Instead, we chose to allow users to login with thier MIT email, which would then also be used as contact information and still checks that they are related to MIT.
- How do we retrieve MIT data, like class numers and dorm locations? We could hard code this ourselves in the app, we could ask users to input it as they go, or we could pull from one of IS&Ts resources. The cleanest way would probably be to pull from IS&Ts resources, but that would expand the scope of the app quite significantly. Else we would have to ask for a lot of information from users, which significantly takes away from the experience of the user.