Parcel Service

Neelima Sahu / Advanced Database Module

June 2022

Guided by Professor Frank Hefter

HOCHSCHULE HEIDELBERG
SRH
Intelligence in Learning

Table of Contents

Introduction 3

Details 6

Database deployment 9

Application 12

GITHUB Path 14

Evaluation 15

References 16

- Docker Orientation and setup 16
- Spring Boot Reference Documentation 16
- Google Maps Platform Documentation 16

Table 1: Roles and Responsibilites. 3

Table 2 : Tools Description 4

Figure 1: DFD 6

Figure 2: Node module dependency in React Project 7

Figure 3: UML Backend 8

Figure 4: Application Demo 9

Figure 5: Docker Commands 10

Figure 6: UML Database 11

Figure 7: Application Home page 13

INTRODUCTION

The key component of a parcel service provider is parcel delivery. The parcel service provider hands over the sent parcel to the destination in this phase. The shipment is legally delivered only when the parcel or small box is handed over to the recipient in person, delivered to the Parcel Shop, or left at an agreed place.

The Parcel Service application is similar to DHL service, in that shipping orders can be created and tracked using GPS coordinates provided by Google Maps API.

ORGANIZATION

2.1 Roles and Responsibilities

Tasks	Team Member [Neelima]
Brainstorming	Responsible
Requirements Gathering	Responsible
Selection of Database	Responsible
Database Setup	Responsible
UML Diagram	Responsible
Report Documentation	Responsible
User Story	Responsible
Use Case 1	Responsible
Use Case 2	Responsible

Table 1: Roles and Responsibilites.

2.2 Tools Used

Area	Tools
Databases	MongoDB Driver MongoDB Compass Docker Desktop
Framework	React JS Framework Spring Boot Framework Docker Desktop
IDE	IntelliJ Visual Studio Code
Version Control	Git Github
Documentation	Microsoft Word Adobe PDF
UML	IntelliJ
DFD	Lucidchart

Table 2 : Tools Description

2.3 Timeline

Participants: Professor Frank Hefter & Neelima Sahu.

Meetings platform: MS Teams.

Week 1

Meeting 1	25th May 2022
Action	Project Selection
Remarks	Approved
Actions	Installation Docker, DB Setup and Technologies selection.

Week 2

Meeting 2	27th May 2022
Action	Displayed prototype and discussed use case
Remarks	Clarification upon use case by Professor and altered the usecase.
Actions	Looked upon Maps API and implementation methods.

Week 3

Meeting 3	02 June 2022
Action	Displayed Progress report
Remarks	Approved as use case was finalized with database.
Actions	Looked further upon plotting a

	path using Google Maps API

Week 4

Action	Continued to work on illustrating a smooth path on google maps and cleaned code base.

DETAILS

3.1 User Story

An administrator or logged-in user would like to be able to place shipment orders and track them using a tracking ID.

3.2 Identified Use Case

An order should get generated with a unique ID whose location should be tracked with the gps coordinates that are provided by google maps API.

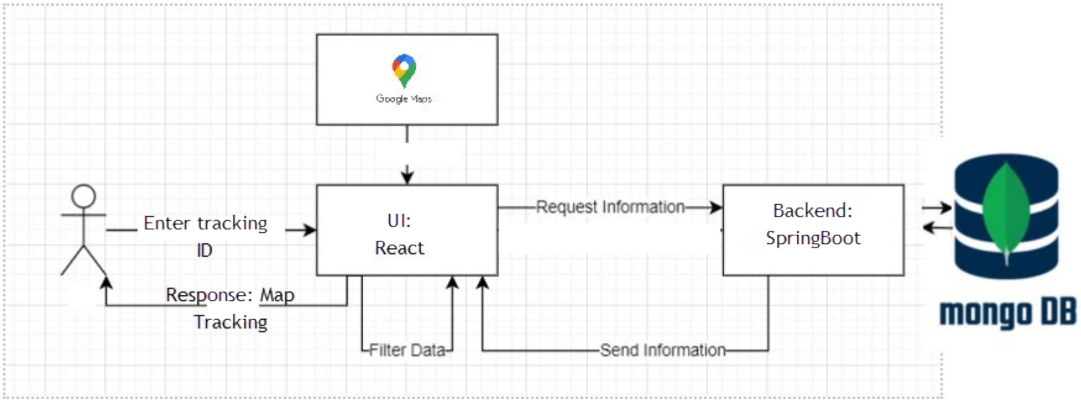


Figure 1: DFD

The React project was built using libraries like chakra UI, which has various new tags to display the UI of the google maps suitable for the intended task. Following is the module dependency displaying the classes built in JavaScript to accomplish the task.

