|  |
| --- |
| Fontys Hogescholen |
| Software Design |
| Swagon – car rental service |

|  |
| --- |
| Viktor Naydenov  Supervisor:[Your tutor/supervisor’s name]  Eindhoven, 06.10.2020 |

Table of Contents

[1 Introduction 3](#_Toc31980902)

[1.1 Document Purpose 3](#_Toc31980903)

[1.2 Document Overview 3](#_Toc31980904)

[2 System Overview 4](#_Toc31980905)

[3 System Architecture 5](#_Toc31980906)

[3.1 Architectural Design 5](#_Toc31980907)

[3.2 Design Rationale 5](#_Toc31980908)

[4 Data Design 6](#_Toc31980909)

[5 GUI 7](#_Toc31980910)

[5.1 General Overview 7](#_Toc31980911)

[5.2 User Interface Action 7](#_Toc31980912)

6 Front-end framework 8

6.1 Selection of criteria 9

7. API Endopoints……………………………………………………………………………………………………………………………10

8. ORM Tool…………………………………………………………………………………………………………………………………..11

9. Software Quality Metrics tool overview………………………………………………………………………………………12

# Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information.

## Document Purpose

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to built. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

## Document Overview

## Section 1 – Contains information about the software design document

## Section 2 – Gives overview of what the software product is going to look like

## Section 3 – Contains UML diagram about the architecture of the product

## Section 4 – Contains information about the data of the Software product and information about the database.

## Section 5 – Contains information about how the product will look like. It also includes some graphical illustrations.

Section 6 – Is about the front end choice.

Section 7 – API Endpoints

Section 8 - SonarQube

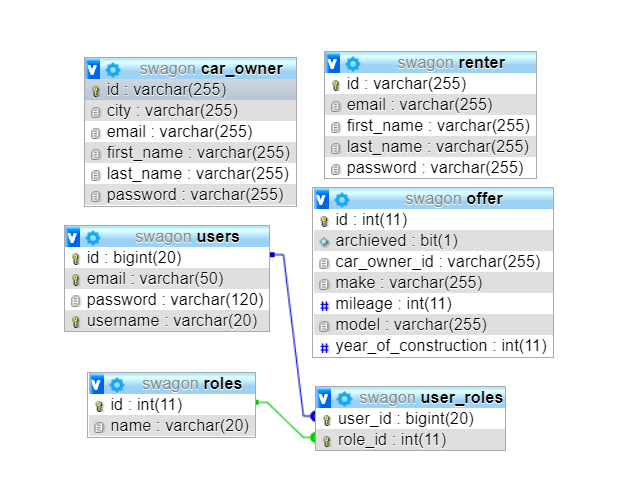
# System Overview

Swagon Is web based service for car rentals. Something remarkable is that everybody can use it. You cant give your personal car for short period to somebody in exchange for money. There is user registration available. In your profile section there are records of all your car rentals. You can browse through the different car offers, sort them by price, location etc. You can also filter the results by make, model, mileage and other parameters that will help you to find the best car for your needs. Another feature is rental experience review. You can live feedback with information how satisfied you were with the car rental.

## 

# Data Design

This is my screenshot of my current database tables. My project is in early phase of developing so in the process of working on the project the database tables and its properties will increase rapidly. For the beginning I am storing only the general data needed for the API to work.

