

AnaEE Interface Manual

Beau De Clercq

July 2019

Contents

1	Setting Up for the first time	3
1.1	Docker	3
1.2	Creating the databases	3
2	Usage	4
2.1	Home	5
2.2	Users tab	6
2.3	Vehicles tab	7
2.3.1	Adding a vehicle	8
2.3.2	Removing a vehicle	9
2.3.3	Vehicle details	10
2.4	Stops tab	11
2.4.1	View all stops	12
2.4.2	Search for a stop	13
2.4.3	Search for a stop: per province	14
2.4.4	Search for a stop: per line	15
2.4.5	Search for a stop: per town	16
2.4.6	Stop details	17
2.5	Ratings tab	18
2.5.1	Rating a vehicle	19
2.5.2	Rating a stop	20
2.5.3	Updating a review	20

3	Architecture	21
3.1	Users service	21
3.2	Vehicles service	21
3.3	Stops service	21
3.4	Ratings service	21
3.5	Interface service	21
3.6	Diagram	22
4	Tools used	23

1 Setting Up for the first time

1.1 Docker

In order for this interface to work, a working version of Docker needs to be installed and running on your machine.

For this, please follow the steps found at <https://docs.docker.com/get-started/>.

Linux users will need to run the `install.sh` script to ensure a correct working.

1.2 Creating the databases

Once Docker is up and running it is time to make everything ready for use. The first thing you need to do is execute the `run.sh` script on Mac/Linux while Windows users will need to execute the `run.bat` script.

When the services are up and running the user will need to open a second terminal and execute the `create_dbs.sh` script on Mac/Linux or the `create_dbs.bat` script on Windows. This will create the necessary tables and also fill the database with all stops available at that time in order to prevent long loading times when using the app.

2 Usage

After completing the set-up, the app can be launched by running `run` script for the corresponding OS and will be available at `http://localhost:5000/`. Following pages will describe how the different aspects can be used.

2.1 Home

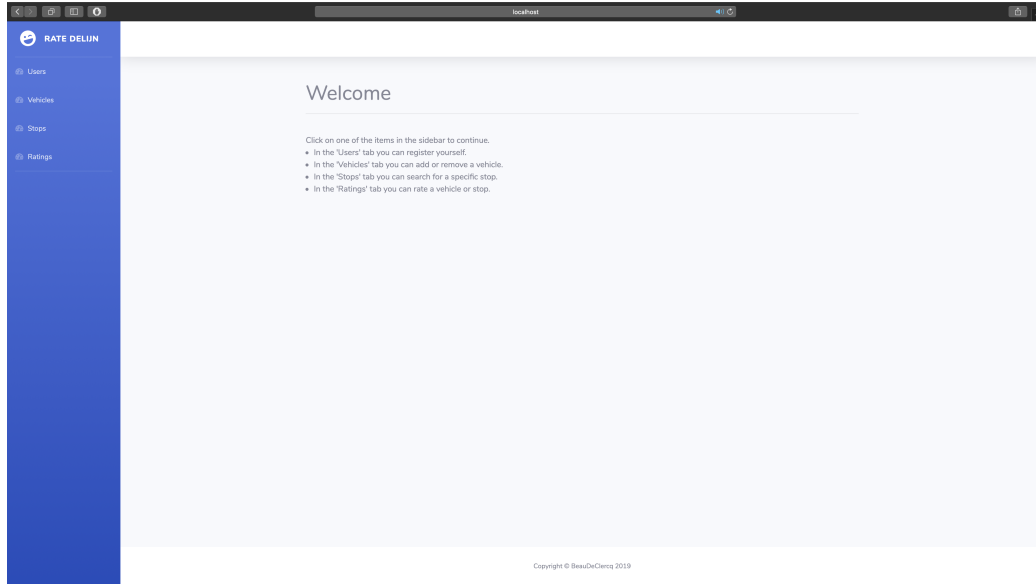
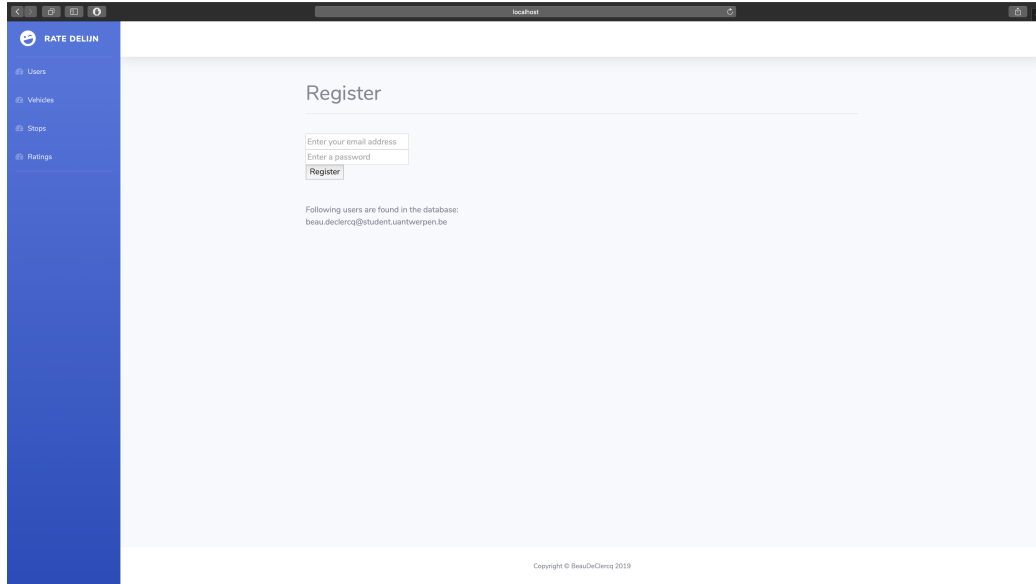


Figure 1: Home screen

The home page of the app provides a small description of what can be done in any of the individual tabs.