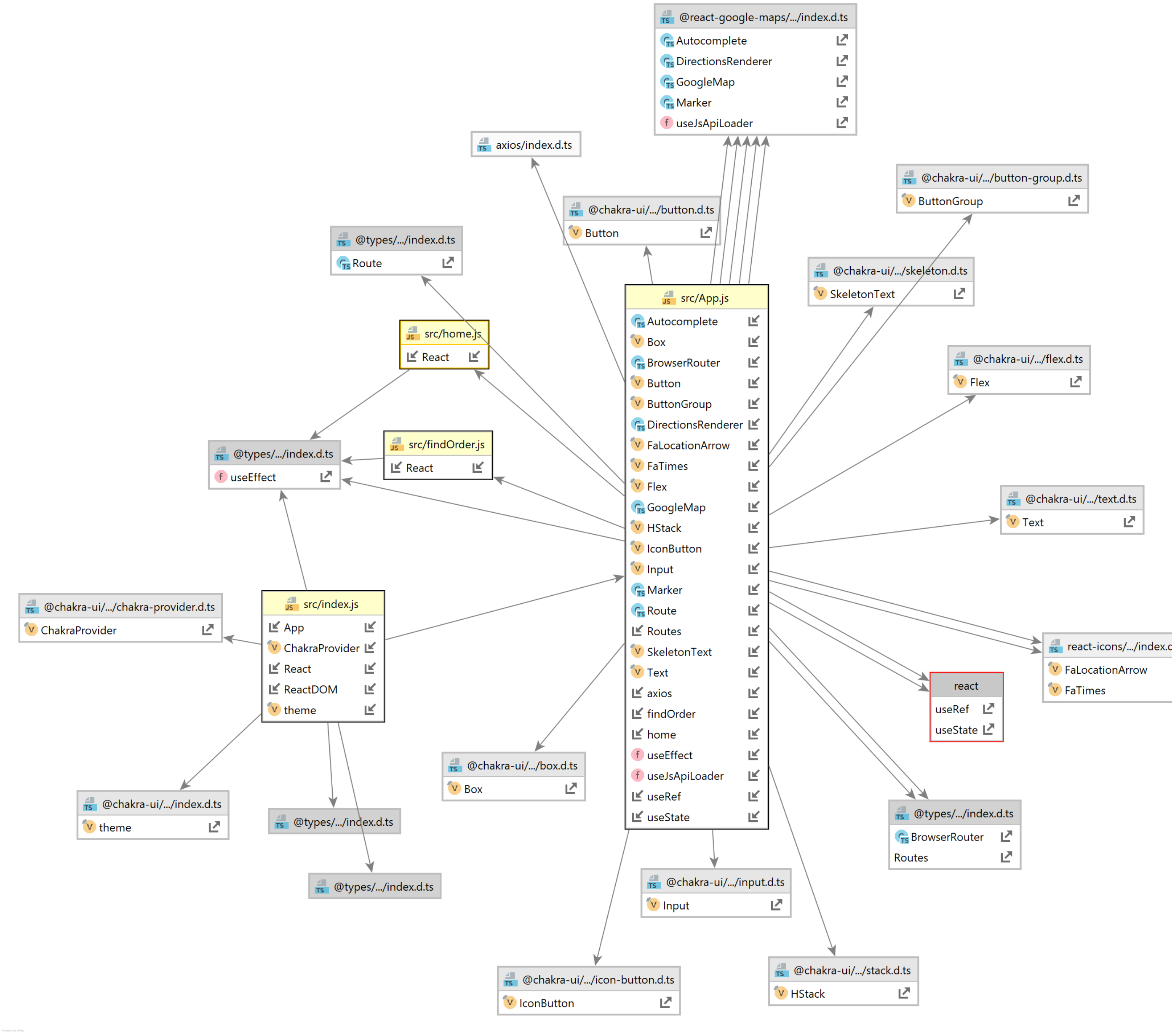


Figure 2: Node module dependency in React Project

The Backend was built using Spring boot framework and following is the UML describing the entire architecture.

