



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

КУРСОВАЯ РАБОТА

по дисциплине: Шаблоны программных платформ языка Джава
по профилю: Разработка программных продуктов и проектирование информационных систем
направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Веб-приложение «Экомаркет»

Студент: Подгорная Ольга Александровна

Группа: ИКБО-13-19

Работа представлена к защите _____ (дата) _____ /Подгорная О.А./
(подпись и ф.и.о. студента)

Руководитель: старший преподаватель, Зорина Наталья Валентиновна

Работа допущена к защите _____ (дата) _____ /Зорина Н.В./
(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: _____

_____/_____,
_____/_____,

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ЗАДАНИЕ
на выполнение курсовой работы

по дисциплине: Шаблоны программных платформ языка Джава
по профилю: Разработка программных продуктов и проектирование информационных систем
направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Подгорная Ольга Александровна
Группа: ИКБО-13-19
Срок представления к защите: 20.05.2021
Руководитель: Зорина Наталья Валентиновна

Тема: Приложение «Экомаркет»

Исходные данные: индивидуальное задание на разработку; документация по Spring Framework и JEE, документация по языку Java 14; инструменты и технологии: JDK 14, создание Spring MVC web-приложений, RESTful web-сервисов, Spring ORM, Gradle, YouTrack, GitHub, IntelliJIDEA. Нормативный документ: инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18.

Перечень вопросов, подлежащих разработке, и обязательного графического материала:
1. Провести анализ предметной области и формирование основных требований к приложению.
2. Обосновать выбор средств ведения разработки. 3. Разработать приложение с использованием фреймворка Spring и выбранной технологии и инструментария. 4. Провести тестирование приложения. 5. Оформить пояснительную записку по курсовой работе. 6. Провести анализ текста на антиплагиат. 7. Создать презентацию по выполненной курсовой работе.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: _____ /Р. Г. Болбаков/, «_____» _____ 2021 г.

Задание на КР выдал: _____ /Н.В. Зорина/, «_____» _____ 2021 г.

Задание на КР получил: _____ /О.А. Подгорная/, «_____» _____ 2021 г.

Подгорная О.А. «Веб-приложение «Экомаркет»» / Курсовая работа по дисциплине «Шаблоны программных платформ языка Джава» профиля «Разработка программных продуктов и проектирование информационных систем» направления профессиональной подготовки бакалавриата 09.03.04 «Программная инженерия» (4-й семестр) / руководитель старший преподаватель Н. В. Зорина / кафедра Инструментального и Прикладного Программного Обеспечения института Информационных Технологий РТУ МИРЭА – с. 27, табл. 4, рис. 13, ист. 10.

Целью работы является создание веб-приложения «Экомаркет».

В рамках работы осуществлены краткий анализ аналогов инструментов для создания веб-приложения, написание программного кода, создание веб-страниц и тестирование программного продукта.

Podgornaya O.A. «Web application “Ecomarket”» / Coursework in the discipline «Java software platform templates» of the profile «Software development and information systems design» courses undergraduate 09.03.04. “Software Engineering” (4th semester) / head associate senior lecturer N.V. Zorina/ department IPPO of institute IT RTU MIREA.

The aim of the work is to create web application “Ecomarket”.

As part of the work, a brief analysis of the analogs of instruments for developing web application, writing the program code, creating web pages and testing the program product.

СОДЕРЖАНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ.....	5
ВВЕДЕНИЕ	6
1. СБОР И АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ	7
1. 1. Описание предметной области	7
1. 2. Выбор технологий разработки веб-приложения.....	7
Выводы к разделу 1	13
2. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ	14
2. 1 Проектирование программного продукта	14
2. 2 Структура программного продукта.....	14
Выводы к разделу 2	22
3. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА.....	23
3. 1. Тестовый план	23
3. 2. Тест кейсы.....	25
Выводы к разделу 3	32
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	34

СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

- 1) HTML – HyperText Markup Language - стандартизированный язык разметки документов во Всемирной паутине
- 2) СУБД – Система Управления Базами Данных
- 3) IDE -- Integrated Development Environment -- Интегрированная среда разработки
- 4) JVM – Java Virtual Machine – Виртуальная машина Java
- 5) MVC – Model-View-Controller – Модель-Представление-Контроллер
- 6) REST – Representational State Transfer – Передача состояния представления
- 7) ORM – Object-Relational Mapping – Объектно-реляционное отображение

ВВЕДЕНИЕ

В отличие от статичных веб-сайтов, веб-приложения позволяют реализовывать работу на серверной части, включающую в себя хранение и защиту данных, доступ к ним, а также ответы на запросы. Помимо этого, при переходах между разделами, отправки форм и обновления данных, страница в браузере будет перезагружаться, каждый раз генерируя новый HTML- файл.

Целью данной курсовой работы является создание веб-приложения «Экомаркет», представляющего из себя интернет-магазин, и для ее достижения требуется выполнение следующих задач:

- Непосредственно разработка веб-приложения с использованием Spring Framework
- Поддержка СУБД
- Проведение тестирования приложения

В качестве информационной базы использовались Интернет-ресурсы, касающиеся как темы предмета исследования, так и разработки, в том числе официальная документация технологий и элементов инструментария.

Данная курсовая работа включает в себя следующие разделы:

- Анализ требований к программному продукту;
- Выбор технологий разработки веб-приложения;
- Определение стадий разработки;
- Формирование тестового плана;
- Формирование тест-кейсов.

1. СБОР И АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ

1. 1. Описание предметной области

Предметной областью для данной курсовой работы является исследование веб-ресурсов по продаже эко-товаров.

В последние годы возрастает заинтересованность людей в разумном потреблении, в следствие чего повышается спрос на экологичные товары. Согласно опросам [1], молодое поколение обеспокоенно проблемами экологии, поэтому, вероятно, данный спрос будет лишь увеличиваться. Желание отказаться от пластика мотивирует людей выбирать экомаркеты, или рынки, вместо привычных супермаркетов.

Экомаркет как таковой представляет из себя магазин экологически чистых товаров, то есть тех товаров, которые не оказывают вредного воздействия на окружающую среду и человека. Это могут быть привычные вещи, такие как пакеты, предназначенные для многоразового использования, а также пищевая продукция, выращенная без использования вредных удобрений и реализуемая без перерабатываемой пластиковой упаковки.

Наиболее привычным форматом экомаркета является торговая площадь под открытым небом либо на какой-либо территории, однако в условиях ограничения передвижений и скоплений людей, а также всеобщего увеличения предпочтения онлайн-покупкам, имеет смысл реализация эко-продуктов в Интернете. Вследствие этого, веб-приложение экомаркет должно полноценно заменить оффлайн покупку товаров, давая возможность пользователям изучить товары на экране своего устройства.

1. 2. Выбор технологий разработки веб-приложения

В качестве среды разработки будет использоваться среда IntelliJ IDEA (Рисунок 1). В данной среде предоставляется обширная помощь в работе с фреймворком Spring, включающая в себя настройку проекта, выделение кода, контекстные действия, встроенную документацию, удобную навигацию и настраиваемые конфигурации запуска [2]. Также IntelliJ IDEA автоматически

обнаруживает файлы конфигурации Spring, и IDE сообщает, если проект требует каких-либо корректировок.

