Where2Go
A Generalized Mapping Application for Non-profits
External Advisors: Dr. Sangeeta Vohra, Dr. Chris Murphy
Team 30

Katie Jiang kajiang@seas

Stephanie Shi stephshi@seas

Erik Zhao erikzhao@seas

Vicente Guallpa guallpav@seas

Abstract

Non-profit organizations have location-based resources to help their community but lack a solution that is both inexpensive and effective to display this information. Lists poorly convey geographical information, Google Maps is not customizable, and custom applications are expensive to develop. Where2Go is a web application that allows non-profits to bulk upload custom resource information onto their own map. Users can search and filter locations using this custom resource information. We evaluated the speed and difficulty for the most common non-profit and end user actions. Our evaluations show that these actions are quick and easy to complete. As a result, people will be able to more quickly and easily find resources that best fits their needs. Non-profits now have a solution that is both inexpensive and effective.

1 Overview of the Problem

Non-profit organizations have location-based resources to help their community but lack a solution that is both inexpensive and effective to display this information. Some organizations use list on a web page as an inexpensive solution, but lists do not effectively communicate geographical information to users. To better communicate this information, some organizations also use Google Maps. This tool is not customizable and so users cannot filter locations using custom resource information. Moreover, Google maps is not free. Organizations could hire web developers to achieve their goal, but this would incur high fixed and maintenance costs.

2 Overview of the Need

Non-profit organizations need a solution that is both affordable and effective. The solution must be easily accessible for both non-profits and users seeking their resources. It must also be easy to use and require no in-depth technical knowledge of web applications. The solution must be equipped with a map view so that users can easily locate resources. The map must be able to display custom resource information effectively and enable users to filter locations based on that information. Additionally, each organization must have their own map and have their account be password secured. There must also be an establish channel of communication for any issues a non-profit or end user may have. The solution must be in the form of a web application.

3 Market Research

At the beginning of the project, we conducted an over-the-phone preliminary survey with multiple non-profit organizations such as SERVE Philadelphia. The purpose of our survey was to investigate how they display their information and resources in a map view. We identified multiple obstacles in their approach such as high costs and lack of customization. Additionally, we met with a select few to design a mock solution that would address these issues. What they were looking for in a better solution were lower costs, ease of usability, and customizability as mentioned in Section 2. Much of our knowledge came from two of our group members' (Katie and Stephanie) previous work on a project through a Penn student organization.

Based on this feedback our goal was to create a generalizable map-based product to enable non-profits without technology capacity to easily display custom location-based data on a map view. Our solution would allow non-profits to sign up and request a subdomain, deploy an instance of the map application with a single click, and receive admin privileges to that subdomain. The map application itself would grant admins the ability to bulk upload and edit resource location data, and enable users to view, search, filter, and get directions to resources.

4 Market Segment

According to the National Center for Charitable Statistics, approximately 1.56 million non-profits were registered with the IRS in 2015 (McKeever, 2019). Our primary target market is small, local non-profits, since they are best served by Where2Go. Approximately 210,670 organizations had expenditures under $500,000 (McKeever, 2019). Using a conservative estimate of 1% of these non-profits who would use the paid tier of our product, this would generate an annual revenue stream of approximately $42,134, which is more than sufficient to fund server and development costs.

5 Competition

As discussed in Section 1 existing solutions are either inexpensive or effective, but not both. The most common solutions used today are lists, Google Maps, and custom web applications. Geographic Information Systems (GIS) are also worth mentioning since some non-profits use this solution.

5.1 Lists

Non-profit organizations use lists because they incur no additional costs and are easy to implement with some knowledge of HTML. Although lists are inexpensive they are not effective. Specific location-based information may vary between resources and this is not communicated effectively through lists. Additionally, the lack of a map view makes it difficult for users to find the nearest resource.

For example, Free Dental Care provides free or low cost dental services to low-income residents. In an effort to better inform the community about their services, Free Dental Care displays individual clinic information in a list on their website (Free Dental Care, 2020). Since not all clinics have identical services, prices, nor availabilities it is difficult for a user to find the closest clinic that meets their medical and financial needs.

5.2 Google Maps

Google Maps is an effective tool to display geographical resource locations, but it does not effectively display custom information about those locations. Additionally, users cannot filter locations using that custom resource information.

For example, Healthy Minds Philly partners with multiple LGBTQ+ non-profit organizations to direct users in need of their resources. In addition to using a list, Healthy Minds Philly also uses Google Maps to display the location of LGBTQ+ resource centers. Each LGBTQ+ offers varying resources such as legal name and gender change or professional development training. Since these services are not reflected on Google Maps users in need of these specific services would spend more time attempting to identify all the LGTBQ+ centers that offer these services (Healthy Minds Philly, 2020).

