

Overview

In J.K. Rowling's series *Harry Potter*, a Marauder's Map is a magical object that lets characters automatically see the location and movement of other characters on a map. This allows whoever owns the map to both seek out people they want to find, and avoid others they don't want to run into.

Often at MIT, students find themselves in the first scenario where they would like to find a friend for collaboration, event hangouts, or other purposes. We would like to create a "magical" Marauder's Map for MIT, to allow users to find friends nearby by publishing their locations and seeing the locations and statuses of other users.

Problem Statement

Though students would like to know where their friends are, it is often hard for a user to collect this information on their own. Current solutions that exist for meeting friends include the methods in the following table:

Current Solution	Downside
Calling or texting friends	The user has no way of knowing who to contact first or who might be in the area. MIT students are also often reluctant to be the one to reach out first.
Planning ahead	At MIT, people usually do things at the last minute.
Foursquare	Exists mainly to broadcast where you have been, not where you currently are.
Facebook	Not a clear way to see which friends are near you.
Swarm	Does not include a map and is for everyone. Marauder's MIT will be more visually useful and only for MIT students
WeChat	Not enough privacy: allows everyone to see where you are which is creepy

Purpose and Goals

Our app is designed to retrieve users' locations and to broadcast this information to their friends in an easy-to-read format. The key purpose of this app is to let a user know which of their friends are in a specific area at a specific time. The user can then get in contact with these specific friends to meet up with them. This solves the problem of not knowing which friends it would be most worthwhile to contact first and saves the user time and effort.

Essence of the Design

Design Criteria

Safe:

Only available on MIT campus, only available to friends, only available with certificates

Scalable:

As more users join, the app needs to remain usable. To solve issues created by crowding, we can use the following approaches:

1. To make the visuals functional, we can enable a zoom-in feature on the map.
2. To help find a certain user, a search by name function would be useful.

Intuitive:

To encourage people to join and continue using the app, we hope to make the user interface as intuitive as possible by using pre-existing tropes. For example, users will be able to zoom in on the map with the customary finger-spreading motion. Green will encode the available status, and red will encode busy.

Accurate and Up-to-date:

A user's location on the Marauder's map should reflect his or her precise geo-location within 10 feet of the user's actual location, within the last minute.

Key Concepts

The main concepts we plan on implementing for our application include the following:

Apparition

When a user checks in, he or she is known to have "apparated". This concept allows users broadcast to their friends where they currently are, or where they plan on being in the

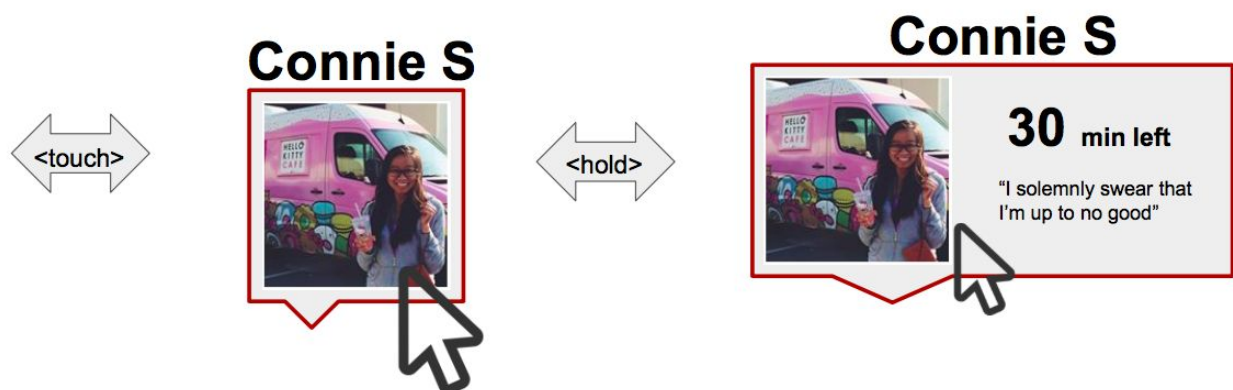
future. The main purpose of the Marauder's MIT application is to broadcast where a user currently is, and the concept of Apparition allows us to do this.

