

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



# Индивидуален проект

Разработка на уеб приложения с JAVA,  
летен семестър, 2022/2023

Система за предоставяне на  
почистване

Изготвил:  
Волен Чавдаров Димитров,  
71994, ИС, IV курс

## СЪДЪРЖАНИЕ:

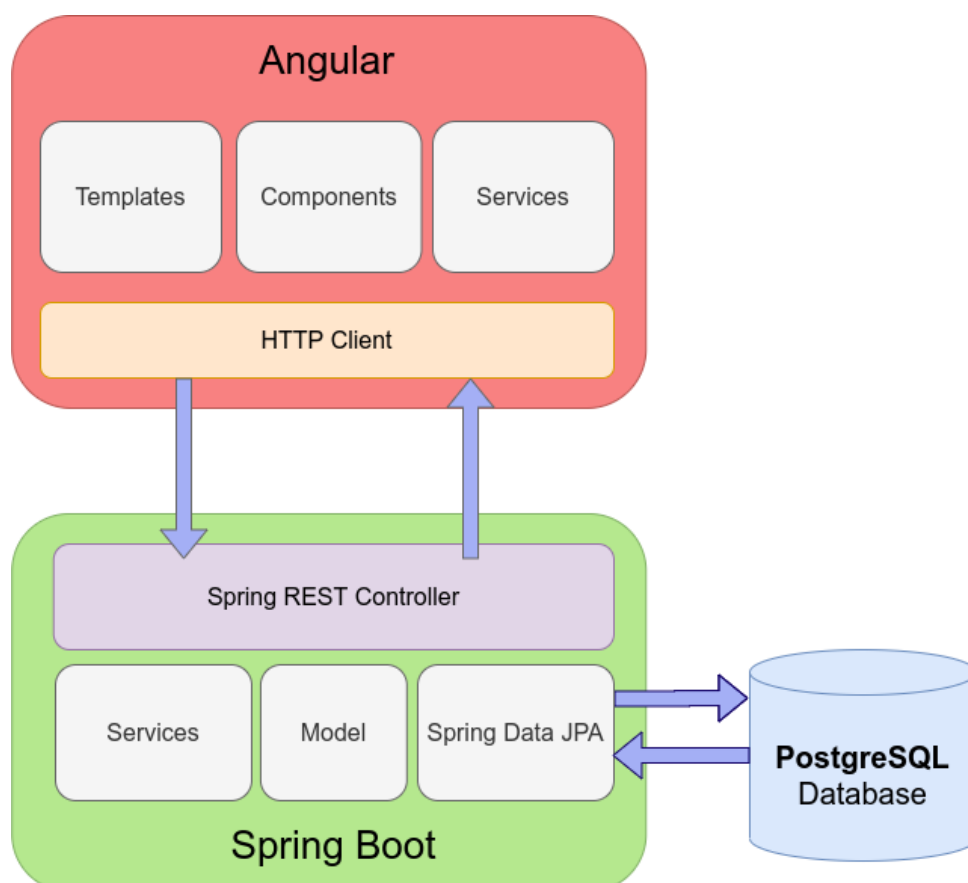
1. Описание на учебния проект
2. Архитектура и дизайн на системата
3. Описание на цялостната реализация
4. Обхват
5. Описание на компонентите в слоя на потребителския интерфейс
6. Описание на компонентите в слоя на приложението/бизнес логиката
7. Описание на слоя на данните

# **1. Описание на учебния проект**

Системата за предоставяне на услуги, свързани с почистване, има за цел да улесни и привлече клиенти, които имат нужда от подобна услуга и желаят тя да бъде възможно най-достъпна за тях. Системата дава възможност както за вътрешно администриране на работническата среда, така и за поръчки и доставки. Всеки клиент има възможност да разгледа каталог от услуги и техника, за която може да заплати и да наеме за определен момент. От друга страна, работниците могат да я използват, за да проследяват поръчките, които трябва да се направят, а началникът им има възможност да ги разпределя при наемане и да следи предлаганите услуги.

## 2. Архитектура и дизайн на системата

Системата се реализира посредством съвременните Java и JavaScript технологии, за да може системата да бъде лесно скалирана, променяна и поддържана. Потребителският интерфейс е направен с Angular, базата чрез PostgreSQL, изградена в Docker контейнер, а мостът помежду им – със Spring Boot.