2.2 Users tab



The screenshot shows a web browser window with the URL 'localhost'. On the left is a blue sidebar with the 'RATE DELIJN' logo and a menu containing 'Users', 'Vehicles', 'Stops', and 'Ratings'. The 'Users' tab is selected. The main content area is titled 'Register' and contains a form with two input fields: 'Enter your email address' and 'Enter a password'. Below these fields is a 'Register' button. Underneath the button, a message states: 'Following users are found in the database: beau.declercq@student.uantwerpen.be'. At the bottom of the page, a small copyright notice reads 'Copyright © BeauDeclercq 2019'.

Figure 2: Users tab

In the **Users** tab, users can register themselves by filling in their email address and a password they want to use throughout the system.

After submitting the form, the user will be taken back to the home page where a message will be shown telling whether the registration has succeeded or not.

Under the registration form the user can view a list of all registered email addresses in the system. This may prevent users from unintentionally registering themselves multiple times.

2.3 Vehicles tab

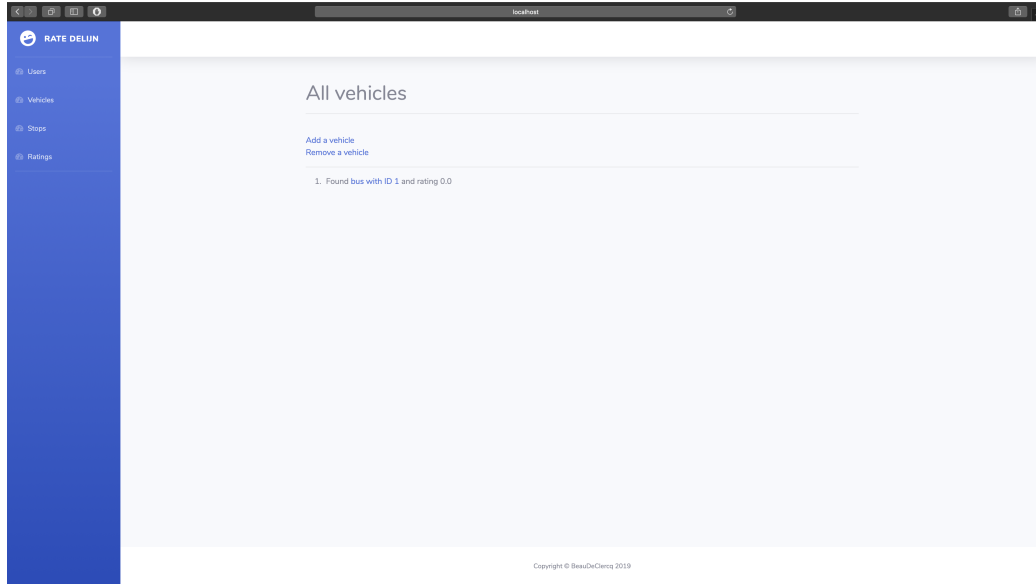
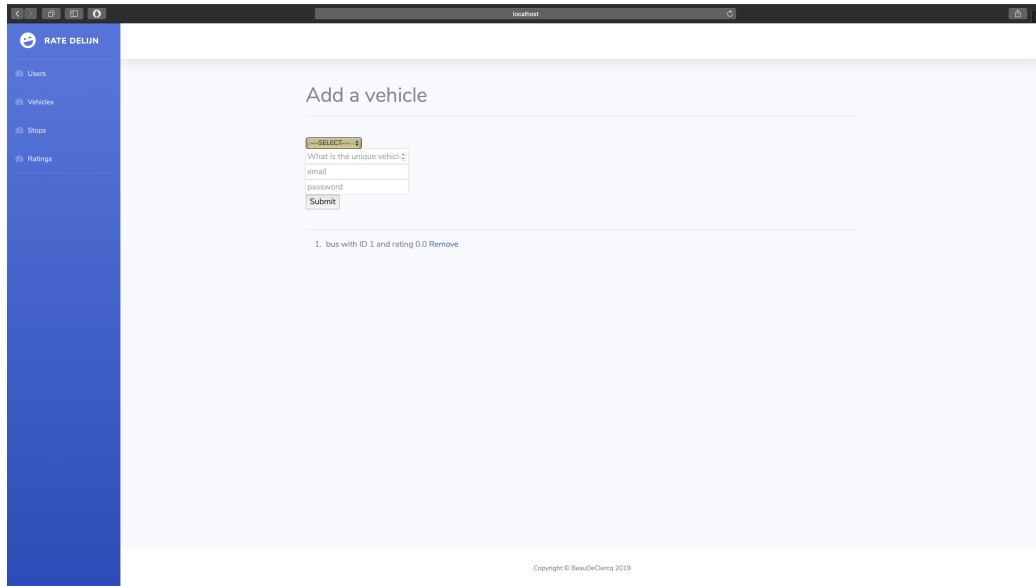


Figure 3: Vehicles tab

In the **Vehicles** tab, registered users can add a new vehicle or remove an existing vehicle.

In this tab, registered and unregistered users alike can also see a list of all vehicles currently registered. By clicking on any vehicle the user will be taken to a page that will provide more details.

