COMPSCI 2XB3:Computer Science Practice and Experience: Binding Theory to Practice
Project Proposal Template

Project Title:	***FireAware***
Lab Section Number:	**L02**
Student Name:	**Jennifer Cheng**
Student Number:	[redacted]
Student Email:	[redacted]

By virtue of submitting this document I electronically sign and date that the work being submitted is my own individual work.

Abstract
As technology evolves, companies incorporate increasingly more electronic elements into their venues. As a result, buildings are at a higher risk of a fire hazard. Although fire inspections should be a regular occurrence, some buildings go by unnoticed, and their fire safety is thus compromised. This project will inform users if a building is deemed to be safe or unsafe based on its previous fire inspection date, as well as the typical frequency of fire inspections. Using open data from New York City, this mobile application allows users to search up a building and retrieve the dates that it has been inspected, as well as a metric of how safe the building is. If a user walks into an unsafe building, the application will alert the user. To ensure accuracy, a reputable map API, Google Maps, will be used. Location-emulating will be used to test that alerts are sent appropriately.

1. Objective
Fire safety is a growing concern as entertainment venues continue to evolve to incorporate more electricity-based components, increasing the risk of fires. This application serves to provide peace of mind to people who are considering going out to a building or venue that has multiple fire hazards by informing the user of the last fire inspection date, as well as the next expected one.

2. Motivation
On January 5, 2019, five teenage girls were killed in a fire that occurred in an escape room (1). A 25-year old man was also seriously injured by the fire (1).

An escape room is a popular attraction that challenges teams to find their way out of a locked by solving riddles within a given time limit. The teenagers were killed due to failure in meeting safety standards, such as a providing an escape route, in the venue (1). These were preventable deaths that could have been easily been averted by having a means of leaving the escape room, as they were locked inside. Similarly, although the full story is not known regarding the man's injuries, they could have been prevented or lessened by ensuring that the building had adequate fire-prevention supplies (extinguishers, fire blankets, etc.).

Fire safety is a serious concern. Almost every single building is at risk of catching on fire, whether it's from a heating element, or a faulty outlet. Every person should be able to easily identify how fire-safe a building may be. In an ideal world, all buildings should have a sign that

states the date of their last fire inspection check. However, it is evident that not all buildings get regular inspections; some go by unnoticed, leading to a concern in fire safety standards.

To alleviate these concerns, an application that tracks the dates of each recorded fire inspection can be created. This application is meant for everyone, as fire safety is a universal concern. Users should be able to see the date of the last fire inspection of a building in their area, as well as previous check-up dates. In addition to recording all of the previous dates, the application will also predict the date of the next inspection to see (relatively) how safe the building is. With this application, the users can feel more assured that their entertainment venue is safe, so they can enjoy themselves to the fullest.

3. Prior Work

Links to previous work:
https://play.google.com/store/apps/details?id=org.fruct.oss.firepoint&hl=en_US
https://play.google.com/store/apps/details?id=com.life360.android.safetymapd

Currently, there are no projects or applications that do exactly what this proposed product does. However, there are similar ones. One is called Firepoint (2), which maps all of the fire outbreaks around the world. The application also gives fire safety tips and other educational tidbits. The proposed project, however, tracks fire safety inspections. Both of the products have fire safety in mind, but with different approaches. Since Firepoint already tracks fires around the world, it could easily build in the functionality of being able to check when a building was last inspected for fire safety (or vice-versa.)

Another product that exists is called Life360 (3). This application tracks the GPS location of a phone so that family members and friends can be assured that their loved one is safe. It offers functionality of being able to emergency call a representative from the company, in the case that they are in danger (3). Like FireAware, Life360 has safety as its main priority, mainly using location services to ensure that the user is not in an area that is dangerous. Being able to quickly call the fire department would be a good addition to this project.

4. Input/output and proposed solutions

Dataset from New York City open data:
Mandatory Inspections by Fire Companies: https://data.cityofnewyork.us/Public-Safety/Mandatory-Inspections-by-Fire-Companies/kfgh-h6re (4)
This dataset contains a list of all fire inspections done by fire companies in New York City since 2004; this is the dataset to be used for the entirety of the application.

The application should generate some data type, possibly a hash map, containing buildings with their corresponding inspection dates, as well as other important information (e.g. geographical location). This output will be used when plotting each location on the map to show the number and dates of previous inspections done.

