Milestone 2 Report

SnowAngels - Gamification of Snow Shoveling

Team Members:

Avinash Thangali - at553@cornell.edu
Brendan Fox - bf264@cornell.edu
Ivana Terziyska - it76@cornell.edu
Jeen Yan Toh - jt547@cornell.edu
Jiahua Li (Cathy) - jl3253@cornell.edu
Martin Eizayaga - me393@cornell.edu
Mishcat Ibrahim - mi243@cornell.edu
Quadri Bada - qb26@cornell.edu

The Client

Dan Cogan, Chief of Staff, City of Ithaca - dcogan@cityofithaca.org
Ducson Nguyen, Common Council member and Software Engineer - dnguyen@cityofithaca.org

Report Summary/Objective

This report is being written to discuss the progress we have made over the past several weeks and to discuss what we are planning on working on going forward.

Progress has been made on both the frontend and the backend portions of this project. In the backend, databases have been created and populated with sample data. Flask endpoints have been created to allow the frontend to access the data in the database, with several endpoints allowing for creating, accessing, and removing data already made. The frontend has done user testing with paper prototypes for several facets of the user interface, and is using the feedback to improve it. Additionally, they have begun programming the actual interface.

For the next milestone, the backend and frontend teams both plan on finishing up everything we have listed for milestone 3 as well as ensuring every part of milestone 2 is functional and fully tested.

Requirements Analysis

We received feedback of our feasibility study from the professor and the client. After discussing and analyzing the feedback, we refined our requirements for Version 1 of the application as follows:

- Map Interface
  - All Ithaca corners must be visible on the interactive map
  - Users must be able to click on corners and make reports about the snow conditions of corners
- Leaderboard
  - The user interface must display the top users per day, per storm, and per season
- Subscription/Notifications
  - Users must be allowed to subscribe to corners to get notifications about them
  - Users can choose to subscribe to corners along a route and/or choose individual corners
- Voting
  - Users can vote if a corner is properly shoveled
  - Points are awarded regardless for version 1

Backend Team Updates

Database

The backend team focused on the following requirements: Leaderboard, Subscriptions, and Voting. We received sample data of the city intersections and storm basins from the GIS team at the City of Ithaca. Then, we designed a database management system as shown in the entity-relationship diagram below according to the relationships between the intersections (named corners) and the users.

