6.170 FINAL PROJECT DESIGN

Jointly authored by Lucy Yu, Chongyuan Xiang, Yang Chen, Andre Mroz

1 | OVERVIEW

1.1 PURPOSES AND GOALS

Brief description of system to be built

DOWN2GO is an application for planning day itineraries with groups of people. We will build an app that aids groups of people to collaboratively create a schedule for the day. The app will feature a map, shared within a group, that users can put pinpoints on, which will update an agenda. There will also be a feature to chat with users in the group, and a feature to evaluate the scheduling feasibility of an agenda (whether or not there is time to visit some place given its distance to the previous and subsequent event).

Key goals and purposes (what problem does it solve?)

Often, when a group of friends wants to create an itinerary for the day (of places to visit), they need to use multiple mediums of communication (such as cell phone, email, internet chat, etc), as well as separate location services (ie. Google Maps, Apple Maps, etc) to communicate, visualize their events, and contextualize their events within the day. When planning a day in a group, one has to make some considerations:

- Who do I want to do these activities with?
- How will we communicate with each other in an organized way?
- How can we give everyone the opportunity to give their input?
- What if people all contribute different ideas? How will we know what the current state of the plan is? Is it immediately obvious if some ideas are not compatible (due to time conflict/locations are too far from each other) so that we can resolve these problems in advance?
- Does everyone know where a certain place is located?
- Does everyone know when we agreed to meet at a place?
- How do we keep our plan relatively fluid and open to change?

Keeping track of all of these locations and timeframes becomes an unnecessary inconvenience. The main goal of this app is to abstract all these concerns. More specifically/modularly, DOWN2Go aims to:

- Remove the need for a person to remember who is in their group and make sure all information is communicated to everyone in the group simultaneously.
- Remove the need for people to research where places are/record or memorize times, especially if they change.
- Allow users to view the places in question on a map (visual form) as well as in a time-organized list (text-form).
- Allow users to quickly identify if visiting a certain sequence of locations at some specified times is a bad idea (because they are too far apart/close in time)
- Allow users to do separate tasks -- plan and discuss-- without switching windows/apps.
- Provide a pleasant and easily-usable interface to do all of these things.

Motivation for development (eg, deficiencies of existing solutions)
Some existing solutions to this problem include the following:

1. "Roadtrippers" (https://roadtrippers.com/)

   Roadtrippers also has Google maps integration but they focus on the idea of a roadtrip (ie. the first thing you are required to fill out is a start and end destination). This is not what we desire because we want the definining identifier to be the group of friends, not the geographic path. There is also no obvious "group" (cabal) sharing concept on Roadtrippers and it is more focused on an individual creating a plan.

2. Tripomatic (http://www.tripomatic.com), Google Maps

   These are other sites that integrate Google Maps but again, there is no emphasis on collaboration/sharing or discussing with a group. Tripomatic focuses on trips for multiple days so you cannot specify specific times, only "Day 1" or "Day 2" of your trip. On Google Maps you can pinpoint places, but there is no agenda (time-sorting or time-listing) functionality.

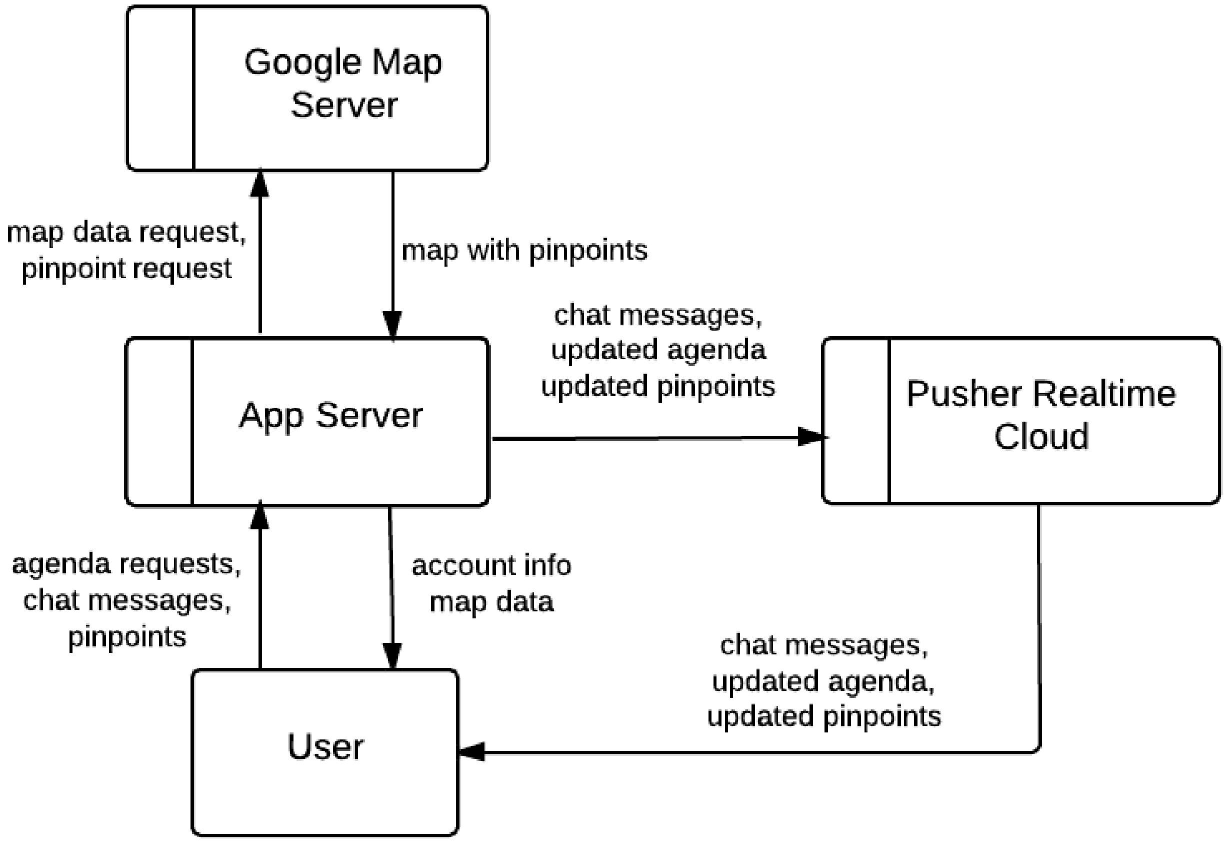
3. Email, Facebook messaging, text messaging, etc