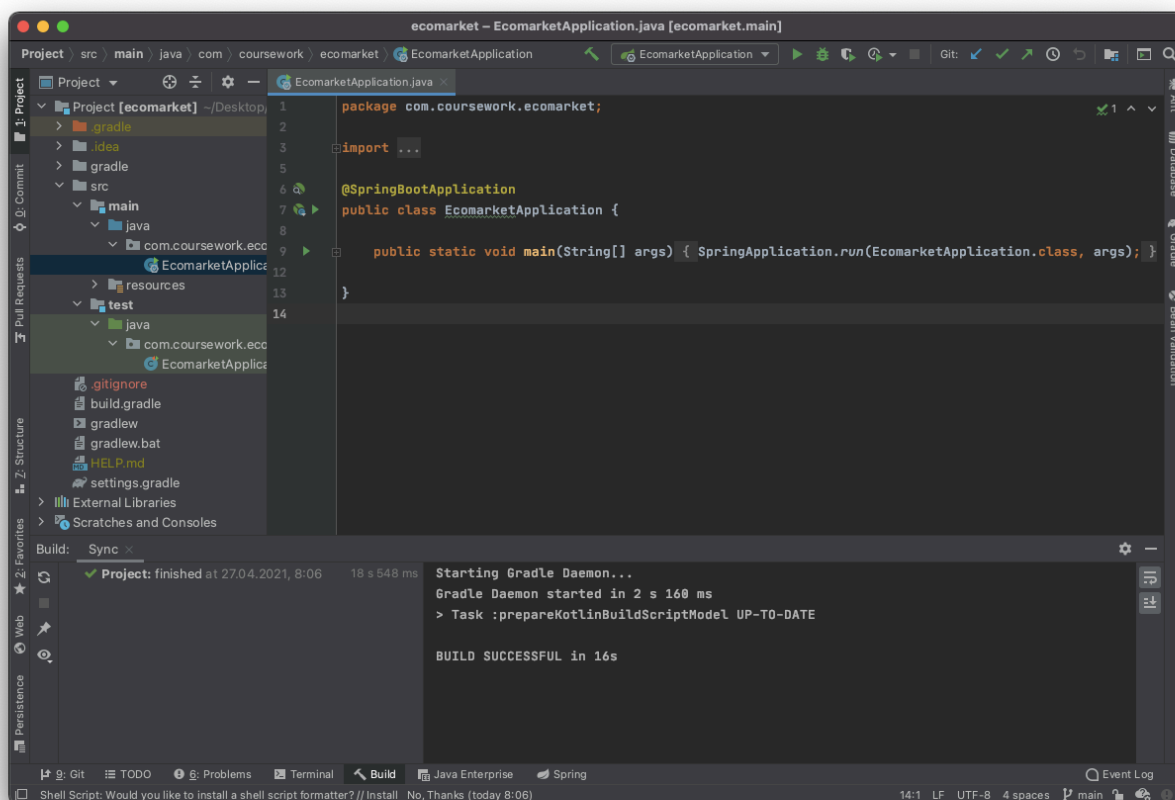


Рисунок 1 – пример проекта в IntelliJ IDEA

В качестве языка программирования для данного курсового проекта будет использоваться Java 14, одна из последних версий языка Java. Помимо Java, Spring также поддерживает такие языки программирования, как Kotlin и Apache Groovy [3].

Kotlin -- это статически типизированный язык, ориентированный на JVM (и другие платформы), который позволяет писать краткий и элегантный код, обеспечивая при этом отличную совместимость с библиотеками, написанными на Java.

Groovy — это мощный, необязательно типизированный и динамический язык с возможностями статической типизации и статической компиляции. Он предлагает краткий синтаксис и легко интегрируется с любым существующим Java-приложением.

В качестве инструмента автоматизации сборки будет использоваться Gradle. Он обладает такими качествами, как высокая производительность, расширяемость, сканирование сборки. Сама сборка осуществляется путем выполнения задач, смоделированных как ациклический граф. После создания этого графа, Gradle определяет, в каком порядке их необходимо запускать, а затем приступает к выполнению [4]. На рисунке 2 представлен скриншот файла `gradle.build`, являющимся обязательным файлом скрипта конфигурации сборки.