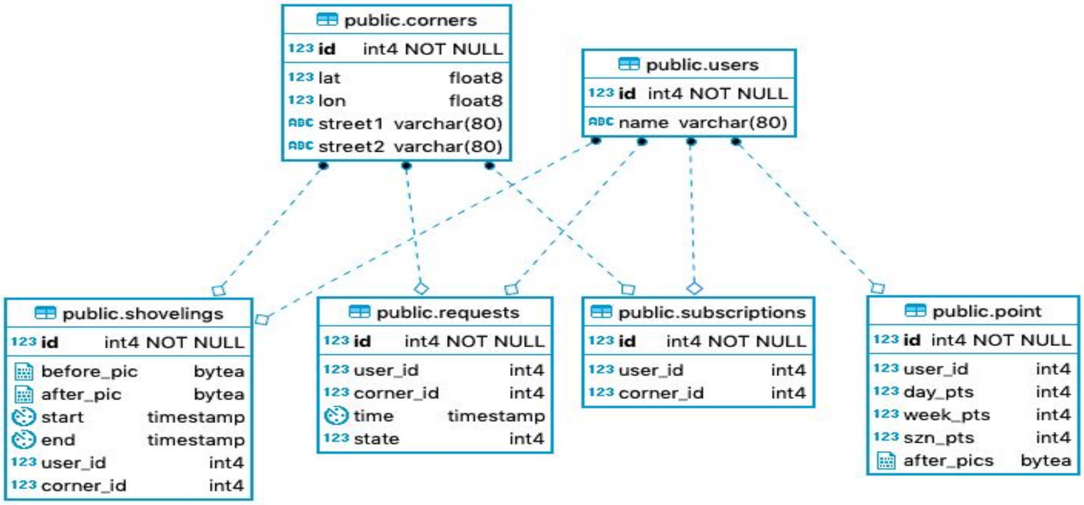


Figure 1: Entity-Relationship Diagram

We started to implement the tables and relationships using SQLalchemy and PostgreSQL. In addition, we added endpoints to add and retrieve data from the system, and manually tested the endpoints with Postman. We also decided to add a function to import a dataset from a local file, after the GIS team suggested that the complete dataset will be given to us as a local file. Thinking ahead for future additions to the frontend, we have tried to make our system as flexible as possible, which will help if we need to implement additional functionalities.

After the presentation to the client and professor, we received feedback about how we should handle the handover of software to the client once the project is over. With that issue in mind, we will continue to inform the GIS team and the client of the software that we are using to see if our program can be run easily on the client's side. If there are differences between the client's technology and ours, we will either alter our code or find a way to make our program easily transferable.

Endpoints

We wrote REST endpoints to allow users to interact with the database. We currently have POST endpoints that allow for new subscriptions, requests, users, corners, and shovel claims to be added into the database. The subscription POST endpoint allows a user to subscribe to a corner. The requests POST endpoint allows a user to add to the database that a specific corner needs to be shoveled. This will set the state of that corner to 0, meaning someone has made a request. The user POST endpoint allows a new user to be added to the database. The corner POST endpoint allows a corner to be added to the database. We've also written a function that reads GIS information from a file and adds corners to the database automatically. The shovel POST endpoint allows a user to enter a claim that they have shoveled a corner into the database and they immediately get rewarded with points for this action for instant gratification. This then sets the state of the initial request on the corner to be 1. We also wrote an endpoint that would allow the original requester to validate whether the person who claims to have shoveled actually did. If the requester validates it, the state of the request becomes 2, otherwise, the state goes back to 0, the user loses the points they had gained for claiming to shovel the corner, and all users subscribed to the corner will get a notification that the corner needs to be cleared. We also wrote GET endpoints. For example, we have GET endpoints that return all users subscribed to a corner, list all corners a specific user is subscribed to, get the state of a request, get the street names of a corner, get the coordinates of a corner, and get the top users of the day, week, and season. We also have one DELETE endpoint that allows a user to unsubscribe from a corner.

Next Steps

Our immediate next step is to integrate the current endpoints we have with front end. We will work on integrating these endpoints with the user interface such that the event handlers for the user interface will perform as we expect. Also, we will create dummy data in the database in order to give the frontend team data to work with

We also have many game specific next steps:

1. Implement Notification System: We will send notifications to all users who are subscribed to a corner when someone makes a request. We will also send a notification to the original requester with before and after pictures of the corner when someone claims that they have shoveled the corner.
2. Add weights to corners: We will give weights to corners depending on if they are on a priority route or if they have gotten many requests. We will discuss this further with the client
3. Determine how to label specific corners: Since every intersection can have multiple corners, we need to develop a way to give each of these corners a unique, identifying number.
4. Times of before and after pictures for shoveler: We need to determine a way to send the exact time the user uploaded the before and after pictures to the database in order to figure out how long they spent at a corner
5. Allow users to subscribe to a route: Currently we allow users to subscribe to corners, but we'd also like them to be able to specify a route and then automatically have them subscribe to all corners along that route

We will write comprehensive unit tests. Currently, we've been doing manual tests of the endpoints via Postman, however, we will write unit tests.

We will incorporate the dataset from the GIS team at the City of Ithaca. We will figure out how to adjust this dataset to make it easier to incorporate it with the front end teams pins for each corner.

Frontend Team Updates

The front-end team focused on preliminary design for the user interface through paper prototypes for version one functionalities: Reporting and shoveling Corners, Marker Design, Leaderboard Layout, and History Page. Then, we began learning React Native, the framework for the User Interface and implemented the map interface, location tracking, side menu, and markers with labels. We expect to reallocate work to focus more on frontend development and implement a complete preliminary design of the version 1 user interface by the next milestone.

