

P(arty)Set

6.170 Project 4 Problem Analysis

Jessica Andersen, Carolyn Holz, Nick Mohr, Hyungie Sung

PartySet is a service that helps students in the same classes come together to work on their problem sets. These meetings, called *pset parties*, will be shown to relevant students who might wish to join in. PartySet will reduce the efforts needed to meet fellow students to pset with one another.

Current web applications that try to address the same concerns are often unused. These services require a heavy log in process and setting up a profile is often deemed not worth the amount of time and effort it requires. Instead, students find it easier (although more limited) to set up pset groups by word of mouth.

Purposes

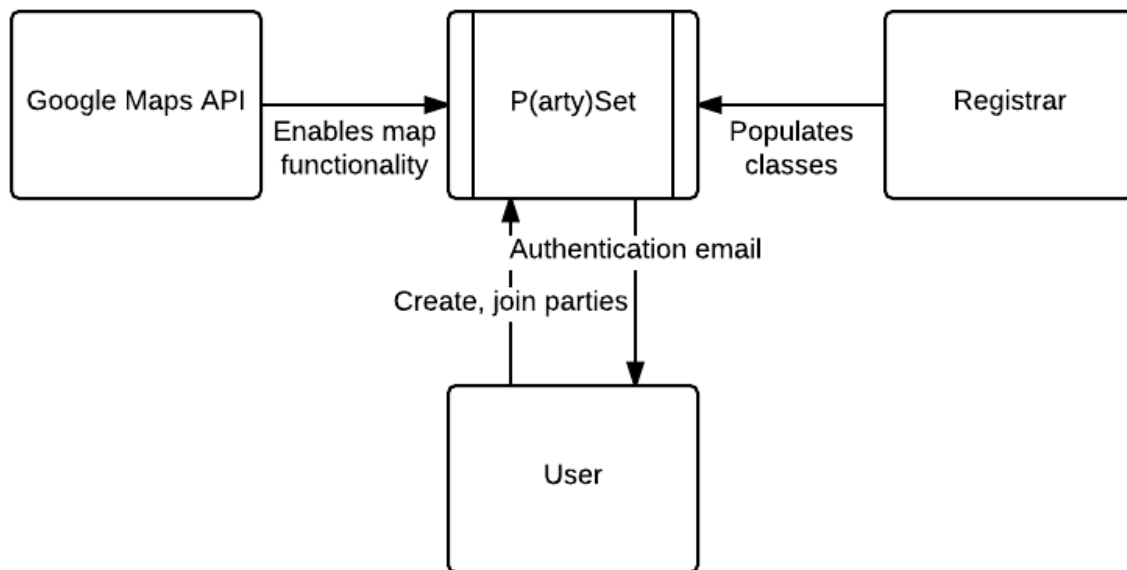
- **Connect people in the same MIT classes to pset together.** Students have missed many opportunities to expand their academic and social circles, not getting the maximum benefits from the environment around them.
- **Help people physically find one another.** It is difficult to plan out where to meet on a campus as big as MIT's. A map with updated locations of events will minimize these efforts any student has to make.
- **Increase effectiveness of MIT students.** Students have wasted much time and efforts in working on psets alone. A combined effort will result in greater efficiency and effectiveness in how each student spends their time.

Concepts

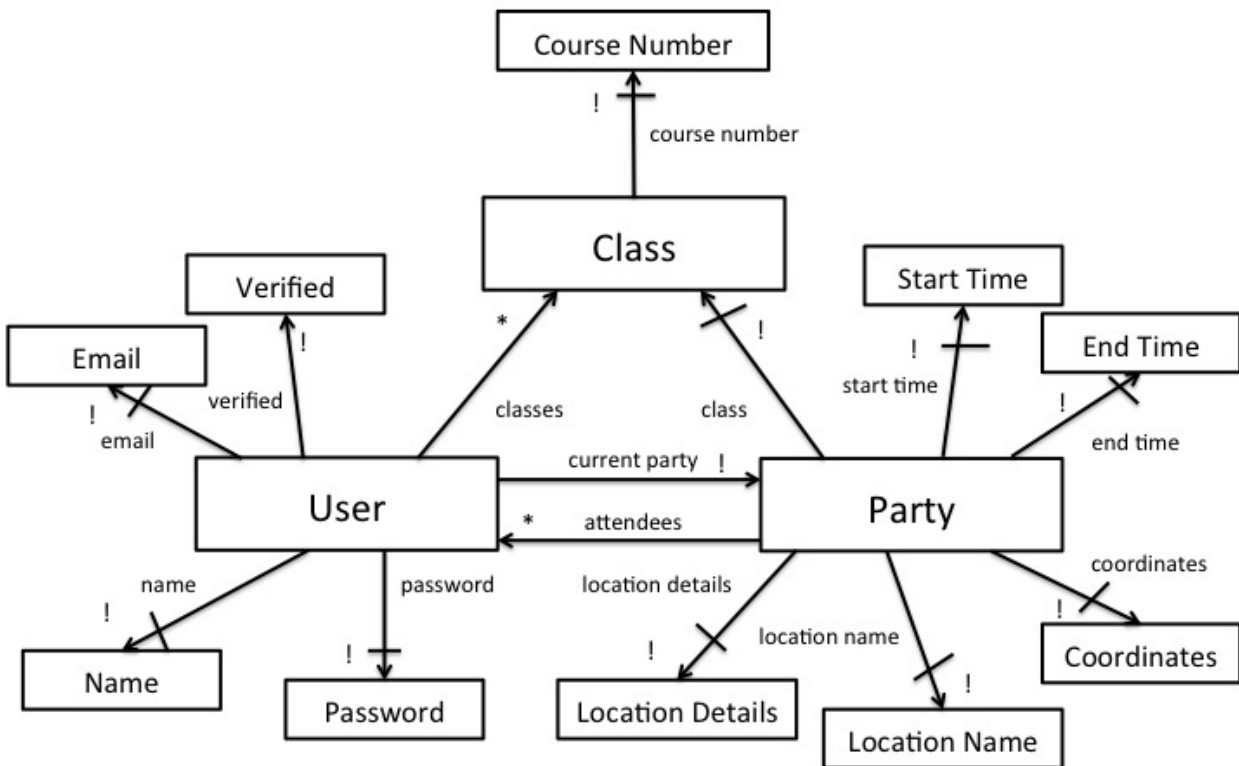
- **Party** - A party is a group of MIT students psetting together at a certain location for a certain class. By having parties, we allow MIT students to search for the classes they are in, fulfilling the first purpose of connecting students in the same classes, and to figure out where people are, fulfilling the purpose of helping people physically find each other. Parties are necessary in order to help people physically find one another.
- **Attendees** - An attendee is someone who is going to a party. This concept is important because we want people to use this as a social networking site, and be able to choose to attend a party based on how many people are there or which people are there. Attendees are necessary in order to connect people in the same MIT classes to pset together.
- **Class List** - The main criterion by which a user finds a party is which class it is for. They will not want to view parties that are not for the classes they are taking. Therefore, each user has a class list of the classes they're currently taking at MIT that

they can delete or add classes to. They'll also be able to filter which classes they see on the map further from that list. Class lists are necessary in order to connect people in the same MIT classes to pset together.

Context Diagram



Data Model



Security Concerns

The security requirements are that:

- Only MIT students are able to view existing pset parties.
- Personal information about users are kept hidden from other users. The only information that should be displayed regarding a user is possibly the location of where they are (and even then, the location is not directly mapped to a specific user).

Potential risks include:

- Fake users signing up for the service for purposes aside from setting up pset parties.
- Malicious users who create nonexistent pset parties in order to distract and deter fellow users.
- Using information about popular locations on campus for evil purposes.

Threat model:

- Can assume little to no interest from criminal figures as only minimal personal information (name and email) is stored.
- Fake users are very likely.

- Impersonating another user is unlikely because there is no benefit to being one user over another.

Mitigations:

- The only communication directly between users is in person. The only other communication is in communicating the location of the party.
- An email authentication system will be established to make sure that users are valid users through a confirmation email.
- Prevent alterations of parties once they are established.

Design Challenges

HTML map element vs. Map API vs. Textual representation:

- The HTML map element will most likely give us the most ability to do whatever we want with the map without extra bulk on top of exactly what we want to do. The main con for the map element is that we don't really have any base functionality and we're building from nothing.
- The map API could be good because a lot of the features we want to make are already built, we would just need to change them slightly for our uses. The main con is that it has that extra bulk on top of it like zooming in and out when we just want to show the MIT campus (just as one example).
- The textual representation would be the simplest when it comes to design/coding time for us. It also kind of defeats the purpose of helping people physically find each other since maps are particularly good for that.
- **Decision:** We decided to go with a map API for this. We felt that while we couldn't do an overlay well in a map api, we could do pins very well and we decided that was what was much better for readability of users. We also think that letting the API do the pins part of the map makes our implementation job really simple.