Moreover, Google Maps is not free. Google Dynamic Maps cost $7 per thousand requests (Google Maps Pricing, 2020). These costs can quickly add up as the traffic through the non-profit's site increases. Although Google Maps is a better alternative to lists, it does not entirely satisfy the needs of non-profits.

5.3 Geographic Information Systems

GIS applications allow users to present spatial or geographic data, create searches, edit data in maps, and more. There are many large players that provide robust geographic information systems, including Autodesk, Esri (Environmental Systems Research Institute), and Integraph. Many of these systems include advanced functionality such as the ability to build completely new maps and real-time analytics. However, because these systems are designed for large enterprises, they provide much more functionality than needed for displaying resources on a map, which requires more maintenance. Furthermore, the cost of using these systems can be as high as $100 per year for basic features (Esri Pricing, 2020).

5.4 Custom Web Application

Some non-profits may attempt to develop their own map application that satisfies the needs discussed in Section 2. This level of involvement required for a custom map application would divert the organization's resources from their mission. Non-profits will incur additional costs for backend and frontend developers as well as other site-related costs. The national average hourly rate for web developers is $85 - $125 (Thumbtack, 2019). Even if they do accomplish to develop their application, it would only resolve the needs for that specific organization.

6 Value Proposition

Where2Go is a generalized resource mapping application for non-profits without extensive technical or financial resources to effectively display resources and information about these resources to their target users, as this is currently an unserved need. Non-profits can register for a free account and manage multiple maps containing relevant resources. Users can easily add and remove new resources. Where2Go features bulk upload and download to streamline an admin users workflow. Where2Go's low cost allows non-profits that rely on bespoke work from web developers to lower their costs, and non-profits that utilize a map view provider such as Google (which is also very expensive!) would be able to better display custom location-based data.

Users are able to search for relevant resources and view their location relative to their own on a map.

Overall, Where2Go is a cost-efficient platform that enables non-profits to better serve their target users by effectively communicating beneficial resources.

7 Customers and Stakeholders

The primary customer for Where2Go are non-profits, ranging from local to national. This is because these organizations can manage resources for their users. Small, local non-profits may have strained budgets and limited technical resources to support the maintenance of a complex application.

Additional stakeholders include the ultimate end users of the product, who are the people that would be using Where2Go instances to search for resources. These users often do not have regular access to the internet and are most interested in a smooth user experience.

8 Cost Model

Because our product is a web application, our primary expense are the server costs, which would come from the costs of hosting on AWS EC2, as well as the Mapbox (our mapping library) API costs.

For our server costs, because we do not require intensive computing resources, we will use the AWS EC2 A1 instances, which are around 40% cheaper than other EC2 instances. To allow our server to scale as the number of users increase, we plan on using an `a1.large` instance, which costs $0.051 per hour.

In addition to server costs, we also have costs associated with our mapping library, Mapbox. We require a Mapbox API call in the following two cases:

1. Every time a map needs to be loaded: this occurs every time a user visits a Where2Go instance (Mapbox GL JS)

2. Every time a user performs a single search query for places, addresses, or points of interest: this occurs every time a user searches for a resource and every time an admin adds a new resource (Mapbox Geocoding API)

Below are the costs for every time Mapbox GL JS initializes on a web app, which for us would be every time a user visits a Where2Go subdomain (a particular organization's map).

Monthly Loads	**Costs per 1,000**
Up to 50,000	Free
50,001 to 100,000	$5.00
100,001 to 200,000	$4.00
200,001 to 1,000,000	$3.00
1,000,000+	Contact Sales

Table 1: Mapbox GL JS API Costs

Monthly Loads	**Costs per 1,000**
Up to 100,000	Free
100,001 to 500,000	$0.75
500,001 to 1,000,000	$0.60
1,000,001 to 5,000,000	$0.45
5,000,000+	Contact Sales

Table 2: Mapbox Geocoding API Costs

For the cost model, we make the following assumptions:

- Each non-profit makes 1.05 maps: we expect most organizations to require 1 map, though some may require 2 or more.

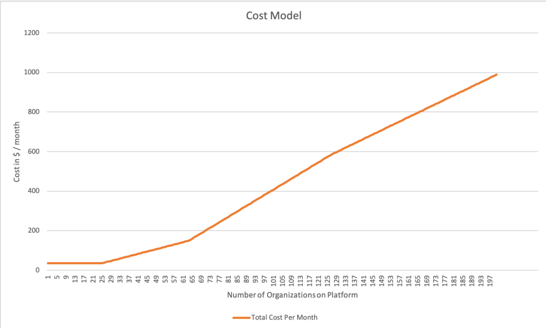
- Each map contains 30 resources.