2.3.1 Adding a vehicle



The screenshot shows a web browser window with the URL 'localhost'. On the left is a blue sidebar with the application logo 'RATE DELIN' and a menu with links to 'Users', 'Vehicles', 'Stops', and 'Ratings'. The main content area is titled 'Add a vehicle' and contains a form with the following elements:

- A dropdown menu with the text 'SELECT --' and a small arrow icon.
- A label 'What is the unique vehicle ID:' followed by a text input field containing the value 'small'.
- A label 'password' followed by a text input field.
- A 'Submit' button.

Below the form, there is a list of existing vehicles:

- 1. bus with ID 1 and rating 0.0 [Remove](#)

At the bottom of the page, there is a small copyright notice: 'Copyright © BeauDeClerq 2019'.

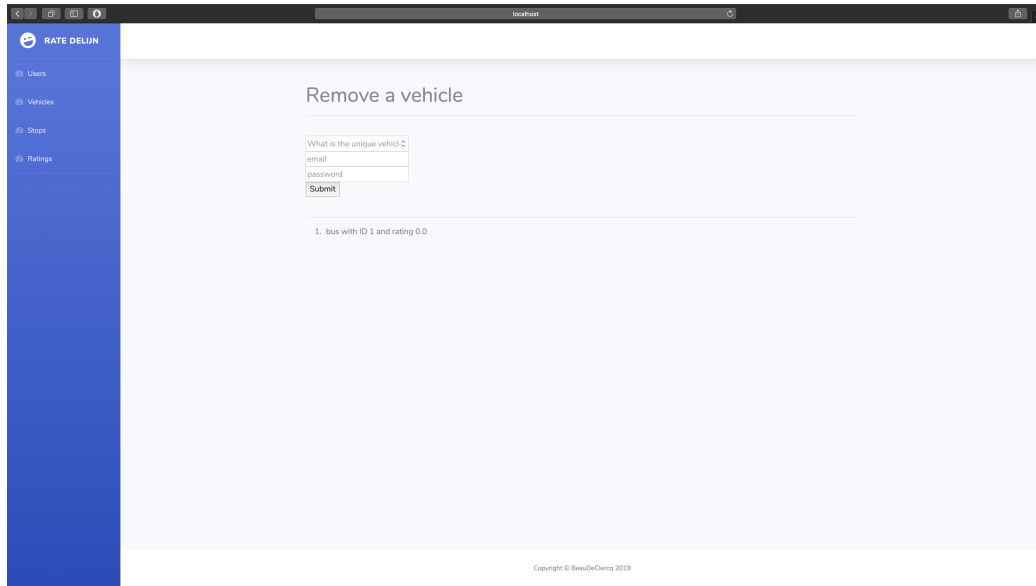
Figure 4: Adding a vehicle

A user can add a vehicle to the database by filling in the required fields and authenticating. Once the form is submitted, the user will be redirected to the home page where a message will be shown telling whether the operation was successful or not.

Underneath the form, users can see a list of all vehicles currently registered in the system to prevent them from entering a vehicle with the same ID multiple times.

If a user thinks that a vehicle has been added with a wrong ID, that vehicle can be removed by clicking the 'Remove' button and following the removal procedure.

2.3.2 Removing a vehicle



The screenshot shows a web browser window with the URL 'localhost'. The application is 'RATE DELIN'. On the left is a blue sidebar with a menu containing 'Users', 'Vehicles', 'Stops', and 'Ratings'. The main content area is titled 'Remove a vehicle'. It contains a form with the label 'What is the unique vehicle?' and two input fields: 'email' and 'password'. Below these fields is a 'Submit' button. Underneath the form, there is a list of vehicles, with the first item being '1. bus with ID 1 and rating 0.0'. At the bottom of the page, there is a small copyright notice: 'Copyright © BeauDeClerq 2019'.

Figure 5: Removing a vehicle

A user can remove a vehicle from the database by filling in the required fields and authenticating. Once the form is submitted, the user will be redirected to the home page where a message will be shown telling whether the operation was successful or not.

2.3.3 Vehicle details

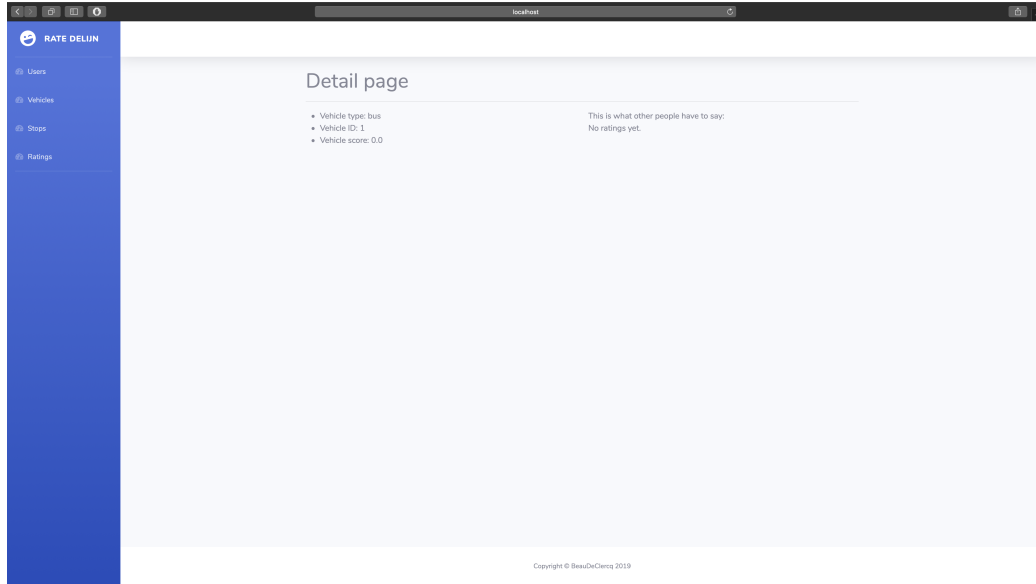


Figure 6: Details a vehicle

On the details page, users can see the details about the vehicle and read all ratings that are currently submitted.