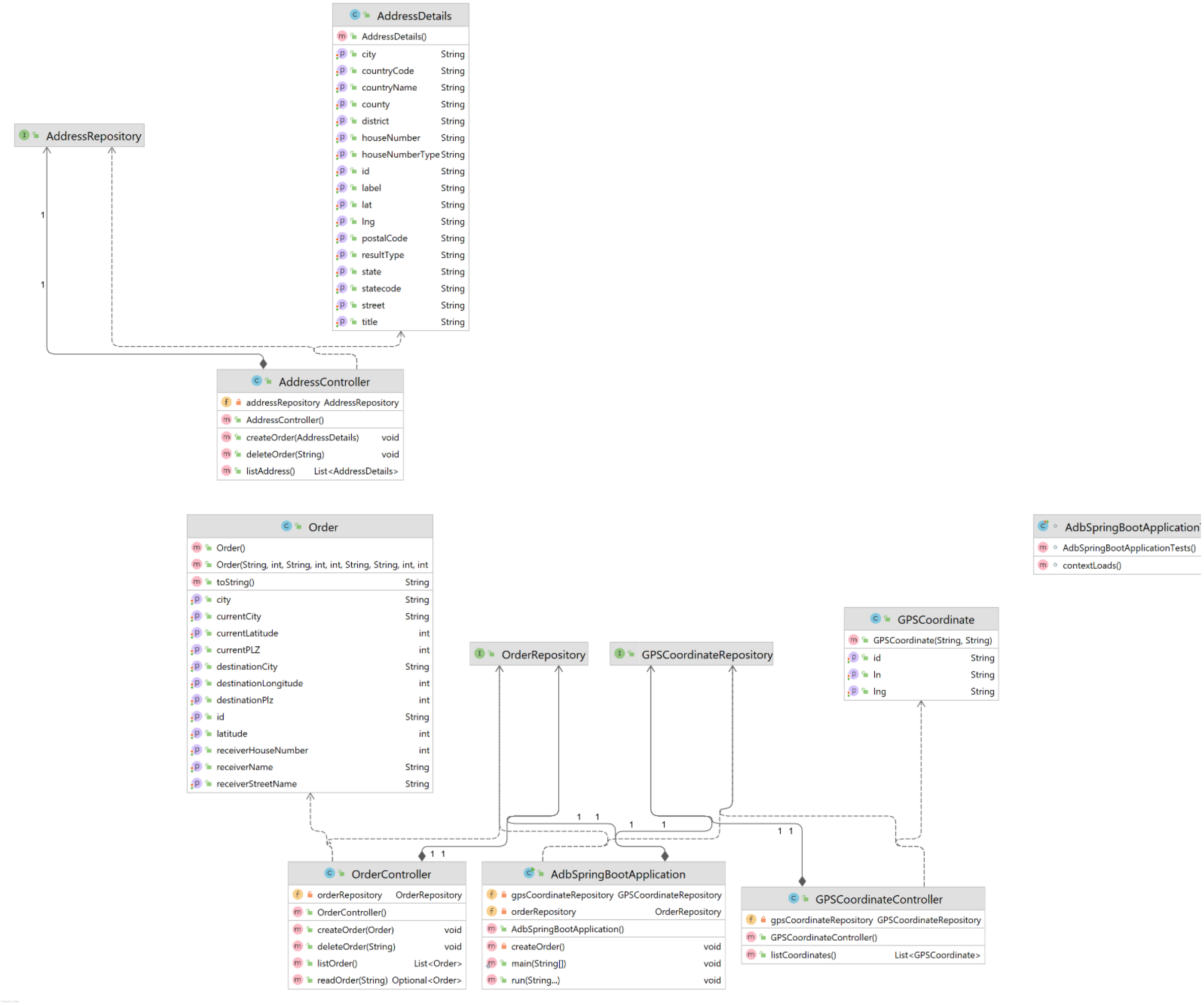


Figure 3: UML Backend

3.3 Database

With Mongo Database as the database continues to grow, we can continue to add more servers. The advantage is that these new servers don't need to be big, expensive machines-they can be cheaper, commodity hardware. Plus, no downtime is required.