- Each map is visited 25 times per day, and on a visit a user performs 5 searches.

- There are 30 days in a month.

- The cost per 1,000 exceeding 1,000,000 is $2.00 for the Mapbox GL JS API, and the cost per 1,000 exceeding $5,000,000 for the Mapbox Geocoding API is $0.30.

Below is our cost model (costs are per month).



As we can see, the cost is low while there are only a few number of organizations using our platform; however, the costs increase significantly as the number of organizations increase. This is because most of the costs can be attributed to the cost of the Mapbox API. As a caveat, we note that it is likely that we would be able to negotiate a lower price of the Mapbox API if needed since our work is benefiting non-profits; we did not include this assumption in our cost model. We also note that it is unlikely we will hit the inflection point by which the costs are extremely high.

9 Revenue Model

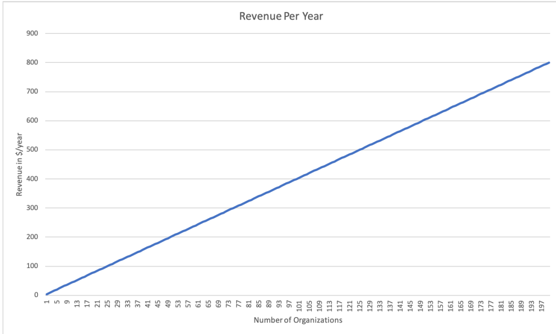
It is not our goal to profit off this product because we believe that the non-profits should utilize their funding to benefit their target users. However, it is also important that the product is maintainable, so we plan on implementing the following:

- Free tier: On a per-map basis, the free tier limits each map to a maximum of 2 "collaborators" (admin users) on a single map and a limit of 25 resources.

- Paid tier: An organization that wishes to invite more than 2 collaborators or add more than 25 resources must pay a nominal fee of $20 per map per year.

We hope that these tiers help support the associated hosting costs and new feature development, and provide value to both small and large organizations.

To build our revenue model, we made the assumption that 20% of the organizations will want to use the paid tier.



10 Product High Level Functionality

The high-level functionality of this product for non-profits is that it allows them to quickly deploy and share links to maps of locational information they have. Users looking for resources will be able to find the links via other pages or receive links from the non-profit, and search for specific resources that they need using any constraints they have.

10.1 Non-profit Flow

Non-profit organizations apply for a Where2Go account and subdomain. Once their application is vetted they receive their subdomain and a new blank map is generated for their subdomain. From there non-profits can upload their resources in bulk as a csv file. They can also add, edit, or remove locations one at a time, change the details they provide about their resources, or edit the schema used for the map's resources. They can manage their account to change their password or email and they can also share the link to their map with other organizations or people.

10.2 User Flow

Users find or are given a link to a non-profit's Where2Go subdomain. There they can see all of the non-profit's resource locations on a map and can enter their location to get results for nearby resources without logging in. They can also filter these resource locations using any custom fields that are specified by the non-profit such as days of operation, types of services offered, or specific requirements.

10.3 Demo

The video demo for the high level functionality can be found at `https://bit.ly/3bUSWyF`.

11 Technical Approach

We used the following technical stack for our project:

- React frontend
- Typescript + Node.js backend
- MongoDB database

This tech stack was chosen to minimize technical problems in the site, either through features of the technologies themselves or our familiarity in them, and ensure the level of flexibility that the product required would be possible. The code repository can be found at http://github.com/katiejiang/where2go.

11.1 Frontend

The frontend was written in React. We chose React for its flexibility, which would make it easier to build frontends supporting user-friendly flows. We used the React Bootstrap library for frontend components.

11.1.1 Mapping

For our mapping library, we chose to use Mapbox over the Google Maps API due to its lower cost. The features on the map pages are created entirely using Mapbox services through the mapbox-react-gl package. In addition, we utilize Mapbox's geocoding service to convert addresses of resources entered by the users into latitude-longitude coordinates that can be used for displaying locations on maps. We utilized a combination of the provided Mapbox client, as well as React wrapper classes built by Uber in their `react-box-gl` library.

11.1.2 Search

For the search bar, we built a React component which integrated the Mapbox Geocoding API to allow users to search for addresses easier. All searching is done on the frontend, so only one server load is required. We also used `Fuse.js` to support fuzzy search to improve the user experience, as users may not know the exact name of what they are looking for.