2.4 Stops tab

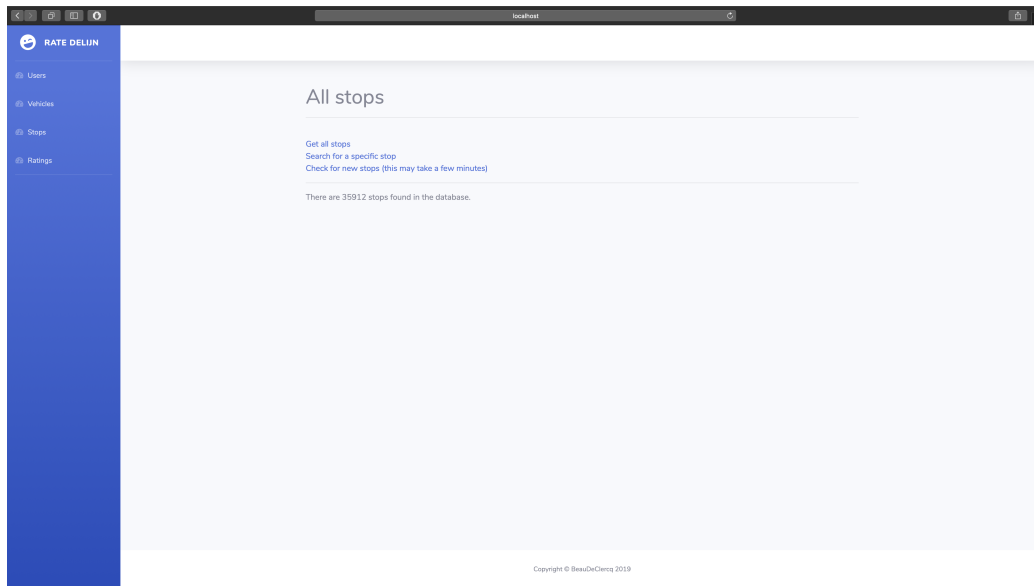


Figure 7: Stops tab

In the **Stops** tab, users can view all stops, search for a specific stop or update the database if they think new stops have been added by DeLijn.

2.4.1 View all stops

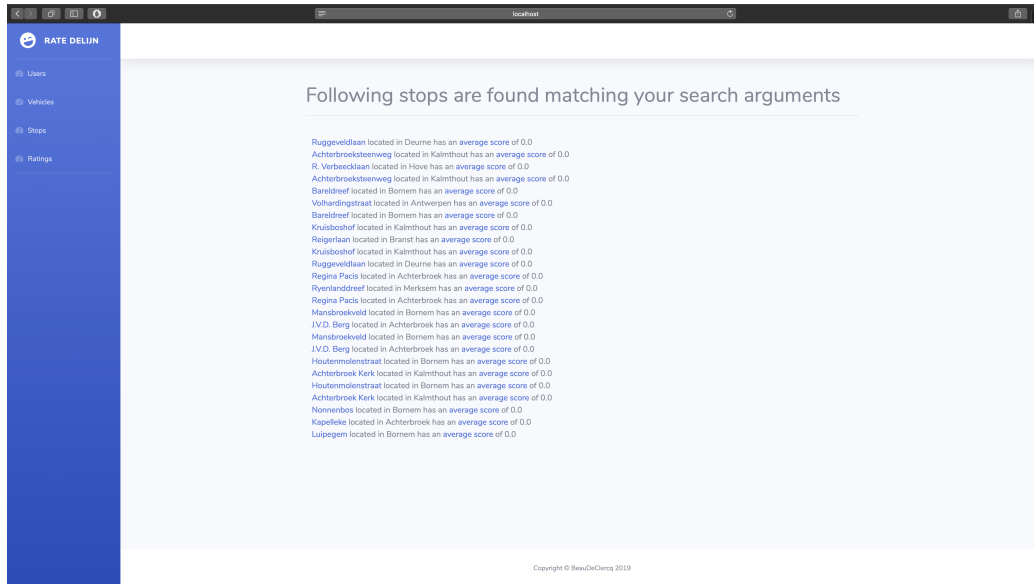


Figure 8: List of all stops

On this page, the user will get an unsorted list of all stops currently found in the database.

Clicking on the name of the stop will take the user to the rating form for the stop, clicking on the average score will take the user to the detail page for that stop.

Using this list is not advised as it may take a minute to load all records.