### **3. Описание на цялостната реализация**

Приложението има 3 режима на работа в зависимост от ролята на регистрирания потребител. Например, началникът може да влезе с админски профил и да създаде профили за работниците си като ги поставя в съответен екип. Може да разглежда поръчките за деня и екипите, които трябва да ги изпълняват. Съществува и опцията за разглеждане на обратна връзка, както от клиенти, така и от служители. По този начин той ще е в течение на всякакви проблеми и неща, за които трябва да се работи в посока подобряване. Системата му позволява и да добавя нови услуги и техника, които фирмата предоставя за наемане.

Служителите могат да разгледат поръчките, които имат да извършат за деня и да оставят обратна връзка в случай на проблем. Това е лесен начин за уведомление на началника.

Клиентите имат възможност да разгледат наличните продукти и да ги добавят към своята кошница за поръчка, да отбележат желано време за доставка, да попълнят нужните данни и да завършват поръчката си като вариантите за плащане са 2 – в брой при доставка или с карта чрез интегриране на външна система за оторизаиране на плащането. Клиентите също могат да оставят своята обратна връзка в случай на повреда, неизпълнени очаквания или нещо друго.

## 4. Обхват

- Регистрация
- Вход в системата като:
  - Администратор
  - Работник
  - Клиент
- Разглеждане на услуги и техника
- Добавяне на продукти в кошницата
- Избор на време за доставка
- Плащане с карта или опция за плащане в брой при доставка
- Оставяне на обратна връзка
- Проследяване на поръчките за деня
- Добавяне на служител
- Добавяне на услуга
- Добавяне на техника
- Разглеждане на обратната връзка

## **5. Описание на компонентите в слоя на потребителския интерфейс**

Angular е мощно средство за създаване на Single-Page Web приложения и всичко се случва чрез модули, съставени от компоненти, които се рендерират при определен адрес, а между адресите се преминава с помощта на Router. По този начин приложението е много по-бързо и интуитивно, защото компонентите се рендерират и пререндерират при експлицитна промяна. В съответната система почти всяка страница е представена от отделен компонент:

- CartComponent – страница с кошница
- CatalogueComponent – страница с каталог от продукти и услуги
- EquipmentComponent – страница за добавяне на нов продукт
- FeedbackComponent – страница за изпращане на обратна връзка
- ViewFeedbacksComponent – страница за визуализиране на обраната връзка



- HomePageComponent – начална страница
- LoginComponent – страница за вход на потребител
- ModerateComponent – страница с опции за модериране като администратор
- ViewOrdersComponent – страница за визуализиране на поръчките за деня
- PaymentComponent – страница за добавяне на данни на карта за плащане
- RegistrationComponent – страница за регистрация
- ServiceComponent – страница за създаване на нов сървиз от администратор
- ToolbarComponent – вечно рендериран обект, който ни помага да навигираме през приложението като меню
- AppComponent – стартова точка за нашето приложение

Модели за приемане и изпращане на заявки в по-чист вид чрез обекти със съответстващи член данни на тези в backend-а:

- Equipment – съдържа интерфейс за Equipment и NewEquipment
- Feedback - съдържа интерфейс за NewFeedback и Feedback
- Order - съдържа интерфейс за Order и ViewOrder
- Service - съдържа интерфейс за NewService и Service
- User - съдържа интерфейс за User и LoginUser
- 

Services:

- BIApiService – осъществява звките към backend-а
- loadingService – осъществява проверките за зареждане през приложението
- localStorageService – осъществява използването на localStorage в браузъра

## **6. Описание на компонентите в слоя на приложението/бизнес логиката**

Бизнес логиката изпълнява определени операции по подадените му от интерфейса данни и ги запазва в базата, както и в обратна посока, от базата към интерфейса. За целта използваме Spring Boot Framework с Java и JDBC, за да навържем таблиците към определени класове Entity-та, репрезентиращи entity-тата в таблиците в базата. Цялата връзка се извършва чрез определени анотации и интерфейси, с които надграждаме JpaRepository. Операциите, които прилагаме, извършваме чрез Service-и, използващи Repository интерфейсите. Всеки end-point, който искаме да достъпваме чрез интерфейса имплементираме чрез REST API Controller-и. Някои обекти, които получаваме, налагат

създаването на различни модели от тези, които използваме за директни репрезентатори на entity-тата. В Spring конфигурираме и базата, която е композирана в Docker контейнер, давайки нужните credentials.

## 7. Описание на слоя на данните

За база от данни приложението използва релационна база чрез PostgreSQL.

E/R Диаграма:

