# Crime Respective Route Generation in Dense Urban Areas

Anabelle Chang

Advisor: Professor Brian W. Kernighan

Submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Arts
Department of Computer Science
Princeton University

May 2021

I hereby declare that I am the sole author of this thesis.

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Anabelle Chang

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Anabelle Chang

# Abstract

Individuals who walk as their primary source of transportation may experience concerns about their safety, especially as sexual harassment and assault continue to occur and menace society. In light of raising concern about this issue, there have been a host of solutions that attempt to enhance urban safety and security; however many of them are lacking in functionality and features to sufficiently do so. This thesis seeks to help individuals feel safer in major metropolitan cities by creating an application that provides users with navigation routes along which there are minimal amounts of crime.

Acknowledgements

To my incredibly kind thesis advisor, Professor Brian W. Kernighan, thank you for your support and advice, not only for this thesis, but for the entire duration of my Princeton career. I'm eternally grateful for your encouragement and warmth, and I hope your walks along Springdale will always be full of excitement and wildlife!

To my parents, thank you for all the sacrifices you both have made to open these opportunities for Alison and me. Your unwavering support, unparalleled compassion and care, and steadfast belief in me are things I will forever cherish. Alison, the last four years at Princeton would not have been the same without you. Thank you for being my rock even in the darkest of moments (and when you had so many things going on for yourself) and for your sweet kindness - I'm lucky to call you my sister.

Golf has been an indelible part of the majority of my life, and it has been an even more irremovable aspect of my college experience in the last four years. To all my teammates, thank you for the love and laughter on and off the course, for making all those off campus trips tremendously special, and for making the golf course feel like home. To Coach DeSanty, thank you for all the sagacious advice and for your confidence and optimism even when morale was low.

Finally, to all the friends that I have made in my time at Princeton - thank you. Each and every one of you have immensely added to my experience, and I'm so very grateful to have such steadfast forces, even through the most tumultuous of times, in my life. I love you all.

To my loving family

# Contents

# List of Figures

# Chapter 1

# Introduction and Background

## 1.1 Problem Description

Crime is unavoidable; the amount of crime in an area is typically a consideration in decision making for individuals who walk as their primary source of transportation or for individuals who are new to cities. Studies even go as far as to show that women are far less likely to feel safe while engaged in normal activities and six in 10 women preemptively taking measures to protect themselves (Figures 1.1, 1.2) [1]. This statistic and many more described in Chapter 1.2 begets the need for a resource that actively mitigates risk and promotes safety for these individuals.

## 1.2 Problem Background

The motivation for this project stems from the lack of a safety resource for those living in largely urban or metropolitan areas. In my own experience, I have had uncomfortable encounters with random strangers while walking in urban areas and being navigated by the Google Maps or Apple Maps applications. Statistically, this holds also holds true; in 2015, Professor Beth Livingston of Cornell University conducted a research study with data from over 16,600 women in 42 cities globally

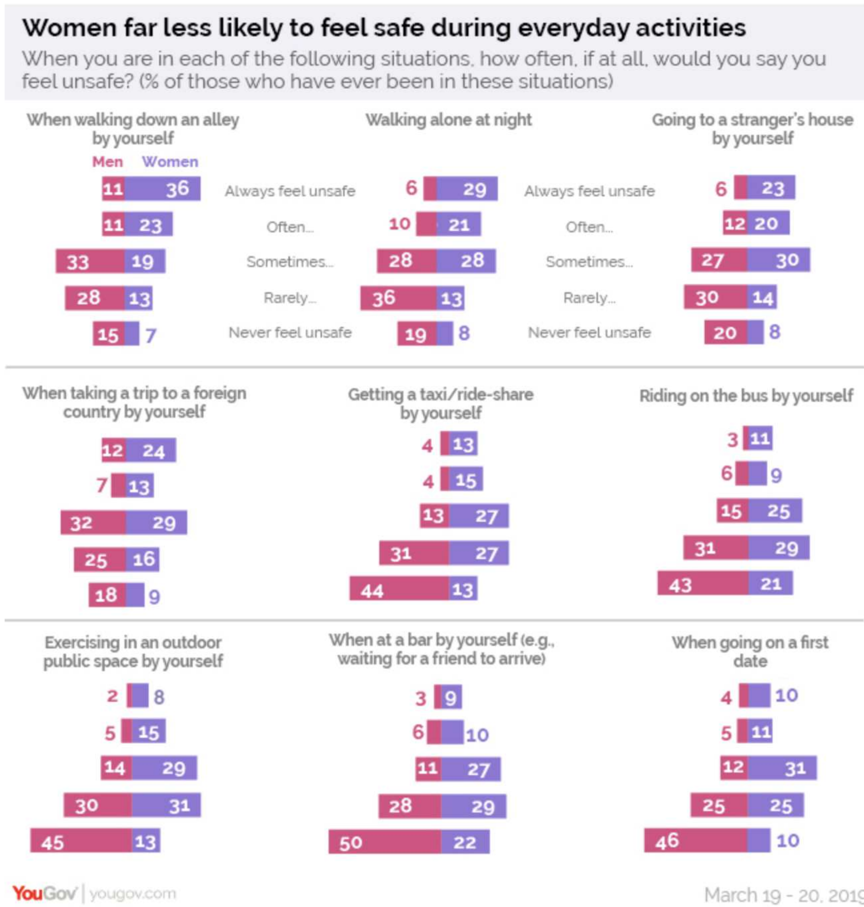


Figure 1.1: How safe women and men feel during typical everyday activities. [1]

and her findings indicated that 85% of women in the United States experience street harassment before the age of 17 and 77% under age 40 reported being followed by a male or group of males in a way that made them feel uncomfortable and unsafe [4].

Unfortunately, this problem has no signs of slowing down. In a 2018 survey launched by the nonprofit Stop Street Harassment, it was discovered that 81% of women and 43% of men had experienced some form of sexual harassment in their lifetime. Out of these harassment experiences, 77% of women were verbally assaulted, 51% claimed unwelcome sexual advances and touching, 34% were physically followed, and 30% experienced unwanted sexual assault. A majority of these experiences - approximately 66% for women - were borne out of being in public spaces. Further

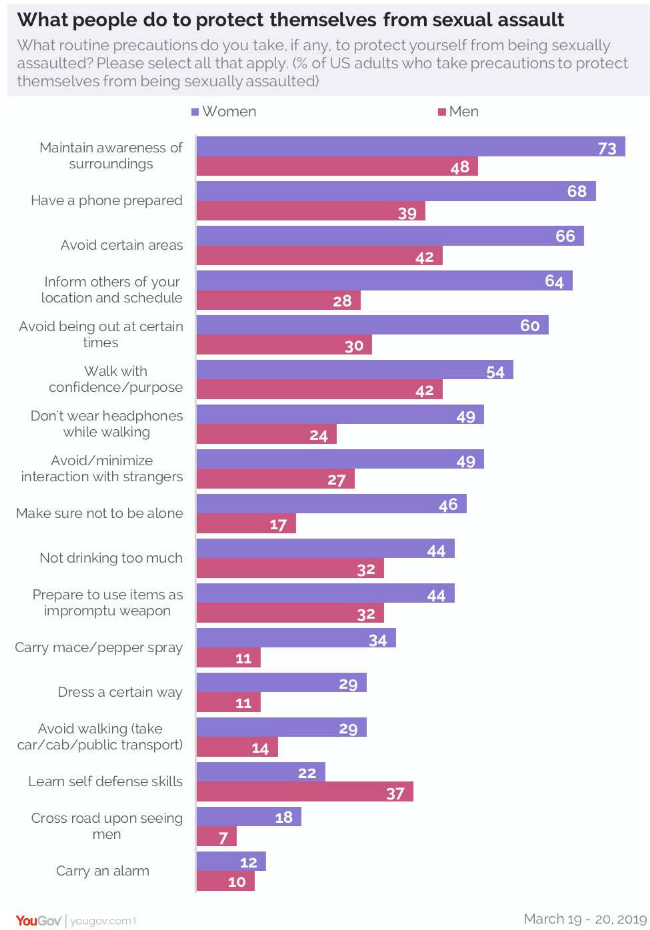


Figure 1.2: Precautions taken preemptively to protect against sexual assault. [1]

surveys from the same study indicated that these experience had long-lasting negative effects on these women, including increased feelings of anxiety or depression and negative life changes, such as stopping a hobby or seeking a new job (Figure 1.3) [2].

According to FBI data, while violent crimes such as murder, robbery, and manslaughter have steadily decreased, the rate of rape in the United States has slowly risen as an overall trend [5]. Additionally, rape is the most unreported violent crime, suggesting that the numbers are higher than those currently reported [5]. This data is concerning, since it suggests the lack of protective resources for those who may be vulnerable to such a transgression.

Even more recently, there has been a surge of COVID-19 related hate crime against

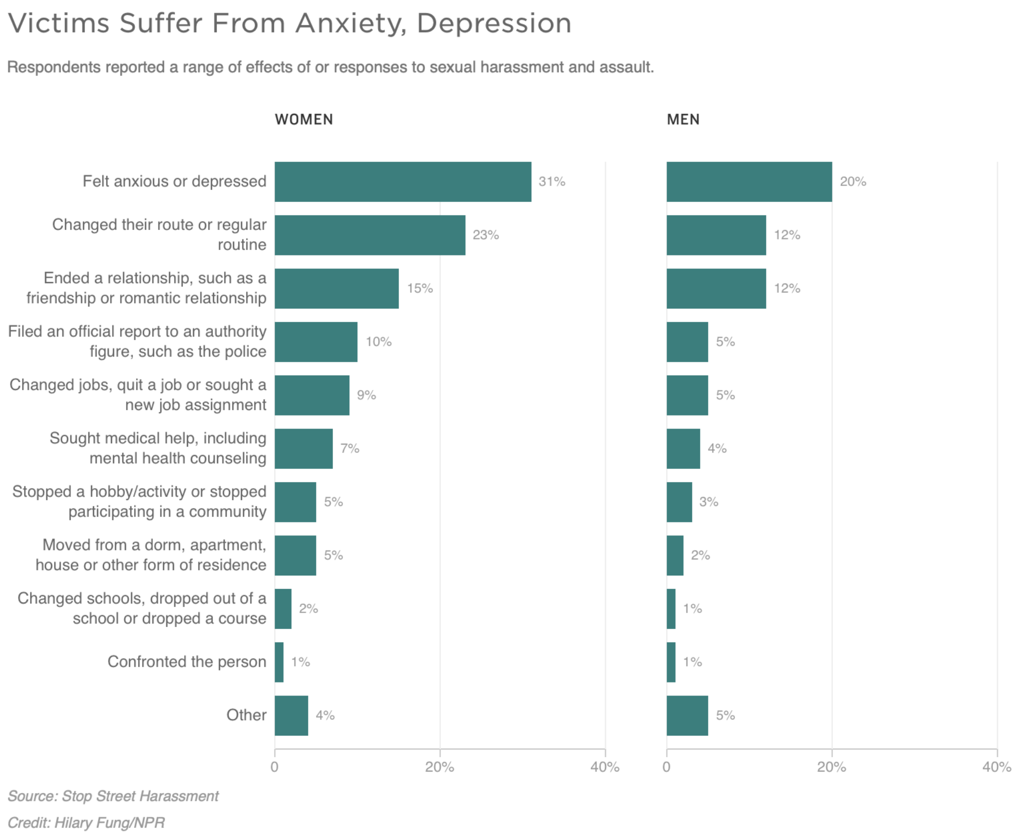


Figure 1.3: Surveyed lasting effects of harassment and assault [2].