2.4.2 Search for a stop

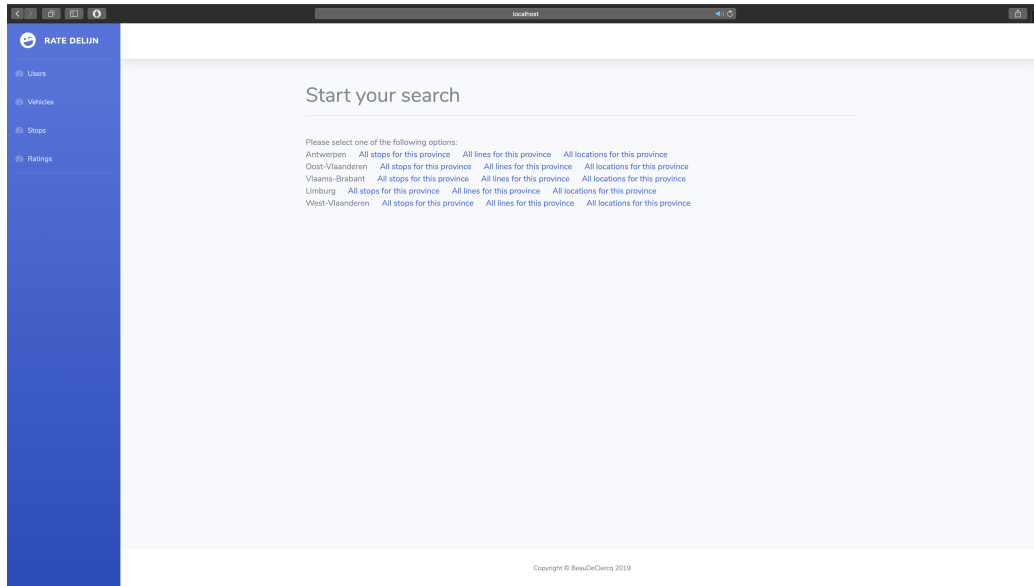


Figure 9: Start searching for a stop

There are a few options for a user who want to search for a specific stop:

- Per province: if the user wants a list of all stops for a province, this is the desired option.
- Per line: if the user wants a list of all stops for a certain line, this is the desired option.
- Per town: if the user wants a list of all stops for a certain location, this is the desired option.

2.4.3 Search for a stop: per province

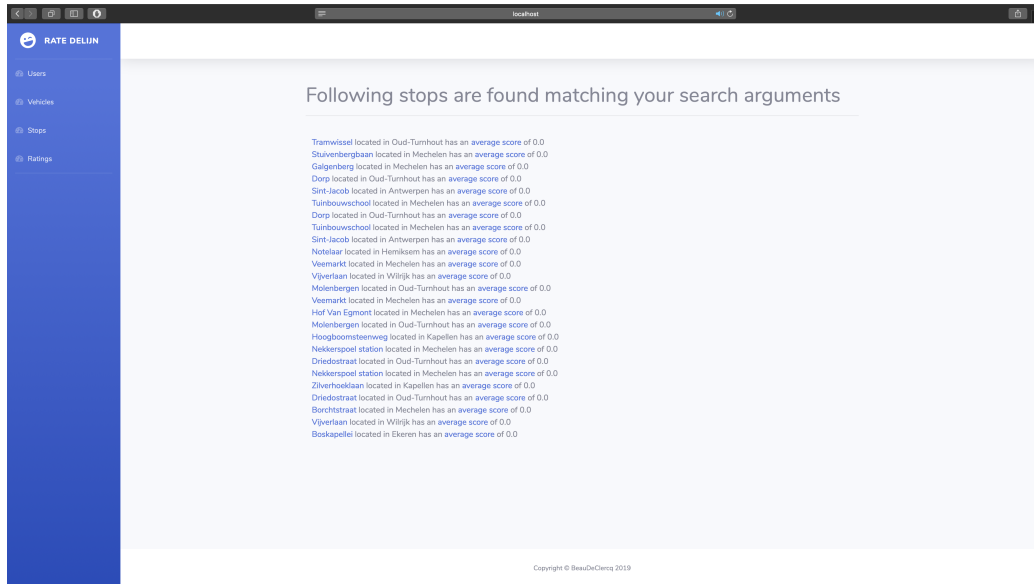


Figure 10: Stops per province

If the option ‘All stops for this province’ was selected, the user will be shown an unordered list of all stops for the chosen province.

Clicking on the name of the stop will take the user to the rating form for the stop, clicking on the average score will take the user to the detail page for that stop.

2.4.4 Search for a stop: per line

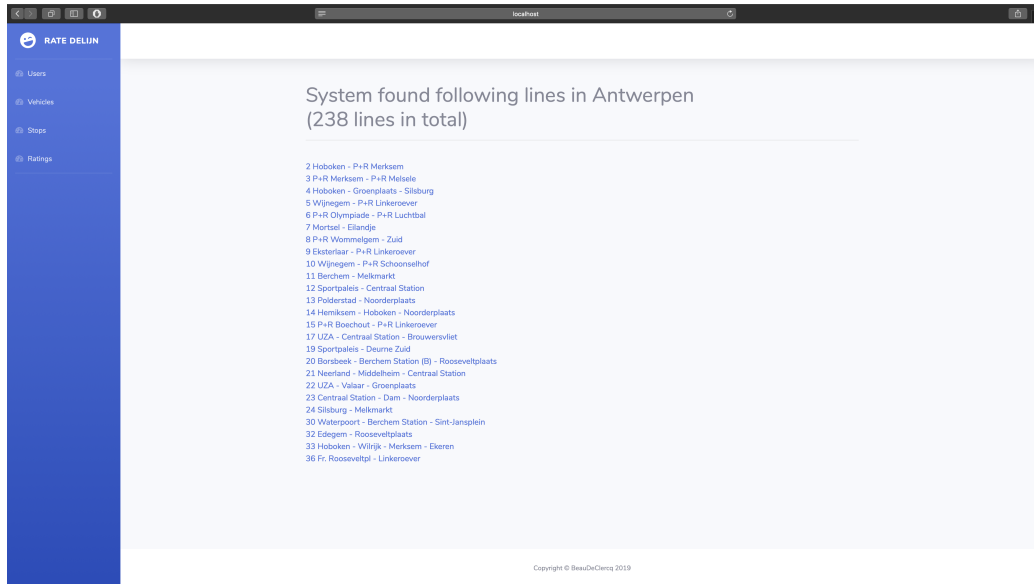


Figure 11: Lines per province

If the option ‘All lines for this province’ was selected, the user will be shown a list of all lines for the chosen province.

