Sayyara

Team 16 MERN>MEAN

Introduction

The Problem

Your vehicle has broken down - *You need a repair ASAP!*

What are the options for a *price-conscious* vehicle owner?

- Search for nearby automotive repair shops - start making phone calls
- Sit through countless automated menus hoping to talk to a human being
- Repeat your problem and vehicle information every time you get through
- Receive a quote, write it down, rinse and repeat…
- Finally, go with the best price

*Potentially waste **hours** of valuable time*

The Solution

Centralize and *streamline* the automotive repair process

Sayyara functions as your hub for vehicle repair

Without ever leaving the app, the user can:

- Browse repair shops and view their services
- Request quotes for their issue from any participating shop
- Monitor the status of their quotes - All price estimates are consolidated in one place
- Schedule an appointment directly after accepting an estimate

*No more waiting on hold - Shops **compete** for your business!*

Customers Benefit

- Save time, stress and hassle
- Every part of the process is in one place.
- See all available options at a glance
- Easily compare quote estimates for the best deal

Shop Owners Benefit

- See your upcoming appointments on the homepage
- Give price estimates with ease
- See relevant client info, including vehicle specifications and contact information
- Get right down to business, cut out unnecessary verbal communication

Demo

Process and Application Access

Frontend

- Ahsan
- Hamoon
- Nate

Backend

- Uthman
- Jamie

Team Process

- Regular standup meetings to discuss next steps, impediments, etc.
- Semi-regular meetings with the project partner to discuss progress
- Daily coordination within frontend and backend groups
- Frequent coordination between frontend and backend groups
- Github Actions used to automatically run unit tests
- Docker used for the backend

Accessing the Application

- Give the project partner access to a Github repository containing the application code
- Provide detailed instructions on how to run the application locally
- Facilitate a deployed version of the application with a guide on how to update the deployed version

Technical Design

Frontend

TS TypeScript

React

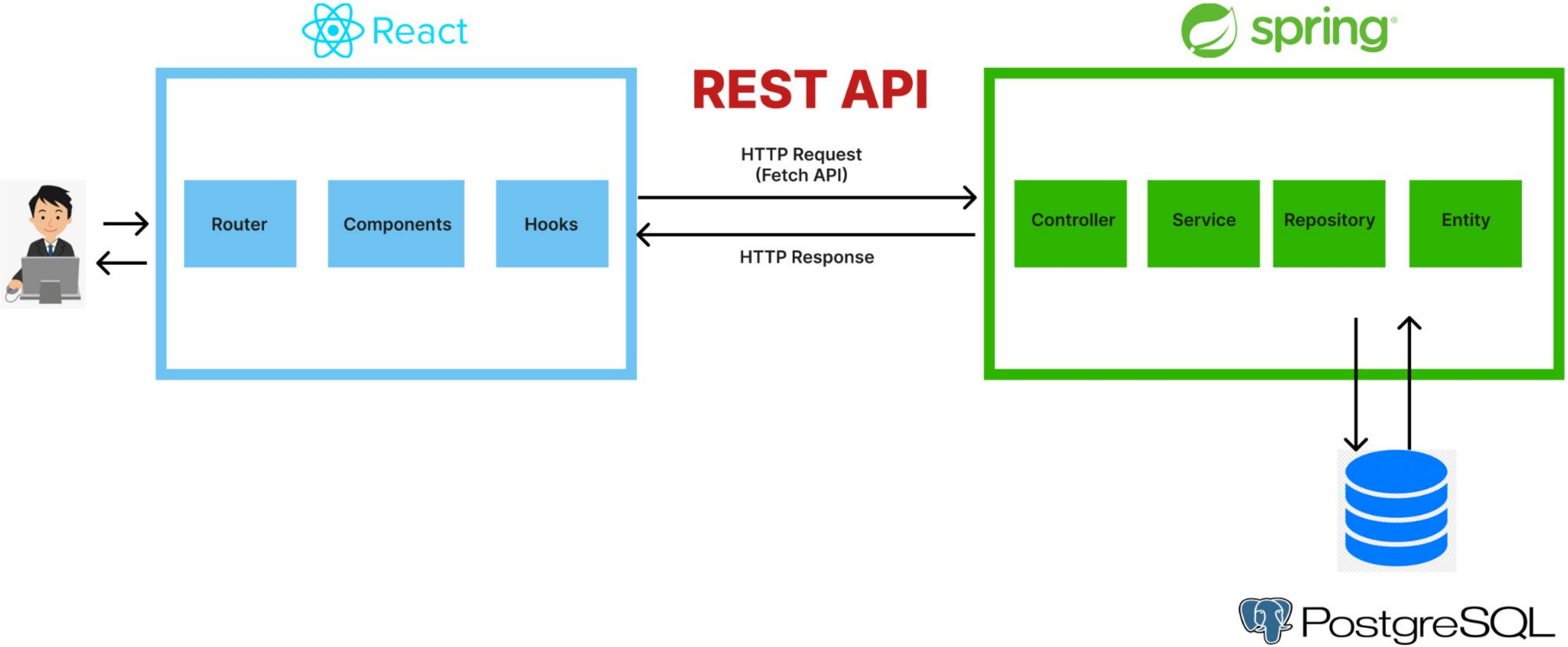
tailwindcss

Backend

Java™

spring®

PostgreSQL



React

REST API

spring

Router

Components

Hooks

HTTP Request
(Fetch API)