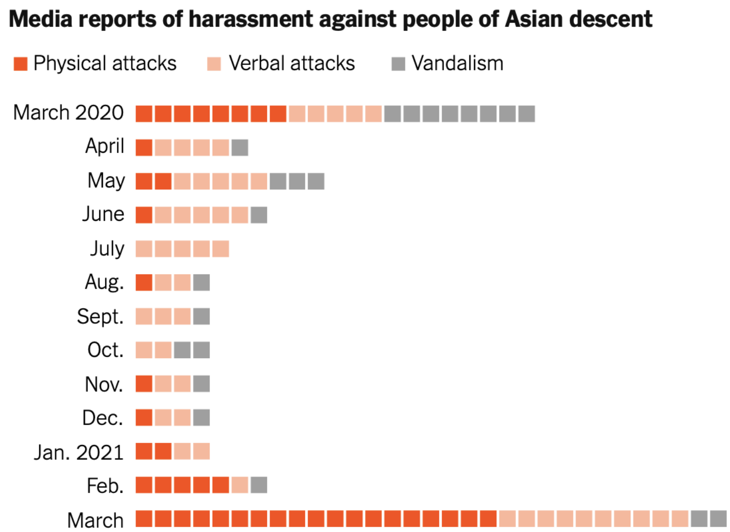


Figure 1.4: Rising hate crimes targeting Asian Americans [3].

Asian Americans. In late March 2021, a gunman opened fire in three spas in the Atlanta area, killing 8 women [6]. In addition to this, there have been a countless number of physical attacks against Asian elderlies, including a "fatal attack of a Thai man in January 2021, as well as the assaults of a 91 year old man in Oakland's Chinatown, and a 89 year old woman in Brooklyn" [3]; this trend of targeted Asian hate crimes has been steadily rising (Figure 1.4).

These hate crimes have been reportedly occurring in large cities such as New York City, Los Angeles, Seattle, Boston, San Jose, and Dallas. The incidents are also typically physical and verbal attacks in public settings, such as on subways, in stores, on the street, or store-front vandalism on Asian-owned businesses [3]. This is a deeply worrying issue, as the attacks continue to escalate in number and severity without resolve.

What strikes as most concerning is that these types of attacks - the verbal, sexual, and physical assaults - all occur in visibly public spaces, meaning that the attackers are not fearful of being witnessed in the crime and have no moral bearings of being good citizens, treating others with respect, and following the law. Additionally, as rape and hate crimes continue to trend upward, it is becoming more and more evident that efforts with our current law enforcement and activism to try to curb these issues have been fruitless.

## 1.3 Motivation and Goal

The steady street harassment statistic and rising number of hate crimes lends to these being issues that are both pervasive and unyielding. And, as an Asian woman, I find myself increasingly fearful of walking around in cities because of the potential of being harmed or harassed. The goal of this project is twofold: first, to create an application that draws from and displays crime data on a map and allows users to map a route that maximizes his/her safety, and second, to thoroughly document the process of creating such an application with reasoning behind design decisions, to create a "handbook" of some sort. By doing so, the end product will be an application that hopefully helps individuals feel more at ease when walking alone, as well as a way for programmers to learn from my experience and develop an intuition to ask the pertinent questions when making design and algorithmic decisions.

# Chapter 2

# Related Work

## 2.1 Existing Solutions

Several applications have been released in attempt to help communities stay aware of nearby crime [7].

### 2.1.1 LexisNexis Community Crime Map

LexisNexis Community Crime Map plots reported crimes and provides a breakdown of the crime statistics for a given city; users are able to enter an address and see the nearby crimes, including assault, burglary, and theft (Figure 2.1) [8]. Users are able to click on each marker on the map and the relevant offense information will be revealed. While the map has a wide variety of city data available, upon further inspection, much of the data use outdated reports between 2016 and 2017.

The problem with LexisNexis is that the data is outdated and the interface is not the most user friendly. It is also a rather passive solution, since it does not directly help users improve safety but simply informs them of the criminal activity in their area.

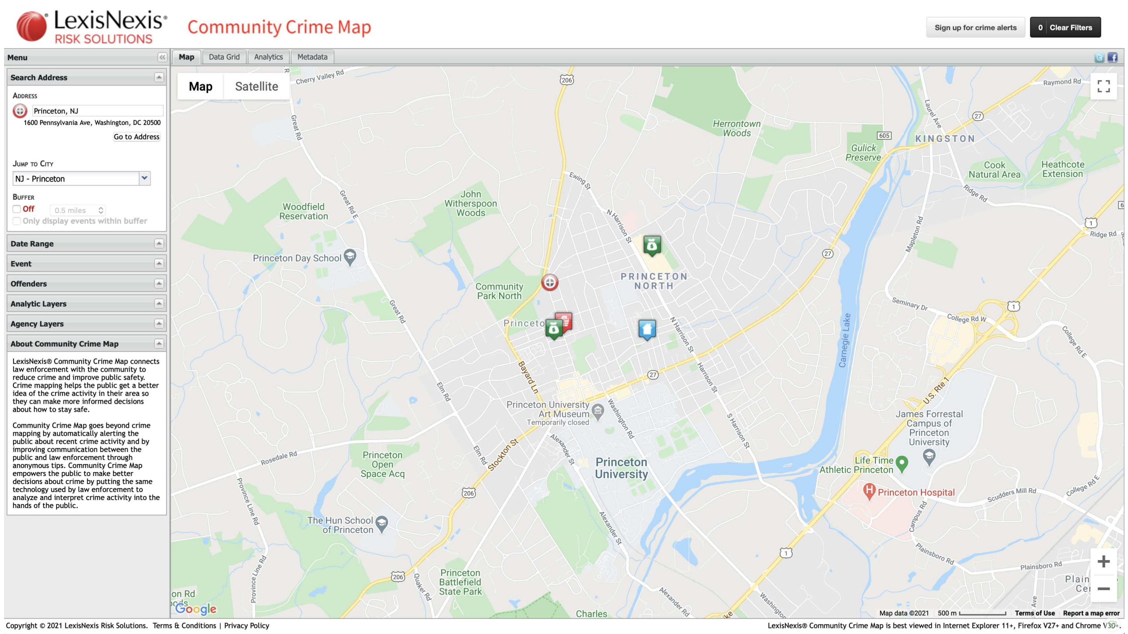


Figure 2.1: LexisNexis Community Crime Map example.

### 2.1.2 Neighborhood Scout Community Crime Map

Neighborhood Scout also uses crime data to create a density map overlay for large cities (Figure 2.2). In order to see the crime statistics of specific regions and neighborhoods, users have to pay a subscription to access. The website prides itself for its "uniquely accurate" data with a "relational database to assess the true count of reported crimes in a locality" and their data processing method allows the website to "accurately fill in the holes based on the crime experience of many like locales, and provide accurate crime data for anywhere in the U.S." [9]. The types of data used to populate the Neighborhood Scout map are: burglary, larceny-theft, vehicle theft, homicide, rape, armed robbery, and aggravated assault.

Neighborhood Scout is an excellent idea in principle, but it is incredibly inaccessible to the general public with the requirement of an expensive paid subscription. When used, it could exacerbate the issue of over policing, since the website directly states that it creates specific Crime Risk Ratings for each neighborhood report, essentially drawing potentially unfair and biased conclusions from inherently

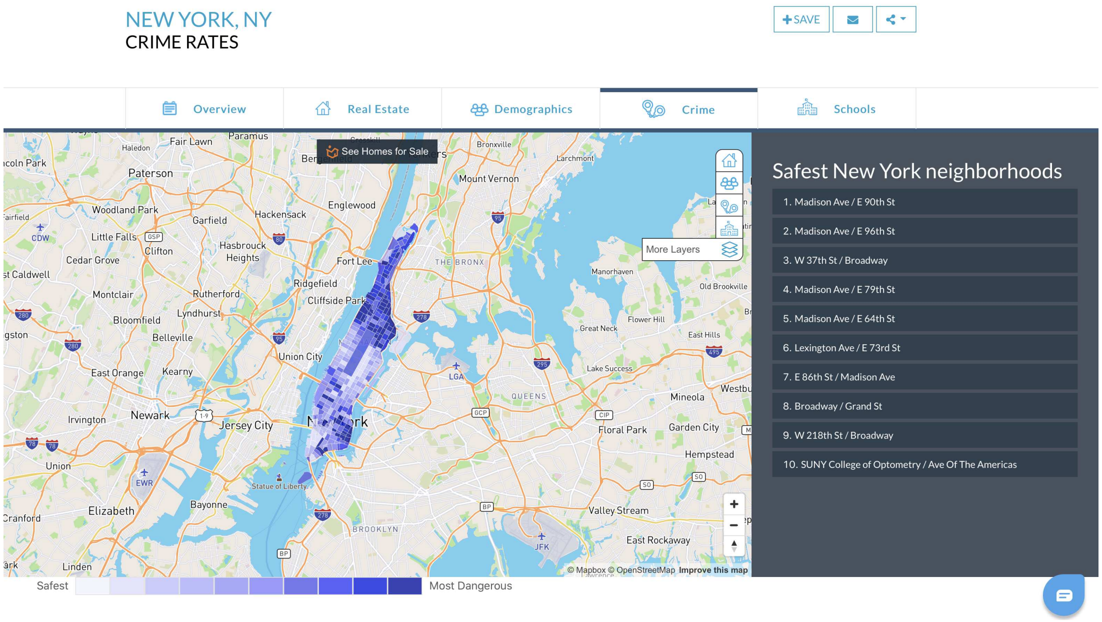


Figure 2.2: Neighborhood Scout crime density map example.

discriminatory data. [9].

### 2.1.3 CitizenCOP

CitizenCOP is an India-based application that allows users to report nearby crime and communicate with local law enforcement for limited cities, such as Bhopal, Indore, Jhansi, Raipur, Noida, Bengaluru, and Navi Mumbai [10]. For cities not supported by CitizenCOP, users can access the "Lite Version," in which they are limited are able to sending SOS messages and their location to contacts of their choosing, rather than contacting the police, reporting incidents, and viewing nearby incidents, features which are available in the "Full Version" (Figure 2.3). CitizenCOP offers a more direct solution and encourages quick responses to and awareness of crime, allowing users to contact help when in danger and to view nearby reports.

This application promotes user awareness of their surroundings, but in my personal opinion, offers a solution too late – only when the user him/herself is a victim or witness of a crime happening.

