4 Design

4.1 Design Context

4.1.1 Broader Context

Below is a table containing examples of the ways our project impacts areas outside of the application:

Area	Impact
Public health, safety, and welfare	Increases pollution from more travelers visiting points of interest. May increase the spread of illnesses. Can be used by emergency response vehicles in case of widespread disasters.
Global, cultural, and social	Encourages more exploration. Gives travelers a more diverse set of places to visit.
Environmental	Increased energy use from running servers computing recommended routes.
Economic	Promotes tourist exploration. Can improve efficiency of rideshare routing. Beneficial to local economies.

4.1.2 Prior Work/Solutions

With the rise of smartphones over the last couple decades, an increasing number of users have begun to use keyword-based search for content that is often related to their geographic location. This has sparked research focused on integrating location into keyword-based querying of geo-textual content (Chen et al., 2021). An example of such research is from a paper titled, *KORS: Keyword-aware Optimal Route Search System*, which outlines an algorithm to answer queries that find routes such that they cover a set of user-specified keywords, a specified budget constraint is satisfied, and an objective score of the routes is optimized (Cao et al.). The shortcoming of this approach is that it generates a route that brings the user to the specified keywords, but does not take into account the diversity of these keywords/PoIs.

There are many products on the market today that visualize routes between destinations and provide nearby PoIs for people to visit. Some of the more notable products in this area include Google Maps and Roadtrippers. Google Maps is known for its efficient and detailed route planning that allows users to see things like travel time, different routes, and nearby PoIs that the user may like based on their data. Roadtrippers is well known for their travel guides where they provide PoIs that are within a given range along the route the user is taking as well as hotels for the user to stay at along the way.

Neither of the two solutions in the previous paragraph take into account PoI diversity when generating routes for the users to follow. This may lead to routes or travel guides that do not allow users to really see what the area they are visiting has to offer. Our solution fixes this problem by creating routes that maximize the PoI diversity based on constraints given by the user. In addition to this our solution has the following pros and cons as compared to the other products:

- Pros
    - Route maximizes PoI diversity based on given constraints
    - Specify certain PoI categories to visit
    - Does not store user information
- Cons
    - Limited to routes in NYC
    - Limited to starting from a known location

4.1.3 Technical Complexity

For the complexity of the project, we will be incorporating multiple data types into the project like spatial, temporal, and semantic data. We will also be utilizing multiple components/subsystems that make up the design of the project. We will have a frontend that contains the visualization of the project with a map API like MapBox. We will also have a backend that contains the API for the routing algorithms that are developed for us and incorporate that into the visualization. We will be coupling mathematical algorithmic solutions with index structure for efficient accessing of the data, and we will also be incorporating these solutions into the visualization which introduces technical complexity. These algorithms exceed current routing solutions by providing a route that is within a given range and diverse enough for the user.

4.2 Design Exploration

4.2.1 Design Decisions

Below are some key design decisions that will be decided on in the following sections:

Mapping API

> There are a lot of possible services to use when it comes to displaying the routes in our application. The map is the center of the visualization, so it is important we select a service that meets our needs of readability, simplicity, and expandability.

Programming Language

> Everyone on our team has different prior experiences so it is important that we play into everyone's strengths and select languages and frameworks that elevate the success of our team and provide the best opportunities for a good product.

Aesthetic

> The choice of aesthetic, color, and font for the application is important because they affect how well our application's needs are met. The application should look good and be easy to use for our users using it as tourists while users using it as an educational tool for the algorithms need it to be informative and presentable.

4.2.2 Ideation

In the lotus blossom activity pictured below, we have brainstormed for ideas that involve the functional software as well as the UI of our application. We divided our functional software ideas into the categories: FrontEnd, BackEnd, and the Mapping API. As for the design component, we separated these ideas by the aesthetic, color scheme, and font of the application.

Front end: For the front end, we came to the consensus that the main three frameworks that we would look at were React, Angular, and Vue. Since our team knew we wanted to create a website, these three frameworks are very popular and widely used in web development. Through these choices, we evaluated our team's experience and skill sets to determine that React was the best option. In addition to the framework, we also had to decide between JavaScript and TypeScript. After discussing with the team, we decided to use TypeScript because of the several benefits of it allowing us to have static typing and can point out compilation errors at the time of development.

Back end: As for the backend, we have a plethora of options like Java, C#, Python, and COBOL. As a whole, our members have worked on various projects and classes with Java. As a result we unanimously decided to use Java as our back end language.

Mapping API: The Mapping API of choice had several options such as MapBox, Google Maps, Open Street Maps, ArcGIS, and BingMaps. Looking at google maps, we knew that holding an account could possibly create additional expenses or may not cover a whole semester for free. Other applications like ArcGIS are massive software which could require additional training and learning. One factor that played a huge role was that our client had suggested that we use MapBox. With careful consideration and looking further into the mapping api choices, we decided to go with MapBox.

UI and Design: For the design components, we thought of several ideas involving the aesthetic, color scheme, and font. With so many options, it was quite difficult for our team to decide on the exact look of our application. Through brainstorming, we realized there were several factors to consider when selecting these components. One of the issues we realized we could run into was the map colors clashing with the main colors of the app. There were other problems that we could see arising such as using a font that is too difficult to read on different backgrounds. As a team, we have decided to hold off on the decision until we familiarize ourselves with the map options and other design options that we have not explored yet. Creating the lotus blossom for this component helped our team realize other considerations of what we need to be aware of before following through with these design decisions.

4.2.3 Decision-Making and Trade-Off

Below are some trade-offs for each of the ideated options:

Mapping API

> Certain GIS Tools have strict API limitations (such as Google or Bing Maps). Others may be very complex or hard to pick up, and may have too many or too few features for what we need (such as ArcGIS or OpenStreetMap). Because of the huge number of online tutorials available, and the recommendation of our client, we chose MapBox.

Programming Language

> Our team has experience with a very wide variety of programming languages. Because of the complexity of the project, we felt we needed something higher level, but not too high such that it is not performant. We also felt we needed to choose something we were comfortable with using and we were experienced with. For this reason, we chose Java for the backend and REACT for the frontend.

Aesthetics

> Decisions on aesthetics have a profound impact on the user experience. Blue light and complex fonts may be stressful or exhausting to a user. Poorly chosen colors may blend with the map. It is for this reason that we have decided on a simpler aesthetic, but our exact decisions on font and colors are yet to be determined because we would like to get an idea of how they would look with the mapping API first.

Development Environment

Due to most of the team being most familiar with development on Windows, we will be doing all development on DOS based systems. Deciding on a single, uniform development environment is important so that all system calls will work on all systems. If we ever intended to extend our project support to UNIX based systems, we would need to rewrite all of our system calls which could be time consuming, however it would greatly increase the number of compatible systems and the portability of our software.

4.3 References

Chen, Z., Chen, L., Cong, G. et al. Location- and keyword-based querying of geo-textual data: a survey. The VLDB Journal 30, 603–640 (2021). https://doi.org/10.1007/s00778-021-00661-w

X. Cao, L. Chen, G. Cong, J. Guan, N. -T. Phan and X. Xiao, "KORS: Keyword-aware Optimal Route Search System," 2013 IEEE 29th International Conference on Data Engineering (ICDE), 2013, pp. 1340-1343, doi: 10.1109/ICDE.2013.6544939.