HTTP Response

Controller

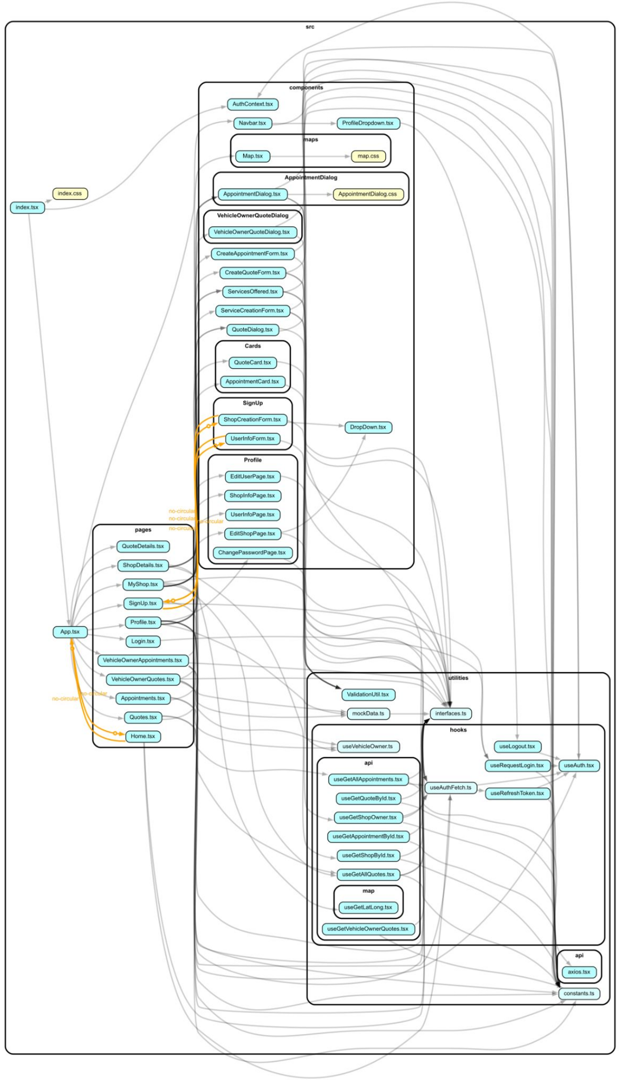
Service

Repository

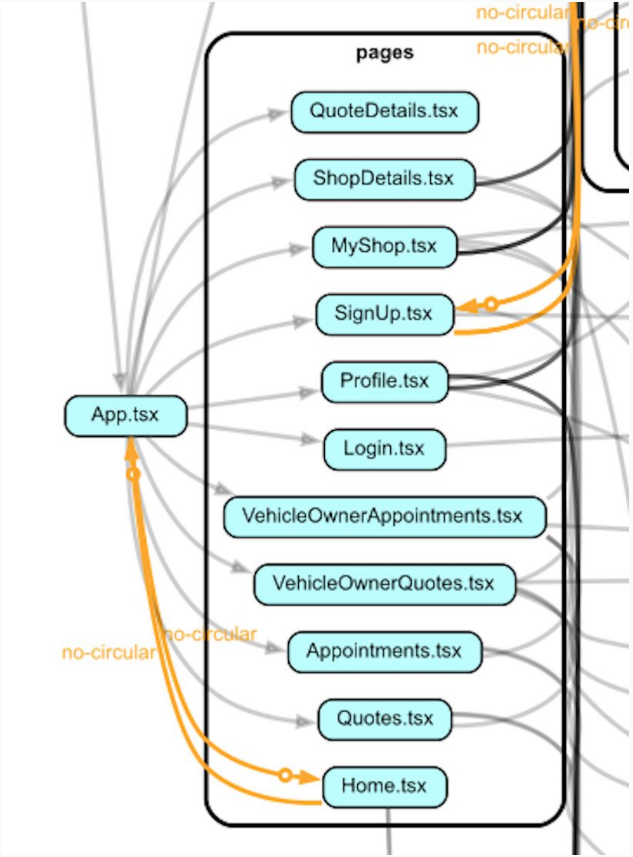
Entity

PostgreSQL

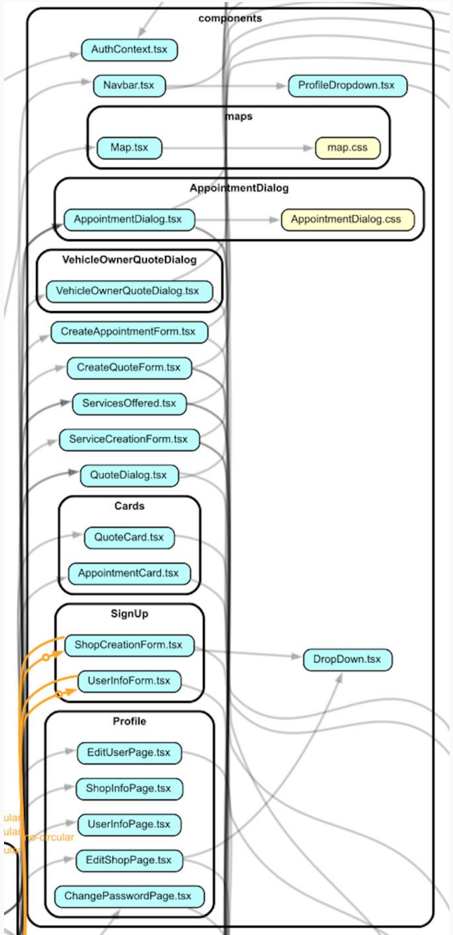
Frontend Architecture



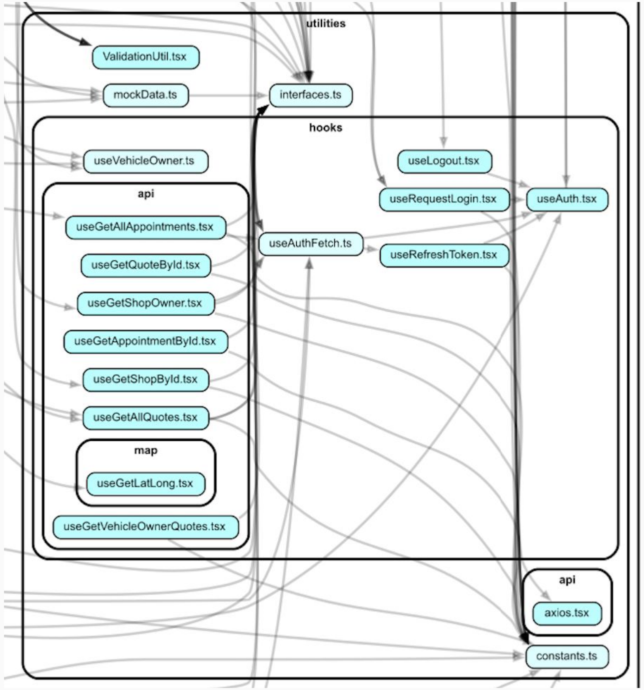
Pages



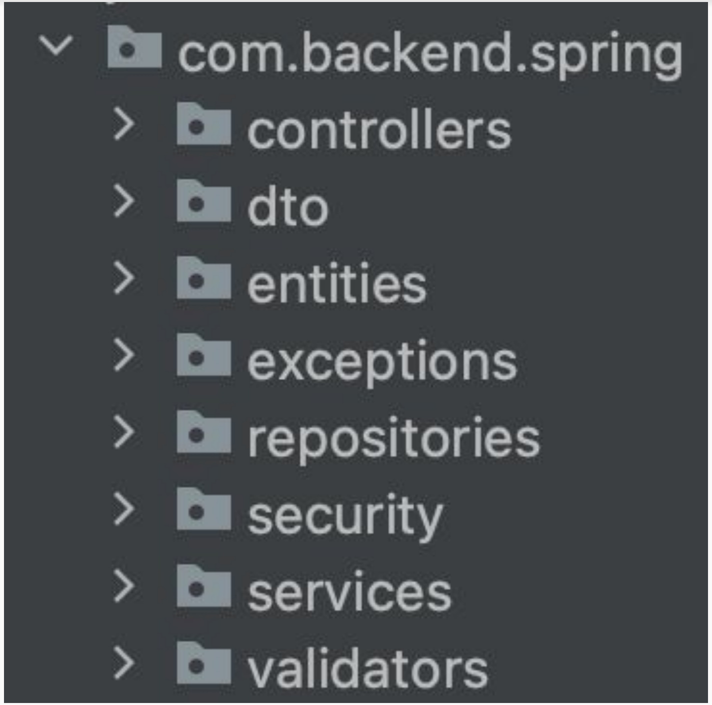
Components



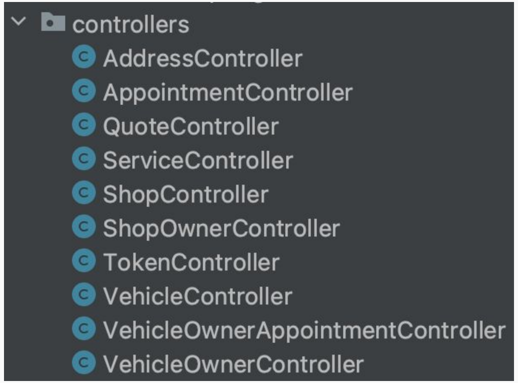
Utilities



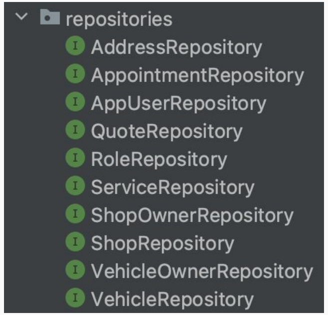
Backend Architecture



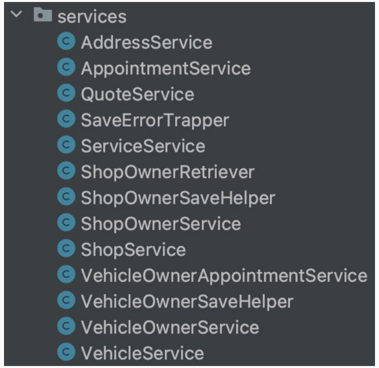
```
com.backend.spring
  controllers
  dto
  entities
  exceptions
  repositories
  security
  services
  validators
```



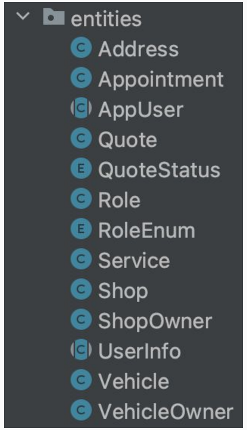
```
controllers
  AddressController
  AppointmentController
  QuoteController
  ServiceController
  ShopController
  ShopOwnerController
  TokenController
  VehicleController
  VehicleOwnerAppointmentController
  VehicleOwnerController
```



```
repositories
  AddressRepository
  AppointmentRepository
  AppUserRepository
  QuoteRepository
  RoleRepository
  ServiceRepository
  ShopOwnerRepository
  ShopRepository
  VehicleOwnerRepository
  VehicleRepository
```