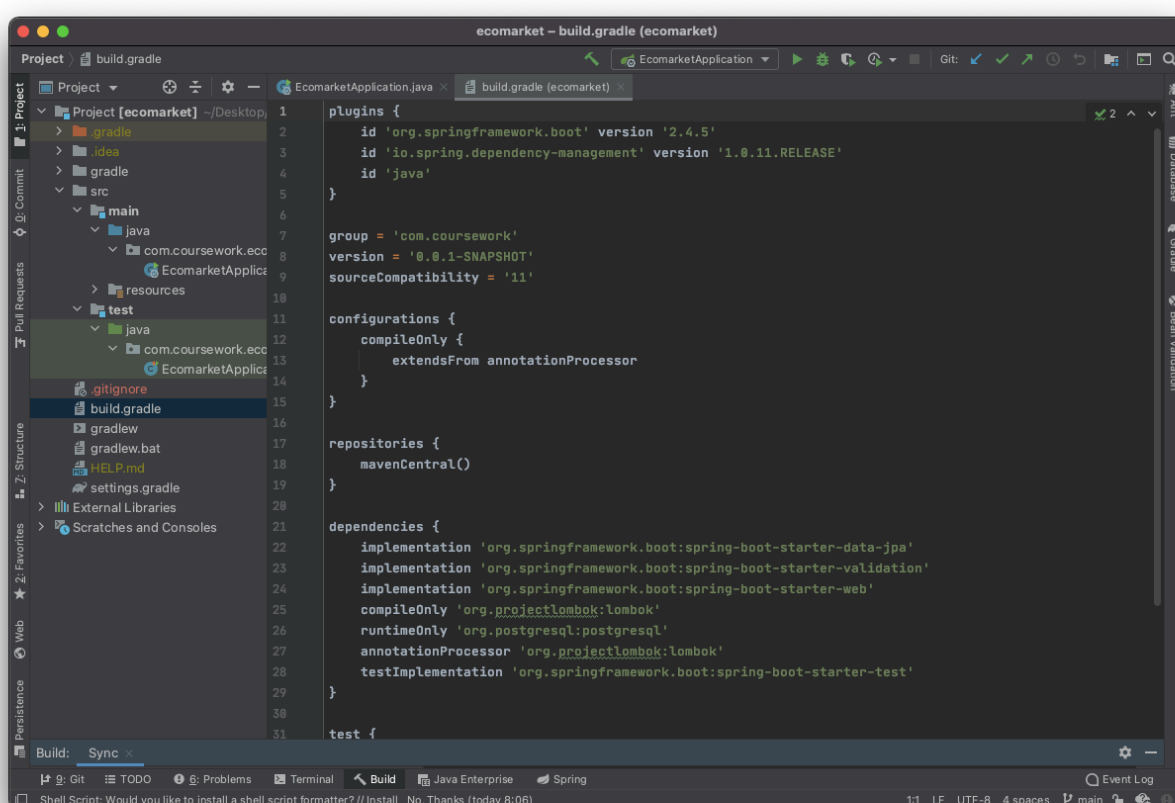


Рисунок 2 – пример файла `gradle.build`

Помимо Gradle, существуют и другие системы сборки. Самой популярной из них является Maven, функции которого схожи с функциями Gradle. Для обеих систем центральным репозиторием зависимостей является `mavencentral`. Главное отличие – формат файла конфигурации. В Maven он представлен в виде `pom.xml` файла, где данные прописываются в виде тегов. Пример `pom.xml` файла представлен на рисунке 3.

```

1. <project>
2.   ...
3.   <properties>
4.     <mavenVersion>3.0</mavenVersion>
5.   </properties>
6.
7.   <dependencies>
8.     <dependency>
9.       <groupId>org.apache.maven</groupId>
10.      <artifactId>maven-artifact</artifactId>
11.      <version>${mavenVersion}</version>
12.    </dependency>
13.    <dependency>
14.      <groupId>org.apache.maven</groupId>
15.      <artifactId>maven-core</artifactId>
16.      <version>${mavenVersion}</version>
17.    </dependency>
18.  </dependencies>
19.  ...
20. </project>

```

Рисунок 3 – пример файла pom.xml

В качестве сервиса онлайн-хостинга исходного кода будет использоваться GitHub -- самый популярный веб-хостинг IT-проектов. С помощью него удобно отслеживать версии приложения, а также он позволяет другим пользователям просматривать содержание файлов.

Иным программным обеспечением для контроля версий выступает GitLab, который также поддерживает систему отслеживания ошибок, создание заданий для групп, учитывает рабочее время, обладает функциями защиты ветвей, блокировки файлов, персонализации уведомлений и т.д.