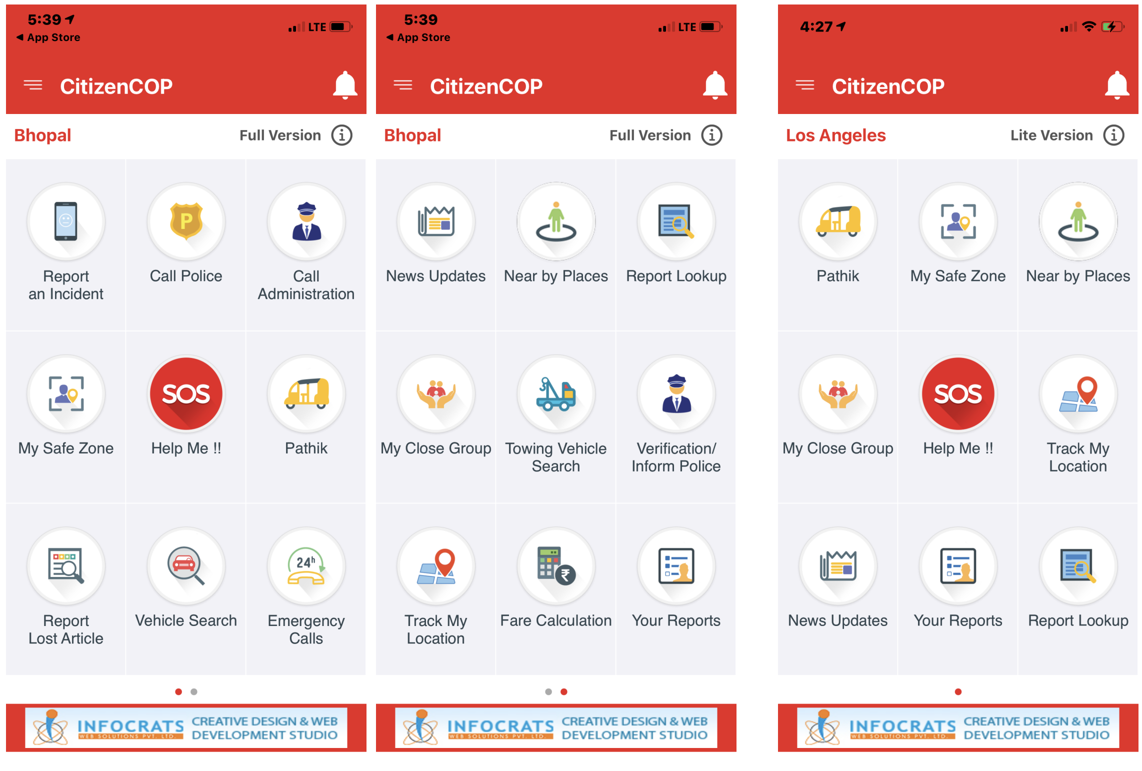


Figure 2.3: CitizenCOP options menu for the "Full" (left) and "Lite" (right) versions of the app.

### 2.1.4 Google Maps

There have also been attempts to directly solve the specific problem of finding safer paths in navigation apps. In 2019, developers found through the Google Maps beta code that Google was going to implement a "lighting layer" for the application, which would help users navigate themselves through areas with brighter walkways [11]. Unfortunately, two years later, this feature is still not available to the general public. Even if it were, however, it is unclear whether or not the lighting would be factored into navigation routes for pedestrians.

### 2.1.5 Brightpath

Last but not least, there are a host of online solutions that try to create safer walking paths. An example of this is Brightpath, a navigation application created through a hackathon. Brightpath takes the Google Maps approach of using the lighting of roads to output well-lit walking paths in London, helping users avoid darker passages where sexual assault incidents are more likely to occur. This clever solution parses through open source map data, processes it to remove dark passages, and finds the shortest path between source and destination using the A* algorithm [12], [13]. While this is an excellent resource for newcomers to a city or anyone who is walking alone at night, it fails to acknowledge that even well-lit areas can be highly trafficked by sex offenders.

# Chapter 3

# Approach

With the recent increase in violent attacks against Asians and the historic trends of physical and sexual assault against women, this project attempts to solve one aspect of the larger issue of crime by focusing on maximizing safety for individuals in metropolitan areas. This is done by aggregating crime data for four large urban cities, plotting it in a map, then running an algorithm that determines the least crime ridden route from origin to destination.

Initially, this project aimed to be an iOS application. However, unforeseen issues with the iOS MapBox SDK forced me to pivot this project in the direction of a web application. While web apps have certain limitations, creating one broadened the horizon in terms of accessibility, since web apps can be easily bookmarked to a home screen from devices of all types. This would allow anyone with internet access to use the application.

The completed project can be visited at:

https://belle-chang.github.io/crime-navigation

As seen in Figure 3.1, the user is taken to a web page highlighting the crime in four major cities: San Francisco, Fort Worth, Chicago, and Boston. Upon clicking a city in the lower left hand menu, the map will fly to a zoomed in view of the selected city

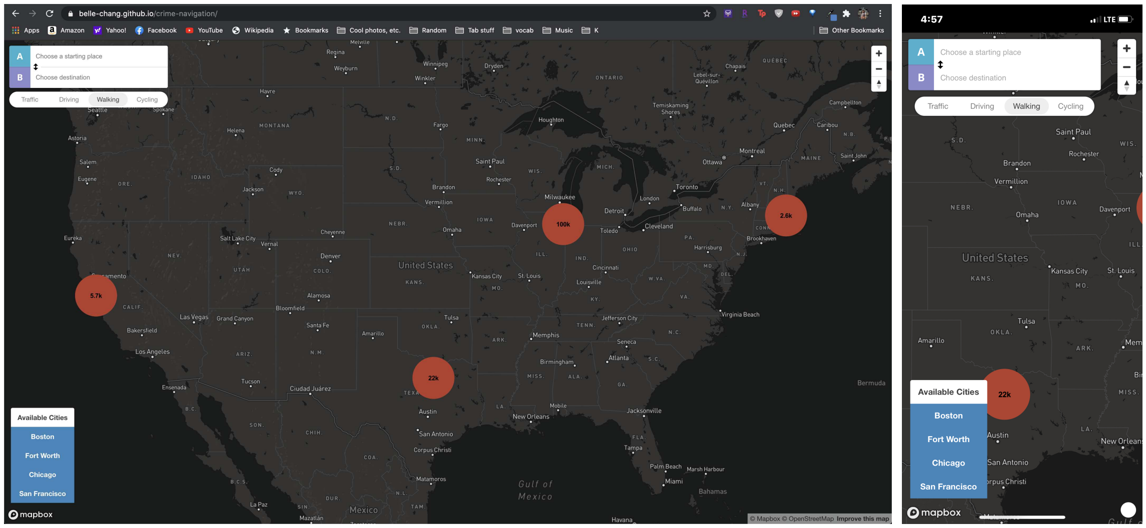


Figure 3.1: Start screen in the desktop (left) and mobile (right) views.

(Figure 3.2). The user can then input an origin and destination, and the application will output the calculated route with the lowest number of intersections with the addresses of reported crimes (Figure 3.3).

This approach differs from the previously discussed in Chapter 2, since the web app is an active solution that can be used by the general population free of charge. There are currently a very limited amount navigation applications on the market that create routes with consideration to human safety. It explores a relatively untapped domain and provides a successful way to better navigate a metropolitan city, as a modern resolution to improve human welfare.

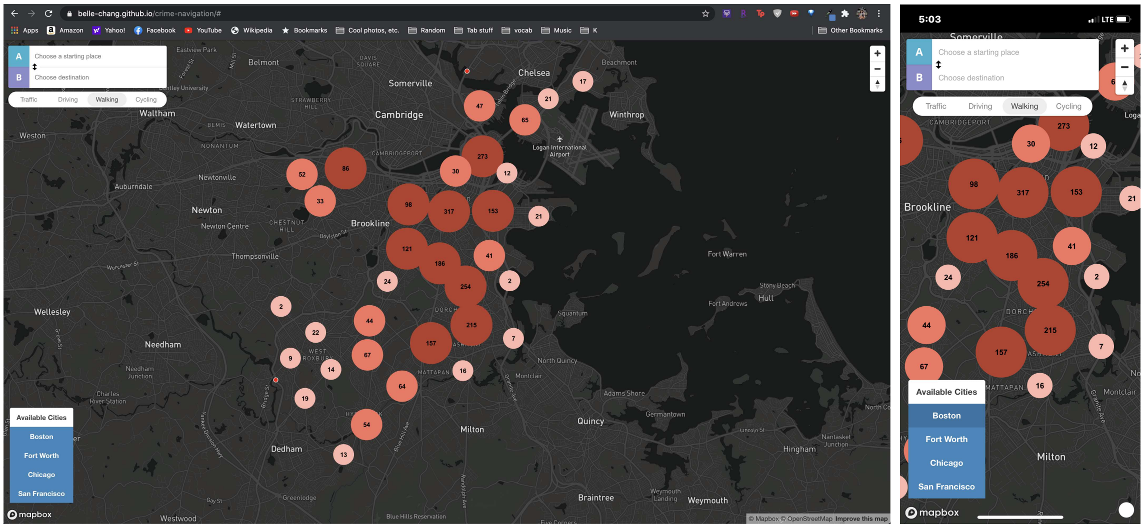

Figure 3.2: Crimes reported in Boston in the desktop (left) and mobile (right) views.

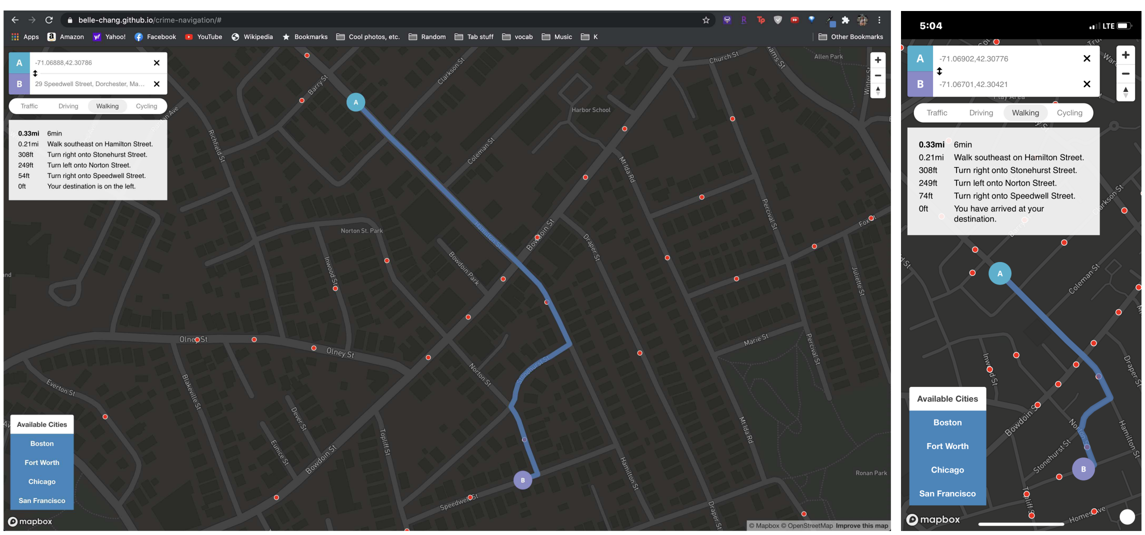

Figure 3.3: Example of a lowest crime affinity route in the desktop (left) and mobile (right) views.

# Chapter 4

# Implementation and Documentation

## 4.1 Implementation

### 4.1.1 System Overview

The web application uses React, Turf.js, Mapbox GL JS, and the Mapbox Directions plugin. On the higher level, the app takes in an origin and a destination as input, and outputs a route between the two points that minimizes the amount of run ins with the locations of reported crimes. A visual of this can be seen at: https://bit.ly/3doeK7x.

### 4.1.2 Tools

#### Figma

Figma is a popular online website used by many designers to create both high and low fidelity mock ups, focusing on user interface and experience. Through Figma, designers can easily mock up the flow of an proposed app, hone in on the look and feel of the app, and ultimately, develop a cohesive and homogeneous design and digital identity of a product. For the purpose of this application, Figma was used to develop an initial prototype of the app design to gain a better understanding of the resources and technologies necessary for its creation. This interface prototype later proved to be handy when considering how to make the web application compatible with mobile devices.

#### Mapbox

Mapbox is a powerful tool that is user friendly and offers a generous free tier with access to all features for web apps with up to 50,000 loads, which was perfect for the purposes of this project. It has many functionalities necessary for the web app, including custom layers, maps, and navigation.

Mapbox is the backbone of this project; all the data is plotted onto the map, utilizing the ability to organize points and customize layers, and the custom route is calculated using the Mapbox directions function of adding waypoints.