Mongo database is suitable for the task for the following reasons:

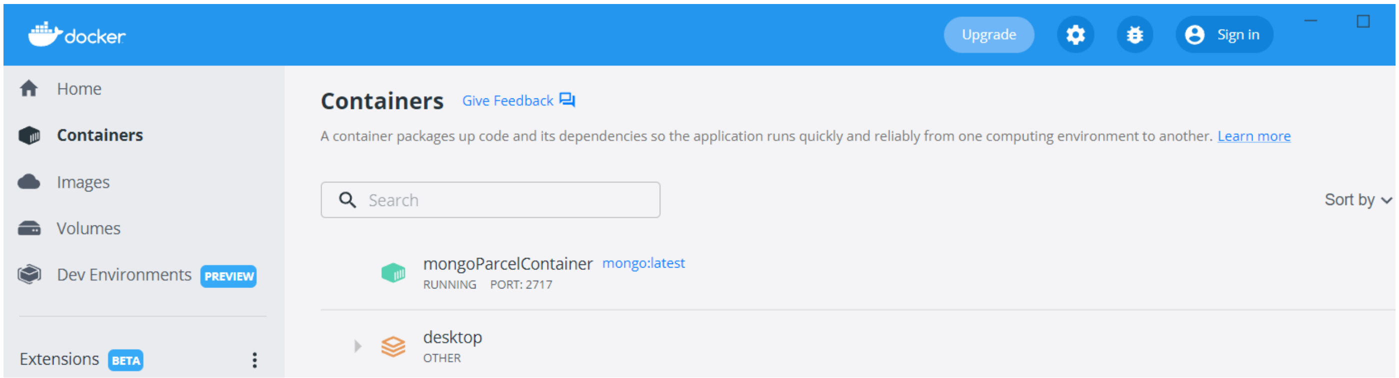
1. Scale Cheaper
2. Query Faster

3. Pivot Easier
4. Program Faster

To achieve the task

DATABASE DEPLOYMENT

The database runs in the container at port 2717 as described in the following screenshot:



Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

mongoParcelContainer mongo:latest
RUNNING PORT: 2717

desktop
OTHER

Figure 4: Application Demo

4.1 Commands used:

- To Create a volume:

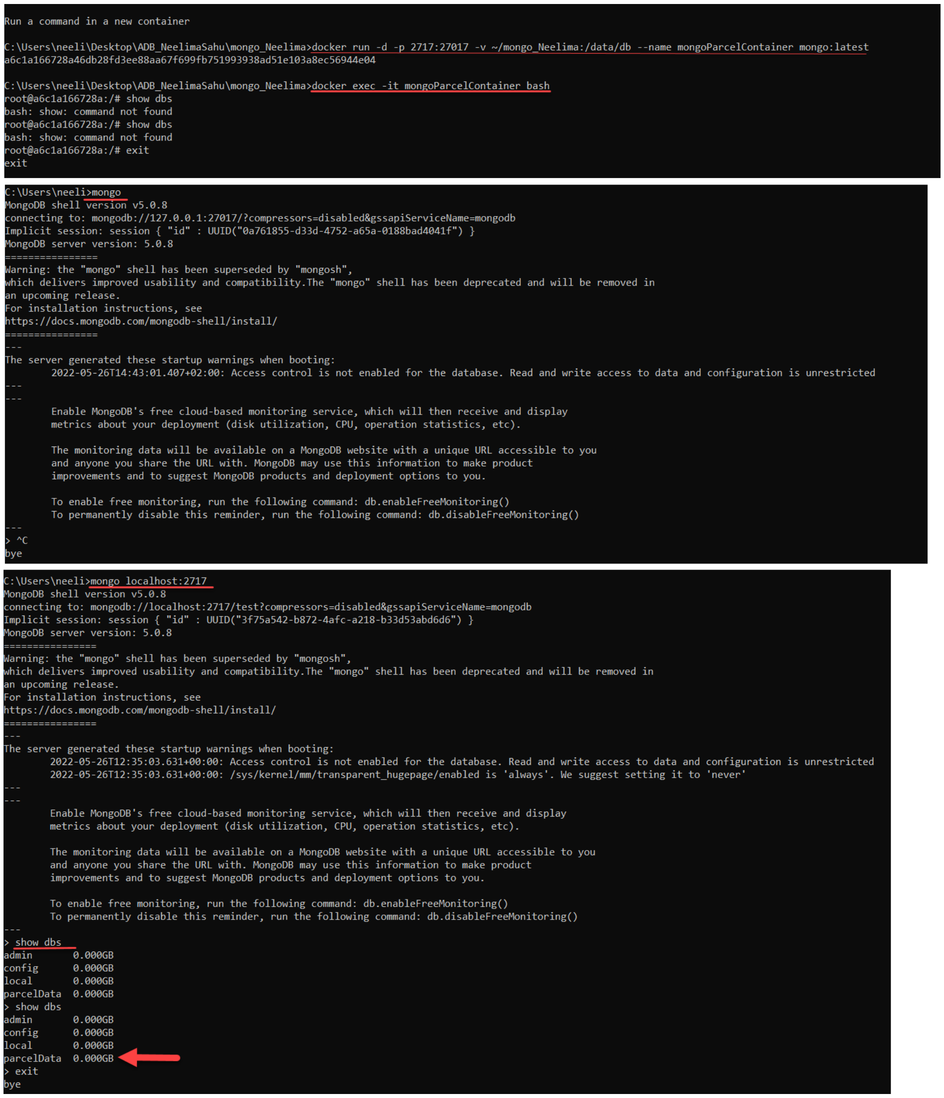
```
docker run -d -p 2717:27017 -v ~/mongo_Neelima:/data/db --name mongoParcelContainer mongo:latest
```