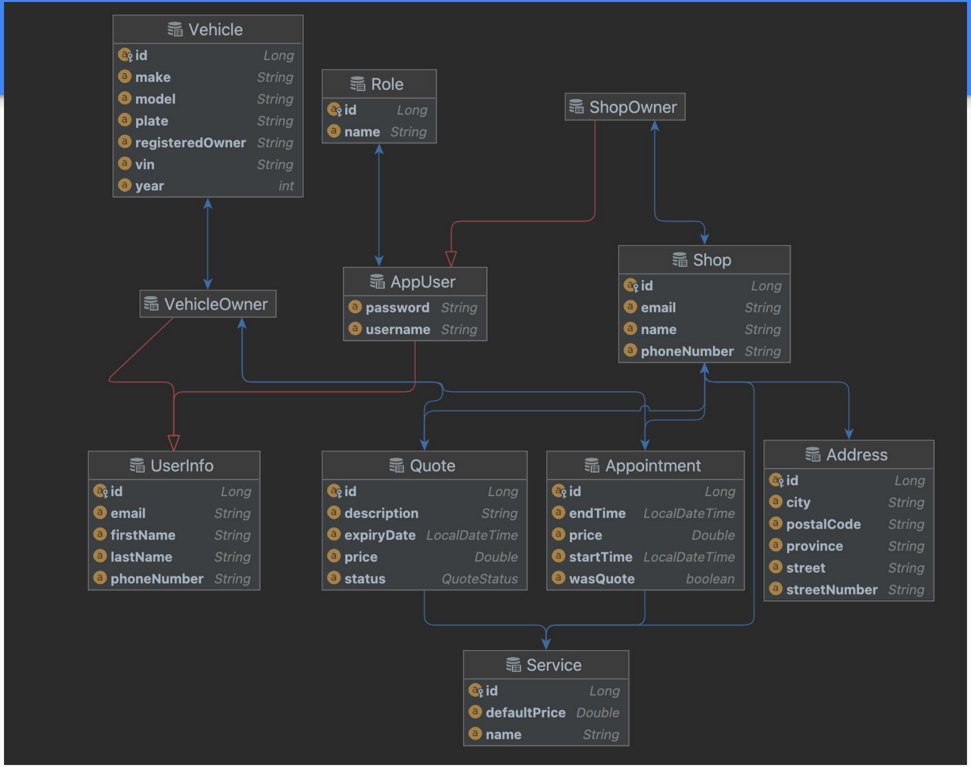


```
services
  AddressService
  AppointmentService
  QuoteService
  SaveErrorTrapper
  ServiceService
  ShopOwnerRetriever
  ShopOwnerSaveHelper
  ShopOwnerService
  ShopService
  VehicleOwnerAppointmentService
  VehicleOwnerSaveHelper
  VehicleOwnerService
  VehicleService
```



```
entities
  Address
  Appointment
  AppUser
  Quote
  QuoteStatus
  Role
  RoleEnum
  Service
  Shop
  ShopOwner
  UserInfo
  Vehicle
  VehicleOwner
```

Database Tables



Vehicle	
id	Long
make	String
model	String
plate	String
registeredOwner	String
vin	String
year	int

Role	
id	Long
name	String

ShopOwner

VehicleOwner

AppUser	
password	String
username	String

Shop	
id	Long
email	String
name	String
phoneNumber	String

UserInfo	
id	Long
email	String
firstName	String
lastName	String
phoneNumber	String

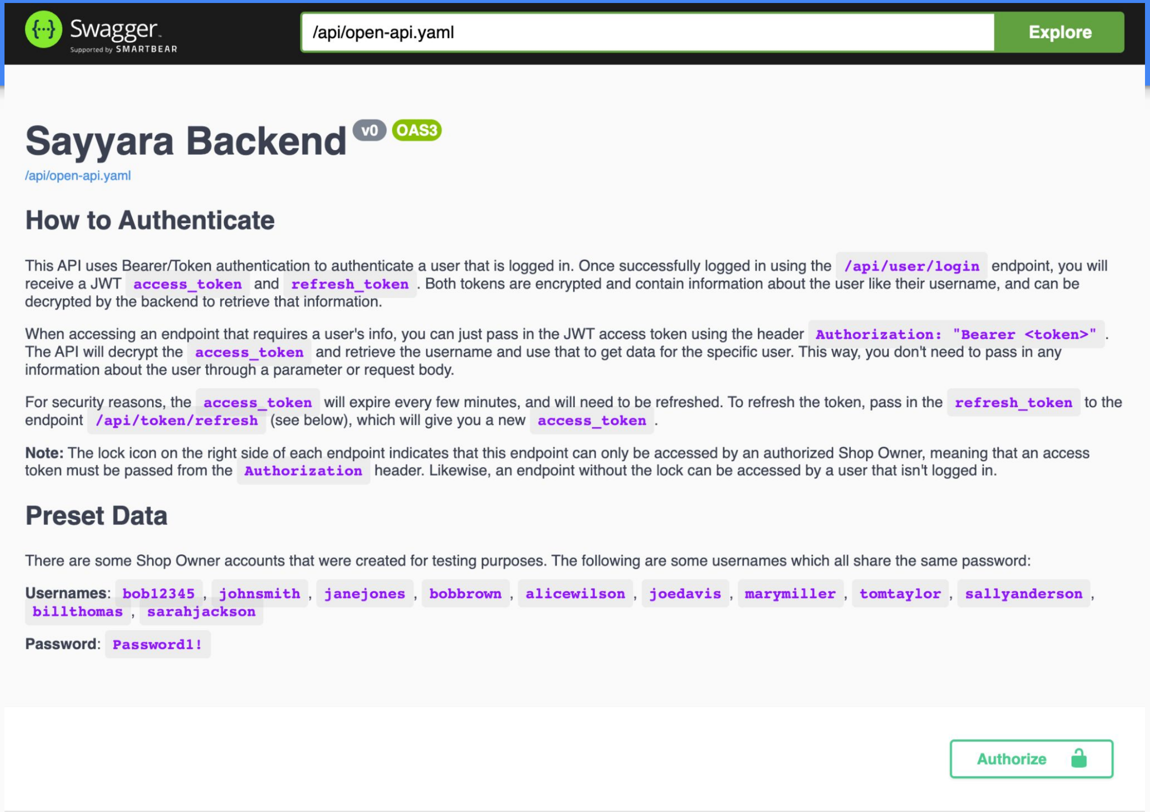
Quote	
id	Long
description	String
expiryDate	LocalDateTime
price	Double
status	QuoteStatus