   Most people probably default to this medium because it is so simple / everyone already has an email/phone number/Facebook account, etc. However, this can get disorganized quickly and often users have to scroll up and read each email/message to extract all the relevant information. People click "reply all" to make

sure everyone gets each response but this clutters up people's inboxes. One can easily miss a place/time in a long email thread. Individuals might not know where something is located, requiring them to open a new window or tab and search it on a map.
It is also more frustrating to resolve a conflict about an idea for a place to visit because empirically, in these situations, there will be much back-and-forth debate and everyone has to agree on a decision and come to a consensus or people are unsure if the event will even take place. We believe this may end in a missed opportunity of friends wanting to do hang out together but unable to come to a consensus. In our app, the idea is that with a pinpoint/agenda, a decision has been made by default. Users can then discuss/change it, but there still exists a certain verdict/ruling that gets rid of some uncertainty/ambituity.

All of these solutions fail to deliver a truly simple way of planning events for one day with a group of people. Since other apps seem to be commonly deficient in a group functionality, we are motivated to manifest this as a defining concept (the "cabal") in our app. DOWN2GO's goal is to be simple, quick, and universal. Another motivation is to combine the ability to visualize events/list events/discuss ideas, all related to the day in question, in one place that users can refer back to in order to extract all the necessary and relevant information.

1.2 CONTEXT DIAGRAM


Google Map Server
map data request, pinpoint request
map with pinpoints
App Server
chat messages, updated agenda updated pinpoints
Pusher Realtime Cloud
agenda requests, chat messages, pinpoints
account info map data
chat messages, updated agenda, updated pinpoints
User

2 | CONCEPTS

2.1 KEY CONCEPTS

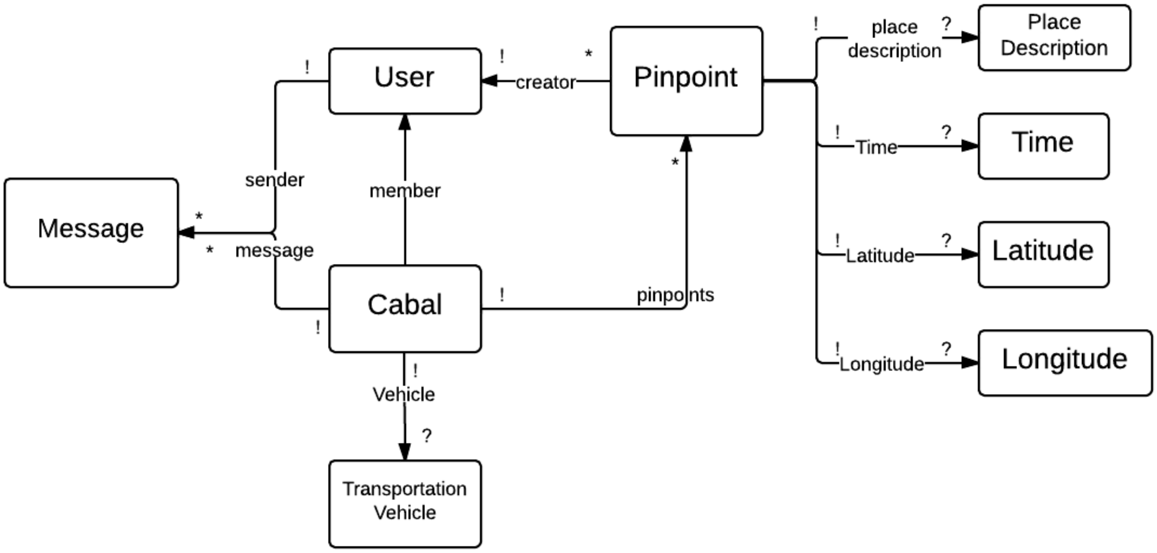
Cabal: A group of users that join together to create an agenda for a specific period of time. A cabal is associated with one agenda, which means if the same group of users want to plan for two trips, there should be different cabals. A user can initialize the cabal by specifing the time period and inviting some other users. After that, all the users in the cabal can invite more users and they do not have the ability to kick out others.

Pinpoints: A user create a pinpoint by doing a geo search on the map. After that, all the users in the cabal can specify when they want to be there. Also, they can give a human-readable name for the place.

Agenda: The agenda should be a sorted list of places, and each place contains the time to arrive there and what to do there. The system will give an evaluation on agenda by calculating the time cost between two places and checking if that fits the schedule.

Real-time chat: The chat functions as a typical online chat. Users post messages from a text box, which all other users viewing the page can see appear in a small frame, in real-time.

2.2 DATA MODEL



** In our design, one cabal has exactly one agenda, and an agenda has zero or more pinpoints. We do not have agenda in our data model because we will not store that in the database. Instead, we will create a self.agenda() method in the Cabal model, which returns a list of sorted pinpoints.*

Changes since last phase:

- Add a transportation vehicle attribute to cabal, because we want to estimate the agenda based on users' transportation vehicles (The users have a car or have to use public transportation)
- For a pinpoint, change "place" into "latitude" and "longitude"

3 | BEHAVIOR

3.1 FEATURE DESCRIPTION

- **Group planning.** Organize a group of your friends to work on a hangout plan.
- **Real time discussion.** Discuss with your friends about your thoughts on the plan using our real time chat.
- **Find your spot on the map.** Search for locations using google maps api and place your activities on the map with markers which include details about the activity and time.
- **Automatic agenda generator.** The system generates an agenda based on your marked activities for you to save and print for later reference.
- **Agenda Evaluation.** The system will give an evaluation on agenda by calculating the time cost between any two places and checking if that fits the schedule.
- **(Potential) Trip Suggestion** Give suggestions about how to plan the trip based on people's interest, location and time.
- **(Potential) Time Conflict** Allow users in a cabal have conflict schedule, and we allow each user to export their own sub schedule.

3.2 SECURITY CONCERNS

The Security requirements are that:

- Only the invited members of an cabal can see and engage in a group's activity including chat, map related actions and agenda management.
- Users have to be logged in to take any additional action.
- The user who created each chat message and each map pinpoint is shown to the rest of the users in the cabal.

Threat Model:

- Can assume no interest from state actors or criminal syndicates, since only minimal profile information stored (name and email).
- Since a cabal is assumed to be consisting of people who already plan to hang out with each other and assumably familiar with each other, fraud is unlikely.

Potential Risks and Mitigations:

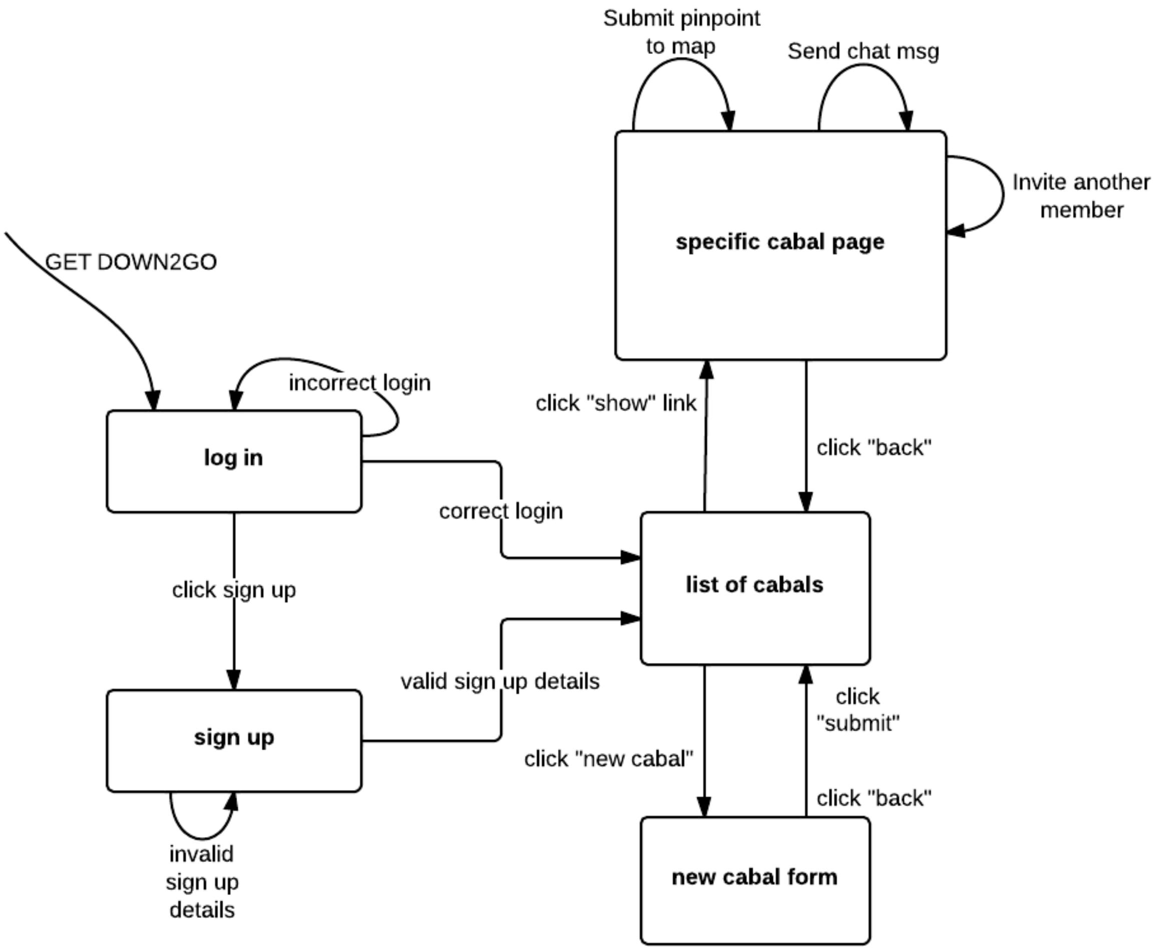
- Malicious users may attempt to access other people's agendas for private data.
    - We will limit to access to each group's activities and data to users who are

participants in the group.

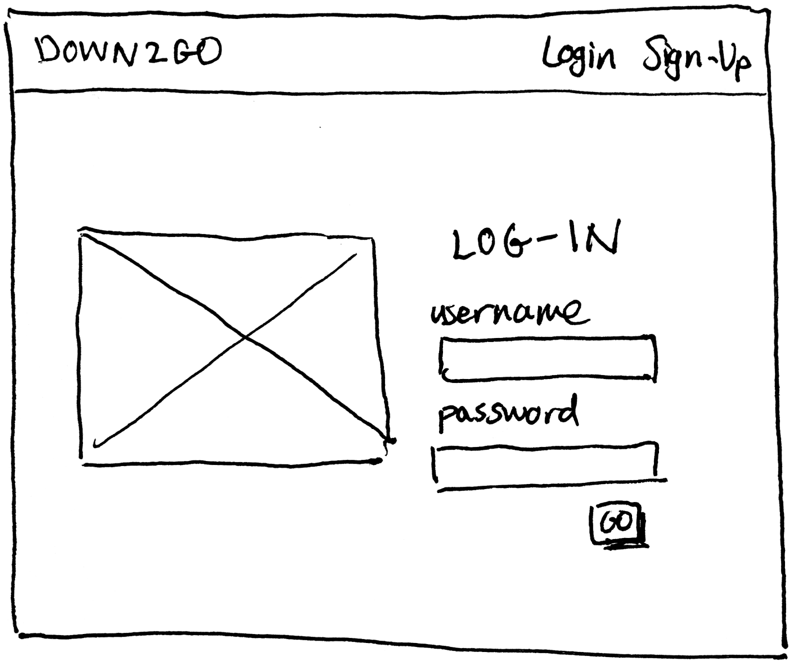
- Participants in a cabal can edit markers and the agenda freely such that someone in the group may disrupt the normal activity of the whole group.
  - At the moment, we do not plan to limit the freedom of activity of each user as long as they have been invited to the certain cabal. Beyond the MVP, we might implement some features to ask for the other users' consent before making any changes to the cabal agenda.
  - For the trip-planning cabals, assume that all members know each other well, and communicate out-of- band, so little risk of fraud.
- Chat messages are sent without being encrypted at the moment.
  - If we have time, we will implement an encryption method in the server side, and push encrypted messages to Pusher. When the user received encrypted message, implement a decryption method in the client side.
- Rails's built-in security features will protect the server against CSRF, database injection and javascript injection.