1) **Paper Prototypes**
    a) ***Reporting and Shoveling Corners***
        i) Objective
        The objective of this user testing was to refine and shovel submission and shovel reporting functionalities.

ii) Process

Users were asked to report and shovel a corner by clicking a corner and interacting with the pop up screen.

iii) Takeaways

The terminology “begin shovel” and “report” was the clearest for users to shovel and report a corner. It was observed that users may need to redo their request, indicating a need for a redo request screen after requests to shovel. Keeping track of data before, during, and after shoveling, such as voting, user location, pictures of the corner, and time at a location was determined to be necessary for verify that the request is valid or the corner is shoveled.

b) Marker Design

i) Objective

The objective of this user testing is to develop a way for users to locate corners and distinguish unshoveled corners from clear corners.

ii) Process

Users were presented two marker designs individually -- one with a heat map of marker density that zooms into individual dots-- and the other with a number indicating the number of reports, where corners with more reports have proportionally larger markers. For both designs, red markers indicate there is at least one report and blue indicates there are no reports. Users were asked follow-up questions about what they believe each component represents, as well as to compare the aesthetic and ease of understanding of both designs.

iii) Takeaways

Users concluded that the design that displays the number of reports was clearer, but the heat map and dot design had more aesthetic appeal. However, they thought both were intuitive and easy to understand. Some criticism of the current design indicating that red markers require action (reporting), and the constraint that some users may be colorblind and not be able to differentiate the corners.

c) Leaderboard Layout

i) Objective

The objective of this user testing was to determine an optimal display for the leaderboard.

ii) Process

Users were asked to navigate to the leaderboard using the sidebar. Then, they were presented with either a chart or bar graph format and asked which they prefer. They were also asked to navigate using a drop down menu to sort scores by the last week, month, year, and all time. They were then asked for feedback on what the default screen should be, and navigate back to the map.

iii) Takeaways

Users were able to intuitively navigate to the leaderboard using the navigation side bar, thus we will implement this our design of the side bar. They generally preferred the bar graph layout, with the top 5 scores displayed. However, our client expressed interest in displaying the information similar to Strava, a running app that displays top scores in a chart format, with the top ten displayed, and a separate panel with the user's scores displayed between the two ahead and behind them, thus we will adopt this design instead .The user testers and the client agreed that the user's score should be displayed to indicate their standing, thus we will display another chart according to the above description to include both the top scores and the user's place. Finally, users were able to navigate back to the map using the map icon. Additional paper prototypes will be necessary to refine the design relative to the client's suggestion. In the future, we will consult with the client before proceeding with user testing and design choices.

d) **History Page**

i) Objective

Design a user friendly page to show the history of shovels for users.

ii) Process

Users were shown two app designs for finding the history of their shovels and the history page layout. The page layout was made to show the details and information around each shovel (time stamp, location, google map visual). We gave each user 10 minutes to look at both versions of the app and write down any notes about what features they liked and disliked. After this, we combined the best features of all the users into one app.

iii) Takeaways

Firstly, users had a dislike for the terminology used to title the page. Most of them preferred “History” than “My Shovels”. Users also preferred the history navigation bar on the side of the app rather than the top because it was too cluttered for them and interfered with the app. In addition, users also appreciated the two categories “All” and “Favorites”. Lastly, they enjoyed a simple preview for shovels that when clicked, the details and map location of the shovel appears.