Appointment	
id	Long
endTime	LocalDateTime
price	Double
startTime	LocalDateTime
wasQuote	boolean

Address	
id	Long
city	String
postalCode	String
province	String
street	String
streetNumber	String

Service	
id	Long
defaultPrice	Double
name	String

Documentation



Sayyara Backend v0 OAS3

/api/open-api.yaml

How to Authenticate

This API uses Bearer/Token authentication to authenticate a user that is logged in. Once successfully logged in using the `/api/user/login` endpoint, you will receive a JWT `access_token` and `refresh_token`. Both tokens are encrypted and contain information about the user like their username, and can be decrypted by the backend to retrieve that information.

When accessing an endpoint that requires a user's info, you can just pass in the JWT access token using the header `Authorization: "Bearer <token>"`. The API will decrypt the `access_token` and retrieve the username and use that to get data for the specific user. This way, you don't need to pass in any information about the user through a parameter or request body.

For security reasons, the `access_token` will expire every few minutes, and will need to be refreshed. To refresh the token, pass in the `refresh_token` to the endpoint `/api/token/refresh` (see below), which will give you a new `access_token`.

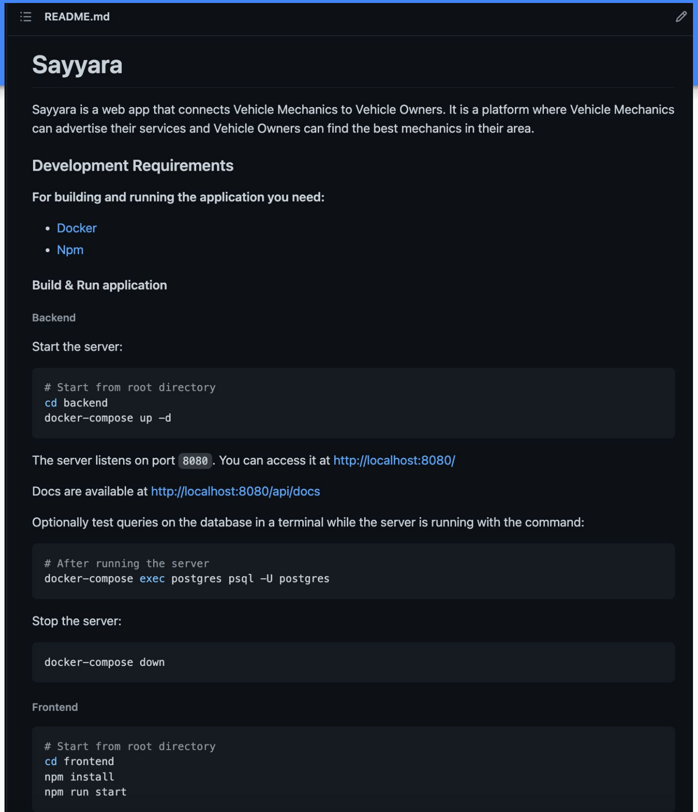
Note: The lock icon on the right side of each endpoint indicates that this endpoint can only be accessed by an authorized Shop Owner, meaning that an access token must be passed from the `Authorization` header. Likewise, an endpoint without the lock can be accessed by a user that isn't logged in.

Preset Data

There are some Shop Owner accounts that were created for testing purposes. The following are some usernames which all share the same password:

Usernames: `bob12345`, `johnsmith`, `janejones`, `bobbrown`, `alicewilson`, `joedavis`, `marymiller`, `tomtaylor`, `sallyanderson`, `billthomas`, `sarahjackson`

Password: `Password1!`



Sayyara

Sayyara is a web app that connects Vehicle Mechanics to Vehicle Owners. It is a platform where Vehicle Mechanics can advertise their services and Vehicle Owners can find the best mechanics in their area.