In order to apparate, a user needs to allow the Geolocation API to access his or her location. If this location is within the borders of MIT campus, the Marauder's MIT Map will display the user's profile picture at that place. After implementing our MVP, we will consider adding a "time-travel" feature which allows a user to manually check in to a location they intend to be in the future. The map would then have the option to show where users intend to be at a certain time in the future and a user can use this information to make plans.

Friends

Due to the fact that people may not want everyone on MIT campus to know where they are, we are introducing the concept of "Friends". Users can send a friend request, accept or ignore a friend request, and delete other users. Exchange of location information is essentially limited to between friends. This means that when a user apparates to a location, this is shared only with other users on their friends list, and when a user checks their Marauder's MIT map, they can only see the profiles of where their friends are located. The motivation behind the friends concept is to allow users to know and choose exactly with whom they are sharing their location information.

Status

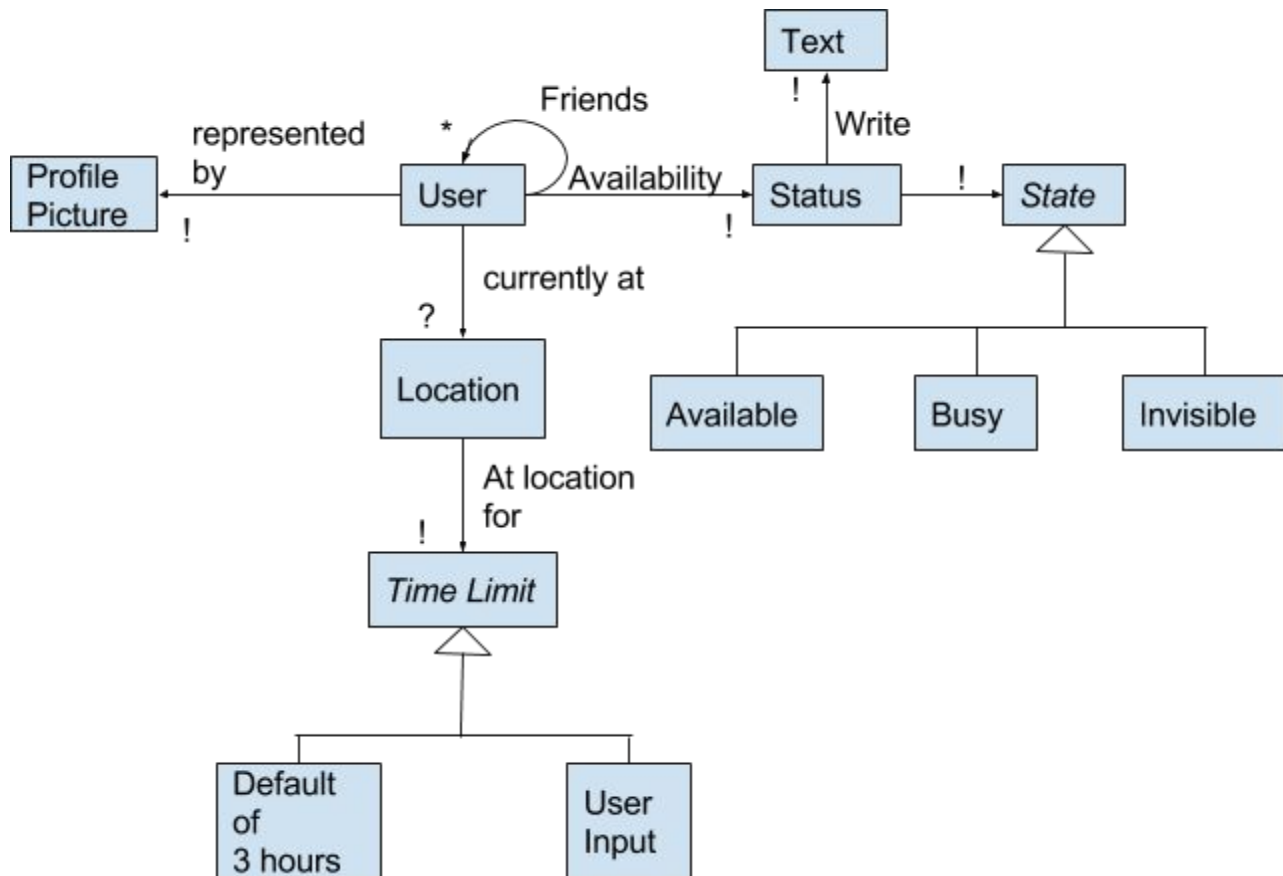


MIT students are known for their collaborative spirit, and students often love working together on psets, even if it's not necessarily the same pset. Students may want to distinguish between broadcasting their location in order to hang out, and letting people know where they are if they want to work together. Because our application is automatically tracking a user's location, we will also have an "invisibility cloak" feature in which a user can choose not to broadcast their location at all. Thus, a user will always be in one of three states - available, busy, and invisible. In addition to a state, a user's status will include a text, which is defaulted to

“I solemnly swear that I’m up to no good”. A user has the option to edit this text to show a bit of personality or elaborate on their state: e.g. “working on 170 pset, anyone wanna join? :)”