This application should also generate the average amount of time done between inspections, dependent on geographical location and the building's own previous inspection dates.

The buildings will be plotted onto a map user interface, similar to Google Maps. When clicking on a building, it will display detailed information about previous fire inspection dates, as well as the company that performed the inspection. The points should be able to be remotely accessed (i.e. can be searched by name, the user does not have to be near the building to get information.) There should also be location tracking (should the user enable it) to send notifications when a user walks into a potentially hazardous building.

To determine if a building may have hazards in fire safety, the average time between inspections (based on geographic location) is calculated. From there, a "cautious" time interval can be decided upon based on the average time between inspections. This "cautious" time interval warns users when a building is due for a safety inspection, but is still relatively safe enough because it hasn't passed the expected inspection time. The exact algorithm isn't yet defined, but it will be based on the building's previous inspection dates as well as the inspection dates of buildings in its vicinity.

This application should, ideally, be very user-friendly and be able to identify dangerous locations nearly instantaneously.

On top of the main functionality of the application described previously, there is potential to use this product to help evaluate the fire safety of an area (borough). The output that the product can generate is a metric of how safe some radius (say, 5km) is around a given location.

Combined with an application that checks crime statistics by area, this output can be a powerful tool to locate safer areas to live in New York City, for users who may wish to live there but want to make an informed choice. It is also a way for fire companies as well as the government to track how they are doing with regards to fire inspections, and if the city is up to par.

5. Algorithmic challenges:

To create this application, the dataset would need to be sorted by building number. Merge sort would, by nature, retain the order of inspection dates (as they are already listed from oldest-newest in the dataset); however, it is slower than quicksort (5, 6). Quicksort may mix up the order of inspection dates, which is critical to be kept in order. However, once the building numbers are sorted (and thus grouped), it should be simple to sort the inspection dates, as the data will be reduced into many small groups.

Searching needs to be implemented to be able to group buildings by geographic location, both to find the time between inspections by different locations, as well as being able to find each building when implementing the user interface that plots locations on a map.

Graphing algorithms can be used to find the shortest path between two buildings (7). With the help of a map API, such as Google Maps, it can be used to find alternative venues that are more fire-safe than their original plan (e.g. finding the closest nightclub that has been inspected for fire safety recently).

6. Project plan

Week	Milestone
1	Parse through data into an acceptable format and figure out the best format to keep data in
2	Implement sorting and searching algorithm for data to be used in plotting points on a map and retrieving their information
3	Implement graphing algorithm on data and create a function to find average number of years between inspections by geographic location
4	Touch up any algorithmic issues, start using external APIs to plot points on a map and create a user interface
5	Finish making working prototype
6	Fix any bugs and refine user interface to be clear and presentable
7	Finish presentation and documentation; have a demo-able product

References

(1) BBC. (2019, January 07). Escape room fire: Poland closes 13 venues after five teenagers die. Retrieved February 1, 2019, from https://www.bbc.com/news/world-europe-46781545FRUCT. (2014, September 2). Firepoint. Retrieved February 1, 2019, from https://play.google.com/store/apps/details?id=org.fruct.oss.firepoint&hl=en_US

(2) Life360. (2019, January 29). Family Locator - GPS Tracker. Retrieved February 1, 2019, from https://play.google.com/store/apps/details?id=com.life360.android.safetymapd

(3) NYC Open Data. (2018, January 24). Mandatory Inspections by Fire Companies. Retrieved February 1, 2019, from https://data.cityofnewyork.us/Public-Safety/Mandatory-Inspections-by-Fire-Companies/kfgh-h6re

(4) Prajapati, K. (2012). Performance of Efficient Sorting Algorithms for duplicate data. *INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING,MANAGEMENT & APPLIED SCIENCE.,1*(10). Retrieved February 1, 2019, from https://www.researchgate.net/publication/235413510_Performance_of_Efficient_Sorting_Algorithms_for_duplicate_data.

(5) Sedgewick, R., & Wayne, K. (2015). *Algorithms*(4th ed.). Upper Saddle River, NJ: Addison-Wesley.

(6) Chaterjee, U. (2014, July 10). *Applications of Three Common Graph Theoretical Algorithms*[Research paper]. Retrieved February 1, 2019, from https://static1.squarespace.com/static/56b6357e01dbaea0266fe701/t/56edad36859fd01fef0b906e/1458416953536/Chatterjee_Applications_of_Three_Common_Graph_Theoretic_Algorithms.pdf