Development Requirements

For building and running the application you need:

- Docker
- Npm

Build & Run application

Backend

Start the server:

```
# Start from root directory
cd backend
docker-compose up -d
```

The server listens on port `8080`. You can access it at http://localhost:8080/

Docs are available at http://localhost:8080/api/docs

Optionally test queries on the database in a terminal while the server is running with the command:

```
# After running the server
docker-compose exec postgres psql -U postgres
```

Stop the server:

```
docker-compose down
```

Frontend

```
# Start from root directory
cd frontend
npm install
npm run start
```

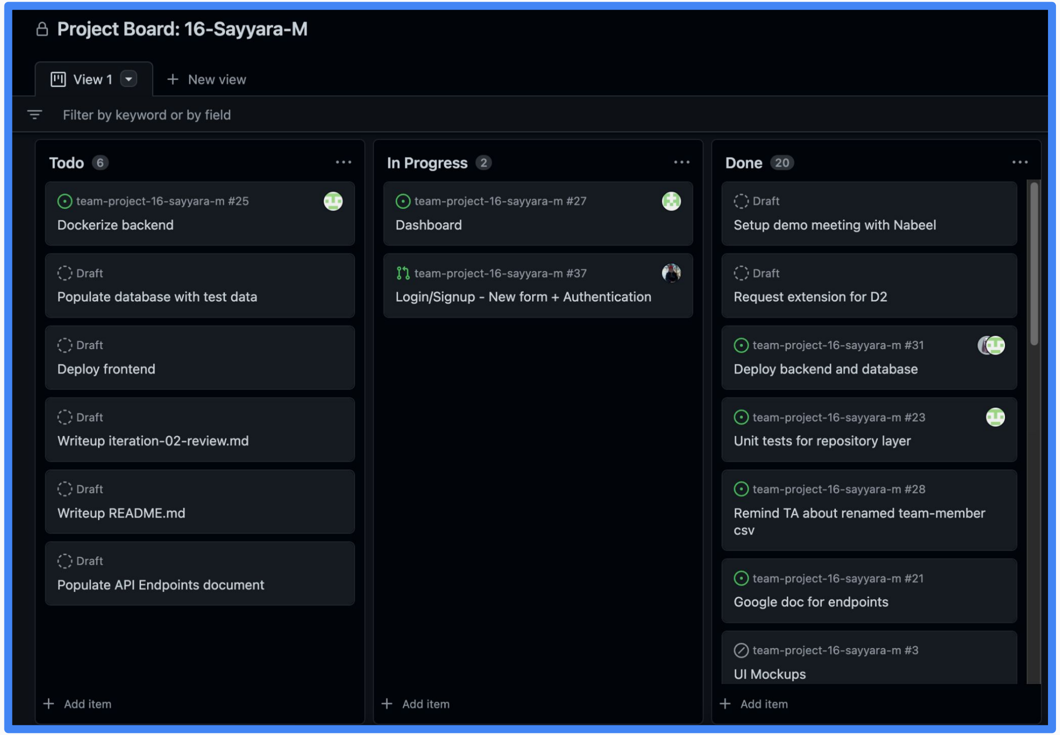
Key Learnings

Things That Went Well

GitHub Project Board

Miro Board

Swagger API Docs



Project Board: 16-Sayyara-M

Todo 6

- team-project-16-sayyara-m #25 Dockerize backend
- Draft Populate database with test data
- Draft Deploy frontend
- Draft Writeup iteration-02-review.md
- Draft Writeup README.md
- Draft Populate API Endpoints document

In Progress 2

- team-project-16-sayyara-m #27 Dashboard
- team-project-16-sayyara-m #37 Login/Signup - New form + Authentication

Done 20

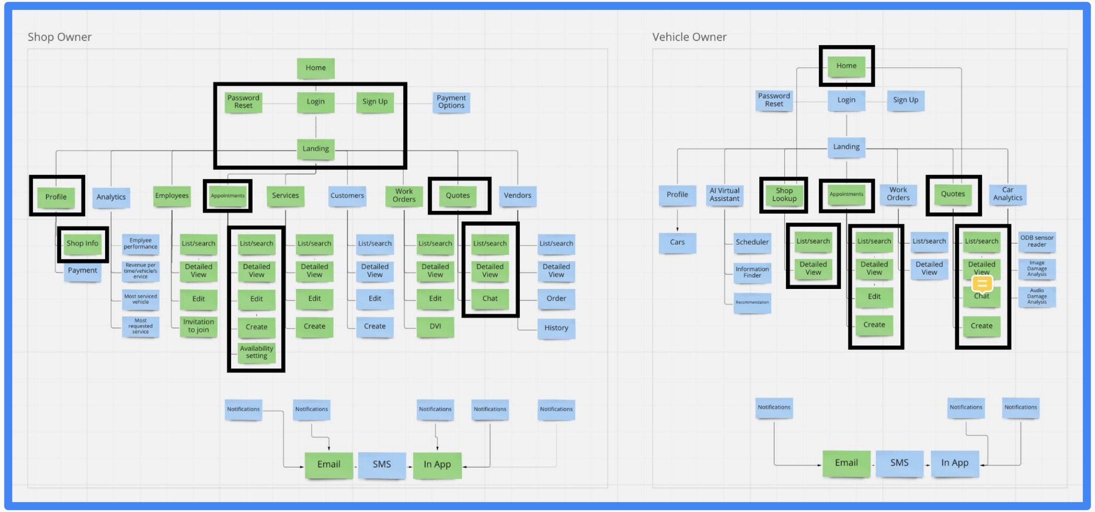
- Draft Setup demo meeting with Nabeel
- Draft Request extension for D2
- team-project-16-sayyara-m #31 Deploy backend and database
- team-project-16-sayyara-m #23 Unit tests for repository layer
- team-project-16-sayyara-m #28 Remind TA about renamed team-member csv
- team-project-16-sayyara-m #21 Google doc for endpoints
- team-project-16-sayyara-m #3 UI Mockups