#### Turf.js

Turf.js is an "advanced geospatial analysis for browsers and Node.js" that I discovered in the Mapbox tutorials page [14], [15]. It was immensely useful in route calculation for this project, since it offers functions that output the points within a polygon, the points intersecting a line, and a set of points with a given buffer around each of them.

### 4.1.3 Method

#### Mockup

During the initial approach of this project, I decided to create mock ups using Figma's design capabilities. While I did not ultimately create an iOS application, this

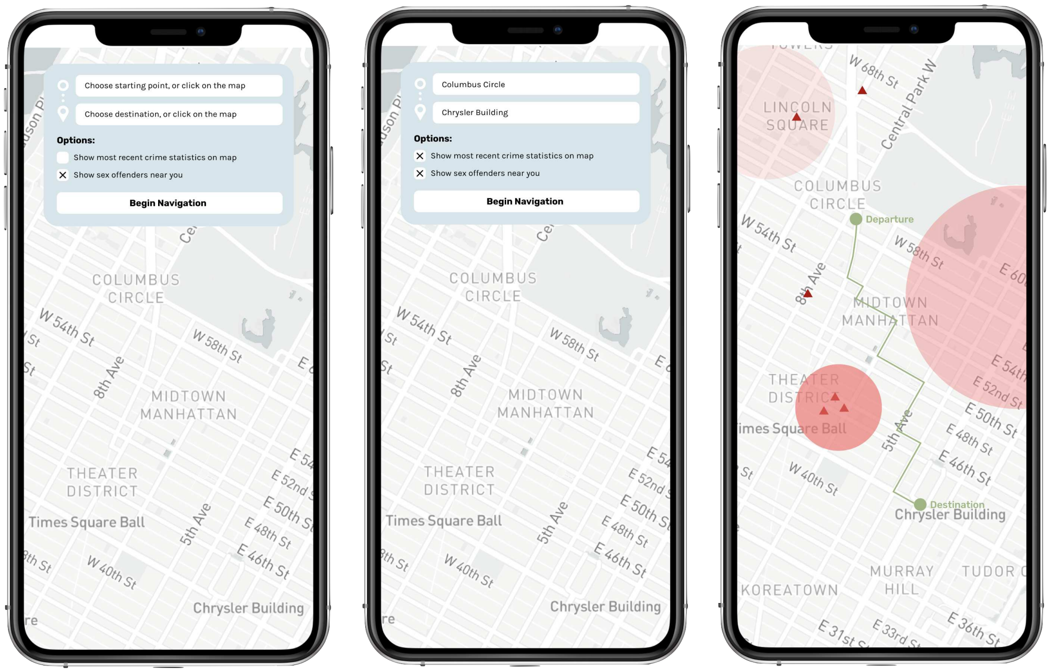


Figure 4.1: Proposed application design using Figma.

was still an integral part of the process because it allowed me to flesh out my ideas and have a clear understanding of what type of data and technologies, databases, and APIs I would need for the app. The mockup also proved to be useful when conceptualizing the interface for the web application to be compatible with mobile devices.

The initial design of the application displayed the locations of the reported crimes and organized areas as "hot spots," in which there would be circles with varying shades of red to indicate locations with higher reported crimes (Figure 4.1). It allowed users to see addresses of nearby sex offenders, a feature inspired by familywatchdog.us, a service that plots the homes of nearby sex offenders, but this was left as a stretch goal. By simply creating this mock-up, I was able to piece together that I would need some sort of map API that would allow for adding layers, location points, and navigation, as well as crime data with addresses or latitude and longitude that would allow me to plot it on a map.

## Data Collection

To create this application, I first found a database that aggregated reported crimes for 16 major U.S. cities [16]. Each of these data sets were taken from the city's public domains, converted into a common format, and normalized to follow the Uniform Crime Reporting National Incident-Based Reporting System (NIBRS) codes. I then selected four cities that I would incorporate into my application: Boston, Fort Worth, Chicago, and San Francisco. These cities were selected due to a number of factors. First, the size of each data file per city was an important consideration for the map's buffering and loading time. Second, I wanted to be able to test my application, and I knew that during this remote semester, I would have a few friends in each.

## Data Preprocessing

After selecting the data, I cleaned up the data to remove all duplicates from the CSV file and to include all violent crime, physical and sexual assault, robbery, homicide, kidnapping/abduction, larceny/theft, weapons law violations, disorderly conduct, peeping toms, drunkenness, and driving under the influence (Appendix A.1). The crimes that were omitted were offenses that did not seem like much of a threat to pedestrians: arson and monetary related crimes, such as bribery, forgery, embezzlement, blackmail, and gambling. This was done to reduce the data so it could be more quickly rendered in the map. After doing so, I converted it into a geoJSON `FeatureCollection`, maintaining the `[longitude, latitude]` and select properties, to be compatible with Mapbox (Figure 4.2).

After visualizing the data in Mapbox studio, it became apparent that many crimes fell under multiple offense codes, causing them to be repeatedly and unnecessarily plotted in the map. The data was then processed again to remove these multiples by checking the coordinates, each with six digits of precision, as well as the date and time of the committed offense.

```
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [
                    -71.058397,
                    42.287877
                ]
            },
            "properties": {
                "offense_code": "13C",
                "offense_type": "intimidation",
                "date": "2019-01-01 00:00"
            }
        }
    ]
}
```

Figure 4.2: Example geoJSON value.

### Mapbox Integration

Each data set (Boston, Fort Worth, Chicago, and San Francisco) was populated in the map. Because I wanted to create a way for users to visualize crime concentration, I decided to cluster the points (Figure 4.3). Essentially, this would group the points into circle markers varying in color and size based on the number of neighbors within a certain radius (Appendix A.2). Clustering also proved to be useful in optimizing render time, as I continued to add city data to the map.

### Route Calculation

Route calculation was by far the most nuanced portion of this project; the final algorithm optimized user safety and duration for each route. There were a few approaches that I ran through before settling for the current solution:

1. **Weighted Roads** Similarly to how typical map applications bias routes toward or against highways and toll roads, I wanted to be able to add a weight to

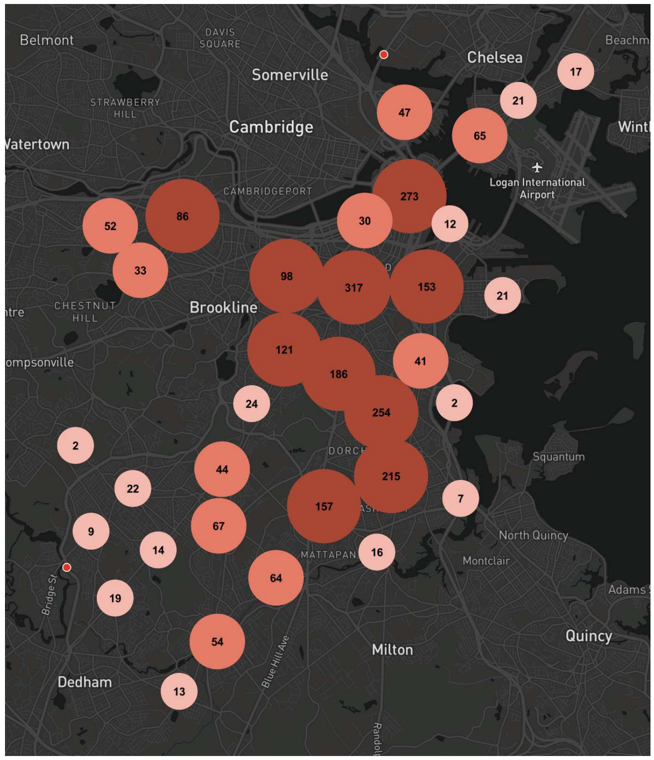


Figure 4.3: Visualization of crime in Boston, clustered hierarchically.

each road indicating how crime-heavy it was. This, however, would distribute the weight across the entire road even if crime is more concentrated on certain parts of it. This was also infeasible, since Mapbox does not allow for altering road properties.

2. **Route Avoidance** Further searching let me to the HERE Routing API, the seemingly only API that allows for route calculation avoiding the inputted bounding boxes [17]. Using this approach would require me to programatically draw bounding boxes around rectangular clusters for data sets in which the points are dispersed relatively randomly, an NP-hard problem. The other option would be to manually draw these bounding boxes, which would be tedious for the 100k points in Chicago. Upon deeper musing, I quickly realized that it would be near impossible to avoid 100% of the plotted crimes.

The final algorithm adapts code from a Mapbox tutorial on routefinding [15] and uses Turf.js and the Mapbox Directions plugin to output the route with the least amount of plotted crimes out of a maximum of 50 possible routes, each of which biases toward walkways and away from alleyways, a setting in the Mapbox directions control (Appendix A.3). Prior to any route calculation, the `turf.buffer()` function was used to create a buffer of 7 meters around each data point to make it "intersectable" with the route line. Once the destination was set in the application, a bounding box was created around the route line. To maintain the route with the least number of intersections between the points and the route line, I used `turf.lineIntersect()` and updated the `min_intersections` route, `num_intersections`, and `time` variables accordingly. If the route was determined to have an intersection, the bounding box was expanded by a factor (0.025) of the attempt count and a random way point was selected from the bounding box to be added to the route for recalculation.

### Optimizing Route Calculation

The above solution proved suitable for cities with less than 10000 data points. However, when tested for cities with many more crimes, such as Fort Worth which had 22000 data points, it sometimes took up to two minutes to obtain a route. To optimize the solution, I quickly realized that I did not need to check if the route line intersected with all data points for the respective city; instead, I only had to check if the line intersected with the data points within the route's bounding box. Luckily, Turf provides a useful function, `turf.pointsWithinPolygon()`, to determine which points in the city collapse into the bounding box. After doing this, I ran the `turf.buffer()` function to create the 7 meter radius around each point and followed the remainder of the algorithm. This easily sped up the algorithm, and it can easily run on the data sets I have, which range from having 2600 to 100000 points.

### Route Pruning

By the nature of adding random waypoints to the map, the outputted route would occasionally have redundant directions; that is, it would navigate the user up and down the same road (Figure 4.4). To account for this, I added the following logic to the route generation algorithm:

1. Check if the current route has the same or fewer number of intersections as the current `min_intersections` route.
   (a) If the current route has the same number of intersections, check if its duration is less than that of the `min_intersections` route.
      i. If the duration of the current route is less, update the `min_intersections`, `num_intersections`, and `time` variables.
2. If the current route has fewer number of intersections as the current `min_intersections` route, update the `min_intersections`, `num_intersections`, and `time` variables.

This, for the most part, took care of the redundant directions.

### UI

This project was to be accessible on both mobile and desktop devices. In order to make the website compatible with the mobile view, I wanted to create a website that was clean and uncluttered. I used JavaScript logic to make the menu bar collapsible and placed the navigation controls in the top left hand corner of the screen. The directions are displayed in a rectangular box in the top left hand corner that can be easily read.