- To enter the container:

```
docker exec -it mongoParcelContainer bash
```

- To RUN Mongo inside the container:

```
root@a6c1a166728a:/# mongo
```



```
Run a command in a new container

C:\Users\neeli\Desktop\ADB_NeelimaSahu\mongo_Neelima>docker run -d -p 2717:27017 -v ~/mongo_Neelima:/data/db --name mongoParcelContainer mongo:latest
a6c1a166728a46db28fd3ee88aa67f699fb751993938ad51e103a8ec56944e04

C:\Users\neeli\Desktop\ADB_NeelimaSahu\mongo_Neelima>docker exec -it mongoParcelContainer bash
root@a6c1a166728a:/# show dbs
bash: show: command not found
root@a6c1a166728a:/# show dbs
bash: show: command not found
root@a6c1a166728a:/# exit
exit
```

```
C:\Users\neeli>mongo
MongoDB shell version v5.0.8
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0a761855-d33d-4752-a65a-0188bad4041f") }
MongoDB server version: 5.0.8
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-05-26T14:43:01.407+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> ^C
bye
```

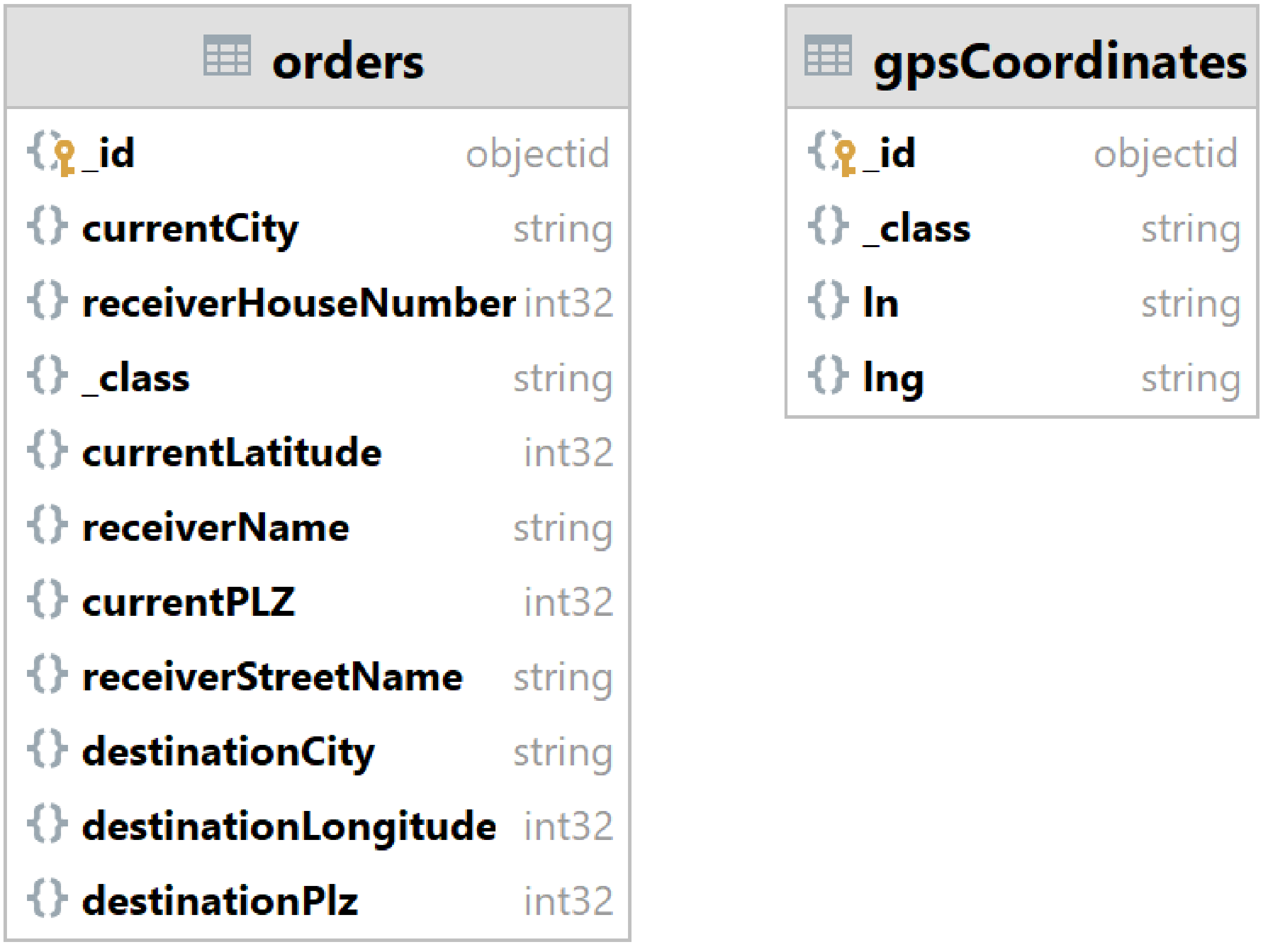
```
C:\Users\neeli>mongo localhost:2717
MongoDB shell version v5.0.8
connecting to: mongodb://localhost:2717/test?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3f75a542-b872-4afc-a218-b33d53abd6d6") }
MongoDB server version: 5.0.8
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-05-26T12:35:03.631+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2022-05-26T12:35:03.631+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
admin       0.000GB
config      0.000GB
local       0.000GB
parcelData  0.000GB
> show dbs
admin       0.000GB
config      0.000GB
local       0.000GB
parcelData  0.000GB
> exit
bye
```