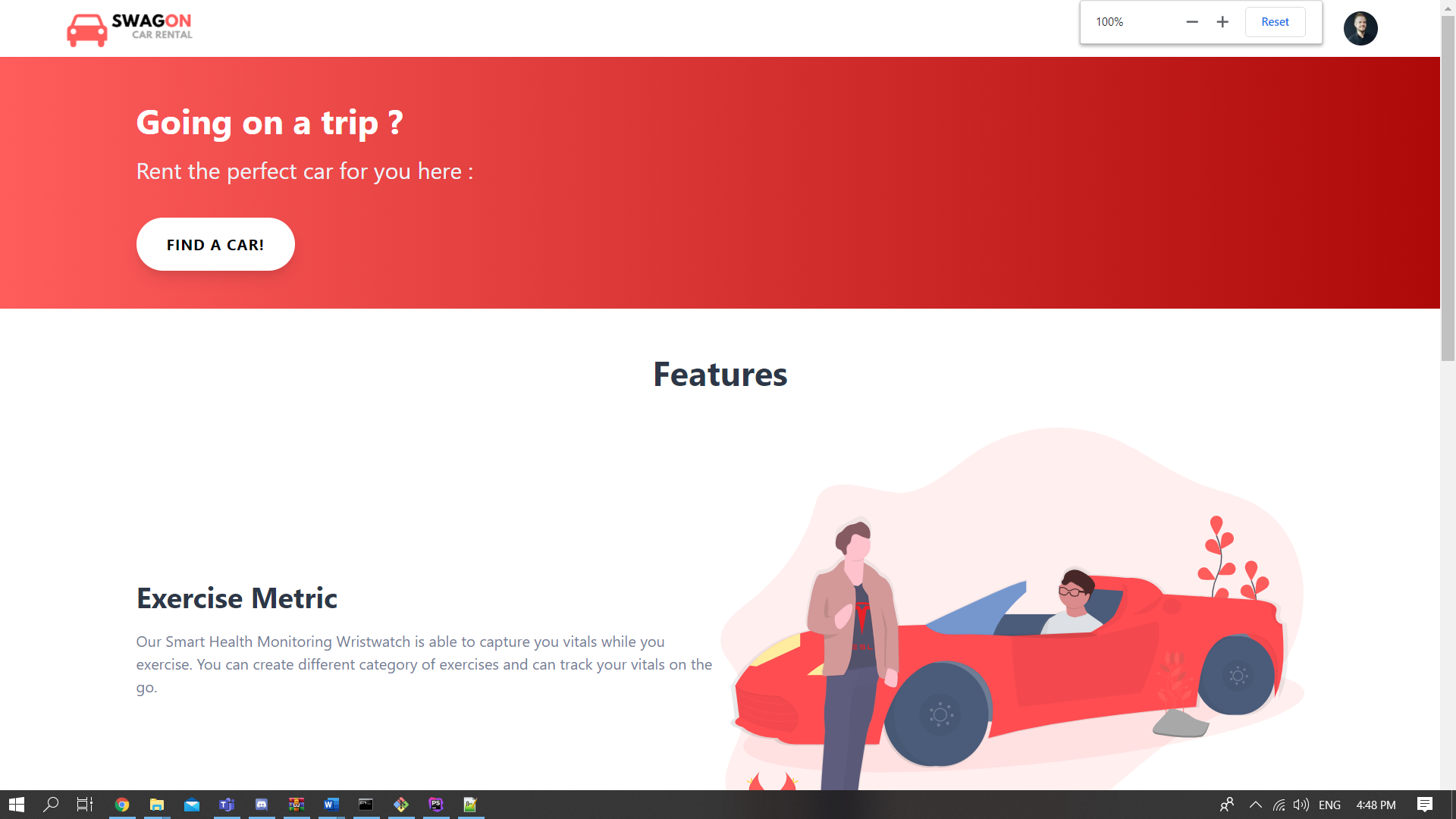


# GUI

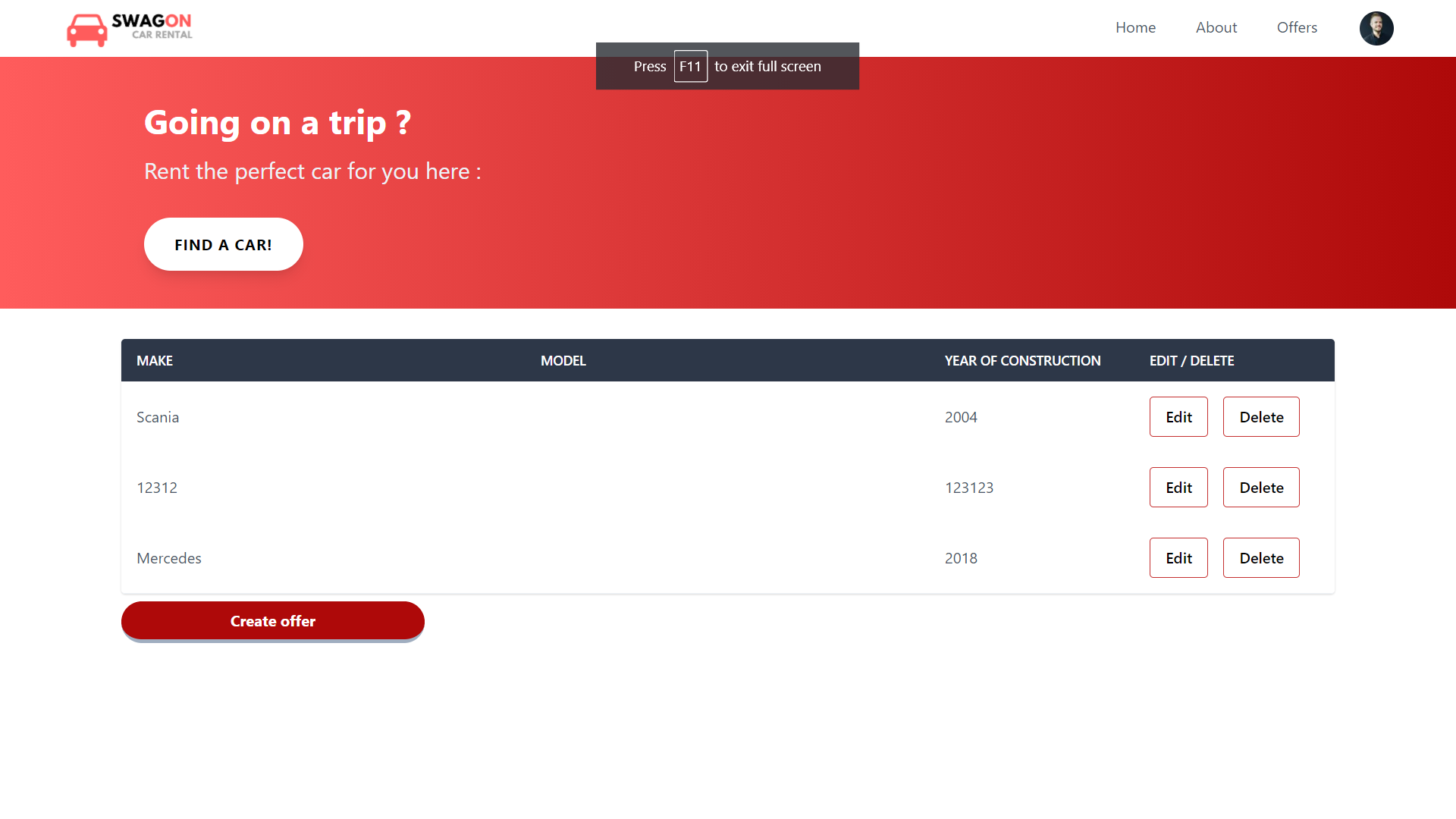
## General overview

Swagon is web based software application which is used to give your car for rent or rent somebody’s personal car. As a user who want to rent a car for a short period I can register and log in into Swagon. Then I can search for a car through different parameters or just browse through the pages with cars for rent. When I choose the best car for me I can message the owner and then book it. As a user who want to give my car for rent I can register and the login into the site and I can create new car rental offer with the details of my car and the car pick up location.

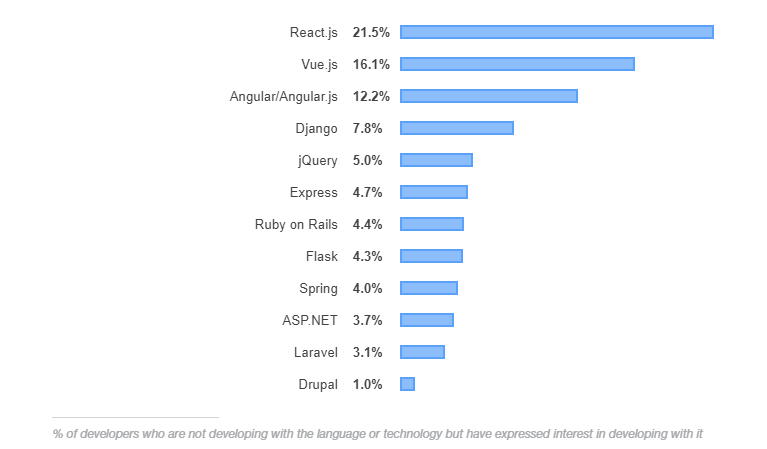
## User interface action







# Selection of criteria

According to survey conducted in In 2019 developers community picked up the top front-end frameworks. Considering the constantly growing interest to React, Vue.js, and Angular, the trends will keep the same for the next few years. 

That’s why I chose to compare the top three front end technologies that are most used nowadays – React, Vue, and Angular.

React :

React is the front-end framework created and developed by Facebook. The team was interested in obtaining high performance by building an effective UI. Jordan Walke proposed to create the library based on XHP and JavaScript combination in 2011. As a result, they received the library to build the web interface with JavaScript where the news update happens at the same time while customers use the chat.

Pros :

* Virtual DOM improves both the experience of the user and the work of the developer – Virtual DOM helps to update any user’s changes without the other parts’ interference by applying isolated components. It greatly helps to smooth the experience of all participants in real time mode.
* Virtual DOM improves both the experience of the user and the work of the developer – Virtual DOM helps to update any user’s changes without the other parts’ interference by applying isolated components. It greatly helps to smooth the experience of all participants in real time mode.