Selecting one of these lines will give the user a list of all available stops for that line. Clicking on the name of the stop will take the user to the rating form for the stop, clicking on the average score will take the user to the detail page for that stop.

2.4.5 Search for a stop: per town

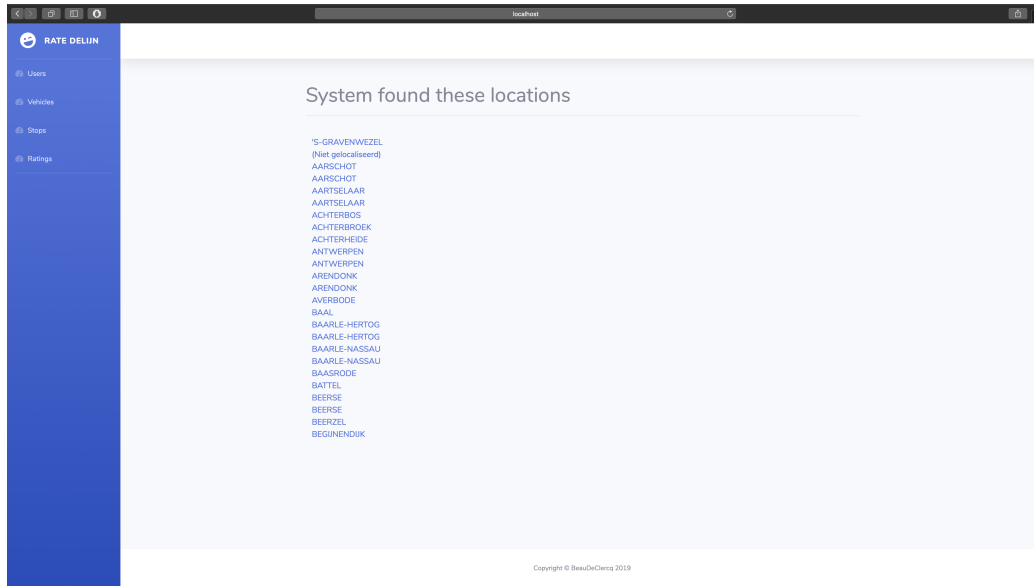


Figure 12: Locations per province

If the option ‘All locations for this province’ was selected, the user will be shown a list of all towns for the chosen province.

Selecting one of these locations will give the user a list of all available stops in that location. Clicking on the name of the stop will take the user to the rating form for the stop, clicking on the average score will take the user to the detail page for that stop.

2.4.6 Stop details

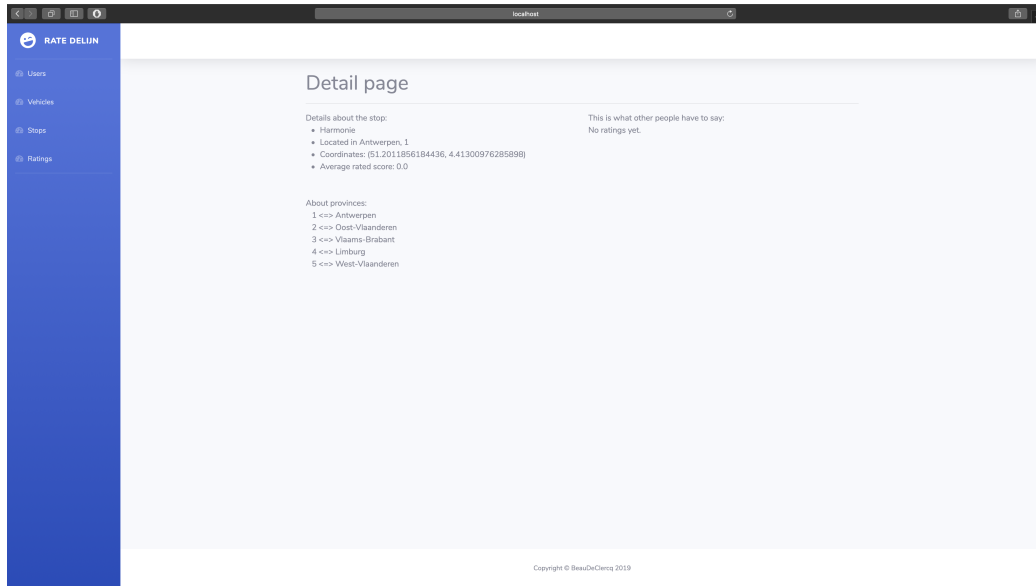


Figure 13: Details for a stop

On the details page, a user can see all details about a stop and read all ratings it has received.