When user touches an image, the image enlarges and the friend’s name appears. When the user continues to press and hold the image, the status and the time left that the friend will be at that location for will also show up.

Data Model



Security Concerns

Since our app involves storing and broadcasting the locations, we need to make sure this information is not leaked or abused. To mitigate this, the first security technique we need to implement is storing a user’s information in a protected account. Logins will be done with MIT certificates to ensure that this app is only available to members of the MIT community. The concept of friends is also designed to protect information. A user’s location will only be broadcast to their friends, and becoming friends with someone is a mutual agreement. This should prevent a user’s information from getting leaked to potentially malicious people. Users

can also remove friends that they no longer want to broadcast their location to. If users are constantly being harassed/friended by the same people, they can report or have them blocked from the site.

Standard Web Attacks: The majority of the functionality of our site does not require the user to input text queries, so many of the standard web attacks are mitigated. The only time text input is available is when users input text based statuses, and during login. During these times we will make sure we to sanitize all text inputs and check for buffer overflow issues.

Threat model:

Unauthenticated users can add/remove another user's friends, or access location information that they should not be privy to. They can also tamper with users' locations to display inaccurate information.

Users with basic credentials can stalk users that they are friends with or harass users by repeatedly trying to friend them.

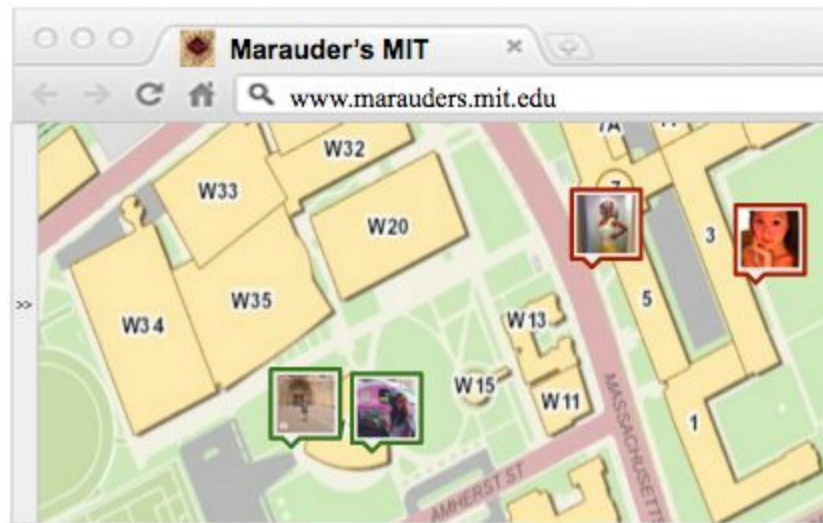
The only possibly sensitive information stored by the application is name and location, so it is unlikely to attract interest from hackers.

