



**Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА – Российский технологический университет»**

РТУ МИРЭА

Институт информационных технологий

Кафедра инструментального и прикладного программного обеспечения

Дисциплина «Разработка серверных частей интернет-ресурсов»

КУРСОВАЯ РАБОТА

«Веб-сервис по анализу цен интернет-магазинов»

Студент: Доронин К.С.

Группа: ИКБО-16-21

Руководитель: старший преподаватель Сеницын А.В.

Москва 2023



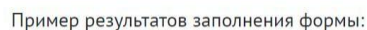
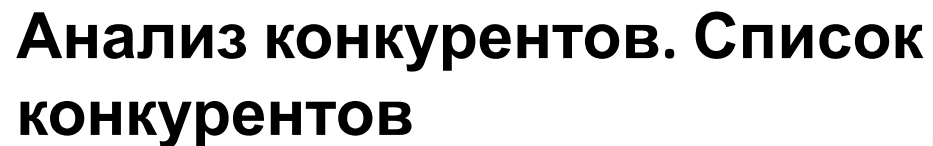
Цели и задачи

Изначальная цель проекта:

Проектирование веб-приложения для интернет-магазина.

Поставленная цель была подразделена на такие задачи:

1. Провести **анализ** предметной области и сформировать функциональные требования к приложению;
2. Выбрать **технологии разработки** и обосновать их использование;
3. Разработать **архитектуру** веб-приложения;
4. Реализовать слой логики **базы данных**;
5. Разработать **серверную часть** приложения с использованием выбранных технологий;
6. Разработать **клиентскую часть** приложения с использованием выбранных технологий.



pricecop



Анализ конкурентов. Функциональные требования

Программный продукт должен включать:

1. Информацию о товарах, их цену, название и ссылку на интернет-ресурс, где их можно приобрести.
2. Сводная таблица с анализом продуктов, наибольшая, наименьшая и средняя текущие цены.
3. Авторизация по электронной почте и регистрация новых пользователей. В личном кабинете должны присутствовать данные о проведенных анализах.
4. Возможность делать запрос на анализ определенного товара, обновлять анализ и удалять.



Технологии разработки

Серверная часть приложения:

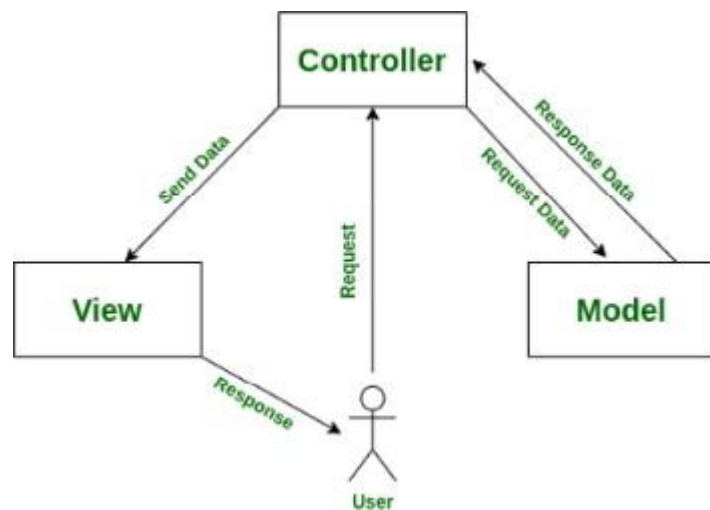
1. Язык программирования Java
2. Фреймворк Spring
3. Spring Security
4. Spring MVC
5. JSON Web Tokens (JWT)
6. Maven
7. PostgreSQL
8. IntelliJ IDEA

Клиентская часть приложения:

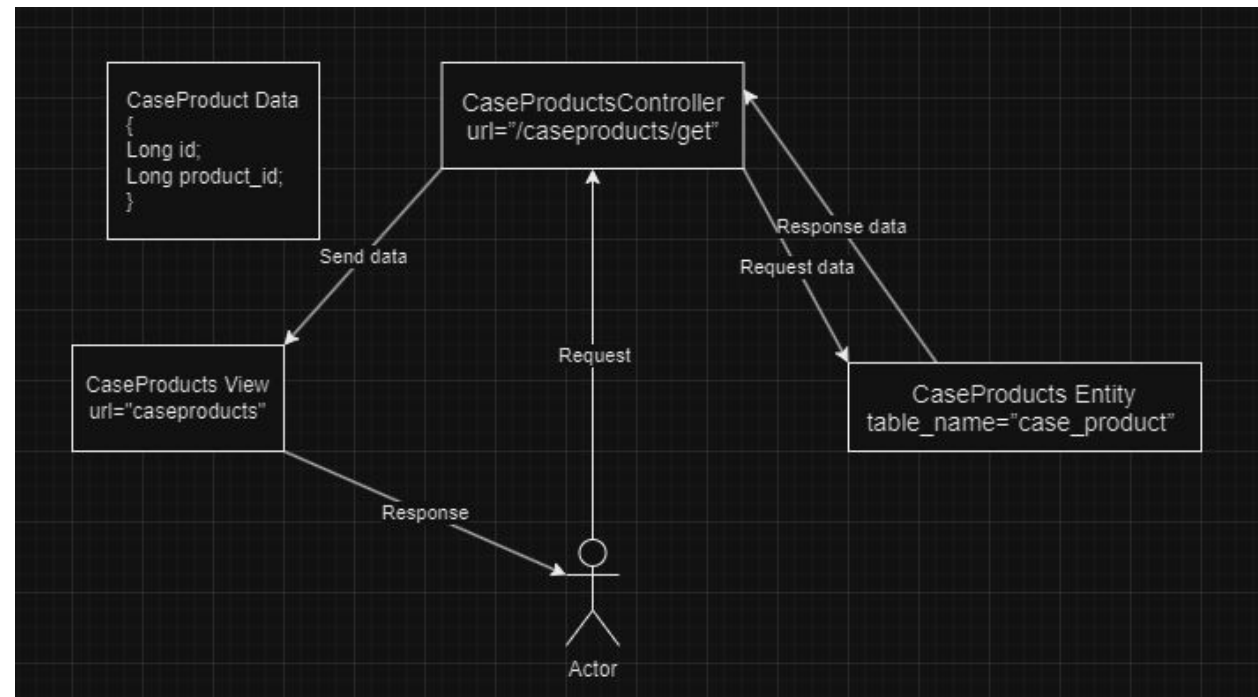
1. HTML
2. CSS
3. JavaScript
4. React
5. React Router
6. Axios



Выбранная архитектура



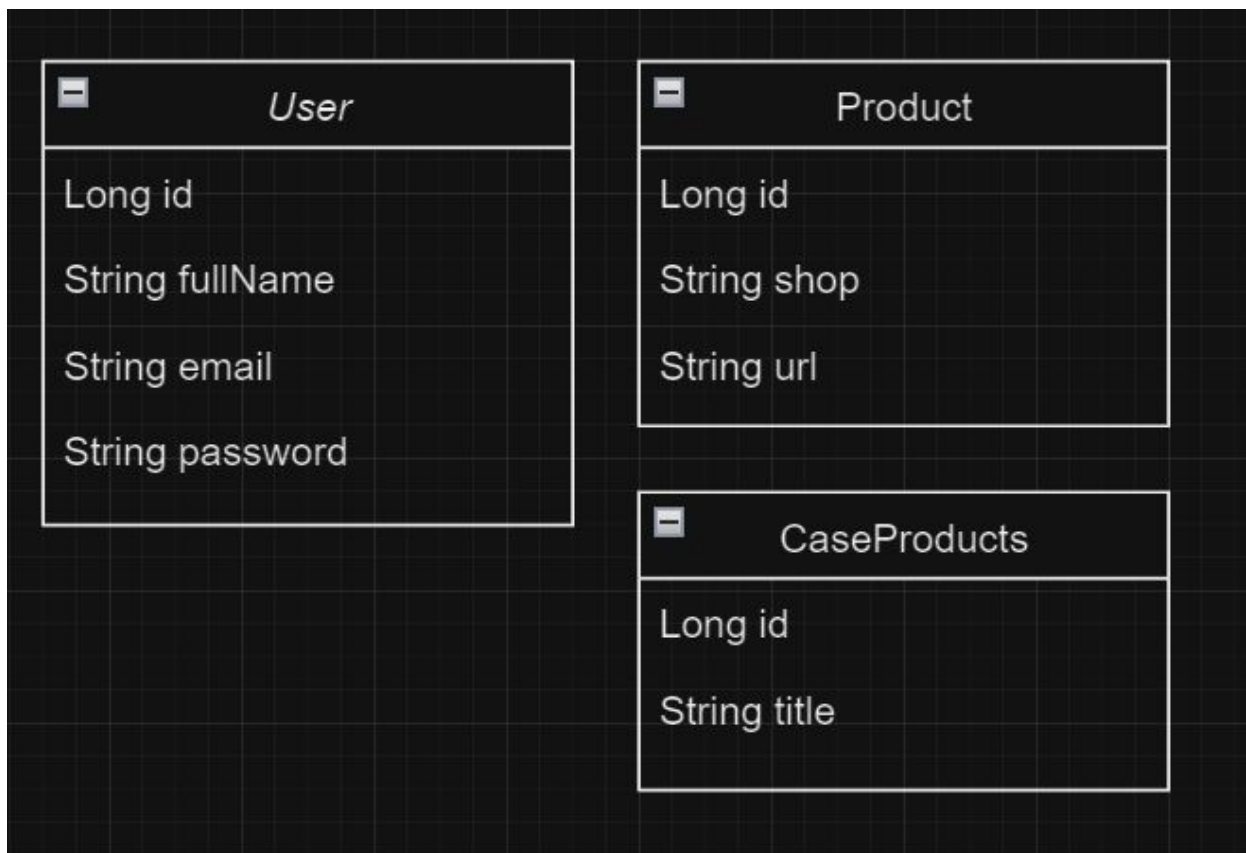
Общий вид архитектуры MVC



MVC на примере группы товаров

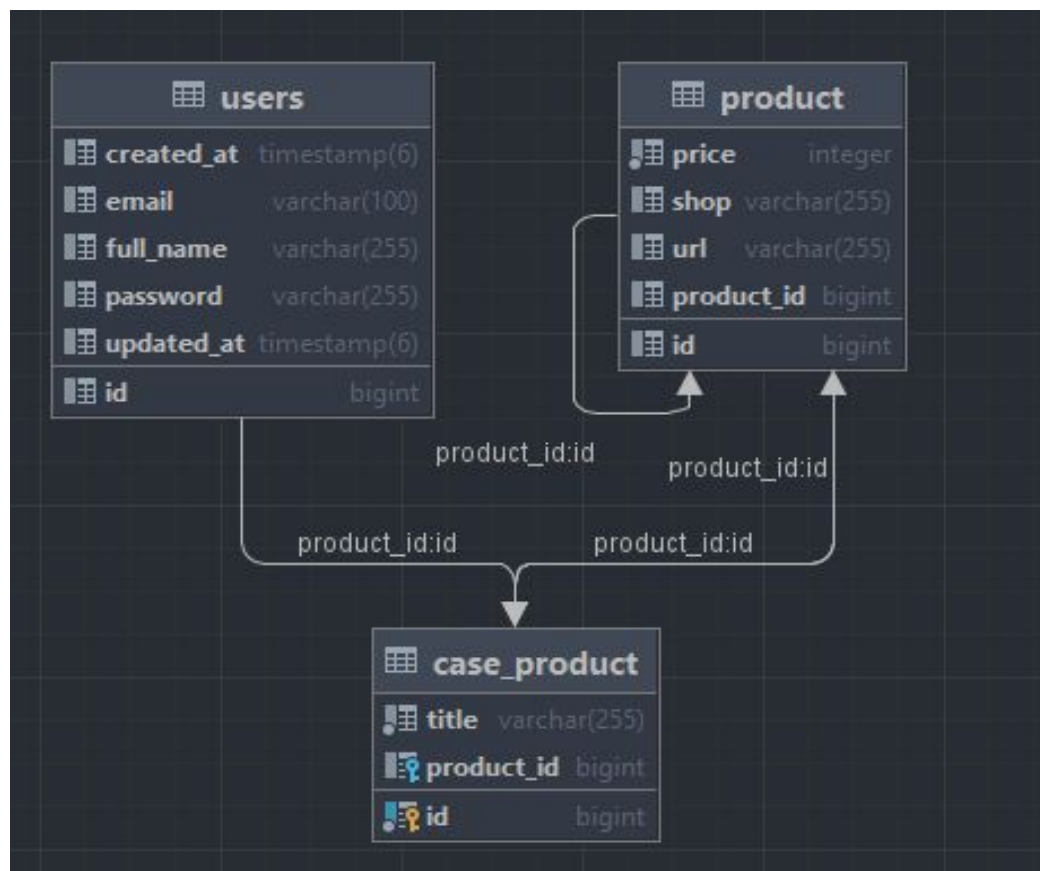


Структура базы данных. Отдельные сущности





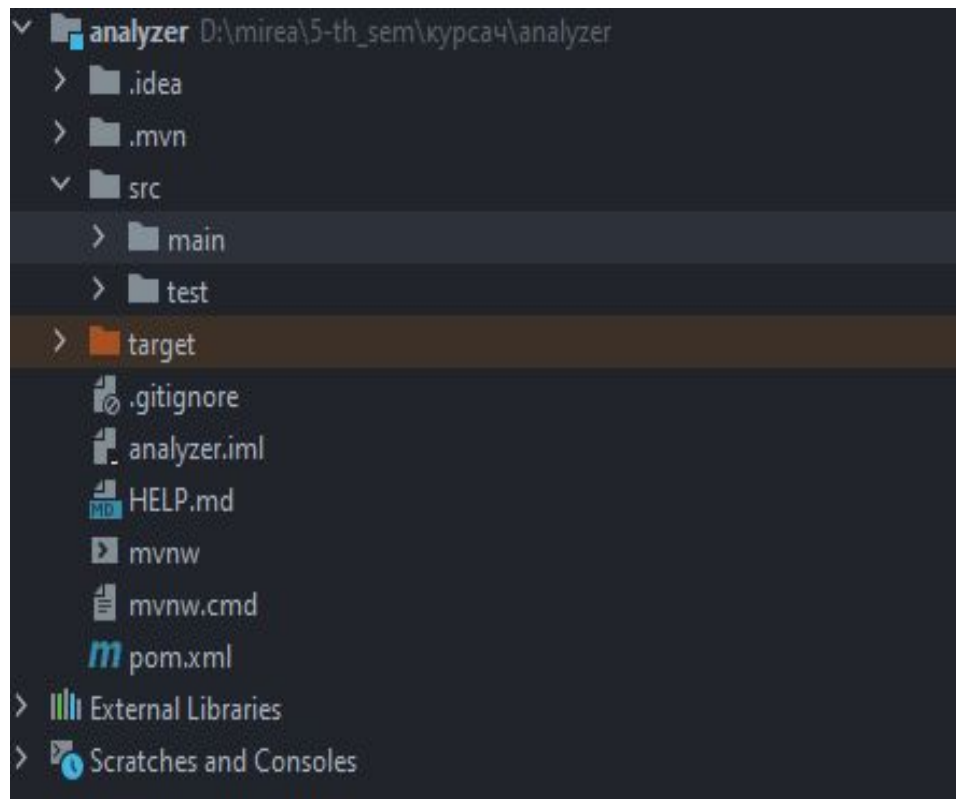
Структура базы данных. Конечная ER-диаграмма



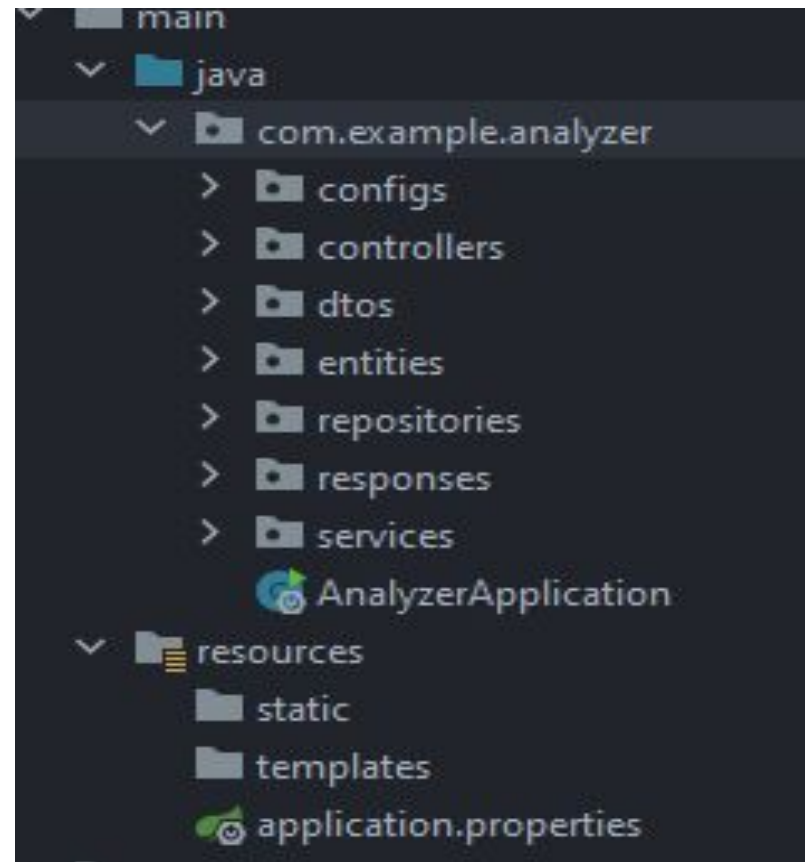
ER-диаграмма спроектированной базы данных



Разработка серверной части. Структура проекта



Общая
структура



Основные
файлы



Разработка серверной части. Конфигурация

```
spring.datasource.url=jdbc:postgresql://localhost:5432/analyzer
spring.datasource.username=postgres
spring.datasource.password=postgres

spring.datasource.driver-class-name=org.postgresql.Driver
spring.jpa.database=postgresql
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update

server.port=8005

security.jwt.secret-key=3cfa76ef14937c1c0ea519f8fc057a80fcd04a7420f8e8bcd0a7567c272e007b
security.jwt.expiration-time=3600000
```

Свойства всего
приложения

```
@Service
public class JwtService {

    1 usage
    @Value("3cfa76ef14937c1c0ea519f8fc057a80fcd04a7420f8e8bcd0a7567c2...")
    private String secretKey;

    2 usages
    @Value("3600000")
    private long jwtExpiration;

    2 usages  ▲ Konstantin
    public String extractUsername(String token) { return extractClaim(token, Claims::getSubject); }

    2 usages  ▲ Konstantin
    public <T> T extractClaim(String token, Function<Claims, T> claimsResolver) {
        final Claims claims = extractAllClaims(token);
        return claimsResolver.apply(claims);
    }

    1 usage  ▲ Konstantin
    public String generateToken(UserDetails userDetails) {
        return generateToken(new HashMap<>(), userDetails);
    }

    1 usage  ▲ Konstantin
    public String generateToken(Map<String, Object> extraClaims, UserDetails userDetails) {
        return buildToken(extraClaims, userDetails, jwtExpiration);
    }

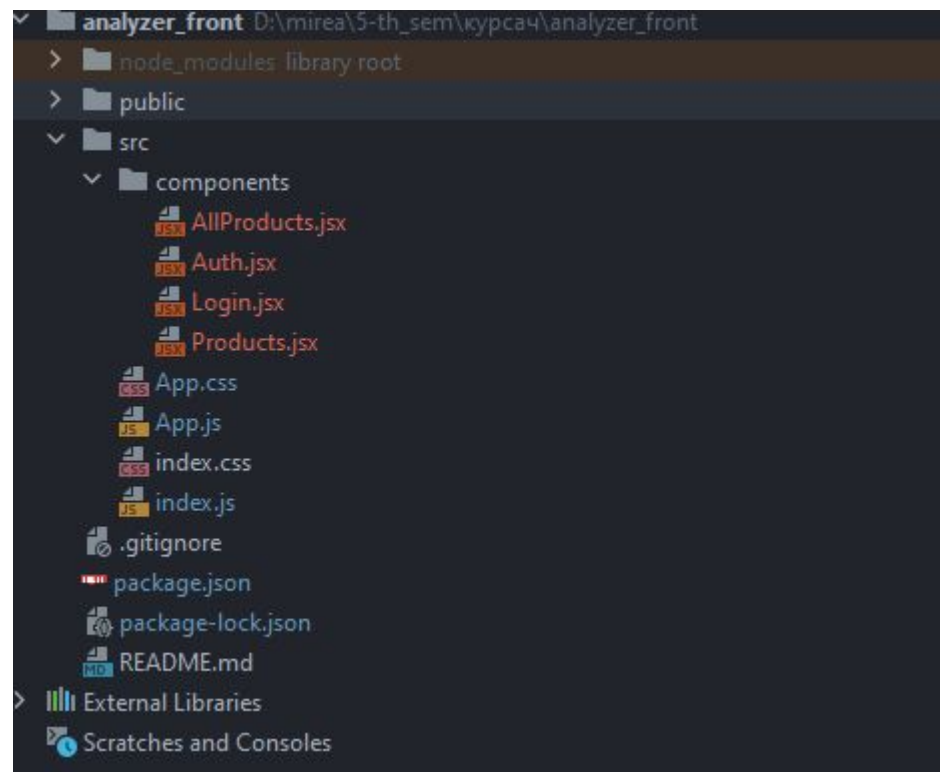
    1 usage  ▲ Konstantin
    public long getExpirationTime() { return jwtExpiration; }

    1 usage  ▲ Konstantin
    private String buildToken(
        Map<String, Object> extraClaims,
        UserDetails userDetails,
```

Обработка
JWT



Разработка клиентской части. Структура проекта



Общая
структура



Разработка клиентской части. Вид конечного приложения

Поиск

Search *

Поиск

[назад](#)[Выйти](#)

Ваши выбранные товары

Название	Ссылка	Обновить	Удалить
apple iphone 14	Перейти	Обновить	Удалить
apple iphone 15	Перейти	Обновить	Удалить

Основная
страница

apple iphone 14

Наименьшая цена		Средняя цена		Наибольшая цена	
68999		80520		113099	
Магазин	Название	Ссылка		Цена	
Biggeek	apple iphone 14	Ссылка		69990	
istudio	apple iphone 14	Ссылка		113099	
mobinot	apple iphone 14	Ссылка		68999	
i-point	apple iphone 14	Ссылка		69990	

Страница просмотра
продукта



Разработка клиентской части. Вид конечного приложения

Войти

Email *
Password *

[Зарегистрироваться](#)

ВОЙТИ

Страница
ауентификации

Регистрация

Name *
Email *
Password *

[Войти](#)

ЗАРЕГИСТРИРОВАТЬСЯ

Страница
регистрации



Результаты

- проведен анализ предметной области и на его основе поставлены функциональные требования,
- выбраны и обоснованы технологии для разработки,
- реализована архитектура MVC для серверной части веб-приложения,
- разработан слой логики базы данных,
- разработана серверная часть приложения «Анализатора цен интернет-магазинов»
- разработана клиентская часть приложения «Анализатора цен интернет-магазинов»



URL хранилища с кодом серверной части:

https://github.com/LIIuko/analyzer_backend

URL хранилища с кодом клиентской части:

https://github.com/LIIuko/analyzer_front

**СПАСИБО ЗА
ВНИМАНИЕ!**