3.3 USER INTERFACE

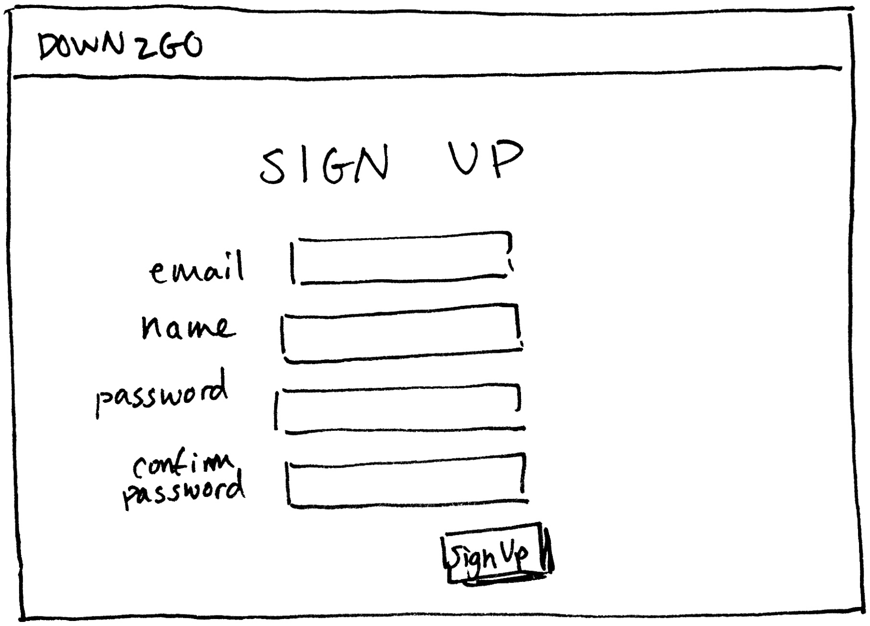
Page Flow



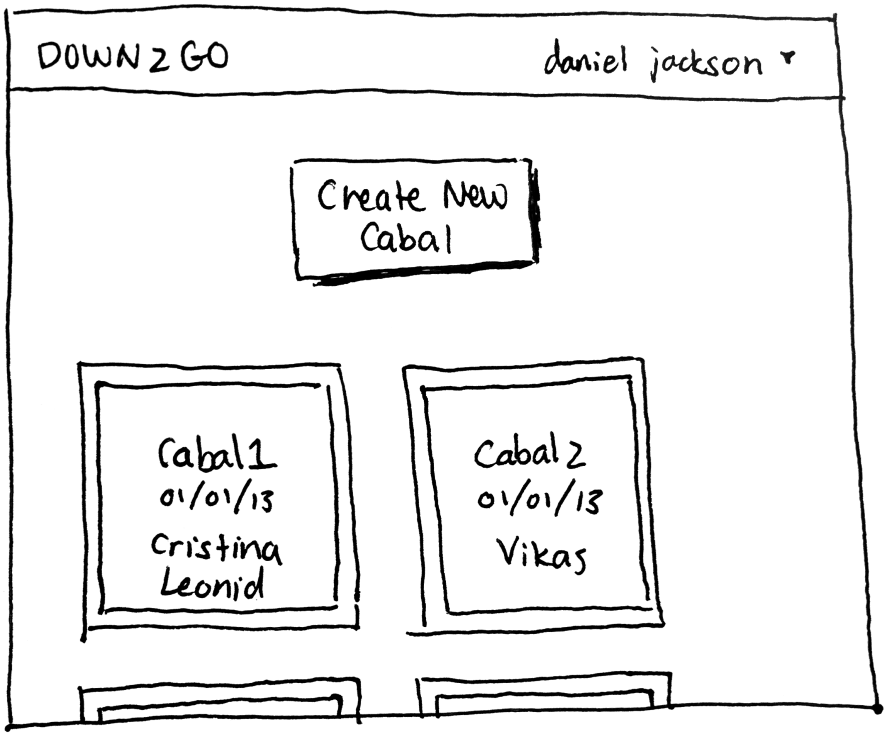
Login Page



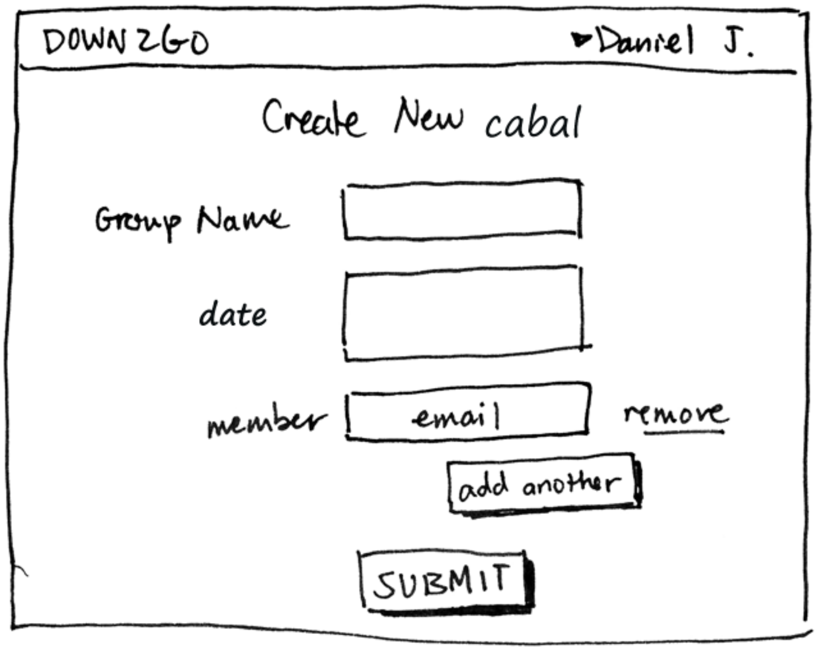
Sign Up Page



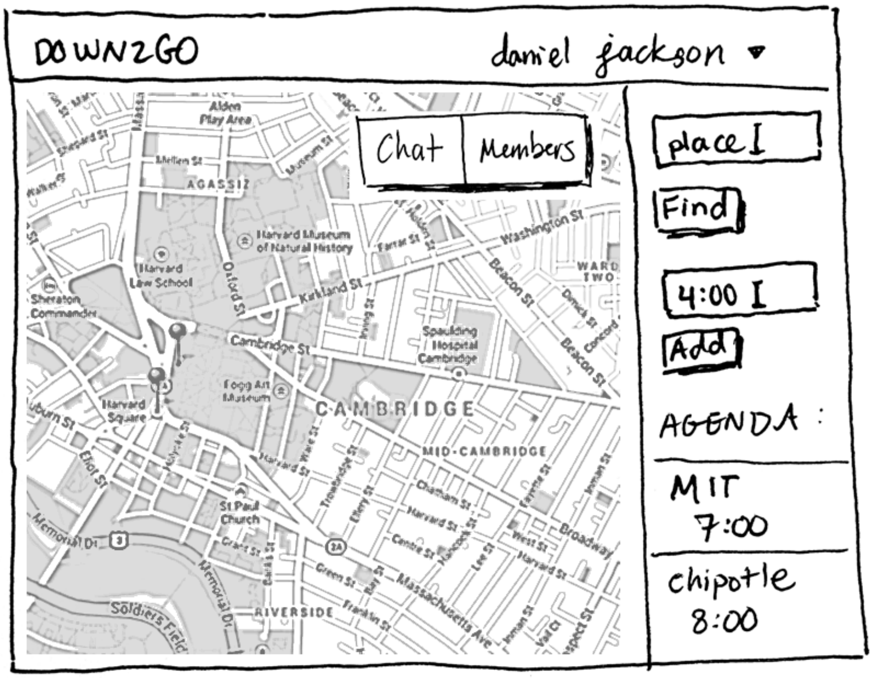
List of Cabals Page



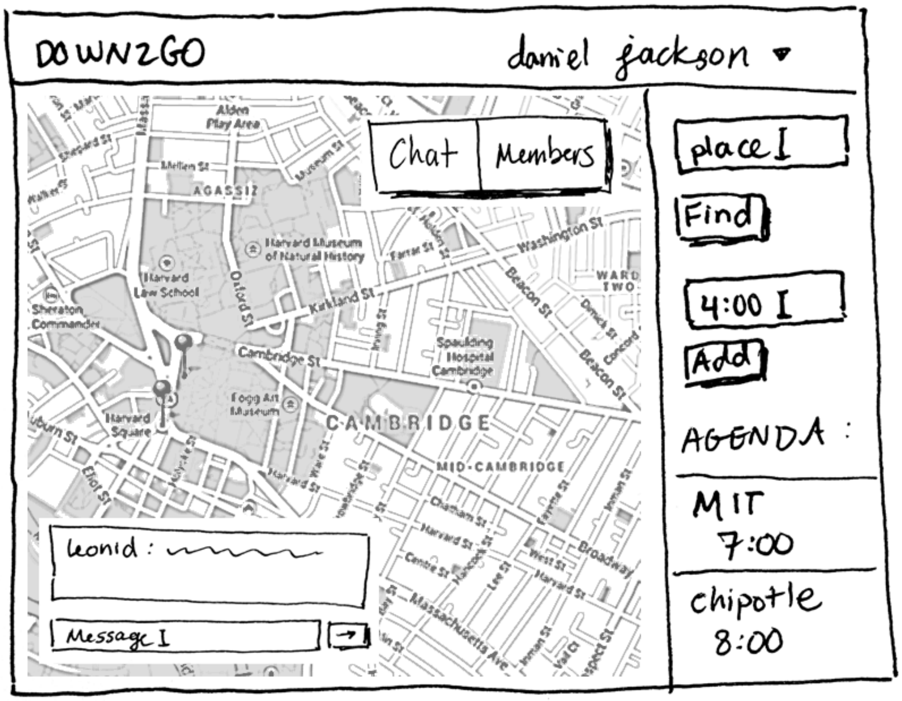
Create New Cabal Page



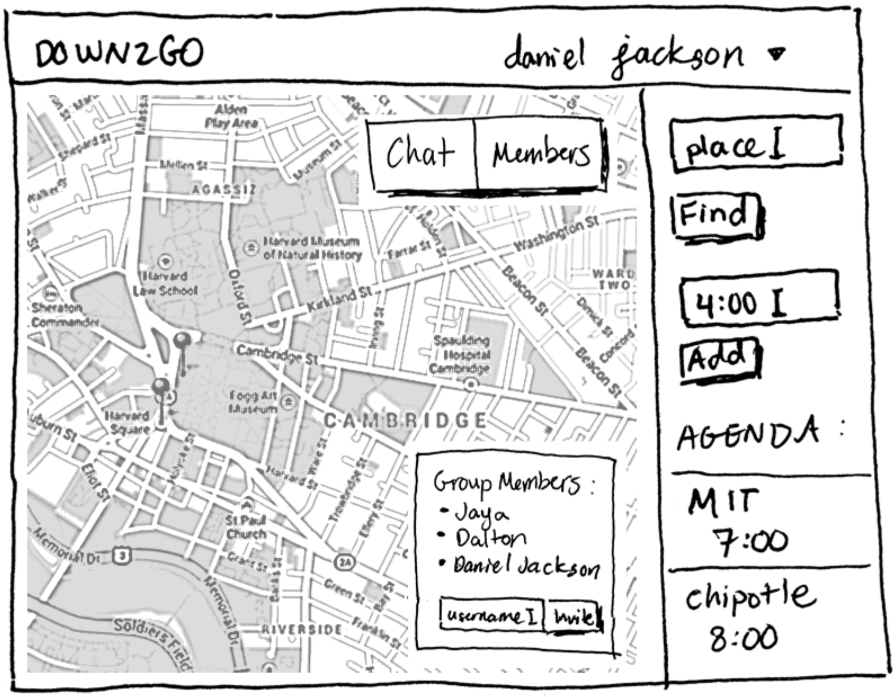
Page for a specific cabal



Page for a specific cabal with chat visible



Page for a specific cabal with "members" visible



4 | DESIGN CHALLENGES

4.1 BEHAVIORAL CHALLENGES

Whether to bring people to a decision?

- **Choice 1**: Make all group members decide on whether to accept or decline a modification to the agenda pinpoints before actually finalizing the change. Chronologically overlapping events in the agenda are not allowed.