2.5 Ratings tab

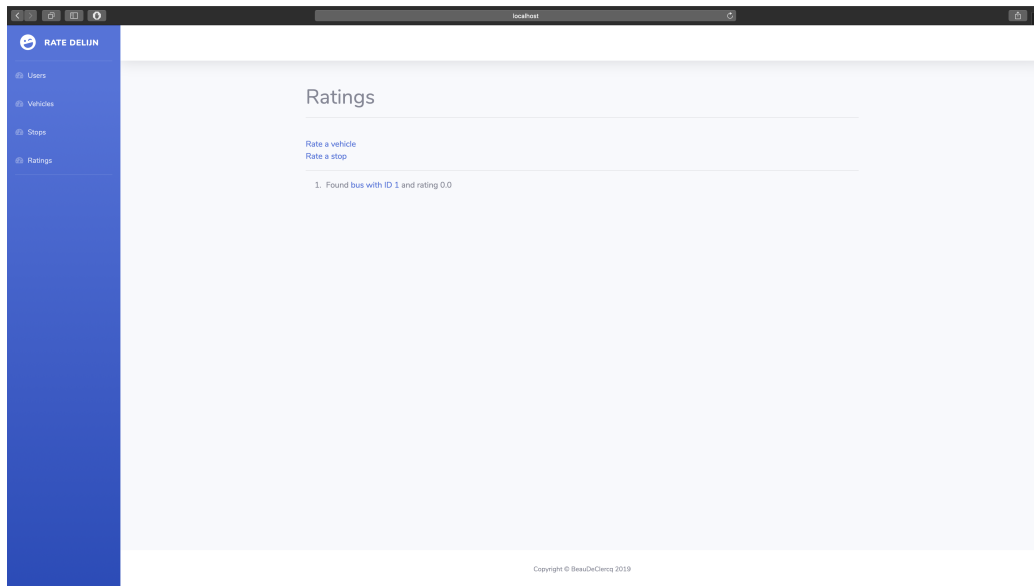
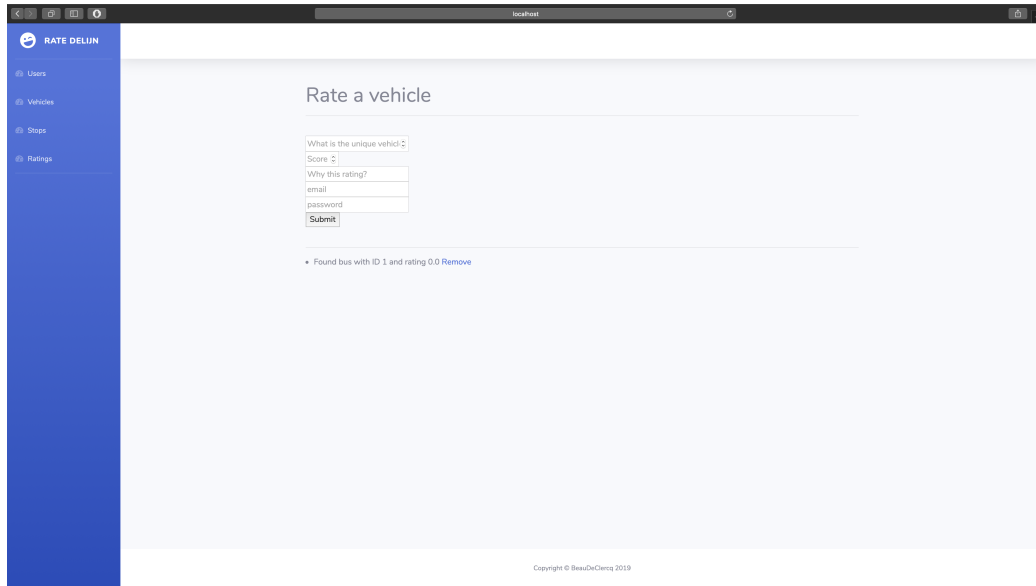


Figure 14: Ratings tab

In the **Ratings** tab, users can choose to rate a vehicle or stop. Clicking on a vehicle from the list will take the user to the details page for that vehicle.

2.5.1 Rating a vehicle



The screenshot shows a web browser window with the address bar set to 'localhost'. The application is titled 'RATE DELIN' in the top left corner. A blue sidebar on the left contains navigation links: 'Users', 'Vehicles', 'Stops', and 'Ratings'. The main content area is titled 'Rate a vehicle' and contains a form with the following fields: 'What is the unique vehicle?' (a dropdown menu), 'Score' (a numeric input field), 'Why this rating?' (a text area), 'email' (an email input field), and 'password' (a password input field). A 'Submit' button is located below the password field. Below the form, there is a message: 'Found bus with ID 1 and rating 0.0 Remove'. At the bottom of the page, there is a small copyright notice: 'Copyright © BeauDeClerq 2019'.

Figure 15: Rating a vehicle

A vehicle can be rated by filling in the form and authenticating. Once the form is submitted, the user will be redirected to the home page where a message will be shown telling whether the operation was successful or not.

2.5.2 Rating a stop

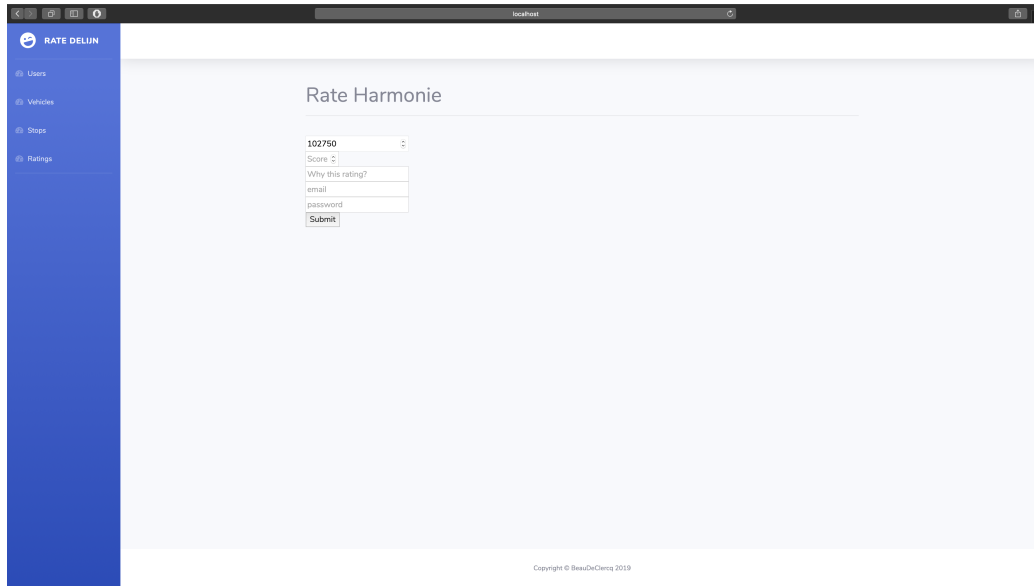
A screenshot of a web browser showing a form titled "Rate Harmonie". The browser's address bar shows "localhost". On the left, there is a blue sidebar with the text "RATE DELIN" and a list of menu items: "Users", "Vehicles", "Stops", and "Ratings". The "Ratings" item is selected. The main content area has a light blue background. At the top of this area is the title "Rate Harmonie". Below the title is a form with the following fields: a dropdown menu showing "102750", a "Score" field with a range indicator, a "Why this rating?" text area, an "email" field, a "password" field, and a "Submit" button. At the bottom of the page, there is a small copyright notice: "Copyright © BeauDeClerq 2019".