Figure 5: Docker Commands

Following is the UML diagram for the Database running under port 2717



orders	
_id	objectid
currentCity	string
receiverHouseNumber	int32
_class	string
currentLatitude	int32
receiverName	string
currentPLZ	int32
receiverStreetName	string
destinationCity	string
destinationLongitude	int32
destinationPlz	int32

gpsCoordinates	
_id	objectid
_class	string
ln	string
lng	string

Figure 6: UML Database

APPLICATION

6.1 Languages used

React JS

React is an open-source JavaScript toolkit for creating simple, quick, and scalable web application frontends that is efficient, declarative, and flexible.

The JSX programming language is used to create React apps. JSX is a simple JavaScript that simplifies HTML quoting and uses the HTML tag's syntax to create subcomponents.

Because of its ability to break down a complex interface and allow users to work on individual components, ReactJS is a more powerful framework.

The main goal of ReactJS is to provide the highest rendering performance possible. Its strength comes from the focus on individual components. Instead of working with the complete web framework, ReactJS allows a developer to split down a complex UI into smaller components.

Spring Boot

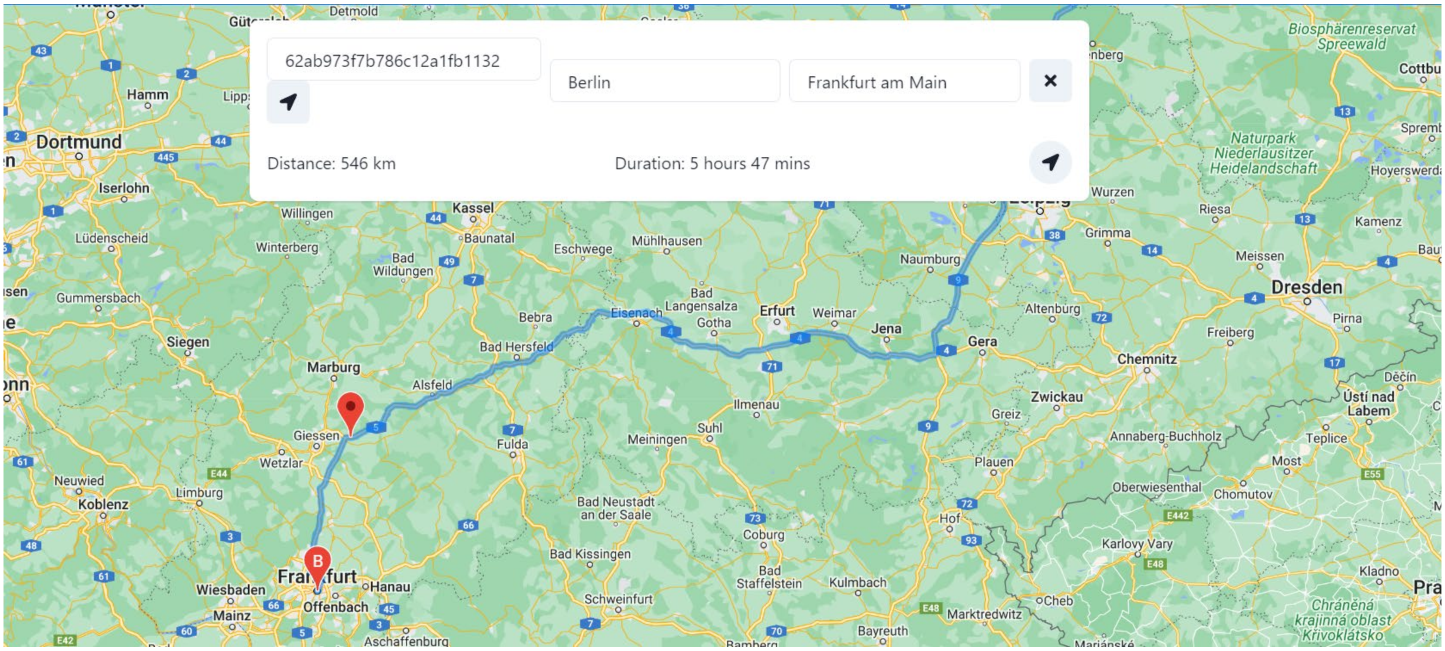
The bootstrapping technique in spring boot makes it possible for users to save space on their devices and load applications quickly.