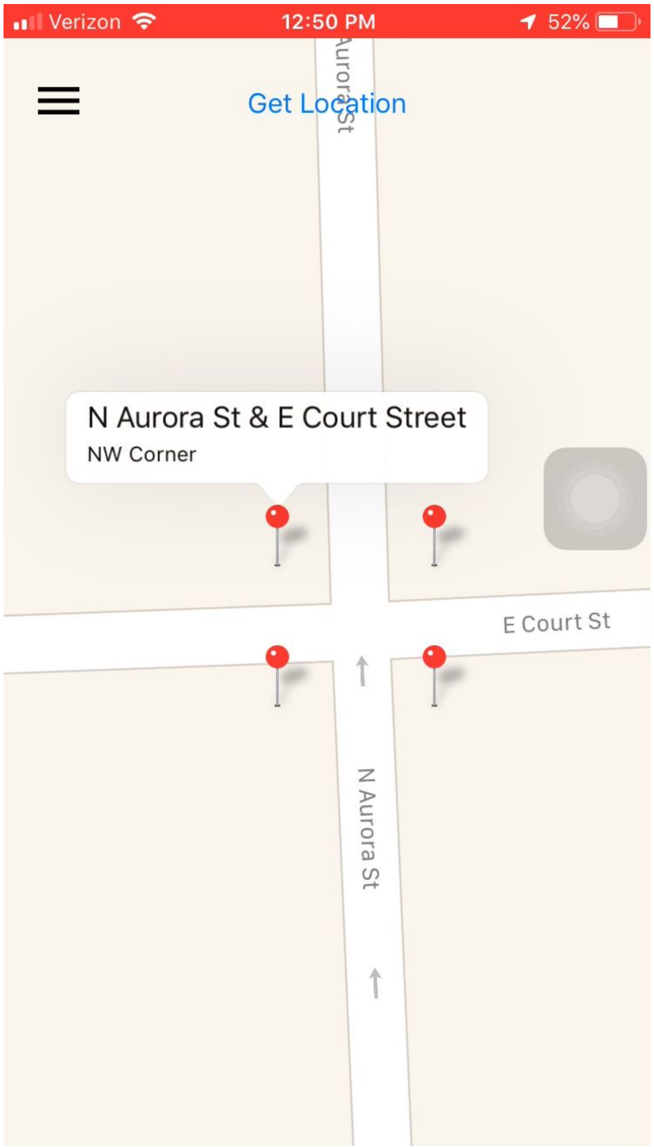
2) UI Functionality

The frontend team has worked on learning and decided to use React Native, a JavaScript framework. We have implemented the map interface using the navigate API, as well as the sidebar menu with navigation options and user information. We also are able to get the user's location and display markers from placeholder data corresponding to corners, displaying information about the corners when they are clicked.