Figure 16: Rating a stop

Selecting the ‘Rate a stop’ option will take the user to the search page for a stop. When the desired stop is found, the user simply needs to click on the name of the stop to be redirected to the rating form.

Once the form is submitted, the user will be redirected to the home page where a message will be shown telling whether the operation was successful or not.

2.5.3 Updating a review

If a user would like to update their review, all they need to do is fill out the form again and submit it.

3 Architecture

The application consists of 5 services: **Users**, **Vehicles**, **Stops**, **Ratings** and **Interface**.

The **Users**, **Vehicles**, **Stops** and **Ratings** service each have their own database.

3.1 Users service

The **Users** service handles all things related to registering and authenticating users. This service makes no calls to any other services.

Email-password combinations are stored in the database belonging to this service.

3.2 Vehicles service

The **Vehicles** service handles all things related to adding and removing a vehicle, updating the average score and retrieving a certain vehicle. This service makes calls to the **Users** and **Ratings** services.

All vehicles are stored in the database belonging to this service.

3.3 Stops service

The **Stops** service handles the searching for a stop and checking for new stops. This service only makes calls to the API of DeLijn and to no other services.

All stops are stored in the database belonging to this service.

3.4 Ratings service

The **Ratings** service handles the rating of vehicles and stops. This service makes calls to the **Vehicles**, **Users** and **Stops** services.

All ratings are stored in the database belonging to this service.

3.5 Interface service

The **Interface** only provides the web pages. This service has no underlying database and makes calls to all other services.

3.6 Diagram

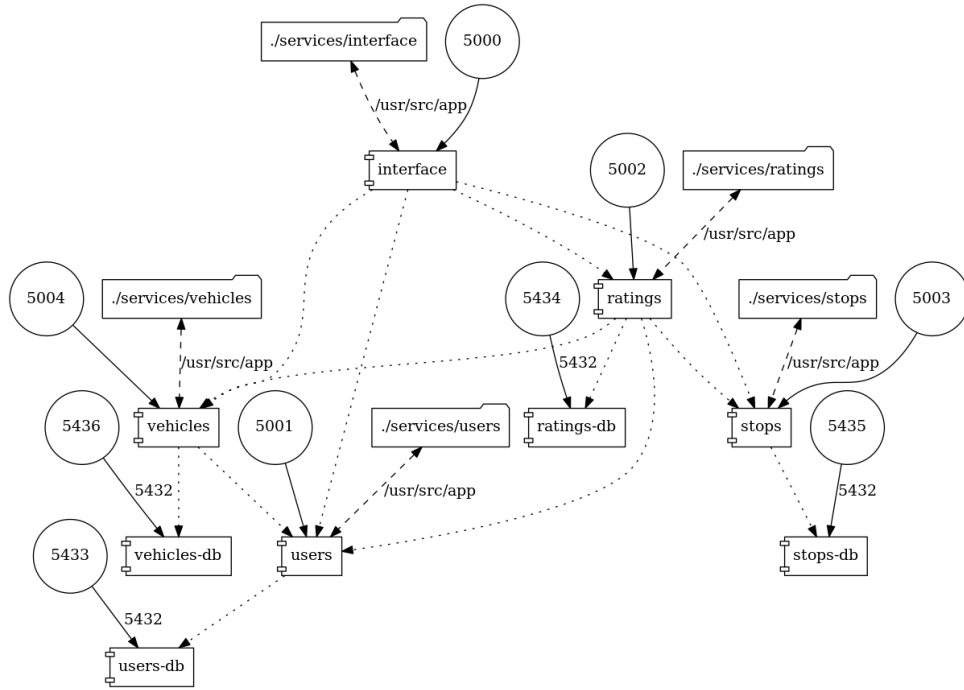


Figure 17: Architecture

This diagram was automatically generated based on the **docker-compose.yml** file by running following command in the root directory:
`docker run --rm -it --name dcv -v $(pwd):/input pmsipilot/docker-compose-viz render .`

4 Tools and choices

4.1 Tools used

Following tools were used in creating this app:

- Basics:
 - Flask: framework of the app.
 - Flask-SQLAlchemy: together with postgresql, this provides the database system of the app.
 - psycopg2: Python-PostgreSQL Database Adapter.
- Others:
 - requests module: makes it easier to perform http calls.

4.2 Choices made

For this project I chose to split the system up in 5 different services because, in my opinion, this best resembled the philosophy of micro-services. Each service can be substituted by another one that provides equal responses to API calls.

Since each service has it's own database, no problems regarding data will arise when choosing to swap services.