    - **Pros**: Only one version of the agenda is generated in any cabal; agendas are unlikely to be destroyed by users with bad intentions; people will be arought to attention each time a change is proposed.
    - **Cons**: More actions required for the users; less flexibility in the planning; can get stuck in a deadlock if a decision acnnot be reached.
- **Choice 2**: Anyone in the cabal can make any direct modifications (add, delete, edit) to the agenda pinpoints , and the group does not have to reach a consensus by requirement of the app at any any time. Chronologically overlapping pinpoints in the agenda are allowed.
    - **Pros**: More flexibility for agenda scheduling; workflow will not be interrupted by the delay of response from cabal members; users can save or print their desired version of the agenda.
    - **Cons**: Need to deal with potential race conditions caused by multiple users editing at the same time; generating different versions of the agenda requires additional logic and UI.
- **Decision**: Anyone can make direct modification at any time.
    - The goal of the app is to help people plan their hangout time more efficeiently. While it seems reasonable that the app have some strategies to force the users to make an agreement, we have determined that the benefits of having them is not worth the cost. For the intended use cases of the site, outing planning would not involve a significant amount of money or privacy, and the experience should be relatively casual. So if the app forces the users to make an agreement, they will feel constrained. We want to give users more flexibility. If they are really into the trip, they will reach an agreement themselves through chat; if they do not care about whether a trip is planned, they can end the conversation any time.

Moderator in the cabal?