Things That Went Well

GitHub Project Board

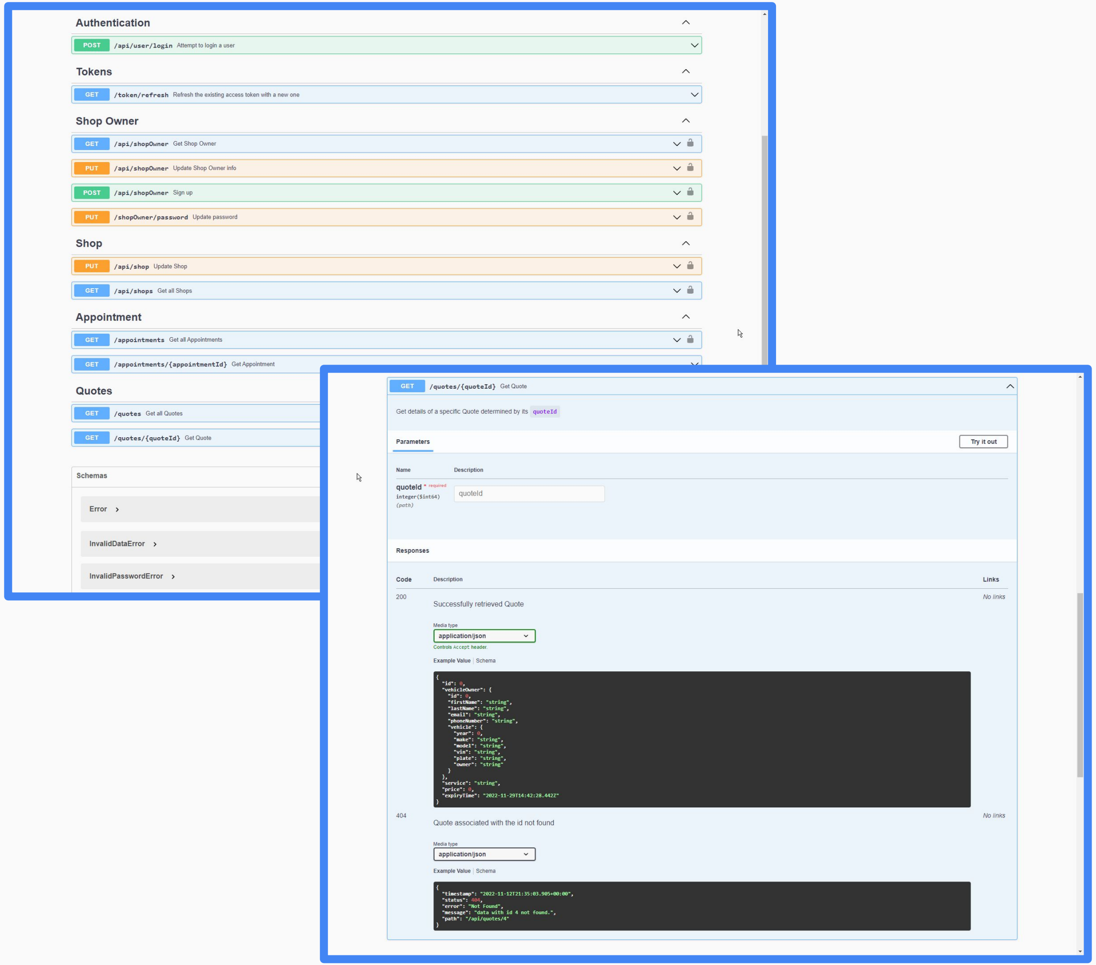
Miro Board

Swagger API Docs



Things That Went Well

- GitHub Project Board
- Miro Board
- Swagger API Docs



Authentication

POST `/api/user/login` Attempt to login a user

Tokens

GET `/token/refresh` Refresh the existing access token with a new one

Shop Owner

GET `/api/shopOwner` Get Shop Owner

PUT `/api/shopOwner` Update Shop Owner info

POST `/api/shopOwner` Sign up

PUT `/shopOwner/password` Update password

Shop

PUT `/api/shop` Update Shop

GET `/api/shops` Get all Shops

Appointment

GET `/appointments` Get all Appointments

GET `/appointments/{appointmentId}` Get Appointment

Quotes

GET `/quotes` Get all Quotes

GET `/quotes/{quoteId}` Get Quote

Schemas

Error ›

InvalidDataError ›

InvalidPasswordError ›

GET `/quotes/{quoteId}` Get Quote

Get details of a specific Quote determined by its `quoteId`

Parameters

Try it out

Name	Description
quoteId * required integer($int64) (path)	quoteId

Responses

Code	Description	Links
200	Successfully retrieved Quote	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "vehicleOwner": {
    "id": 0,
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phoneNumber": "string",
    "vehicle": {
      "year": 0,
      "make": "string",
      "model": "string",
      "vin": "string",
      "plate": "string",
      "owner": "string"
    }
  },
  "service": "string",
  "price": 0,
  "expiryTime": "2022-11-29T14:42:28.442Z"
}
```

Code	Description	Links
404	Quote associated with the id not found	No links

Media type

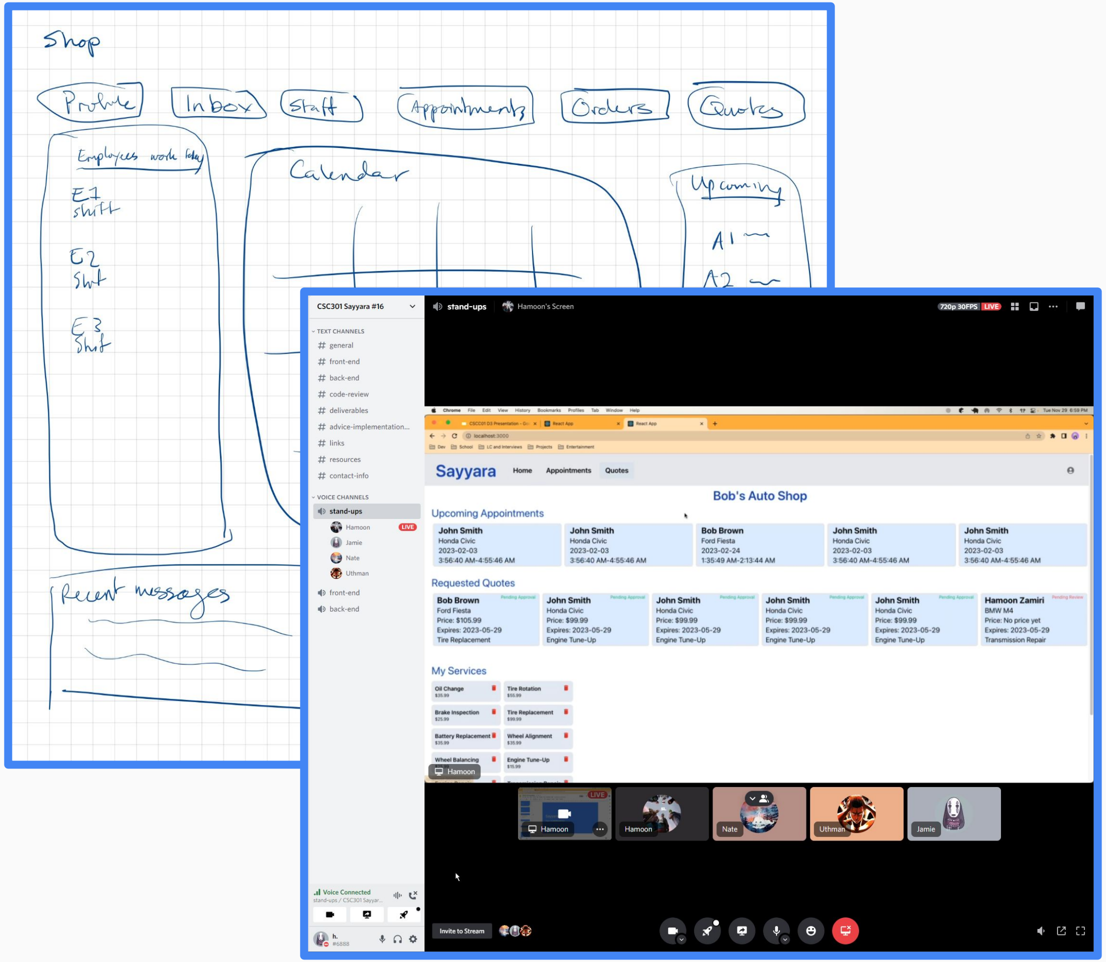
application/json

Example Value | Schema

```
{
  "timestamp": "2022-11-12T21:35:03.905+00:00",