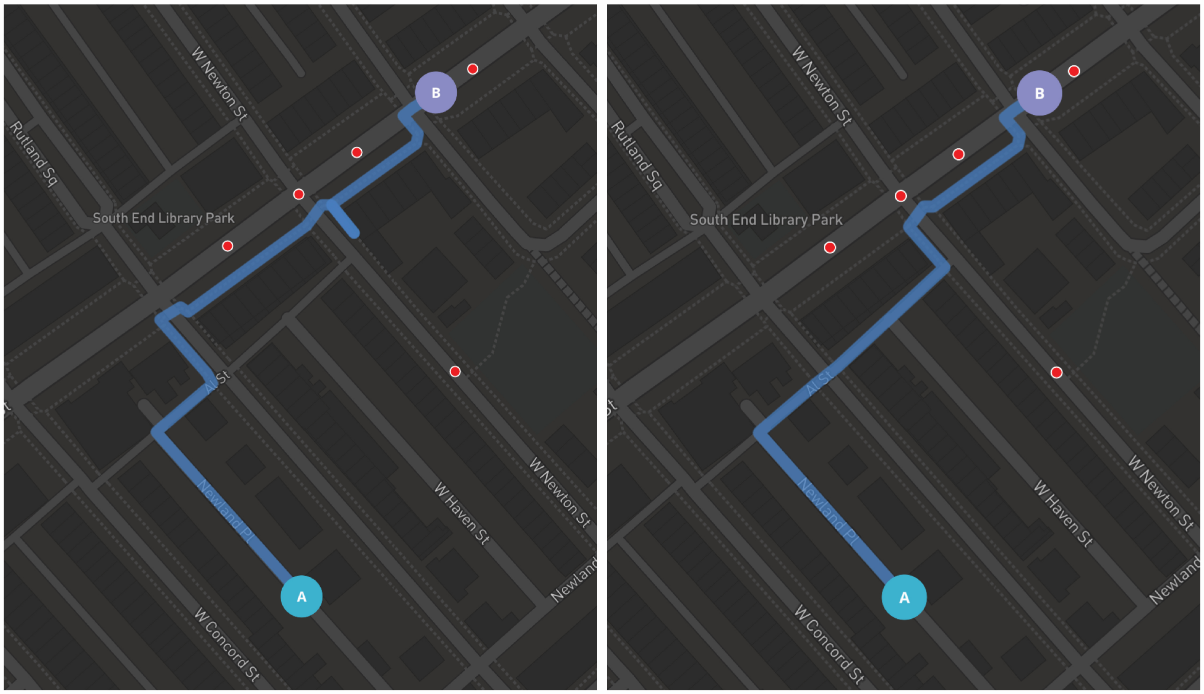


Figure 4.4: Example of a route without (left) and with (right) pruning.

### User Feedback

As with all applications, this was constantly a work-in-progress, and I was open to constructive criticism and feedback from my peers to improve the user experience. In the survey used to evaluate the application, I asked students to provide any comments or thoughts about the application. One of the most popular pieces of feedback was the color scheme used to display the plotted points; many students thought the red and black theme was too intimidating. The final product now uses a red to orange to green color gradient based on the concentration of points in each area, and when zoomed in, the unclustered points are a neutral blue color to not scare users (Figure 4.5). I also lowered the opacity for the circle color, since the fully opaque circles would block useful information like city or region names.

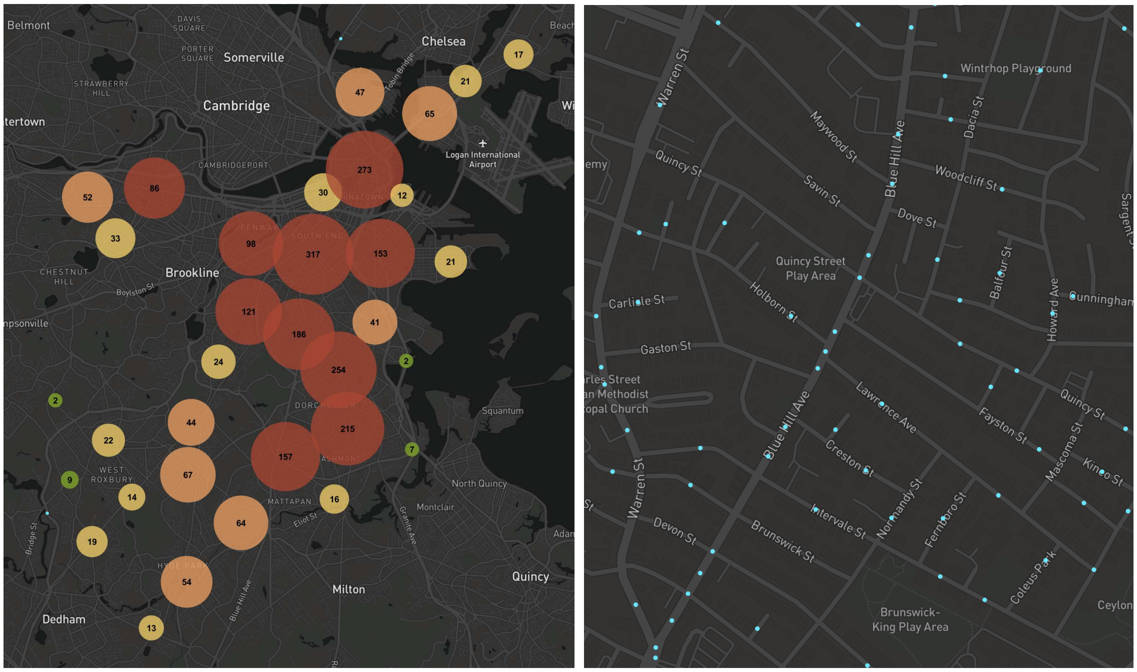


Figure 4.5: Updated color scheme at varying zooms.

## 4.2 Documentation

The second goal of this project was to create a running and thorough documentation of my design decisions and thought process while creating this application to inform others of and teach others from my experience.

### 4.2.1 Idea Selection

The road to creating this application first started with brainstorming. Since this was to be my senior capstone project, I knew I wanted to create something that stemmed from personal experiences, aspirations, and passions – from previous projects, I knew it would be much easier to motivate myself while working on something that I was enthusiastic about. I came up with a few umbrella categories that I would interested in: organ exchange, color theory, golf, human safety, and mindfulness within technology. I then produced a few ideas for each and consulted my advisor, Professor Kernighan, about each of these topics. Together, we eliminated projects that were lackluster and I was tasked with doing more research for the ideas that remained. In terms of selecting the idea I wanted to go with, there were a few questions I considered:

1. Is this idea feasible? This is, by far, the most important question to ask yourself. For example, the organ donation idea would not be possible to fully implement since the necessary data is protected. No one wants to work on a project for a full semester, only to realize the idea is too complicated to individually implement.

2. Is the market over saturated for this idea? It would be redundant to answer a problem for which there are already successfully working solutions.

3. Is there a market for my idea? This is more of an entrepreneurial question, but I think this is still a relevant consideration, especially if you would like tangible user feedback to evaluate the project on.

4. Do you love this idea? Since this is a year long project, it's important to be able to stay excited and passionate about what you're doing...

In speaking to the first two questions, I think it is especially important to select an idea that is both novel and attainable.

### 4.2.2 Design Decisions

#### Google Maps vs. Mapbox

In evaluating whether or not I wanted to use Google Maps or Mapbox, I looked at each service's respective functionalities (APIs and such), availability, and pricing. While Google Maps is more powerful and, perhaps, more accurate (based on previous experiences), the downfall is its cost, which is priced per 1000 requests or calls. I was not sure if I foresaw myself scaling up application during the duration of the project, so I opted to use Mapbox, which offered the same functionalities and better map editability and stylization for free.

### Mobile Application vs. Web Application

My initial idea was to create the same navigation application that minimizes the amount of plotted crimes along the route in an iOS app form. Swift proved tricky to use – creating an iOS application was not without a bit of frustration and annoyance. I got so far as plotting the data points in the map using the Crime-O-Meter API to grab dynamic data, before realizing that such an app would not be possible because of Mapbox's iOS SDK limitations; it would not allow for custom route planning, a feature available for the Android SDK. I was naive to assume that the functionality would be native to all of Mapbox's APIs and SDKs; let this be a lesson to **thoroughly** research the technologies you plan to use prior to starting your project.

Upon this discovery, I had to pivot to creating a web application. Luckily, I have had experience with web development and this process was, for the part, extremely enjoyable. The only obstacle I had was the Crime-O-Meter API failing, since the API key I had limited my access. I took to scouring the internet to find other sources for crime data, ultimately settling for a large data set that aggregated crime data from 16 major cities in the United States.

For the process of developing and coding the web application, I took advantage of the Mapbox blog, tutorials, and documentation to understand how to use each feature. Oftentimes, these resources would have useful skeleton code that I could adapt to my own liking in my application. The following sources were integral to the completion of the website: [18], [19], [20], [15]. In short, maximize the use of the tutorials and documentation. It might sound like cliché advice, but the more you use and familiarize yourself with the technology, the more you learn about and how to use it.

### 4.2.3 Experience with Swift

Because the initial approach with this project was to create an iOS application, I was to use Swift to code my app. While languages and technologies continuously evolve to improve and add features, a transitional experience that is typically hassle-free, my experience with Swift was not case. Swift has recently been upgraded to Swift 5 and along with that, the seed file for every project type has changed in structure, adding a component known as SwiftUI, "an innovative, exceptionally simple way to build user interfaces across all Apple platforms with the power of Swift" [21]. This was simply not my experience because the lack of updated documentation on integrating technologies into Swift with this new file structure and new feature.

While I was trying to add Mapbox to the iOS application, an effort that was unnecessarily difficult, I ran into additional hurdles. Swift 5 is "source compatible with Swift 4.2, but isn't binary compatible with earlier Swift releases," meaning that any application built with previous versions of Swift must be rebuilt with Swift 5 [22], [23]. This meant that the documentation on the Mapbox website for the Maps SDK and the Navigation SDK was no longer up to date with current version of Swift, and I had to outsource to Google to learn how to add the SDKs into the app, a process that was extremely tedious and took many trials and errors.

All in all, I would say my experience with Swift was not very pleasant. Speaking with a few friends that I reached out for help during my short time working with Swift also made me realize that the frustrations I had were not isolated incidents; almost all my friends who had previously worked on a project in Swift considered it a nuisance to work with. While I applaud Apple for its ambitions constantly improve and innovate its own technology, I do believe that this idea of maintaining compatibility with only its in house and native features, in this case the Apple MapKit, and the planned obsolescence – even for its own technology – is frankly, very annoying. On that note, I would highly suggest reviewing the Swift documentation, learning Objective C, and ensuring that all your dependencies are compatible with the latest version of Swift before attempting to start a project with it.

# Chapter 5

# Results and Evaluation

## 5.1 Results

To measure the success of the application, a survey was sent out to the Forbes, Mathey, and Butler residential colleges.

### 5.1.1 Survey

The survey asked students a total of five questions:

1. What are your pronouns? [She/her, He/him, They/them, Other (self input)]
2. How safe do you feel walking around alone in big cities? [Scale of 1-10, with 1 being the most unsafe, 5 being neutral, 10 being the most safe]
3. Would you use a navigation app that provides you the route with the least amount of run ins with reported crimes? [Yes, No, Maybe]
4. What are your thoughts about the application?
5. Would you feel safer with an application like this? [Scale of 1-10, with 1 being the most unsafe, 5 being neutral, 10 being the most safe]

A total of 109 students responded to the survey, 70 of which identify as she/her/hers, 35 as he/him/his, and 4 as other. 60 students would, 43 would maybe, and 6 students would not use the application. Since a sliding scale was used to measure the level of safety students felt while walking around alone in cities, a score less than 5 would indicate feeling unsafe, a score equal to 5 would indicate neutrality, and a score greater than 5 would indicate feeling safe. As expected, a majority of women feels unsafe and a majority of men and they/them/other feels safe walking around alone in cities (Figure 5.1). With this application, the majority of the women's consensus shifts to feeling safe, similarly with the men (Figure 5.1). For the they/them/other category, one student's safety sentiment lowers down to neutral (Figure 5.1). This trend can also be seen clearly in the distribution of safety scores, which shift toward values greater than 5 for all gender categories with application usage (Figure 5.2). In total, 77 (70.6%) students' sentiments increased, 20 (18.3%) decreased, and 12 (11.1%) remained neutral with the use of the application.