Dynamically updating party info on the map: Refresh button vs. reactive data model:

- The refresh button would be the simplest to implement because it would basically just refresh the map part of the website. The really bad part about it is that this slows down the users a lot and the user never knows when refreshing is useless or not. This would lead to an incredibly discouraging user flow for the user (refreshing over and over until they find something).
- A reactive data model would take more development time but would make it much easier for a given user to find what they are looking for (or quickly find that there is not what they are looking for).

- **Decision:** Since the refresh button is so easy to implement, we heavily considered it but in the end we decided it's crucial for the user experience to have the data be pushed to them. We decided on having a reactive data model.

Having users enter their classes when they join vs. Having it as an option:

- This is a traction issue that will be essential to gaining new users and making sure they stay with us. We want to streamline the process so that a user can get in quickly but also use the better features of our app quicker.
- As a new user no one wants to fill out a long profile (adding all of their classes) but with these data the map we show will be much more specific to what our user actually wants to see.
- If we have it as an option our users may never even notice the option and think that our app is really shallow (opposite of deep here) and will not be actually useful to them.
- **Decision:** We decided on a compromise. Rather than making the user slog through making their classes all at once without any results until you submit all your classes, we're going to have a tab on the main page to add classes. When you first load the page no pins will show up but we will point you towards the plus button which you can use to add a class you are a member of.

Satellite pictures vs. Google map pictures vs. Map distributed by MIT:

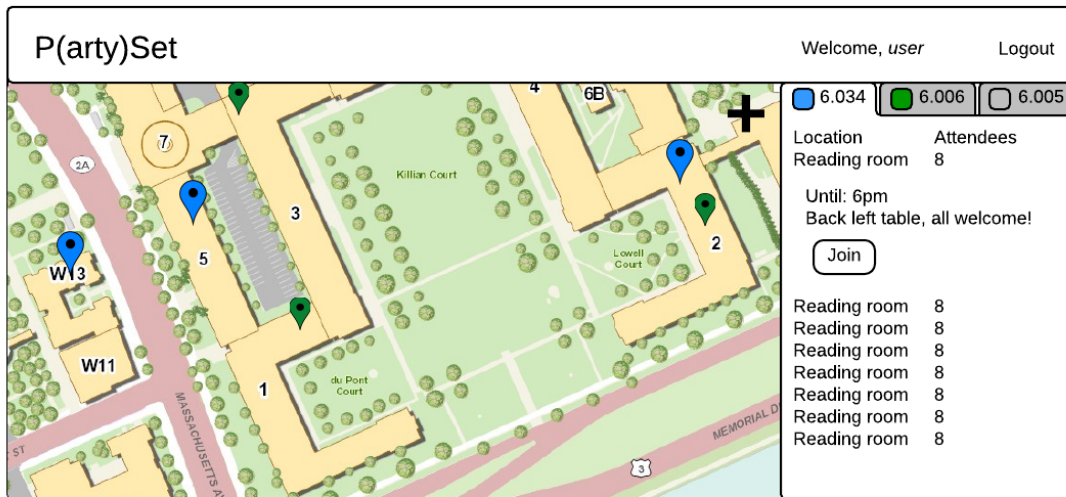
- Satellite pictures would not be labelled nicely and they wouldn't have any separation from building to building making it hard to use for new students who don't know where all the buildings are.
- Google Map pictures have labelling and would be easy to use but the labelling is arbitrary. Some buildings are labelled with building name some are labelled with numbers.
- An MIT map would be very nice but would take the extra work of converting it to whatever form our map api likes. We would also have to tile any picture we want to use if we want zooming to be fast.
- **Decision:** We decided on an MIT map because it makes the labelling of buildings much better which is important for the user because often they need to know what building to label their party as being in. It is also possible to use the google api without using the google maps which is nice.

Authentication - Certificates vs. Confirmation Email:

- Certificates have the potential to be easier for the user. One ok button and we know who you are. One downside is that it's much harder to use it on mobile. It may also be harder to implement because we have to figure out how to identify certificates
- Having a confirmation email to confirm identity would be a bit harder, because the user has to navigate away to even use our service. However it is easier for us to implement and makes the data model simple because we just have a small field of whether or not the user is confirmed.

- **Decision:** we decided on using confirmation email. While it would be nice to use certificates, it's not essential to our goals to make absolutely sure people are who they say they are. Instead we'll just check for an mit email address and that they click on the confirmation link.

Mockup (with standard login page)



MVP Wireframe