Application Details:

We can track the gps coordinates available for a certain order by providing the Order Id created by the backend service for order creation.

In the use case, I've kept the coordinates from Berlin to Frankfurt in an excel sheet that gets uploaded when the app starts.

The application also shows the duration and distance, as well as the units of computation, based on the marker placements and their geographical location.



62ab973f7b786c12a1fb1132

Berlin

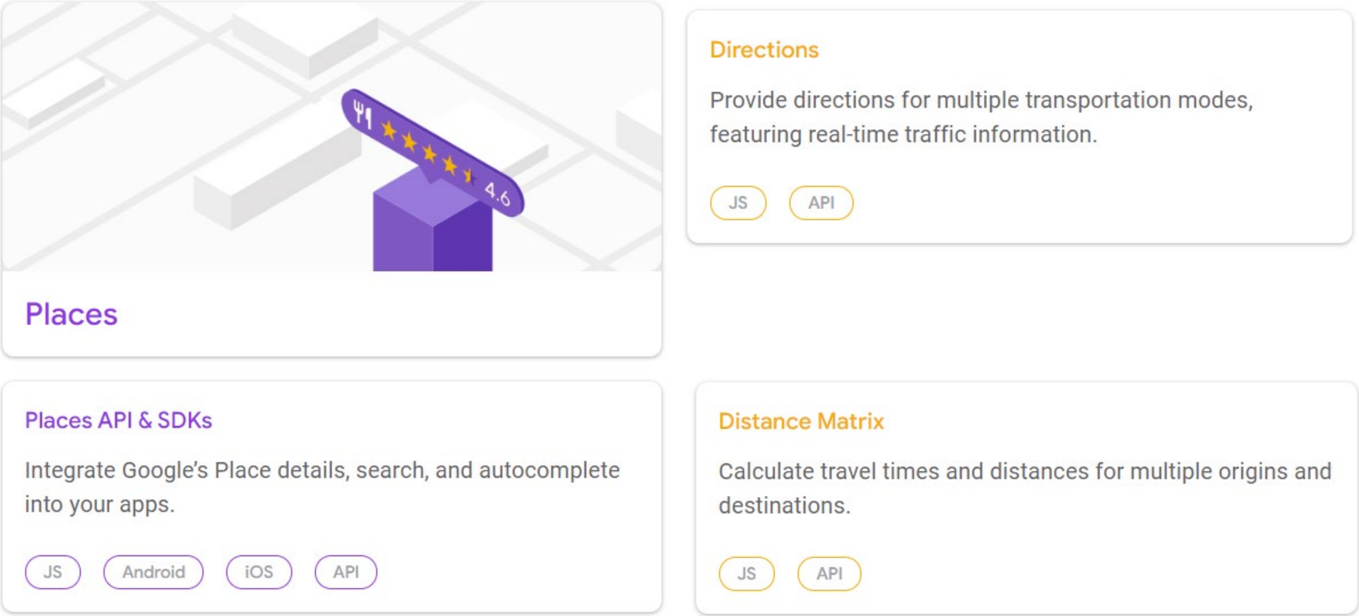
Frankfurt am Main

Distance: 546 km

Duration: 5 hours 47 mins

Figure 7: Application Home page

API's used:



Places

Directions

Provide directions for multiple transportation modes, featuring real-time traffic information.