### 5.1.2 User Feedback

As mentioned previously, the majority of the surveyed students would use the application. There was a large number of comments indicating that the application would be a useful and practical tool for those who intend on walking to certain locations. A couple of students mentioned the worthwhile issue of crime data being inherently skewed to disproportionally report overpoliced communities. A few students commented on how awareness of such crime around them made them feel unsafer than they had initially felt, since they were surprised by the amount of offenses that had been reported around them. Many students provided feedback on how to make the UI easier to navigate and readable, as well as tips on how to make the website more versatile for mobile use.

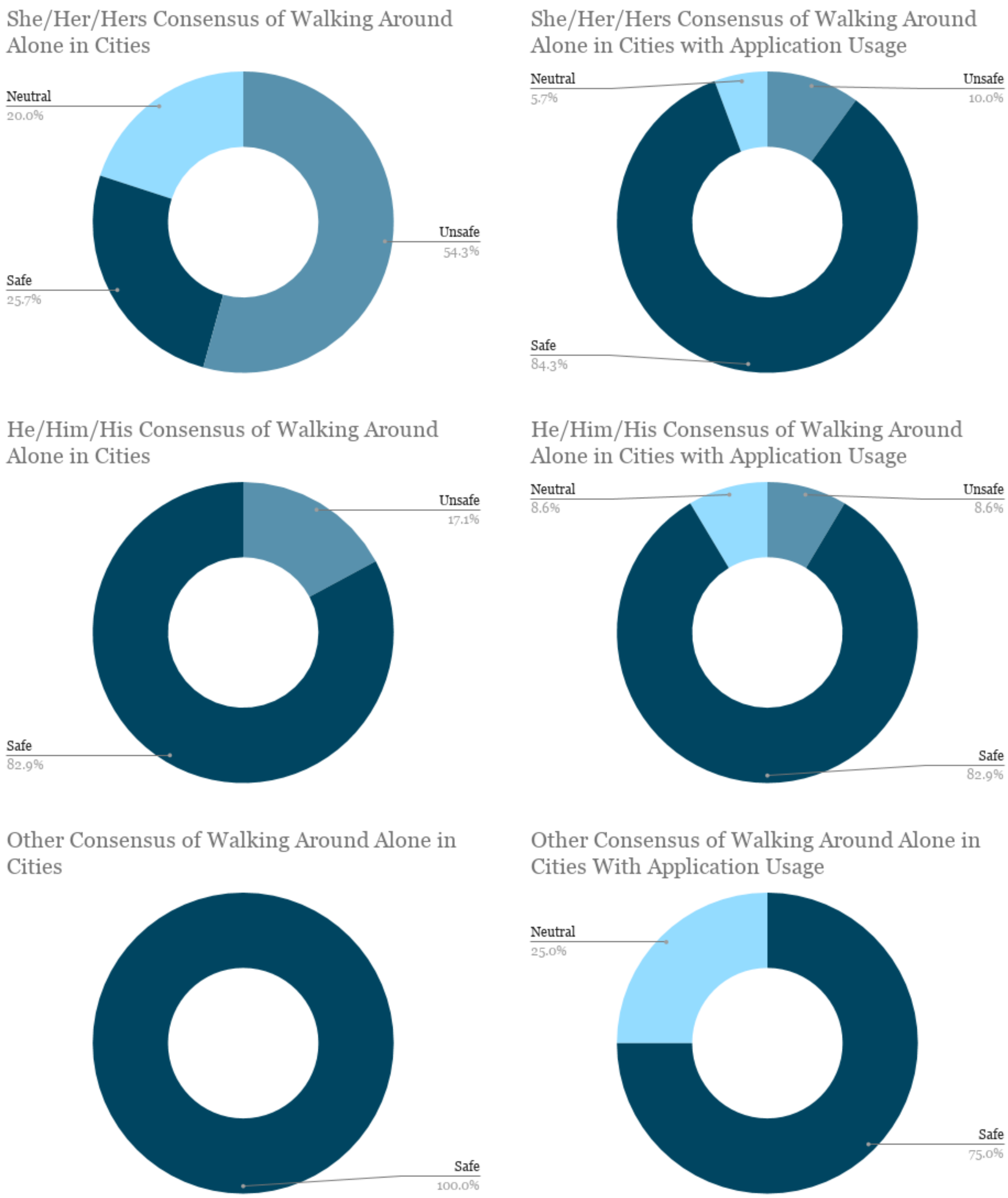


Figure 5.1: General consensus of how students feel while walking around alone in cities without and with the application, divided by genders.

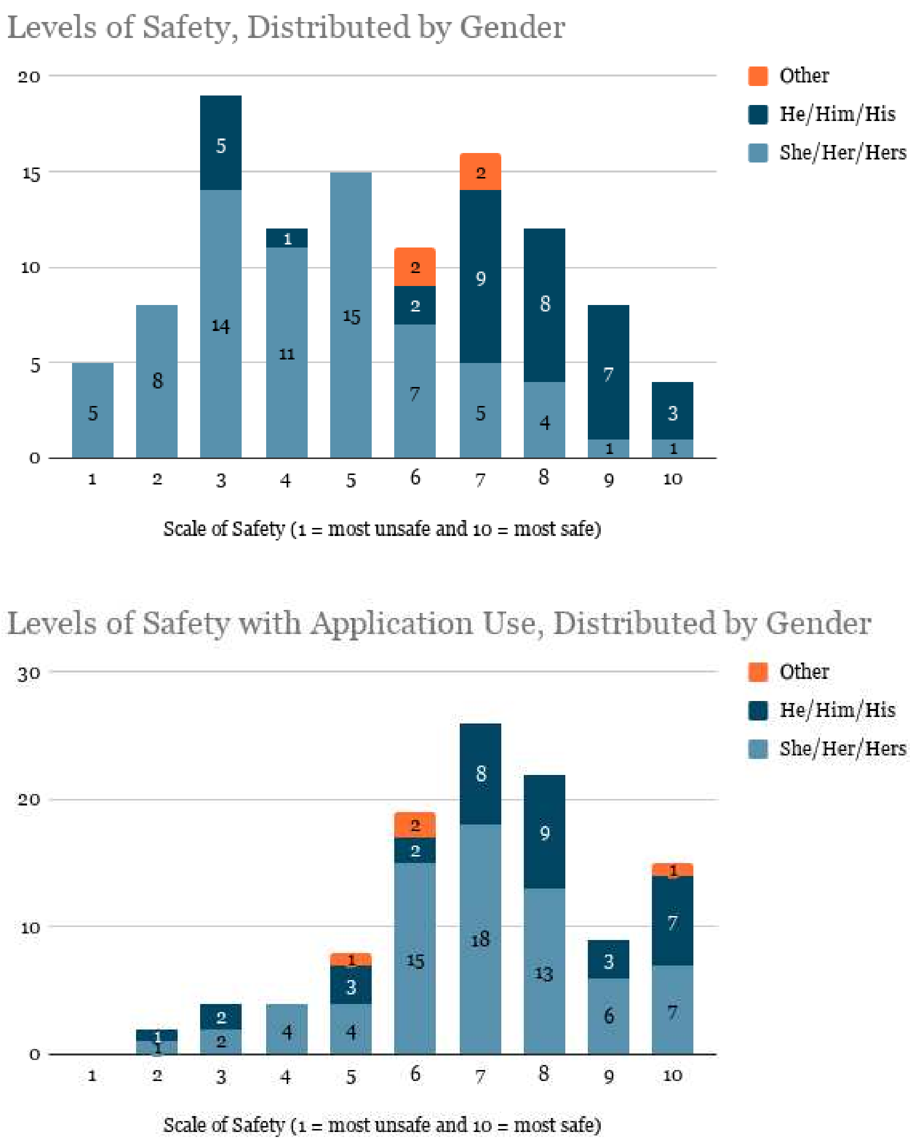


Figure 5.2: Levels of safety, without and with the application.

## 5.2 Evaluation

The initial goal of this project was to help people feel safer while walking alone in metropolitan areas and given the survey results, I would say the website had resounding success. Many students would use the application and felt safer with the availability of such an application (Figure 5.2).

### 5.2.1 Accessibility and Usability

As a free and non-commercial web application, the navigation website offers an accessible way for users to find a route that has the fewest reported crime offenses while navigating around metropolitan cities. It can easily be used in a desktop view, as well as on mobile devices by adding the website to the home screen, and many users applauded its versatility and practicality (Figure 5.3). The navigation app, therefore, has potential as a viable solution for increasing human safety in large cities.

### 5.2.2 Documentation

The documentation portion of this project was highly illuminating for myself; it allowed me to catalog and categorize my ideas, as well as better organize the resources I needed for this project. I was able to approach the project in a more systematic and structured manner, instead of blindly jumping in and beginning to code. While I did not survey my fellow computer science peers on how such documentation would benefit them, I am hopeful that others are able to gain value and insight from my experience creating this project.

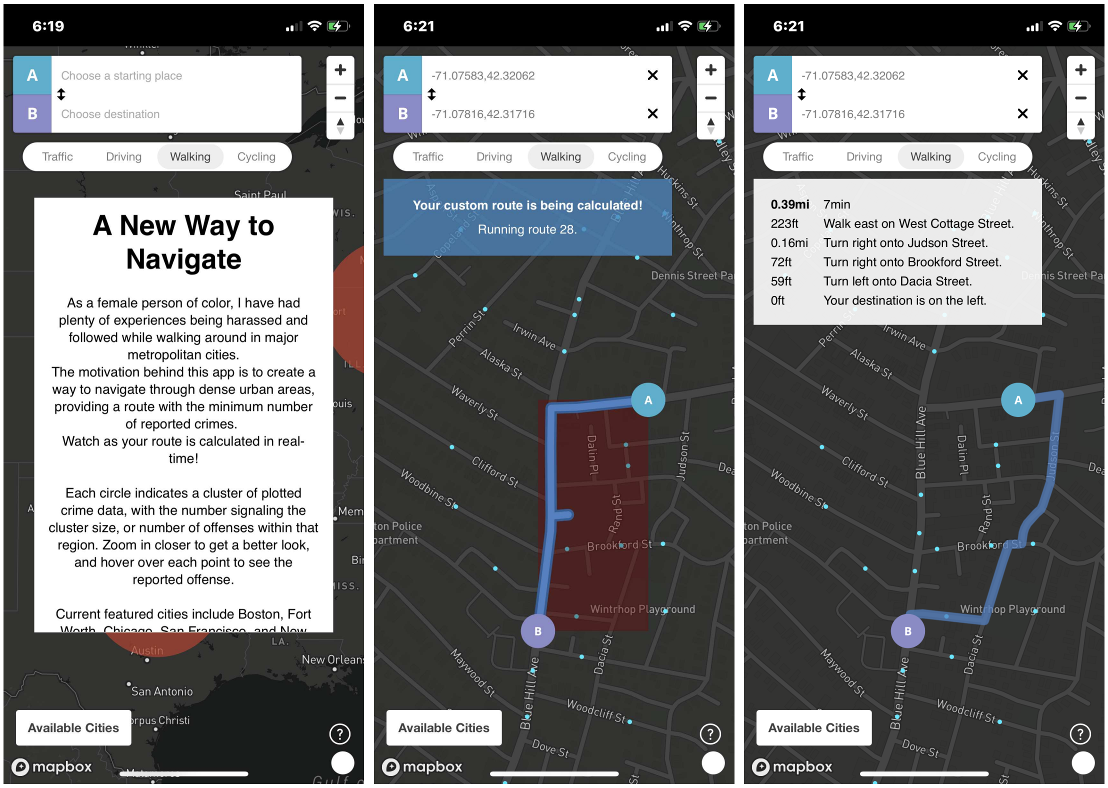


Figure 5.3: Application in the mobile view on an iPhone XS.