Также существуют хостинги для групповой разработки, поэтому их использование не предполагается данной курсовой работой, но их вполне можно использовать как альтернативные. Одной из них является Bitbucket – полностью масштабируемая платформа для разработчиков. Сюда можно легко импортировать репозитории из GitHub, разворачивать модели, отслеживать задачи и многое другое.

Еще одним полезным сервисом является система отслеживания ошибок YouTrack, разработанная компанией JetBrains. Она поддерживает поисковые запросы, манипуляции с наборами задач, автодополнение кода и т.д., обладает встроенной интеграцией с системами контроля версий, описанных выше. Данное программное обеспечение используется, в основном, для корпоративной

деятельности и командных работ, повышая эффективность разработки как группы пользователей, так и каждого сотрудника, позволяя эффективно управлять задачами [6].

Фреймворком для создания веб-приложения является Spring. Он избавляет от повторения рутинных шагов настройки проекта. Общее название утилит, автоматизирующих процедуры настройки и ускоряющих процесс создания и развертывание веб-приложений на базе Spring – Spring Boot. С помощью него разработчикам требуется приложить минимум усилий для настройки веб-приложения и написания кода.

В ходе курса «Шаблоны программных платформ языка Java» были изучены паттерны проектирования, некоторые из которых используются фреймворком Spring[5]: фабричный метод для загрузки бинов с использованием BeanFactory и Application context, паттерн «одиночка» для определения бинов (если не указано обратное), «прототип» для конструирования объектов, которые в последствии могут быть отделены от имплементирующих систем и т.д.

В качестве архитектурного стиля веб-приложения будут использоваться RESTful веб-сервисы. Данный стиль ограничивает и упрощает архитектуру «клиент-сервер» и предназначен для использования протокола связи без сохранения состояния, обычно HTTP. Обмен представлениями ресурсов происходит с помощью стандартизированного интерфейса и протокола. RESTful-приложения обладают некоторыми принципами для простоты, легкости и скорости использования. Так, веб-служба RESTful предоставляет набор ресурсов, идентифицирующих цели взаимодействия с клиентами, обеспечивая глобальное адресное пространство для обнаружения ресурсов и служб. Помимо этого, существует лишь четыре операции для работы: PUT (создает новый ресурс), DELETE (удаляет ресурс), GET (извлекает текущее состояние ресурса) и POST (передает новое состояние ресурсу). При взаимодействии через гиперссылки состояние сохраняется [7].

Технология SpringORM предоставляет шаблоны для работы. В ходе данной курсовой работы будет использоваться библиотека HibernateORM, обладающий расширяемостью, надежностью, масштабируемостью, высокой

производительностью. Данная библиотека является самой популярной реализацией спецификации JPA – Java Persistence API, она позволяет сократить объемы низкоуровневого программирования при работе с реляционными базами данных, может использоваться для создания таблиц в процессе проектирования классов, а также для работы с уже существующими базами. Альтернативными ORM-фреймворками для языка программирования Java являются MyBatis, iBATIS, TopLink, EclipseLink [8].

В качестве системы управления базами данных используется PostgreSQL – мощная объектно-реляционная база данных с открытым исходным кодом, использующая и расширяющая язык SQL в сочетании с множеством функций, включающих в себя безопасное хранение и масштабирование данных. Преимуществами PostgreSQL является, как говорилось ранее, защита целостности данных, создание отказоустойчивой среды, расширяемость типов данных и функций, поддержка разных языков программирования без перекомпиляции данных.