JS API

Places API & SDKs

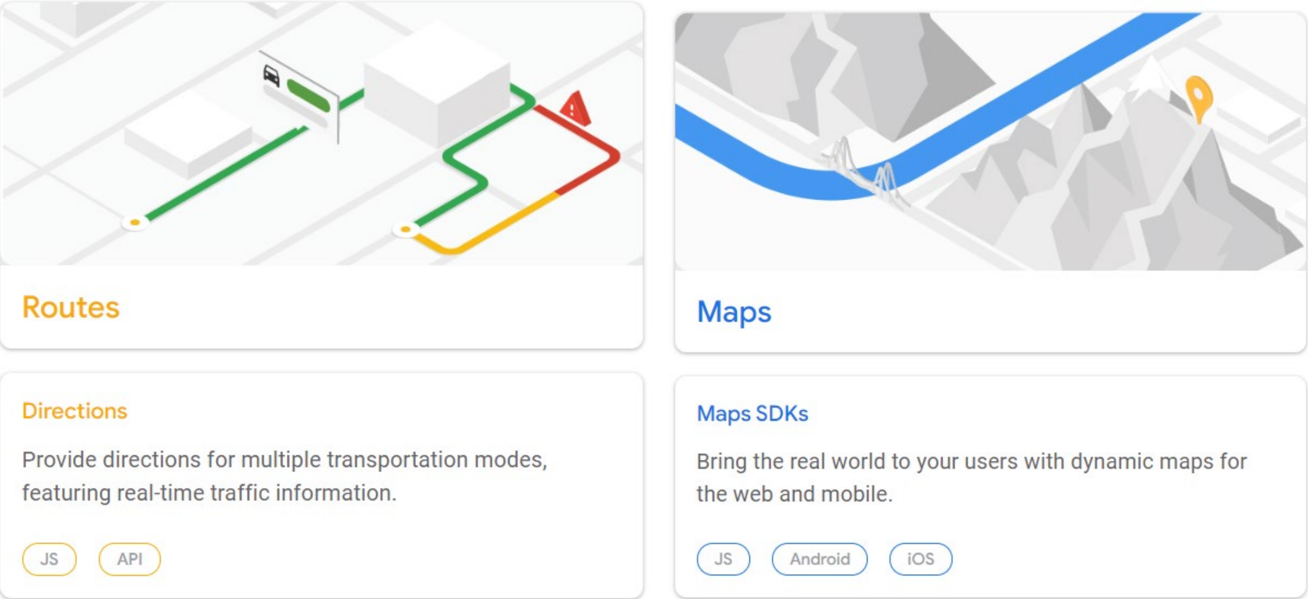
Integrate Google's Place details, search, and autocomplete into your apps.

JS Android iOS API

Distance Matrix

Calculate travel times and distances for multiple origins and destinations.

JS API



Routes

Directions

Provide directions for multiple transportation modes, featuring real-time traffic information.

Maps

Maps SDKs

Bring the real world to your users with dynamic maps for the web and mobile.

GITHUB PATH

https://github.com/neelimasahu-cse/https---github.com-neelimasahu-cse-ADB_TeamGrey_ParcelService_Neelima

EVALUATION

I was able to successfully build and set up the proper data model required for the database structures based on the numerous user stories and use cases given for the application. Some sessions were held with the professor, allowing for the free flow of knowledge and the development of design thinking skills. The mentioned use cases were developed using React.js and Spring Boot to allow users to examine the output of the use cases via Console.

Lessons Learnt

I've learned how to choose and use a database to store information for an application. I gained a thorough understanding of how database models for MongoDB databases might be developed, allowing me to demonstrate the benefits of each database through various use cases. I also learned how to interface with Google Maps' various APIs.

Possible Extensions

- Optimized routes can be selected and chosen for delivery date and time.
- The order ID Creation method can be implemented in React for a better user experience.
- A dashboard with all the various package services can be displayed.

REFERENCES

- Docker Orientation and setup
- https://docs.docker.com/get-started/

- Spring Boot Reference Documentation
- https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

- Google Maps Platform Documentation
- https://developers.google.com/maps/documentation