### 5.2.3 Limitations

#### Data

Crime data was used for illustrative purposes for this project, as the only tangible metric currently available to measure safety. However, due to the inherent implicit bias in every individual (and by the nature of that, police officers), all crime data is inherently suspect since it reflects imperfect societal attitudes and may easily implicitly discriminate against different populations; it is impossible to label any crime data set as objective. While this project only plots the relevant offenses and does not intend to draw any analyses nor conclusions from the data, this raises the issue of over-policing certain communities, which may cause certain regions of particular cities to have more crime data points than others, an extremely important consideration to take into account when using this application.

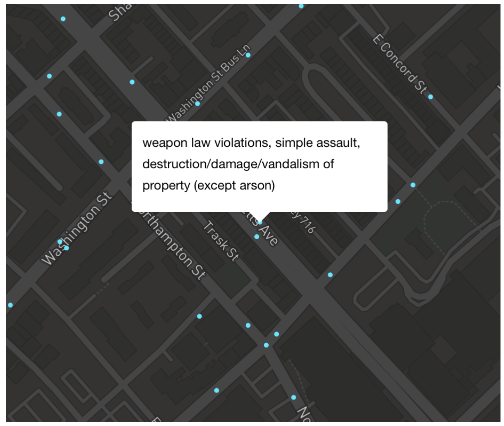


Figure 5.4: Example of tool-tip with offense information for a plotted point.

The data available was extremely limited. Because the Crime-O-Meter API was unsuccessful in consistently scraping crime data with the given API key, I had to use a static database with crime data from 2019. Given the opportunity, I would try to find more recent data or another API that could continuously scrape data to provide the most up to date information.

Many students mentioned that the data was a useful metric to have to make better informed decisions while walking alone at night or when venturing into a new city. To address the issue of biased data, students also suggested to provide details for each crime plotted; the final version of the application has tool-tips on hover (and on click for mobile) for each point that describes the offenses committed at that location (Figure 5.4). Students also commented on using or adding cities that were more likely to have civilians getting around by foot. For example, because Fort Worth is so spread apart, most individuals who live there most likely drive to their destinations, and such an application would not be as useful as for a city like Manhattan.

### Directions Interface

There were a few limitations to the user interface. Although the application could be bookmarked and used in mobile devices, it is not competitive with the native and popular navigation applications, like Apple Maps, Google Maps, and Waze.

Those applications have far more features, including adding stops along a route, searching for nearby locations, and sharing your ETA with contacts. Unfortunately, Mapbox's Directions API is not as powerful and does not have these features for a web application. Users also noticed that inputted landmark names had to be extremely specific - for example, searching for "Laguna" within San Francisco will show "Laguna, Philippines", "Laguna, Santa Catarina, Brazil", and "Laguna Beach, California" as the top three results, even though there is a "Laguna Honda Hospital and Rehabilitation Center" within the city itself (Figure 5.5). Essentially, the Mapbox Directions API does not output the nearest places in the map view for the searched name.

Because the application could not use Mapbox's route object to output the final route, the list of directions provided does not have all the functionalities that a typical navigation application would have. For example, using Mapbox's native directions output would allow the user to click on each direction and the map would fly to the respective street or intersection. This was, for the most part, sufficient information for the surveyed users; however some students mentioned that while it did not deter them from completely using the app, it would be a positive addition.

The Mapbox directions plugin also does not provide for point validation; for example, a user can easily set their origin or destination in a body of water. This presents an issue for determining routes with bounding boxes that include bodies of water, since waypoints are randomly generated and there is no native Mapbox directions function to validate a selected point.

### Data Visualization

Initially, the theme of the application was black and red, with the cluster circles appearing in either dark red, red, or light red, depending on the number of crimes in the area. A few users mentioned that the color scheme was intimidating, so the

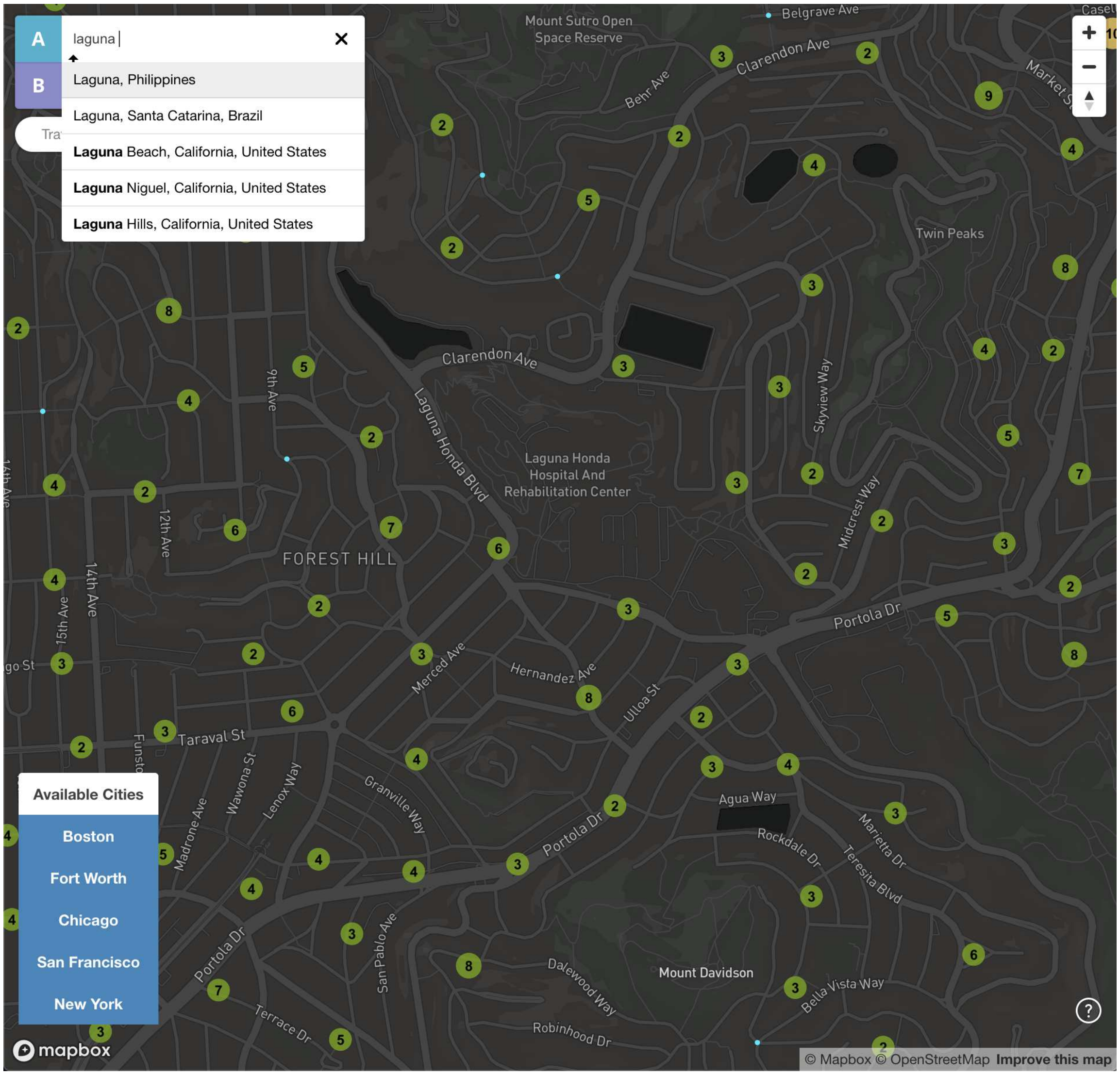


Figure 5.5: Search example using the Mapbox directions plugin.

final product uses a gradient of colors from red to yellow to green, depending on the number of crimes. A good amount of users mentioned that they appreciated how the crime was organized in the clusters of varying size and color; however, there were also responses that indicated confusion about this representation of data. To mitigate this, perhaps I could add a more thorough explanation on the informational popup or represent the data in a different visualization.

### Algorithm

The algorithm was definitely sufficient at providing the desired directions in an efficient way; many students were impressed by its efficiency and speed at calculating the route. However, the algorithm uses a brute force method of finding these directions and there certainly are limitations to the chosen approach. The bounding box method works best for cities that do not have grid-like urban organization, such as Boston. This is because the streets are less organized, meaning that an indeterminate number of streets can diagonally run through a bounding box for a typical route, allowing for a wider variety of random waypoints to be chosen. For grid-like cities, such as Chicago and New York, the bounding box for a route will only contain so many roads that a random waypoint can be chosen from, so there are fewer unique routes calculated.

# Chapter 6

# Discussion

## 6.1 Future Work

While this project as a proof of concept was successful, there are a number of things that can be tweaked or added to improve the application.

### 6.1.1 Data

As mentioned in the previous section, additional "walkable" cities could be added to the website and more recent crime data could be used. I would explore using an actual API, rather than a database to do this and it would be beneficial to allow users to filter the crime data by certain offenses, such as assault, robbery, or weapon law violations.

### 6.1.2 UI/UX

It is typical for an application to only work as well as its underlying dependencies, in this case, the Mapbox API. From the user feedback received, it was clear that this application could not compete against the more popular navigation applications on the market without a few additions to improve the user experience with the Mapbox

Directions API. An incomprehensive list of additional features to add:

1. **Point Validation** This could be incorporated by checking if the point is being projected in an invalid landform, such as the ocean or a lake [24].

2. **Local Search** The Mapbox Geocoding API allows users to search for locations and landmarks within a specified city [25]. This is a feature available on all popular navigation applications, so adding this could really elevate the app.

3. **Route Output** Currently, the calculated route is displayed on the map, with a list of directions appearing under the directions input. To improve user experience, I could add a street-by-street view of the directions that users can click on and interact with, which is how most navigation applications currently take the user through a route.

4. **Data Visualization** A few surveyed users mentioned that the clustering on the map crowded it, making it difficult to tell which areas had less reported crimes than others. Perhaps a better way to visualize the map would be to use a heatmap or voronoi. Using these data visualization methods would lend to easier isolation of areas with less plotted crime.

5. **User Reporting** Since verbal harassment like cat calling is not a formal crime, it would be beneficial to add a way for users to report such an offense. This will be particularly helpful for women in larger cities such as New York, where cat calling is a frequent occurrence.

### 6.1.3 Further Evaluation

While preliminary feedback indicated positive reception of such an application, further evaluation can be done to test its efficiency, efficacy in routing, and usefulness. To draw these conclusions and comparisons with the popular navigation apps, this app would need to be used alongside the popular navigation apps by individuals who live in the cities available for an extended period of time.

## 6.2 Conclusion

For this project, I created an application was designed to increase the safety sentiments for individuals who traverse through metropolitan cities alone. This was done by aggregating crime data for 5 large metropolitan hubs, plotting each point, then creating an algorithm that determines the route along which there are the fewest numbers of reported offenses. Based on surveyed feedback, students felt that it was a novel idea that could specifically help those who feel uneasy in newer cities or those who feel uncomfortable walking around at night in large cities.

