

СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА



Software Requirements Specification

„AutoDeli“

Prepared by:

Владимир Карачомаков, 62395

18.02.2021г

Функционални изисквания

1. REST API трябва да предоставя следните възможности на следните групи потребители:

- 1.1. *Админ трябва да може да:*

- 1.1.1. Да се логва в акаунта си
 - 1.1.2. Да излиза от акаунта си
 - 1.1.3. Редактира акунти, както своя, така и на останалите user-и.
 - 1.1.4. Преглежда всички акаунти
 - 1.1.5. Да добавя нови акаунти
 - 1.1.6. Да добавя нови марки
 - 1.1.7. Да добавя нови модели за марка
 - 1.1.8. Да добавя нови двигатели за марка-модел
 - 1.1.9. Да добавя нови авточасти за марка-модел-двигател
 - 1.1.10. Да добавя нови масла
 - 1.1.11. Да добавя нови акумулатори
 - 1.1.12. Да добавя нови добавки
 - 1.1.13. Да изтрива по-горе посочените

- 1.1. *Регистриран потребител трябва да може да:*

- 1.1.1. Да се логва в акаунта си
 - 1.1.2. Да излиза от акаунта си
 - 1.1.3. Да търси авточасти
 - 1.1.4. Да търси масла
 - 1.1.5. Да търси акумулатори
 - 1.1.6. Да търси добавки
 - 1.1.7. Да добавя по-горе посочените в количка
 - 1.1.8. Да прави поръчка

- 1.2. *Анонимен трябва да може да:*

- 1.2.1. Да се регистрира
 - 1.2.2. Да търси авточасти
 - 1.2.3. Да търси масла
 - 1.2.4. Да търси акумулатори
 - 1.2.5. Да търси добавки

2. WEB апликацията трябва да предоставя възможност за **лесна връзка** с *REST API* и да позволява на потребителя да достъпва посочените за всяка роля в точка 1) данни и функционалности, в зависимост от неговата роля

Нефункционални изисквания

1. **Сигурност (Security)** - Всеки потребител от горе-посочение роли трябва да има достъп само да своите данни и функционалности, без да може да достъпва функционалност и данни, без необходимата оторизация.
2. **Производителност** - Да се постигне възможно най - висока производителност
3. **Наличност** - Да е налична максимално време, с възможно най - кратки прекъсвания на услугата
4. **Модифицируемост** - Сорс кодът на проекта да е организиран така, че лесно да може да се променят, модули, които могат да търпят чести модификации
5. **Тестваемост** - Source кода да е организиран, така че лесно да се тестват отделните модули
6. **Използваемост** - Предоставяне на удобен *WEB* интерфейс за комуникация с REST API частта.

Използвани технологии и модули

1. Реализация на *REST API* със *Spring Boot*:
 - **Java 11**
<https://openjdk.java.net/projects/jdk/11/>
<https://docs.oracle.com/en/java/javase/11/docs/api/index.html>
 - **Spring Boot 2** - <https://spring.io/projects/spring-boot>
 - **Spring Security** - <https://docs.spring.io/spring-security/site/docs/current/reference/htmlsingle/>
 - **Spring MVC** - <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>
 - **Spring MongoDB** - драйвер за синхронна комуникация с база данни MongoDB
<https://docs.spring.io/spring-data/mongodb/docs/current/reference/html/#mongo.core>
 - **Maven** - <https://maven.apache.org/>
 - **Lombok** - <https://projectlombok.org/>

2. Реализация на WEB апликацията с *ReactJs*:

- *ReactJS* – <https://reactjs.org/>
- *Bootstrap* - <https://getbootstrap.com/>
- *Axios* - <https://www.axios.com/>

Инсталация и конфигуриране

Системата е проектирана и тествана на основа *Java* версия 11.

За да се build-не spring back-end апликацията е нужно:

1. Да се изпълни командата „mvn clean install”
2. Да се стартира сървър на проекта

За да се build-не react front-end апликацията е нужно:

3. Да се изпълни командата “npm install” в поддиректорията на проекта “client”
4. Да се изпълни командата “npm run build” в същата директория
5. Да се стартира сървър на проекта с командата “npm start” (стартира се на <http://localhost:3000/>)

Значими API Resources

GET */api/oil - Връща всички масла

GET */api/battery – Връща всички акумулатори

GET*/api/supplement – Връща всички добавки

GET */api/brand/{brandId}/model/{modelId}/engine/{engineId} – Връща всички авточасти за конкретен автомобил

POST */api/cart/autoPart – Добавя авточаст в количката

POST */api/cart/oil– Добавя масло в количката

POST */api/cart/battery – Добавя акумулатор в количката

POST */api/cart/supplement – Добавя добавка в количката

DELETE */api/cart/autoPart/{autoPartId} – Изтрива авточаст от количката

DELETE */api/cart/oil/{oilId} – Изтрива масло от количката

DELETE */api/cart/battery/{batteryId} – Изтрива акумулатор в количката

DELETE */api/cart/supplement/{supplementId} – Изтрива добавка от количката

POST */api/order – Създава нова поръчка и изпраща мейл до админа с нужната информация