Иными СУБД являются [9]:

- Oracle 12c, продукт проверенный временем и обладающий блестящей репутацией, однако его стоимость для организаций может оказаться слишком высокой. Кроме того, данная СУБД требует значительных ресурсов, что означает модернизацию оборудования при необходимости;
- MySQL, одна из самых популярных баз данных для веб-приложений, разработкой и модернизацией которой так же является корпорация Oracle. Ее недостатком является отсутствие встроенного выполнения задач, которые PostgreSQL делает автоматически, например создание инкрементных резервных копий;
- MongoDB поддерживает традиционные документы NoSQL, такие как json, а также обладает скоростью и простотой в использовании. Однако здесь язык SQL не используется в качестве языка запросов;
- Линтер – российская СУБД, реализующая стандарт SQL:2003. Поддерживает большинство операционных систем (Windows, UNIX-

системы), защищает данные от несанкционированного доступа и рекомендована «Единым реестром российских программ». Однако, в случае высокой динамики изменений, падает эффективность работы.

Выводы к разделу 1

В ходе анализа предметной области и технологий разработки веб-приложения были сформированы основные требования для разработки. Они включают в себя:

- Добавление фреймворка Spring в проект;
- Добавление поддержки системы сборки Gradle;
- Добавление необходимых зависимостей проекта, таких как Spring web, Spring data JPA;
- Интеграция RESTful веб-сервисов;
- Подключение базы данных для хранения информации о товарах и пользователях.

2. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ

2.1 Проектирование программного продукта

Разработка программного продукта «Экомаркет» производится по шаблону MVC – Model-View-Controller, разделяющему данные бизнес-логики и пользовательского интерфейса на три компонента: модель, представление и контроллер.

Основными требованиями для данного курсового проекта являются следующие функции:

- Регистрация, вход и выход в аккаунт пользователя;
- Возможность управления категориями, товарами и пользователями для администраторов;
- Добавление товаров в корзину с подсчетом общей стоимости корзины.

Описанные функции могут быть разбиты на подфункции. Так, вход пользователя в аккаунт подразумевает дополнительную логику проверки корректности введенных данных, а регистрация – проверку адреса электронной почты, а именно зарегистрирована ли она.

Администрирование веб-приложение также включает в себя несколько подфункций, работающих с разными таблицами БД. Кроме того, редактирование и добавление товаров подразумевает добавление изображений, по этой причине необходимо отдельно проработать работу с загрузкой файлов.

2.2 Структура программного продукта

2.2.1 Написание основного кода

В качестве моделей выступают три класса:

- Класс User содержит информацию о пользователях, такую как имя, фамилия, адрес электронной почты, пароль и роль;

- Класс `Product` содержит информацию о продуктах: наименование, цена, количество, описание, номер категории, к которой принадлежит данная единица товара;
- Класс `Category` отвечает за информацию о категориях, храня наименование каждой.

Все эти классы связаны между собой для хранения информации в базе данных, работа которой будет рассмотрена в следующей главе.

Для хранения, удаления, сохранения и поиска данных, создадим директорию с репозиториями – интерфейсами, расширяющими `JpaRepository`. Для каждого класса-репозитория используется аннотация `Repository`, позволяющая фреймворку `Spring` использовать их по назначению.

В качестве фасада для бизнес логики веб-приложения, создается директория `services`, в которой обозначены классы-интерфейсы сервисов с использованием аннотации `@Service`, а также их имплементации. В интерфейсах, соответственно, обозначаются функции для работы с моделями, такие как добавление или удаление категории/продукта/пользователя и тд. В классах-имплементациях прописывается непосредственно программный код. Дополнительным сервисом является класс, регулирующий правильную загрузку файлов, а именно изображений товаров. В нем прописывается директория хранения.

Классы-контроллеры, хранящиеся в директории `controllers`, отвечают за отображение данных для пользователей. В данной курсовой работе используются три контроллера:

- `AccountController` отображает данные корзины пользователя, добавляя в нее товары и подсчитывая общую стоимость;
- `AdminController` отображает данные для администрирования веб-приложения, включающие в себя добавление, удаление и изменение продуктов и категорий, а также добавление пользователей, включающее выбор роли (простой пользователь или администратор);

- HomeController регулирует отображение данных на главной странице, т.е. полный список товаров и их отображение по категориям. Также с помощью данного контроллера гость веб-ресурса может войти в свой аккаунт, либо же создать новый.

Для верной работы веб-ресурса создается директория configuration. В ней четыре файла, использующиеся для безопасности хранения данных, конфигурации файлов и регистрации пользователей.