Fuse.js supports a weighted search, which allows the developer to allocate a weight to keys to give them higher (or lower) values in search results. We use the following weighting scheme to prioritize the name and address of a resource.

- If there are more than 2 custom fields: the weight of each custom field is 0.7/(# of custom fields). Name is assigned a weight of 0.2, and address is assigned a weight of 0.1.
- If there are fewer than 2 custom fields: the weight of each custom field is 0.3/(# of custom fields). Name is assigned a weight of 0.4, and address is assigned a weight of 0.3.

This weighting scheme was decided based on user testing. The threshold is set to 0.5 to allow for more fuzzy string matches.

11.2 Backend

We use `Node.js` for our server, with `Express`. The backend was written in Typescript, which adds static typing and is still able to take advantage of the ubiquity of Javascript. This gives us additional avenues pre-runtime to detect potential errors in our code. For user management and authentication, we used `Passport`.

11.3 Network Requests

Communication between the client and server were handled with the open-source `axios` library, which is simple to use, widely adopted, and has built-in protection against cross-site request forgery attacks. Using this library made it easier to ensure the correctness of our code, and more robust from a security standpoint.

11.4 Database

For handling the resource schema, we used MongoDB with `mongoose`, hosted on MongoDB Atlas. We chose to use a NoSQL database because it better supports our need for flexible resource schemas.

11.5 Resource Schema

Building a dynamic resource schema was the most difficult part of our project. We used the contents of the hash map to dynamically generate forms for adding and editing resources, validate input before updating our resource database, filter resources with user-defined parameters, and inform expected resource data structures for bulk upload and download. Then, within each resource document, field data values that conformed to this schema were stored in a hash map from the field name to data value. Making these hash map definitions compatible with our tech stack (Typescript and `mongoose`) was initially challenging, but allowed us to take advantage of type-checking to prevent future errors.

Dynamic resource schemas introduced many additional edge cases we had to think through. For example, how would we update resources if a resource schema changed over time? What if fields were added or deleted? Ultimately, we decided that all resources must keep up-to-date with the current maps resource schema. If a field were deleted from the schema, all the data values associated with that field would also be removed from the maps resources. Similarly, adding a non-optional field would require the map administrator to update all resources to include this new field.

11.6 Test Data Generation

We generated fake test data using the `faker` package. This enabled us to check if interactions with the database were returning the expected responses, when the client attempts to retrieve or input data in the most likely formats we would see, without having to generate hundreds of users, maps, and resources of our own.

12 Technical Evaluation

The goal of technical evaluation was to ensure that our site would not break regardless of user inputs, while also guarding user data and our systems from malicious users.

12.1 Husky Pre-commit Hook

Every time a commit is made, a linter is run before the code is actually committed. If any problems are detected, the commit is aborted and the user is alerted. This ensures that the code committed to the repository meets a minimum quality and correctness standard.

12.2 Integration Testing

Before code was merged into the master branch, we ran through the action flow that users were expected to perform with the newly-added features, to ensure that everything was working as it should in the process we had envisioned. Of course, there were strong possibilities that the user would behave differently, which was what our final technical evaluation component was for.

12.3 Edge Case Inputs

To account for possible malicious inputs, mistakes, or misunderstanding of instructions, we also tested the behaviour of the site with edge case inputs or actions. These could range from attempts at SQL injection, incorrectly-formatted data entries, or attempting to visit certain pages without authenticating. Unexpected site behaviour or failures to prevent certain dangerous actions were addressed with patches.

13 User Evaluation

We evaluated two functional components of our application. Due to the restrictions imposed by COVID-19, we were unable to complete our third user evaluation. For non-profits we evaluated the register, upload, and data manipulation features. For end users we evaluated the search and filter settings features. For efficacy we plan to evaluate Where2Go against Google Maps.

13.1 Non-profit Evaluation

This evaluation is meant to encompass the non-profit experience. The features we tested were register, upload, and data manipulation features.

In previous non-profit evaluations, we discovered that uploading data to maps was too complicated and long. There were also issues with nomenclature and a poor user interface. From the organizations' feedback, we developed a simpler and faster method to upload data and we simplified the user interface.

Due to scheduling conflicts, non-profits were unable to participate in our new evaluation. Instead, we had peers simulate the role of a non-profit.

