Programming Project Databases Intermediate Evaluation Team 6

Datum 8 maart 2020

Onderwerpen

1 Status

1.1 Mockups

There have been made different HTML mockups for most of the pages of our site. Most of them aren't connected yet to the system, but e.g. the login and register page are.

1.1.1 Arno

Taak (DONE): Login page
Taak (DONE): Home page (Logged in)
Taak (DONE): Error 404: Page not found
Taak (DONE): Error 405: Method not allowed
Taak (DONE): W.I.P. page

1.1.2 Sam

Taak (DONE): Home page
Taak (DONE): Settings page

1.1.3 Sien

Taak (DONE): About page
Taak (DONE): Account page

1.1.4 Tim

Taak (DONE): Add Route page
Taak (DONE): A basic slippy map

1.2 Varia

1.2.1 Arno

Taak (DONE): Basic flask tests (incl. various extensions)
Taak (DONE): Register (using the API and using HTML)
Taak (DONE): Deploying our own webapp
Taak (DONE): Bulma Framework tests

1.2.2 Sam

Taak (DONE): Google Cloud Server setup (incl. adding all ssh keys and deploying the template webapp)
Taak (DONE): Material Framework tests

1.2.3 Sien

Taak (DONE): psycopg2 Database tests in Python

1.2.4 Tim

Taak (DONE): ER-Diagram (first version)
Taak (DONE): Bootstrap Framework tests

1.3 Map

At the current moment we have a very basic slippy map used on our website. it uses leaflet for displaying the map on the website, it is a mobile friendly light Javascript based framework to display maps by combing the map data and the map tiles, OpenStreetmaps to get the data for the maps and Mapbox as map tiles so the map has actual graphics. As far as my current understanding of how everything works we can keep using the same rough design for future use, but if this is not the case we can easily swap it out for something else that will suit our needs.

2 Technologies

2.1 Client (Web Design)

We'll use Javascript, CSS and HTML. To make our lives easier, we'll use Bulma as CSS framework.

2.2 Web server

We'll use Flask with a lot of it's extensions, like wtforms for handling form inputs, werkzeug.security for password hashin, FlaskLogin as LoginManager ...

2.3 Data service

We'll use SQLAlchemy and Psycopg2 for handling requests to the database and simplifying the database migrations.

2.4 Relational Database

As described in the assignment, we'll use PostgreSQL.

2.5 Hosting

The site will be hosted with Google Cloud Platform.

2.6 Database Design

We'll use DBdiagram for visualizing our database design.

2.7 Map

The maps on our site will work with Leaflet. We'll also try to experiment a bit with the OpenStreetMap API.

2.8 Version Control System

GitHub is obviously our prefered VCS.

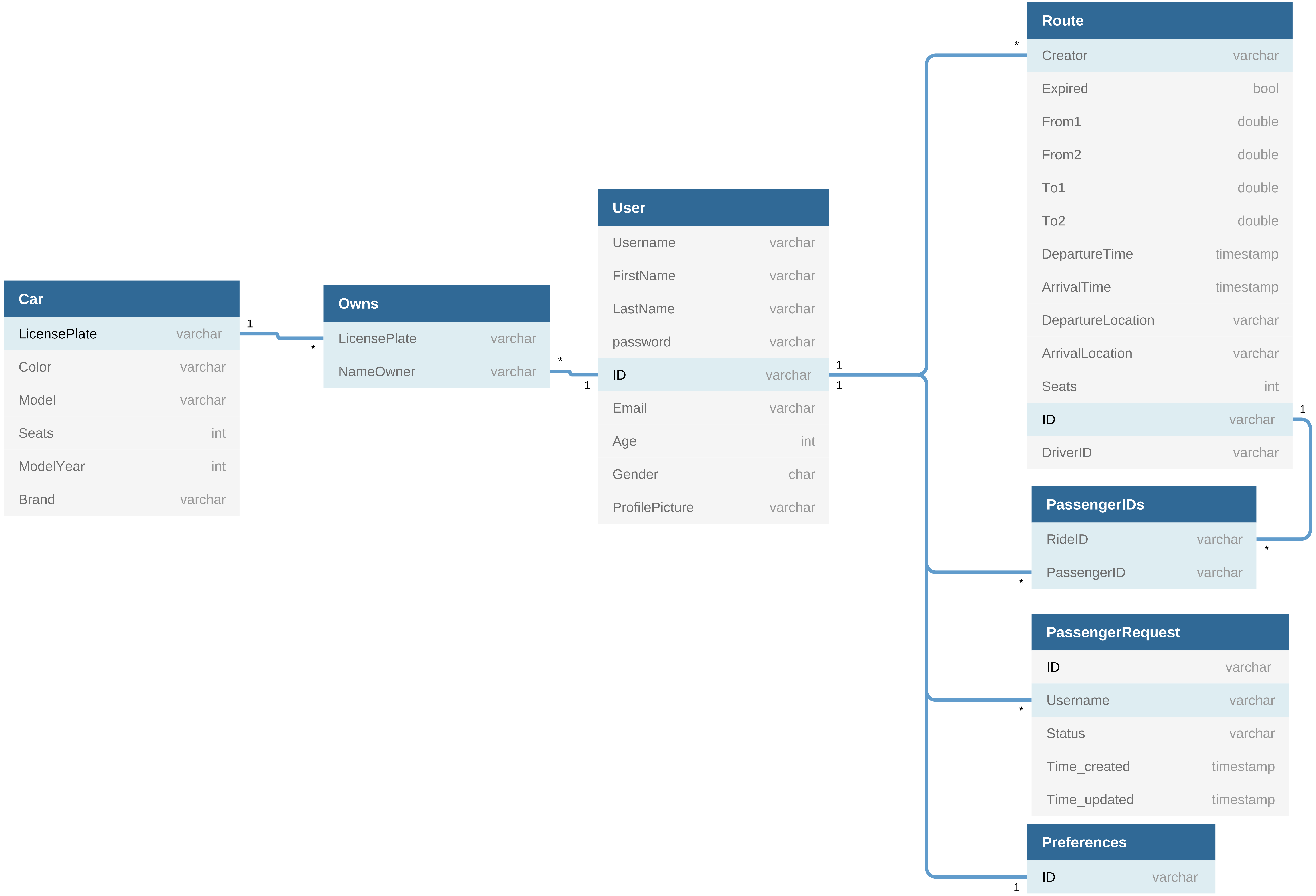
2.9 Planning

We tried out toggl, but realised it was a lot easier to write the planning in our weekly report and use that as reference.

2.10 API

We'll use PostMan to send the specified POST Json request from the ppdb apiary to our site.

Klik hier om mij op te sturen.



Car	
LicensePlate	varchar
Color	varchar
Model	varchar
Seats	int
ModelYear	int
Brand	varchar

Owns	
LicensePlate	varchar
NameOwner	varchar

User	
Username	varchar
FirstName	varchar
LastName	varchar
password	varchar
ID	varchar
Email	varchar
Age	int
Gender	char
ProfilePicture	varchar

Route	
Creator	varchar
Expired	bool
From1	double
From2	double
To1	double
To2	double
DepartureTime	timestamp
ArrivalTime	timestamp
DepartureLocation	varchar
ArrivalLocation	varchar
Seats	int
ID	varchar
DriverID	varchar

PassengerIDs	
RideID	varchar
PassengerID	varchar

PassengerRequest	
ID	varchar
Username	varchar
Status	varchar
Time_created	timestamp
Time_updated	timestamp

Preferences	
ID	varchar

dbdiagram.io