User Interface

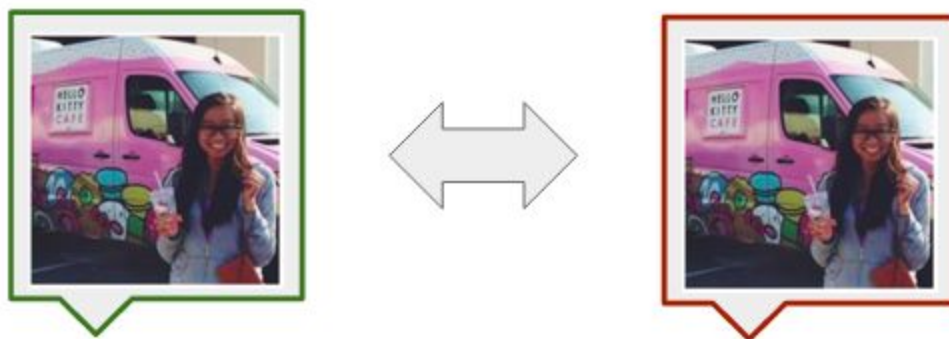
Our application will be a single-page mobile-friendly website. Because there is no transition between screens, we've decided to make screen prototypes instead of transition diagrams.

There will be two main screens: one screen for log-in and register (not shown) and the home-page map shown below. The main functionality of the home page is detailed below.

Main Screen:



Users are visible from a bird's eye view on the map.



The user updates their status from "available" to "busy" which is displayed with either green or red.

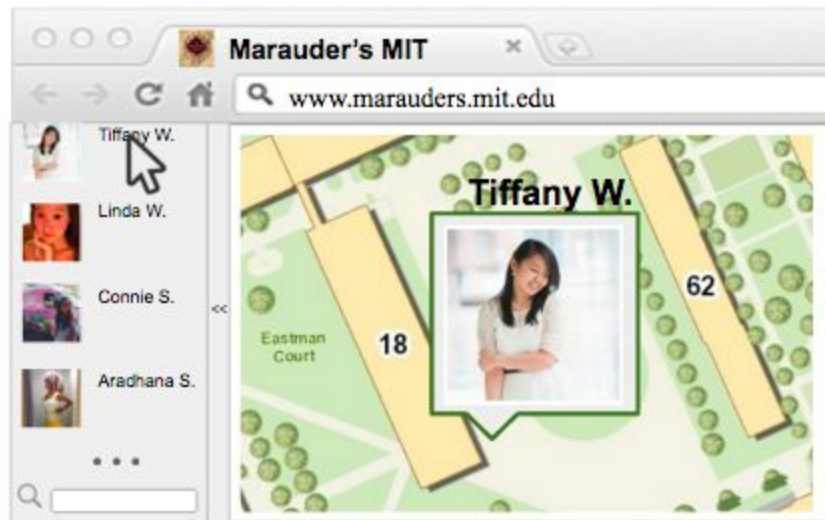


By clicking or pressing on the user, the picture enlarges and the user's name appears