We designed a simulation where we assigned mock non-profits the following task:

As a non-profit I want to:

1. Register for a Where2Go account and generate a new map.

2. Upload one hundred resources to my map.

3. Edit five resources.

We had six mock non-profits split into two groups of three. The first group was assigned to use Google Chrome for the evaluation. The second group was assigned to use Firefox. Although there are several other web browsers, the most common desktop web browsers are Google Chrome and Firefox (Statcounter, 2020). We only used desktop web browsers in our evaluation since those are the needs for non-profits as discussed in Section 2. Before the evaluation, each mock non-profit was shown a tutorial to accomplish each milestone. Each one received an identical csv file that contained one-hundred rows of resource information.

For the evaluation, we measured the amount of time each mock non-profit took to complete each milestone. We took note of any incidents where a mock non-profit was unable to complete their task. We also measured the perceived level of difficulty of each milestone using a Likert scale. Our Likert scale ranged from one to five with one being ”very easy” and five being ”very difficult.”

Based on previous evaluations, our baseline time to complete each milestone is 120, 63, and 300 seconds respectively. We held the same baseline for both web browsers since we had non-profits use Google Chrome in our previous evaluation. We did not expect significant differences between the performances of both browsers. Since the bulk uploading feature was not available in the previous evaluation, we decided to use the average time to upload one resource as a baseline.

For the Google Chrome group, the average time spent on each milestone was 80, 17, and 250 seconds respectively. The average total time to complete the entire task was 346 seconds. For the Firefox group, the average time spent on each milestone was 84, 16, and 251 seconds respectively. The average total time to complete the entire task was 351 seconds.

Compared to our baseline, mock non-profits were able to complete their task more quickly. We observed that users had a positive reaction to the user interface, which was more simplified and required less user events to complete an action.

The baseline for the level of difficulty for each milestone is 2, 5, and 4 respectively. As mentioned previously, non-profits had a difficult time attempting to upload a resource. For this reason, the second milestone has a higher baseline. Due to user interface issues, non-profits also found it difficult to edit resources on their map but it was not as difficult as uploading a resource. For this reason the third milestone is ranked lower in difficulty relative to the second milestone.

For the Google Chrome group, the average level of difficulty to complete each milestone was 1, 1, and 2.3 respectively. The average level of difficulty to complete the entire task was 1.4. For the Firefox group, the average level of difficulty to complete each milestone was 1.3, 1, and 2 respectively. The average level of difficulty to complete the entire task was 1.4.

Compared to our baseline, mock non-profits had less difficulty completing each task. The most significant positive difference was with the second milestone, since the user simply had to upload a file to upload one-hundred resources.

See Appendix A for more details.

13.2 Finding Resources

This evaluation is meant to encompass the end user experience. The features we tested were the search and filter settings.

In previous end user evaluations we discovered that many had difficulty using the filter settings. The user interface was also an issue. With this feedback we developed a simpler interface and a more intuitive filter settings for maps.

We designed a simulation where we assigned mock end users the following task:

As an end user I want to find the closest resource with the following conditions:

1. The resource is a food access program.

2. The resource is available on Mondays.

We had six mock end users split into two groups of three. The first group was assigned to a map with a high concentration of locations. In their map, there would be five matching locations within a three block radius from their home. The second group was assigned to a map with a wide spread of locations. In their map, there would be five matching locations within a fifteen block radius from their home. We assigned the same home address to each user. Each user had to select the closest matching location to complete the task.

Each group's location data set included an additional five locations that satisfied one of the conditions, but not both. Each group was given the same tutorial for using the search and filter features.

For the evaluation, we measured the amount of time it took for mock users to complete the task. We also took note of any incidents where users were unable to complete their task. We also measured the perceived level of difficulty of the task using a Likert scale. Our Likert scale ranged from one to five with one being ”very easy” and five being ”very difficult.”

Based on current solutions, our baseline time to complete this task is 240 seconds. The reason being that end users have to visit multiple websites on Google Maps to find the custom resource information. Even if a list is used, end users still have to search each item on the list to find that information.

We held the same baseline for both groups since we did not expect significant differences in the results.

For the high concentration group, the average time spent to complete the task was 150 seconds. For the wide spread group, the average time spent to complete the task was 164 seconds. Compared to our baseline, mock end users were able to complete their task more quickly.

Based on current solutions, our baseline level of difficulty to complete this task is 4. Besides having to visit each site to find the custom resource information, users might find it difficult to search a map that has a high concentration of locations. It could also be difficult if the map has a wide spread of locations. We kept the same baseline for both scenarios since each scenario can make it difficult to perform the task.

For the high concentration group, the average level of difficulty to complete the task was 2.3. For the wide spread group, the average level of difficulty to complete the task was 3.3. Compared to our baseline, mock end users found it easier to complete their task. Although the wide spread group found it more difficult than the high concentration group, the results are not in the ”difficult” zone of 4 and 5 on the Likert scale.

