CS 491 Senior Design Project I

Project Specifications Report

Project short-name: Pigeon's Map

Berdan Akyürek 21600904

Ömer Olkun 21100999

Tanay Toksoy 21703919

Abdullah Ayberk Görgün 21201986

Ekin Üstündağ 21602770

Supervisor: Özcan Öztürk

pigeon-s-map.github.io

1. Introduction

Postal workers spend their days on the streets to carry packages to people. They deliver more than a few packages per day. With the increase in the number of packages to deliver, it gets harder to find the best way to visit all places efficiently. Choosing and following a route manually can be both time-consuming and tiring for the postal worker, considering the high chance to fail to find the optimal route with many places to deliver. Also, if a postal worker with a motor vehicle uses an inefficient route, more fuel will be consumed, which will be harmful to the environment.

Furthermore, people's shopping habits changed by Covid-19 pandemic. The tendency of people to do shopping online has increased. As a result of this, the workload of postal workers increased too [1]. That change raised the significance of the route of the delivery.

There is no guarantee that any routes will remain stable in terms of efficiency due to social, environmental, and natural factors in daily life. Therefore, every day even every hour, the optimal route can be changed. This dynamism in the roads creates the need for a software to calculate the optimal route dynamically.

With this project, we wanted to solve these problems and make it easier to find the best route to visit multiple stops with less effort, time, and fuel by calculating the best route for both pedestrian postal workers and postal workers with a vehicle.

1.1 Description

Pigeon's Map is an Android[2] application idea that makes route planning easy and efficient for multiple stops. It is mainly designed for postal workers, but it can be used by anyone who needs to go to multiple places with minimum time and effort.

It will get a list of addresses and their priorities from the user and suggest a route on the map considering each address's priority. If all addresses' priorities are the same, the software will find the most efficient order of addresses. This way, the user will save time and energy.

Also, the user will be able to grade different streets. If a street is marked with low points, the software will suggest a route that does not pass from that street if possible. If it is possible to reach the desired point without passing that particular road, but it makes the whole route much longer, the software may ignore the user's grade on that road. However, if changing the road will not significantly differ, the software will not pick the undesired road.

Also, users will be able to create notes, suggestions, or warnings about a road or a point like Yandex.Navigator[3]. These notes will be public, and each user will see warnings about points on his/her way. In this way, postal workers will be able to communicate with each other and warn each other about their common problems.

Lastly, the software will be able to suggest routes for both vehicles and pedestrians. The software will suggest pedestrianized zones and will ignore road directions, and it will not suggest roads that are closed to pedestrians. For vehicles, it will not suggest pedestrianized zones, and it will consider road directions. This way, the most efficient and safe route will be suggested to the user.

1.2 Constraints

1.2.1 Implementation Constraints

- The software will be written in Java[4].
- An open-source map API like OpenStreetMap[5] will be used to calculate and show the route to the user.
- Object-Oriented Programming coding style will be used.
- To store user warnings and login operation, a server will be used.
- Calculations will be operated on the client's device.

1.2.2 Economic Constraints

- The software will be free to use for everyone without annoying ads.

- A free map API will be used.

- For the website, Github Pages[6] will be used for free.
- There will be a server to hold user data and share user notes between users. A system like Firebase[7] can be used.

1.2.3 Sustainability Constraints

● The software can be updated, and new features can be added by user feedback.

1.2.4 Social Constraints

● Users of the software can communicate with each other by notes and warnings in the application. This function will create social cooperation between people of similar jobs.

1.2.5 Technical Constraints

● The software will be time-efficient because it should not take too much time for usability.
● The software will be efficient in CPU and memory usage because it also should work on devices with low specifications.
● The software will have limited internet usage during traveling because it will mostly work outside, and most of the users will have to use their limited mobile internet.

1.2.6 Language Constraints

● Since English is the most common language globally, the default language of the software will be English.
● Other languages may be added later.

1.2.7 Ethical Constraints

● User notes will be public to other users, but these notes may include inappropriate content. In this case, there will be a report option. This way, people who find a post abusing can report it, and after checking, if it is inappropriate, it will be removed, and action against that user will be taken.

1.2.8 Legal Constraints

● Since we use user personal information and location, we will keep them secure and not share them without user permission.

1.3 Professional and Ethical Issues

1.3.1 Professional Issues

- During every step of the project, we will work professionally.
- The workload will be shared fairly.
- Each group member will respect other people to keep professionalism.

- Online communication tools like Discord[8] will be used for group communication.

1.3.2 Ethical Issues

- We will follow the Code of Ethics by ACM[9].
- We will not share any personal user info.
- We will not use any other people or organization's work. This project will be our own work.
- We will use the location information of users only to offer them a good experience. We will not store or share/sell this information.

2. Requirements

2.1 Functional Requirements

- Users should be able to create an account.
- Users should be able to login with their existing accounts.
- Users should be able to enter multiple addresses.
- There should be a function to set priority for each address.
- There should be a street voting system.
- Users should be able to change/remove their grades on a street.
- Users should be able to create notes to warn others.
- Users should be able to delete their notes.
- Users should be able to report any abusive content.

2.2 Non-Functional Requirements

- The software should be a mobile application.
- The software should work on Android devices.
- The software should be open source.

- The software should be easy to use.
- The software should be user friendly.
- The software should work fast.
- The software should be efficient.
- The software should be secure from outside attacks.
- Some functions may require an internet connection.

3. Conclusion

Finding the optimal route of delivery is one of the essential factors for postal workers. Efficiency and saving time is critical for them. They may fail to know the roads' daily conditions because of unexpected events such as social meetings and lag bursts. Although the time is always essential for the postal workers, now it is even more critical because of the COVID-19 pandemic. Their workloads have increased because of the increase in online shopping. Our application aims to save time for postal workers as much as possible by avoiding excessive routes.

4. References

[1] "Pandemic increases workload, health risks for postal and delivery employees", *PBS NewsHour*. [Online]. Available: https://www.pbs.org/newshour/show/pandemic-increases-workload-health-risks-for-postal-and-delivery-employees. [Accessed: 12-Oct- 2020].

[2] "Android | The platform pushing what's possible", *Android*. [Online]. Available: https://www.android.com/. [Accessed: 11- Oct- 2020].

[3] "Add a road alert - Yandex.Navigator. Help", *Yandex.com*. [Online]. Available: https://yandex.com/support/navigator/events-add.html. [Accessed: 11- Oct- 2020].

[4] "Java", *Java.com*. [Online]. Available: https://www.java.com/en/. [Accessed: 11- Oct- 2020].

[5] "API - OpenStreetMap Wiki", *Wiki.openstreetmap.org*, 2020. [Online]. Available: https://wiki.openstreetmap.org/wiki/API. [Accessed: 11- Oct- 2020].

[6] "GitHub Pages", *GitHub Pages*. [Online]. Available: https://pages.github.com/. [Accessed: 12- Oct- 2020].

[7] "Firebase", *Firebase*. [Online]. Available: https://firebase.google.com/. [Accessed: 11- Oct- 2020].

[8] "Discord | Your Place to Talk and Hang Out", *Discord*. [Online]. Available: https://discord.com/. [Accessed: 12- Oct- 2020].

[9] "Code of Ethics | National Society of Professional Engineers", *Nspe.org*. [Online]. Available: https://www.nspe.org/resources/ethics/code-ethics. [Accessed: 11- Oct- 2020].