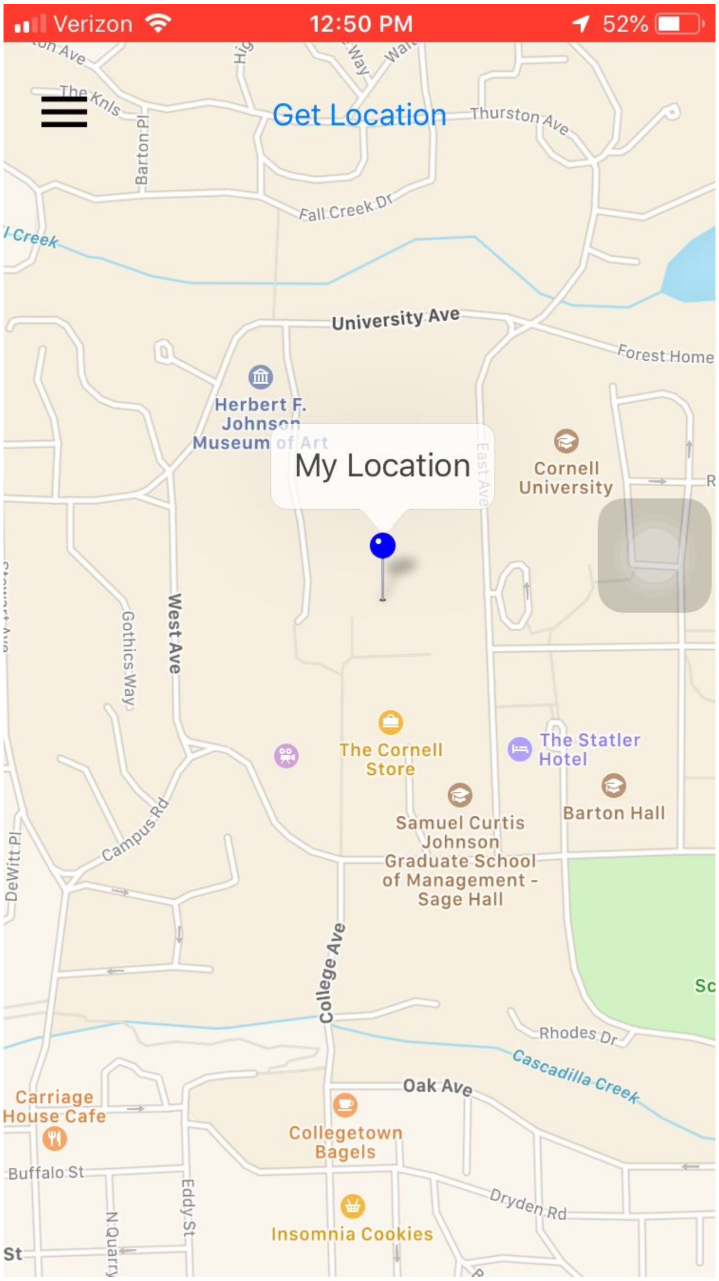
Screenshots

We have included screenshots of the map interface below.

In the first screenshot, the user sees pins on corners. In this version, we have the pins and they show the street names. In the next implementation by the second presentation, the user will be able to click a pin and make a request on that corner to indicate that it needs to be shoveled. Users can zoom into and out of the map in order to click on different pins in Ithaca.



In the second screenshot, the user's location appears as a pin. The color of the pin here is blue. We are using a blue pin to mark two things: the user's location and corners that have no requests on them. One piece of feedback that we got from the presentation is that this is not accessible, as some users may be colorblind. Thus, we are going to consider a different design instead of red vs. blue for the second milestone.



Next Steps

First, we aim to finish the Milestone 2 requirements from the feasibility study, which requires explansin of current features such as a pop-up menu for marker information displaying reports, shovels, and location, and screens for report/shovel and user history. We also aim to implement a static leaderboard with a chart interface, user information, and ability to sort by time frame.

We aim to implement a preliminary, functional design for all version one features by milestone 3, including the leaderboard, notifications and subscriptions, and voting. These features can be static or not fully functional , but the user should be able to navigate to each feature and should represent the overall interface design that will be further tested. To complete the user interface, we may require more team members and time allocated to this task. On both sides, we aim to use GIS data provided by the client, and integrate the backend and frontend sides of the application for a more realistic display.

Also, from the presentation, we learned that it may be better to allocate more of us to frontend. The backend team will wrap up the remaining functions soon, and then help the frontend team implement the functionalities of the app.

Revised Milestone Requirements

After discussing our progress for both the frontend and backend, we decided to update our requirements and target for Milestone 3:

We will place emphasis on finishing milestone 2 functionality as stated in the requirements analysis above. The frontend team will work on developing the user interface for the version 1 features and a prototype for the leaderboard function. The backend team will incorporate the full dataset from the GIS team. As a team, we will begin to integrate the frontend and backend and use real GIS data on both sides. We will also conduct additional user and unit testing.

We reviewed our timeline for the next few milestones as follows.:

- Milestone 3 (3/27) -- Next Presentation
  - Successfully integrate user interface and backend
  - Implement all remaining backend endpoints
  - Create a user interface with all of version 1 functionality
  - Perform user testing to see if any improvements need to be made to the app
  - Meet with the client and assess feasibility of further features with them
- Milestone 4 (4/17) -- Ready for Release
  - App will be updated based on user testing from last milestone

    - Entire application to be tested and debugged
    - Write complete, comprehensive unit tests
    - Functionality of final application to be presented to the client
- Milestone 5 (5/16) -- Final Delivery
    - Changes to the code will be made based on feedback from the client after the functionalities of the app are shown to them
    - All source code and documentation to be shared with the client