- **Choice 1:** Have the creator of the cabal be the moderator, who is the only one with invite and kick user privileges.
    - **Pros:** Smaller chance of users with bad intentions entering a cabal.
    - **Cons**: Less freedom--it would take more effort to invite users since it has to be done through one person.
- **Choice 2**: Everyone who is already in the cabal can invite anyone freely and no kicking mechanism would be provided.
    - **Pros:** Easy to add new members.
    - **Cons:** Once a user with bad intentions entered the cabal, it would be able to invite more without checks.
- **Decision:** Anyone in the cabal can invite people freely and no one can kick.
    - We want to give users more flexibility and makes the app easier to use. Trip planning usually does not involve money or privacy, so the potential risk of privacy leak is low. Also we assume that people already form a group before using our app, so it will be very unlikely they invite a bad user to the cabal. If accidentally, a bad user come into the cabal, people can quit the cabal and create a new one without too much effort.
    - As an example, facebook group chat also do not have a moderator, and people cannot kick out people. Our app and Facebook chat are both social chatting systems, so it is reasonable for us to choose a similar strategy.

Editing pinpoints--who can edit which?

- **Choice 1:** Users can only edit/delete pinpoints they created.
    - **Pros:** Points are less likely to be modified/deleted by mistake by other users; race conditions on individual points are eliminated.
    - **Cons:** Have to go through the pinpoint creator to make a desired change.
- **Choice 2:** Any user can edit any pinpoints created in the cabal.
    - **Pros:** More freedom for the other users; workflow will not be blocked by a single user.
    - **Cons:** Pinpoints may be edited unintededly.
- **Decision:** Any user can edit any pinpoints created in the cabal.
    - Again, our focus is to give users more flexibility. While there are certain risks in allowing users to make modifications to pinpoints that others users created, we'd like to assume that users will communicate about it first through the chat functionalities.

Editing pinpoints--should we allow drag-and-drop?

- **Choice 1:** Pinpoints can be dragged and placed on other positions on the map after they have been created; the location of the pinpoint will be adjusted based on

the dropped position on the map.
    - **Pros:** Users can to adjust locations intuitively.
    - **Cons:** Not completely certain that it can be implemented; might introduce potential race conditions.
- **Choice 2:** Pinpoints are immutable; one has to destroy and recreate the pinpoint if the location is incorrect.
    - **Pros:** Agenda is easier to manage.
    - **Cons:** Making modifications to the location of a pinpoint requires extra steps.
- **Decision:** Immutable pinpoints for the MVP, draggable pinpoints implementation will be attempted for the final version.
    - We did not expect to implement this feature in the minimum viable product because its benefits are not critical to our solution, and the overhead it causes due to potential race conditions is not a problem we have to face. However, we do feel that being able to relocate the pinpoints intuitively is beneficial for the user experience. As long as timer permits, we will attempt to implement this feature.

4.1 CODING CHALLENGES

Whether and how to encrypt messages sent to Pusher?

Currently, we are using Pusher API (http://pusher.com) as a real-time push platform to implement out real-time feature. We need to think about how to encrypt the messages sent to Pusher.

- **Choice 1:** Subscribe to Pusher Encryption Service
    - Pros: No Extra Work
    - Cons: Have to pay more than 40 dollars a month
- **Choice 2:** Implement an encryption method in the server side, and push encrypted messages to Pusher. When the user received encrypted message, implement a decryption method in the client side.
    - Pros: No need to pay extra money
    - Cons: All users in a cabal have to share a same decryption method to decrypt a same message; may be time consuming to implement the encryption and decryption.

- **Decision:** Not planning on implementing an encryption. May implement our own encryption method if time permits.
    - Since we're not doing banking or storing passwords, nonencryption might

not be a huge problem. We will choose not to worry about the problem for now. If we have time, we will implement our own encryption and decryption to avoid paying.

For a specific pinpoint, storing latitude and longitude or address for location representation?

- **Choice 1**: Store the latitude and longitude.
    - **Pros:** Requesting and displaying the pinpoint on the map will be more reliable, since the point is guaranteed to exist.
    - **Cons:** Latitide and longitude are arguably not human readable; will have to retrieve human readable data from geosearch each time we rerender the data.
- **Choice 2:** Store the address.
    - **Pros:** It is human readable, we can directly show it on the agenda.
    - **Cons:** Have to do geosearch to pin points on the map. Not a hundred percent reliable since the address might not exist. People might enter an incomplete address which makes the geo search inaccurate.
- **Decision:** Store the latitude and longitude.
    - We will store latitude and longitude because it is more reliable given that we would need to perform geosearch everytime the pinpoint is rerendered anyway. Also, besides showing an address in the agenda, we will also allow a user to enter a name or short description for the place to make it more human-readable.

For response from the server to the client, what format should be used?

- **Choice 1:** Pure JSON containing the related data.
    - **Pros:** Simple data transfer; easy to debug.
    - **Cons:** Require building and parsing on client and server sides.
- **Choice 2:** Rendering a javascript that operates on the client.
    - **Pros:** Easy to implement; can use rails helpers for easier data transfer.
    - **Cons:** Harder to debug; might expose client implementation.
- **Decision:** Use JSON.
    - Since we use pusher gem to push updates to clients in JSON, we decided to keep the implementation consistent.