13.3 Effectiveness

This evaluation is meant to compare the effectiveness of Where2Go compared to Google Maps. The reason we chose Google Maps as our benchmark for this evaluation is because they are the primary alternative to Where2Go that is also more effective than lists.

We planned to have this evaluation be similar to the one conducted in Section 13.2. The difference would be that we would need to use different data sets.

We would have two groups of three mock end users. One group would be assigned to use Where2Go and the other group would be assigned to use Google Maps. Their task would be to find the closest resource that matches a set of given conditions.

Since our mock data used in Section 13 does not match the data from Google Maps we would need to modify our data set to match that of Google Maps. Our plan is to find a set of locations on Google Maps that could be used for our simulation. From there we would generate the data set for Where2Go and we would also need to add the custom resource information for each location. Since Google Maps is not customizable, it would not have this custom resource information. To get this information, we would need to visit the websites of these locations. Additionally, we would need to modify the home address to better fit the simulation.

For this evaluation, we would measure the time it takes for each group to complete their task. We would also measure the perceived level difficulty that it takes to complete their task. Our Likert scale would have a range of one to five with one being ”very easy” and five being ”very difficult.”

We expect the time it takes for the Where2Go group to complete task to be significantly lower than that of the Google Maps group. This is mainly due to the conditions for the filter settings. Since the Google Maps group would not have this filter tool, they would need to search each location’s website to find this custom resource information. For similar reasons, we expect the perceived level of difficulty to be lower for the Where2Go group.

14 Societal Impact

Where2Go’s services have positive and negative implications for society. The implications are in regard to the use of information, administrators, accessibility, and differentiation. We also provide mitigation strategies for these negative implications.

14.1 Use of information

Where2Go enables non-profits to better communicate resource information to users. Since this information is available to the public there are positive and negative implications.

The positive implication is that users will become more aware of helpful resources. These resources, such as free dental services or therapeutic social meetings, can improve the quality of life for those who need them. By providing this information in a centralized repository, people will spend less time scouring multiple websites attempting to find specific resources.

A concern about the availability of this information is that it can be used as a tool to target non-profits. We can easily imagine a scenario where someone with malicious intent uses our site to target LGBTQ+ non-profits. Although we understand this concern, we also understand that much of this information is already online. Even without Where2Go people will still be able to find these resources, though it would take them more time and effort.

14.2 Non-profit Verification

For our tool to help users find non-profit resources, it is essential that administrators are non-profit organizations. Applicants are verified by Where2Go manually and this method has both positive and negative implications.

A positive implication is that manual verification can bring more security to our system. Having any applicant receive an account puts users at risk from malicious entities or for-profit corporations that may seek to abuse this tool.

One concern is that there can be a mistake in manually verifying an applicant. It might be that a malicious entity posed as a non-profit and somehow bypassed Where2Go security measures and received an account. From there they can upload false or misleading information on their map that can hurt people. To mitigate this, we've established a communication channel with current account holders and users. If anyone has an issue with a specific account holder they can send an email to Where2Go where a re-verification process takes place.

Another concern is that bias will influence the manual verification process. A Where2Go verification member could have a political bias that would prevent them from verifying a legitimate non-profit. To mitigate this, we've also established a communication channel with applicants. An applicant can email Where2Go to settle any disputes. We recognize that this strategy is not the most effective if the entire Where2Go verification team is biased towards one way. For this reason we will stress to the team that the verification only concerns non-profit status. Automating this verification process is a possibility further discussed in Section 15

14.3 Administrators

Since the administrators are in control of what resource information to upload to their maps, they play a central role in Where2Go. This level of control has both positive and negative implications.

The positive implication is that non-profits can make quick edits to the information they provide. Many non-profits have restricted the availabilities of their resources due to the novel corona virus. To protect the wellbeing of their community it is important that they can update this information on their map as quickly as possible.

One concern is that malicious entities might pose as a non-profit to receive administrator privileges. From there they can upload false information that could hurt the community. To mitigate this, applicants will be vetted manually. As the application scales, however, we will need to automate this process.

Another concern is that an administrator may upload erroneous information onto their map. We mitigate this by prompting the user to confirm actions such as adding or deleting information on their map. They can also edit any information to correct errors.

14.4 Accessibility

Since we are serving communities that may be vulnerable, it is important that they have accessibility to our site.

Although visually impaired people will not be able to fully benefit from our map, we have formatted our HTML so that they can navigate through our site. We've also enabled a list view access for their text-to-voice aids.

Another concern is that our site does not accommodate non-English speakers. Some members of the community may not read English proficiently and this can prevent them from benefiting from our site. Although we do not have a strategy to mitigate this, we depend on these members having relatives or friends that can help them navigate through the site. A future mitigation strategy would be to use the Google Translate API which we discuss in Section 15.

Another concern related to the language barrier is the technology barrier. Some users may not be computer literate or even have access to the internet. Since some members of the community may not have own a smart phone, we've created our site to be suited for desktops. That way people can access our site through the libraries' computers. We also provide guides on how to navigate through our site.

Additionally, non-profits will require guidance through the sign up process and using their map. We provide instructions at every step of the process so that administrators can quickly begin using their maps.

14.5 Differentiation

The most common concern about our site is that it is difficult to differentiate its benefits from Google Maps. To mitigate this issue we have provided information on our website to educate our users about the benefits of Where2Go. We also hope that the benefits of our tool will spread word-of-mouth around local communities. We have also implemented our site to have a different appearance from Google Maps.

14.6 Overloading Resources

There is concern about how we would choose which non-profits to display first in search results if our product receives widespread adoption. Since the resources closest to the user are displayed, there might be a possibility that a resource could receive heavier traffic than other well suited resources. This could also be perceived as unfair or suboptimal if there isn't enough resources to help all users directed to them. Although we could modify our ranking system to tackle this concern, we currently do not have a reliable method to track which users go to which resources. For this reason, we give more control to the administrator so that if a resource is being overburdened, they can temporarily remove that location from the map or add a note to the resource information.

15 Discussion and Lessons Learned

In Section 13 we found that mock non-profits could perform their task more quickly and easily than the baseline. This is due to the changes we made to simplify the user interface and the process of performing each task. There are some caveats to our evaluation. The first is that all the mock non-profits were college students between the ages of twenty and twenty-three. Since we expect users of various ages to use Where2Go, we expect discrepancies in the speed and difficulty of each user to complete a task. The second is that for the resource upload, the mock non-profits already had a csv file to upload. Non-profits would be required to have their own csv file to bulk upload. This would significantly increase the time spent in uploading resources, but there are still benefits over one-at-a-time resource uploading. The first is that if the non-profit where to accidentally delete their map, the could create a new map and simply re-upload the csv file they originally used. If the non-profit had made any changes to the map, they could download a formated csv file of their resource information. We also mitigate the difficulties of creating a new csv file by providing a template on our site.

As for the end user evaluations, we found that mock end users could perform their task more quickly and easily than the baseline. This is mainly due to the custom resource information filter feature. There are some caveats to our evaluation. The first is that location densities on the map vary depending on the resource being searched and specific conditions imposed through the filter. The second caveat is that the demographic profiles of our participants might not match the demographic profiles of actual end users. As discussed in Section 14.4, there are end users that may not be skilled at surfing the web or reading English proficiently. This could make it more difficult for them to use our filter feature to find specific resources. For now we depend on their English-speaking relatives or friends to help them navigate the site.

Overall our evaluations did show significant improvement in the speed and ease in searching locations with specific filter settings. Based on the results, we were successful in satisfying user needs for an efficient and easy to use solution. From Section 9 we were also successful in satisfying user needs for a solution that is also inexpensive.

There were difficulties in developing our project. The first difficulty we experienced was getting our user interface to satisfy the need of ease of use. Through multiple iterations we were able to simplify each user action so that non-profits could get their map set up as fast as possible. Another difficulty was in adjusting the features we wanted to implement. Like every other senior design group, we were very optimistic about what could be accomplished given the time constraints and course load. We soon faced scheduling conflicts and backlogging of features. After a long discussion, we were able to identify what was most important to develop and what could be left for future iterations which is discussed in Section 15.1. The novel Corona virus also made our project development difficult since we were used to pair programming and now we are quarantined in our homes in different time zones. We also faced difficulties in our developing a dynamic resource schema as discussed in Section 11.5.

15.1 Future Work

There are still more features and improvements we could develop for Where2Go. After we test our application with real non-profits and users, we determine whether a stand alone tool works best. Depending on this evaluation, we would consider developing an extension tool that non-profits can embed on their website. It is also a possibility that a mobile application would best compliment the web application.

In Section 14.4 we mentioned that some users will find it difficult to navigate through our site if they do not read or write English proficiently. A future improvement would be to integrate Google Translate into our site.

We could also develop a scalable verification system for applicants. We considered using a peer verification system where existing account-holding non-profits can verify applicants. The drawback of this strategy is that there might be non-profits with juxtaposing missions. This could create an environment where the verification tool is used to prevent other non-profit organizations from creating an account. The most ideal solution is to utilize the IRS Tax Exempt Organization Search tool to automatically verify non-profits. This strategy would also mitigate the concerns discussed in Section 14.2.

A more ambitious feature to develop is a ranking system that could mitigate the negative implications discussed in Section 14.6. One possibility is attempting to find a way to track users and the resources the use on Where2Go. This would involve extensive changes to our application and will not be accurate enough to provide meaningful changes. Currently, the administrators would be the ones to handle the resource overload by temporarily removing locations. We could develop an automated system whereby an administrator could communicate with Where2Go about an overload issue at one of their locations. From there the system would automatically update rankings to resolve the conflict.

16 Acknowledgments

We want to thank the following non-profit organizations that helped refine our definition of the problem and the needs for a solution: York County Reentry Coalition, SERVE Philadelphia, Galaei, Welcoming Center for New Philadelphians, Aquinas Center, and JUNTOS.

We also want to thank Dr. Sangeeta Vohra and Dr. Chris Murphy for providing us an initial direction with starting our application.

We also want to thank our fellow University of Pennsylvania peers for participating in our mock user evaluations and providing helpful feedback to improve our application.

References

Esri. (2020). Buy GIS Software: ArcGIS Product Pricing - Esri Store. Esri. Retrieved from https://www.esri.com/en-us/store/overview

Free Dental Care. (2020). Philadelphia, Pennsylvania Free Dental Care Clinics. Free Dental Care. Retrieved from https://www.freedentalcare.us/ci/pa-philadelphia

Google. (2020). Pricing Table - Google Maps Platform - Google Cloud. Google. Retrieved from https://cloud.google.com/maps-platform/pricing/sheet/

Healthy Minds Philly. (2020). Resources: LGBTQIA Resources. Healthy Minds Philly. Retrieved from https://healthymindsphilly.org/en/lgbtqia-resources/

McKeever. B. (2019, January 3). The non-profit sector in brief. National Center For Charitable Statistics. Retrieved from https://nccs.urban.org/project/nonprofit-sector-brief

Statcounter. (2020, April 1). Desktop Browser Market Share Worldwide. GlobalStats. Retrieved from https://gs.statcounter.com/browser-market-share/desktop/worldwide

Thumbtack. (2019, September 9). How much does web development cost? Thumbtack. Retrieved from https://www.thumbtack.com/p/website-development-prices

A Appendix A. Mock Non-profit Evaluation Data

Legend
Milestone 1 (M1): Register for Where2Go account and generate a new a map
Milestone 2 (M2): Upload 100 resources to map
Milestone 3 (M3): Edit 5 resources
Group 1 (G1): Google Chrome mock non-profits
Group 2 (G2): Firefox mock non-profits

Time taken to complete each task (seconds)

	G1 user 1	G1 user 2	G1 user 3	Average
M1	85	76	79	80
M2	15	18	17	17
M3	220	274	255	250
Total	320	368	351	346

Time taken to complete each task (seconds)

	G2 user 4	G2 user 5	G2 user 6	Average
M1	91	78	83	84
M2	21	16	12	16
M3	256	244	253	251
Total	368	338	348	351

Level of Difficulty for each task (Likert Scale 1 - 5)

	G1 user 1	G1 user 2	G1 user 3	Average
M1	1	1	1	1
M2	1	1	1	1
M3	2	2	3	2.3
				1.4

Level of Difficulty for each task (Likert Scale 1 - 5)

	G2 user 4	G2 user 5	G2 user 6	Average
M1	2	1	1	1.3
M2	1	1	1	1
M3	3	2	1	2
				1.4

Figure 1: Results from mock non-profit evaluation.

B Appendix B. Mock User Evaluation Data

Legend
Task 1: Find nearest food access program that is open on Mondays
Group 1 (G1): High Concentration Map
Group 2 (G2): Low concentration Map

Time taken to complete each task (seconds)

	G1 user 1	G1 user 2	G1 user 3	Average
Task 1	143	145	162	150

Time taken to complete each task (seconds)

	G2 user 4	G2 user 5	G2 user 6	Average
Task 1	161	164	166	164

Level of Difficulty for each task (Likert Scale 1 - 5)

	G1 user 1	G1 user 2	G1 user 3	Average
Task 1	2	2	3	2.3

Level of Difficulty for each task (Likert Scale 1 - 5)

	G2 user 4	G2 user 5	G2 user 6	Average
Task 1	3	4	3	3.3

Figure 2: Results from mock end user evaluation.