Research document

Maps API Research

For an app that heavily relies on mapping/routing, it is cruicial to pick the optimal map API.

Multi-criteria analysis

Problem Analysis

Possible solutions

Our possible mapping/ routing API candidates are:

- Google maps API
- Mapbox

The reasoning behind having these two API's as potential candidates is due to them being the leading API's in terms of popularity and community support.

Criteria

The map API we seek for should do the following:

- display the maps (roads, buildings, street names, markers, other names)
- support routing from point a to point b (and more points)
- allow to create custom onClick events
- adding new custom markers on the map
- it should be affordable for the client
- user friendliness and size of community
- having additional tools

Comparison

Although both API's have a lot to provide, they differ in performance in specific fields.

Google Maps API is expensive in comparison to MapBox. However, Google Maps has a lot to offer. The main selling point for Google Maps

- how detailed the data is
- the additional features.
- community support

1. When we created a map with MapBox we realised that not all shops were displayed already which means we would need to input all the data of the business' in the Walstraat ourselves in a dataset. (*To see the prototype of MapBox please go to "system-docs/proof-of-concept/MapBox_POC"*)

2. As of today, the Google Maps platform provides all types of API's. Which include not only APIs specifically for the map itself but also geographical data such as timezones and distance matrix's that

could possibly come in handy in the future. Another thing worth mentioning is that Google Maps has different type of views, such as sattalite view or real life view.

3. Although the official documentation for both API's are well organized and detailed, due to the popularity of Google Maps it is easier to find the requried help on online tutorials or engineering platforms such as StackOverFlow.

Scales

According to the criterias, the following fields will used for the scaling.

- Price
- GUI
- Routing
- Custom events
- Custom markers
- Community support

Criterions	Google maps	MapBox
Price	high	low
GUI	+	+
Routing	+	+
Custom events	+	+
Custom markers	+	+
Community support	+	-
More tooling	+	-

Outcome of Maps Research

Google Maps has all sorts of features, higher community support and more data regarding the locations on the map. For these reasons Google Maps is more suitable to use in this project.

Database Choice Research

The application will be used by public and users will have their own accounts. A database is needed in order to save all their data regarding their favorite locations and events. This research is for selecting the optimal type of database for this project.

Here are our candidates:

- A traditional SQL database such as PostgreSQL
- A cloud hosted, NoSQL backend solution, Google Firebase.

1. PostgresSQL: Postgres is a relational SQL database. The downside of using Postgre in this project is that the database must be hosted in some way. This can be achieved by using Docker containers or installing the database on a seperate server. The other downside is that there is no direct way to

communicate with the database, there must be some ORM (such as Sequelize) for querying. Although these are not technically difficult issues, they are time consuming and require more effort in learning.

2. Firebase: Firebase is a backend solution from Google. The Firebase NoSQL Database is cloud hosted meaning it can be reached from any part of the world with no configuration or extra hosting step. This backend solution also includes other functionalites such as authentication tools.

Comparison

Platform

In a large project like this with such limited time, minimising the time conspumption is crucial to deliver a viable product. Firebase is very user-friendly and quick to set up. Another huge benefit of Firebase is that it can also provide Authentication and email service to the project.

One con of using Firebase database however is that it is not free to use. However, after researching the pricing model, we came to the conclusion that the usage of the application will most likely not pass the free usage quota. This means that the client will be paying close to nothing.

Database Model

Firebase uses NoSQL while Postgre is relational.

- NoSQL stores data in key-value pairs. This is optimal for this project because the company API we are using returns JSON, which uses the same method to store data.
- SQL is well structured with tables, rows and columns. This makes is great for complex queries whereas NoSQL is not great for this. However our database schema is quite simple and straightforward which makes this issue irrelevant.

Criterions	PostgreSQL	Firebase
Hosting	Local	Cloud Hosted
No usage of ORM	-	+
Authentication Tooling	-	+
Free-to-use	+	-
Learning curve	-	+

Database Research Conclusion

Firebase is the better option of the two with its tools, easy to use nature and pre-configured setup.