While the application in its MVP fulfilled the initial goal of increasing human safety, it's not to say that the project has shortcomings. First and foremost, it's especially important to acknowledge and recognize the internal and implicit biases and human error in crime data, which is currently the only available metric to assess safety. The crime data is used strictly for illustrative purposes; as the nation is currently undergoing a transformation to eliminate police brutality and over policing and to curb discrimination, perhaps more objective data will be available for future use and integration. Secondly, while positive user feedback was obtained from a brief survey, the application will require more thorough and further testing from its target audience to better evaluate its performance.

Holistically, this thesis has been highly enlightening in demonstrating the need for an application such as this one. The number of students that mentioned that they feel unsafe in large cities is a perturbing statistic that echoes that of the larger population. Human safety is a deeply concerning issue that continues to pervade our society today and requires a modern solution that actively helps while people are engaged in normal activities, such as walking around town.

So what does the future of our society look like? While our nation is undergoing a tumultuous awakening in attempt to reconcile meaningfully with its systemically prejudiced policing policies and tactics and its highly discriminatory past in light of the hate crimes that are more frequently occurring today, there is definitely room for growth on the front of protecting human welfare. I can only hope that human safety becomes a priority and that others are urgently working on creating solutions for this deeply rooted issue that continues to suffuse through all levels of society.

# Appendix A

# Code

## A.1 Data Preprocessing

```
geojson = {
    'type': 'FeatureCollection'
    # 'features': [

    # ]
}
features = []
feature = {
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "offense_code": "",
    "offense_type": "",
    "date_single": ""
  }
}

points = {}
# reading csv file
import json
with open(filename, 'r') as csvfile:
    # creating a csv reader object
    csvreader = csv.reader(csvfile)

    # extracting field names through first row
    fields = next(csvreader)

    # extracting each data row one by one
```

```
    index = 0
    for row in csvreader:
        rows.append(row)
        city = row[fields_dict['city_name']]
        cities[city] = cities[city] + 1
        print(cities[city])
        if city != "Boston" and len(points) > 0:
            break
        if city == "Boston":
            rows.append(row)
            feature = {
                "type": "Feature",
                "geometry": {
                    "type": "Point",
                    "coordinates": [125.6, 10.1]
                },
                "properties": {
                    "offense_code": "",
                    "offense_type": "",
                }
            }
            lat = float(row[fields_dict['latitude']])
            lon = float(row[fields_dict['longitude']])
            point = (lon, lat)
            # validate latitude and longitude
            if lat > 90 or lat < -90 or lon > 180 or lon < -180:
                continue
            # keep only specific offenses
            if row[fields_dict['offense_code']] not in offense_codes:
                continue
            # if the same offense has multiple labels
            if point in points:
                feature_index = points[(point)]
                get_feature = features[feature_index]
                get_feature['properties']['offense_count'] += 1
                if row[fields_dict['offense_type']] not in get_feature['properties']['offense_type']:
                    get_feature['properties']['offense_type'] += ", " + row[fields_dict['offense_type']]
                    get_feature['properties']['offense_code'] += ", " + row[fields_dict['offense_code']]
                continue
            # add to data
            points[point] = index
            feature['geometry']['coordinates'] = point
            feature['properties']['offense_code'] = row[fields_dict['offense_code']]
            feature['properties']['offense_type'] = row[fields_dict['offense_type']]
            feature['properties']['offense_count'] = 1
            features.append(feature)
            index += 1

```

```
    geojson['features'] = features
```

## A.2 Map Layers

```
// clusters
map.addLayer({
    id: "clusters",
    source: "crime", // this should be the id of the source
    type: "circle",
    filter: ['has', 'point_count'],
    paint: {
        "circle-opacity": 0.75,
        // vary circle color based on number of points
        'circle-color': [
            'step',
            ['get', 'point_count'],
            '#79ad00', 10,
            '#ffd359', 40,
            '#ff9e59', 75,
            '#b83e2c'
        ],
        // vary circle radius by interpolating number of points
        'circle-radius': [
            'interpolate',
            ['exponential', 1],
            ['get', 'point_count'],
            7, 10,
            15, 20,
            50, 35,
            100, 45,
            200, 50,
            400, 60,
            1000, 70,
            150000, 75
        ]
    },
    layout: {
        // make layer visible by default
        'visibility': 'visible'
    }
});
// label clusters
map.addLayer({
    id: 'cluster-count',
    type: 'symbol',
    source: "crime",
    filter: ['has', 'point_count'],
    layout: {
        'text-field': '{point_count_abbreviated}',
        'text-font': ['Arial Unicode MS Bold'],
```

```
        'text-size': 12,
        'visibility': 'visible'
    }
});
// unclustered points
map.addLayer({
    id: 'unclustered-point',
    type: 'circle',
    source: "crime",
    filter: ['!', ['has', 'point_count']],
    paint: {
        'circle-color': '#14e8ff',
        'circle-radius': 2.5,
        'circle-stroke-width': 5,
        'circle-stroke-opacity': 0
        },
    layout: {
        // make layer visible by default
        'visibility': 'visible'
        }
});
```

## A.3 Route Generation

```
let routes = e.route;

if (counter >= max_attempts) {
    // set route on map, add directions
    e.route = route;
    map.getSource('route_line').setData(min_intersections);
    addDirections(e.route);
}
else {
    routes.forEach((e) => {

        // make route and bbox visible
        map.setLayoutProperty('route_line',
                              'visibility', 'visible');
        map.setLayoutProperty('bounding_box',
                              'visibility', 'visible');

        // get GeoJson LineString feature of route
        let routeLine = polyline.toGeoJSON(e.geometry);

        // create a bbox around this route
        bbox = turf.bbox(routeLine);
        polygon = turf.bboxPolygon(bbox);

        // get pts in the city that are in bbox, check num of isects
        let points_within = turf.pointsWithinPolygon(current_city,
```

```
                                                    polygon);
        // add buffer around points in bbox
        obstacle = turf.buffer(points_within, 7,
                              { units: "meters" });
        let intersects = turf.lineIntersect(obstacle, routeLine);

        // get route with minimum isects
        if (intersects.features.length <= num_intersections) {
            // route pruning
            if (intersects.features.length == num_intersections &&
                e.duration < time) {
                num_intersections = intersects.features.length;
                min_intersections = routeLine;
                time = e.duration
                route = e;
            }
            else if (intersects.features.length
                     < num_intersections) {
                num_intersections = intersects.features.length;
                min_intersections = routeLine;
                time = e.duration
                route = e;
            }
        }

        // update route line and bbox
        map.getSource('route_line').setData(routeLine);
        map.getSource('bounding_box').setData(polygon);

        // check if route isect w/ pts in bbox
        let no_isect = turf.booleanDisjoint(obstacle, routeLine);

        if (no_isect == true) {
            // hide bbox
            map.setLayoutProperty('bounding_box',
                                  'visibility', 'none');
            route = e;

            // reset the counter, min intersections, distance
            counter = 0;
            num_intersections = Infinity;
            time = Infinity;
            addDirections(route);
        }
        else {
            counter++;
            // increase bbox by factor of counter
            polygon = turf.transformScale(polygon, counter * 0.025);
            bbox = turf.bbox(polygon);

            // add a random waypoint to get a route from the
            // directions api
            let random_waypoint = turf.randomPoint(1,
                                                  { bbox: bbox });
```

```
            directions.setWaypoint(
                0,
                random_waypoint['features'][0].geometry.coordinates
            );
        }
    });
}
```

# Bibliography


[1] J. B. Journalist, "Most women say they regularly take steps to avoid being sexually assaulted," Mar 2019. [Online]. Available: https://today.yougov.com/topics/lifestyle/articles-reports/2019/03/28/women-safety-sexual-assault-awareness

[2] R. Chatterjee, "A new survey finds 81 percent of women have experienced sexual harassment," Feb 2018. [Online]. Available: https://www.npr.org/sections/thetwo-way/2018/02/21/587671849/a-new-survey-finds-eighty-percent-of-women-have-experienced-sexual-harassment

[3] W. Cai and Audra, "Punched, kicked, shoved: Documenting the anti-asian violence," Apr 2021. [Online]. Available: https://www.nytimes.com/interactive/2021/04/03/us/anti-asian-attacks.html

[4] "Street harassment statistics," Apr 2015. [Online]. Available: https://www.ilr.cornell.edu/news/street-harassment-statistics

[5] J. Lartey and W. Li, "New fbi data shows violent crime still falling, except rapes," Sep 2019. [Online]. Available: https://www.themarshallproject.org/2019/09/30/new-fbi-data-violent-crime-still-falling

[6] N. B.-B. Richard Fausset and M. Fazio, "8 dead in atlanta spa shootings, with fears of anti-asian bias," Mar 2021. [Online]. Available: https://www.nytimes.com/live/2021/03/17/us/shooting-atlanta-acworth

[7] A. Austrew, “5 best apps and sites to help you find safe neighborhoods,” Jul 2020. [Online]. Available: https://securitynerd.com/best-crime-maps/

[8] “Lexisnexis community crime map.” [Online]. Available: https://communitycrimemap.com/

[9] “Crime rates, statistics and crime data for every address in america.” [Online]. Available: https://www.neighborhoodscout.com/about-the-data/crime-rates

[10] “15 best women safety apps 2021: Redbytes software,” Dec 2020. [Online]. Available: https://www.redbytes.in/best-women-safety-apps/

[11] A. Siddiqui, “Google maps prepares to add a ”lighting” layer to highlight brightly lit streets for safer night travel,” Dec 2019. [Online]. Available: https://www.xda-developers.com/google-maps-prepares-lighting-layer-highlight-brightly-lit-streets-night-travel/

[12] Crowdform, “Building an algorithm to find the safest route home,” Nov 2019. [Online]. Available: https://www.crowdform.co.uk/blog/how-to-lead-people-home-safer-with-routing-algorithms

[13] mfbx9da4, “mfbx9da4/brightpath-backend.” [Online]. Available: https://github.com/mfbx9da4/brightpath-backend

[14] “Turf.” [Online]. Available: https://turfjs.org/

[15] “Route finding with the directions api and turf.js.” [Online]. Available: https://docs.mapbox.com/help/tutorials/route-finder-with-turf-mapbox-directions/

[16] M. Ashby, “Crime open databse (code),” Aug 2020. [Online]. Available: https://osf.io/zyaqn/

[17] "Requesting a route avoiding an area." [Online]. Available: https://developer.here.com/documentation/routing/dev_guide/topics/example-route-avoiding-an-area.html

[18] "Display driving directions: Mapbox gl js." [Online]. Available: https://docs.mapbox.com/mapbox-gl-js/example/mapbox-gl-directions/

[19] "Layers: Style specification: Mapbox gl js." [Online]. Available: https://docs.mapbox.com/mapbox-gl-js/style-spec/layers/

[20] "Display a popup on hover: Mapbox gl js." [Online]. Available: https://docs.mapbox.com/mapbox-gl-js/example/popup-on-hover/

[21] A. Inc., "Xcode - swiftui- apple developer." [Online]. Available: https://developer.apple.com/xcode/swiftui/

[22] C. Pupăză, "What's new in swift 5?" [Online]. Available: https://www.raywenderlich.com/55728-what-s-new-in-swift-5

[23] user18424user18424, "What is the difference between binary and source compatibility conceptually?" Dec 1961. [Online]. Available: https://stackoverflow.com/questions/14908722/what-is-the-difference-between-binary-and-source-compatibility-conceptually

[24] N. W. W. 1 and cammacecammace 3, "Mapbox ability to detect if coordinates is on a body of water?" Jan 1966. [Online]. Available: https://stackoverflow.com/questions/42708403/mapbox-ability-to-detect-if-coordinates-is-on-a-body-of-water

[25] "Local search with the geocoding api: Help." [Online]. Available: https://docs.mapbox.com/help/tutorials/local-search-geocoding-api/