Cons :

* Relatively long learning curve – It is unlikely to learn it in one day. It just requires a little bit more time to be aware of all the details than for example with Angular.
* Lack of documentation due to high pace development – The popularity of this web front-end framework is extremely higher than other ones. It has so many updates and innovations, that sometimes it is hard to find all detailed information.

Vue.js :

Vue.js is the web framework for building user interfaces. It is an independent tool that creates web interfaces and doesn’t require the additional extension. Vue.js was created by Evan You and initially released in February 2014. Probably you might not find the long list of big companies using Vue.js but it is definitely popular among developers and the framework popularity considerably increases.

Pros :

* Clarity and simplicity – This front-end framework has the smallest API surface area. It is quite easy to start not only learning but using it in a short time.
* Detailed and extensive documentation – You can find all the necessary documentation which is up-to-date and well-written. Also, there is always a community of developers who can share their knowledge with you.

Cons :

* Too much flexibility leads to the code irregularities – Some experts have brought up the point that a lot of flexibility is not too good. All web front-end engineers can contribute to its development, so it might cause certain irregularities
* Smaller developers’ community – It is a relatively new and constantly improving framework. Vue.js is mostly supported by the individual developers but their user community fastly increases.

Angular :

Angular or Angular v2 or Angular 2+ is a complete rewrite of AngularJS by the same team in Google who developed the original front-end framework in 2010. It was released in September of 2016 as a completely new framework with new logic and features to address the challenges of web development as we know it today.

Pros :

* Component-based architecture of Angular allows creating the UI with single parts (components) and reuse these components in the app. The elements also simplify user testing and maintenance.
* TypeScript is the core language for Angular and it does compile to JavaScript, making the coding process easier for many engineers due to its improved navigation, and refactoring services.

Cons :

* Angular Complexity – despite the web-based structure, it is quite difficult to manage the components: e.g the devs need to have several files for one Angular component as well as maintain the elements’ lifecycle interfaces.
* Transferring legacy systems from AngularJS to Angular – the difference between AngularJS and Angular is huge, thus it takes a lot of time and effort to migrate the legacy systems.

Credits: stackoverflow.com

Reference: <https://existek.com/blog/top-front-end-frameworks-2020/>

# Api Endpoints

|  |  |  |
| --- | --- | --- |
| **HTTP METHOD** | **URI path** | **Description** |
| GET | Api/Users | Returns all users |
| GET | Api/users/login | User log in |
| GET | Api/users/logout | User log out |
| GET | Api/users/:id | Return user with specified id |
| PUT | Api/users/:id | Modifies a user |
| POST | Api/user | Creates a user |
| DELETE | Api/user/:id | Deletes a user |

|  |  |  |
| --- | --- | --- |
| **HTTP METHOD** | **URI path** | **Description** |
| GET | Api/offers | Returns all offers |
| GET | Api/offers/:id | Return offer with specified id |
| PUT | Api/offers/:id | Modifies an offer |
| POST | Api/offers | Creates an offer |
| DELETE | Api/offers/:id | Deletes an offer |

# ORM Tool Choice

For the Swagon project I am using Hibernate framework. The object-relational mapping frameworks are using technique for converting data between two systems using objected-oriented programming languages. I find working with Hibernate intuitive and easy. Using an XML file, Hibernate can generate skeleton source code for the persistence classes. This is auxiliary when annotations are used. Hibernate can use the XML file or the Java annotations to maintain the database schema. Another good feature of Hibernate is that the framework supports the mapping of custom value types so its possible to override the default SQL type and mapping a single property to multiple columns. Hibernate can significantly reduce development time spent with manual data handling in SQL and JDBC.

|  |  |
| --- | --- |
| **Hibernate ORM** | |
|  | |
| **Developer(s)** | [Red Hat](https://en.wikipedia.org/wiki/Red_Hat) |
| **Initial release** | 23 May 2001; 19 years ago |
| **Stable release** | 5.4.22.Final / September 30, 2020; 37 days ago[[1]](https://en.wikipedia.org/wiki/Hibernate_(framework)#cite_note-1) |
| **Written in** | [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) |
| **Operating system** | [Cross-platform](https://en.wikipedia.org/wiki/Cross-platform) ([JVM](https://en.wikipedia.org/wiki/JVM)) |
| **Platform** | [Java Virtual Machine](https://en.wikipedia.org/wiki/Java_Virtual_Machine) |
| **Type** | [Object-relational mapping](https://en.wikipedia.org/wiki/Object-relational_mapping) |
|  |  |

Reference : <https://en.wikipedia.org/wiki/Hibernate_(framework)>

# Software Quality Metrics tool overview