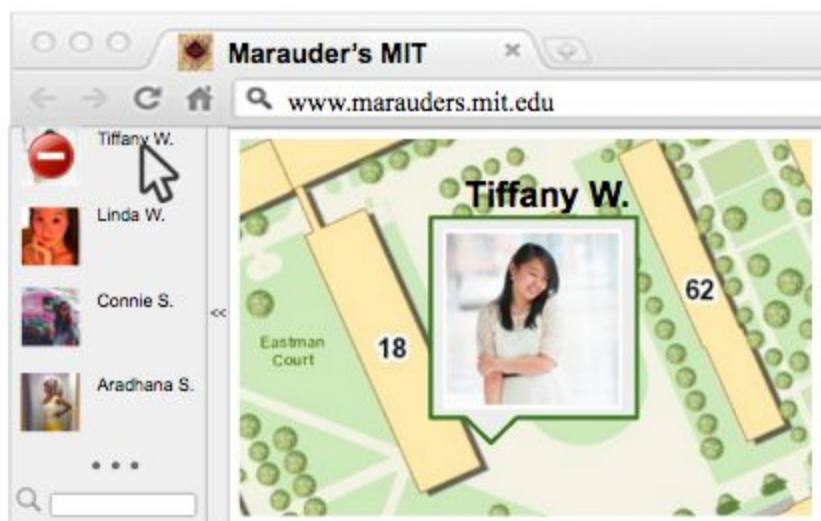


By holding down on the user's image, more information about the user's status and how long they have left at their location is displayed.

Finding Friends:

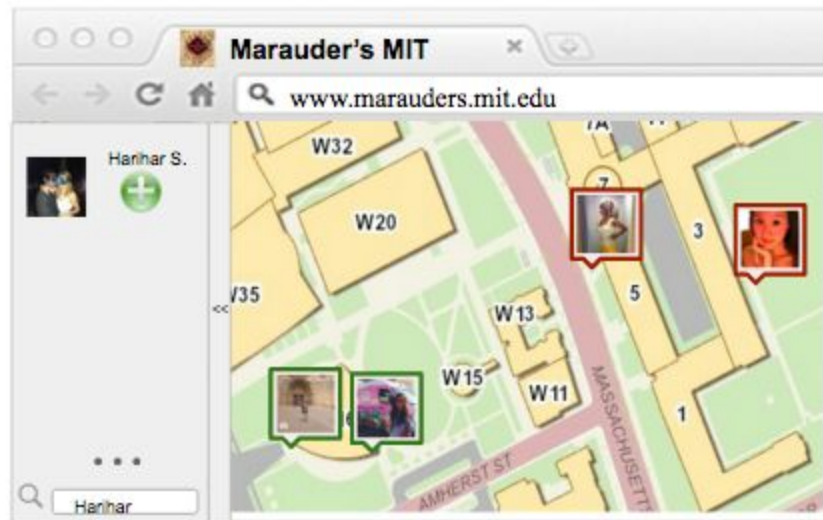


To find a specific friend on the map, the user may navigate to her friends list and click on a friend. The friend will immediately come into focus on the main map.



By holding down or double-clicking on the friend, an option appears to remove the friend from your friends list

Adding Friends:



The user is able to search for a particular name. If the user is not friends with the named person, an option is given to add the friend (which will require that person's approval)

Design Challenges and Mitigations

User and Concept Issues:

Achieving Critical Mass of Users

The success of this app hinges on there being enough users to make joining the app worthwhile. This means that joining the app should be simple and hassle-free. Additionally, as an app designed to help people connect with each other, it would be useful to let people invite their friends to join.

One major challenge will be getting enough users to join our app. We could accomplish this in a number of ways: with marketing, with incentives, or by integrating the app with Facebook. We've decided to integrate the app with Facebook because we believe it will be the best way for the application to go viral and requires little effort on our part. We hope that as facebook users see their friends using the application they will see the benefits of joining and will sign up for the application as well.

Stalking or Harassment

To ensure the safety and comfort of users, we will enable users to:

1. Share location with only their friends. This allows a user to know and edit exactly who they are sharing their information with.
2. Block certain users anonymously to prevent receiving repeated friend requests. This will prevent unwanted attention.
3. Report harassment from other users. If a user is reported for harassing behavior, site managers will contact the user reported. Repeated offenders will be blocked from the app.

UI Issues:

Clown Car Problem

To solve issues created by crowding, we will use the following approaches:

1. To make the visuals functional, we can enable a zoom-in feature on the map. In a fully zoomed out map, the map will display a certain user and a number indicating how many other users are also there. For example, "Connie Siu + 10 others".
2. To help find a certain user's location on the map, a search by name function will be created.

Elevation Issues

(Feature beyond MVP) To help discern which floor a user is located on, we can create different levels for the map for each floor. Broadcasting location by floor could be achieved by manual check-ins or by using the phone to determine which MIT wifi router it is currently closest to.