  "status": 404,
  "error": "Not Found",
  "message": "data with id 4 not found.",
  "path": "/api/quotes/4"
}
```

Things That Could be Improved

- Mockups
- Standups
- Internal Deadlines



Sayyara Home Appointments Quotes

Bob's Auto Shop

Upcoming Appointments

John Smith Honda Civic 2023-02-03 3:56:40 AM-4:55:46 AM	**John Smith** Honda Civic 2023-02-03 3:56:40 AM-4:55:46 AM	**Bob Brown** Ford Fiesta 2023-02-24 1:35:49 AM-2:13:44 AM	**John Smith** Honda Civic 2023-02-03 3:56:40 AM-4:55:46 AM	**John Smith** Honda Civic 2023-02-03 3:56:40 AM-4:55:46 AM

Requested Quotes

Bob Brown Pending Approval Ford Fiesta Price: $105.99 Expires: 2023-05-29 Tire Replacement	**John Smith** Pending Approval Honda Civic Price: $99.99 Expires: 2023-05-29 Engine Tune-Up	**John Smith** Pending Approval Honda Civic Price: $99.99 Expires: 2023-05-29 Engine Tune-Up	**John Smith** Pending Approval Honda Civic Price: $99.99 Expires: 2023-05-29 Engine Tune-Up	**John Smith** Pending Approval Honda Civic Price: $99.99 Expires: 2023-05-29 Engine Tune-Up	**Hamoon Zamiri** Pending Review BMW M4 Price: No price yet Expires: 2023-05-29 Transmission Repair

My Services

Oil Change $35.99	**Tire Rotation** $55.99
Brake Inspection $25.99	**Tire Replacement** $99.99
Battery Replacement $35.99	**Wheel Alignment** $35.99
Wheel Balancing $25.99	**Engine Tune-Up** $15.99

Contributions

Ahsan Saeed: Frontend Developer, UX Validations

Hamoon Zamiri: Frontend Developer, Database Design

Haolin (Jamie) Fan: Backend Infrastructure, API Design

Nathan Raymant: Frontend Developer, UI Design

Uthman Mohamed: Backend, API Docs, and JWT Auth

All: Deliverable reports

Thanks!

Sayyara
Team 16 MERN>MEAN