Класс FileConfiguration, реализующий метод интерфейса WebMvcConfigure addResourceHandlers(ResourceHandlerRegistry registry), позволяет корректно отображать статичные изображения веб-приложения.

Другим важным классом является класс SpringSecurityConfiguration, благодаря которому пользователи могут успешно входить в аккаунт и регистрироваться на веб-ресурсе. В этой же конфигурации происходит хэширование паролей, что увеличивает безопасность.

Остальные два класса, UserPrincipal и UserPrincipalDetailsService, работают непосредственно с аккаунтами пользователей, проверяя их существование и возвращая имена пользователей и пароли, а также сохраняя их данные и роли.

На рисунке 4 представлена структура директории Java-кода проекта:

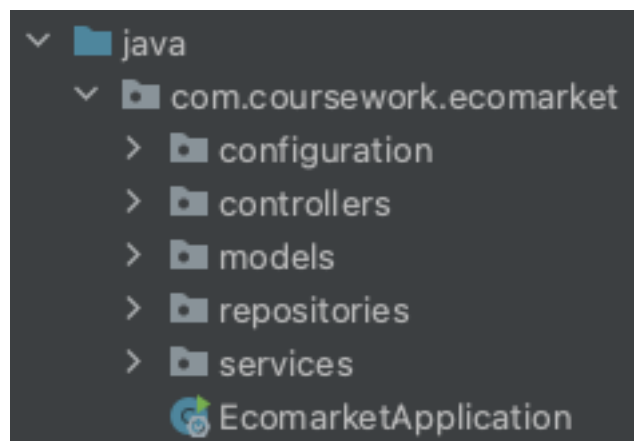


Рисунок 4 – структура директории java

2.2.2 Настройка фреймворков

Для корректной работы проекта необходима правильная настройка его фреймворков. В ходе данной курсовой работы использовались Gradle для сборки веб-приложения, а также Spring.

Конфигурация Gradle, как описывалось в первой части, происходит в файле build.gradle. Помимо стартовых зависимостей, используемых для запуска Spring, использовались также дополнительные (Рисунок 5):

- Lombok – дополнительная библиотека, аннотации которой сокращают количество кода путем автоматического добавления геттеров, сеттеров и конструкторов;
- Движок Postgres соединяет код с базой данных;
- Серверный механизм Thymeleaf дает возможность создавать элегантный и удобный способ шаблонизации, он упрощает работу с файлами-шаблонами, в случае данного проекта – файлами HTML;
- Security-зависимости предоставляют механизмы построения систем аутентификации и авторизации, а также другие возможности обеспечения безопасности веб-приложения.

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-validation'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    compileOnly 'org.projectlombok:lombok'  
    runtimeOnly 'org.postgresql:postgresql'  
    annotationProcessor 'org.projectlombok:lombok'  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'  
    developmentOnly 'org.springframework.boot:spring-boot-devtools'  
    implementation group: 'org.springframework.boot', name: 'spring-boot-starter-security', version: '2.4.5'  
    testImplementation group: 'org.springframework.security', name: 'spring-security-test', version: '5.4.6'  
    implementation group: 'org.thymeleaf.extras', name: 'thymeleaf-extras-springsecurity5', version: '3.0.4.RELEASE'  
}
```

Рисунок 5 – зависимости проекта

Для настройки непосредственно Spring используется файл application.properties (рисунок 6). В нем прописываются конфигурации работы с базами данных, а также настройка загрузки файлов-изображений.

```
#Database and Hibernate configuration
spring.datasource.url=jdbc:postgresql://localhost:5432/ecomarket
spring.datasource.username=postgres
spring.datasource.password=
spring.jpa.open-in-view=false
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true

#Multipart configuration
spring.servlet.multipart.max-file-size=800MB
spring.servlet.multipart.max-request-size=1024MB
```

Рисунок 6 – файл application.properties

2. 2. 3 Работа с базой данных

Подключение к базе данных Postgres происходит в файле application.properties, там же происходит настройка Hibernate – библиотеки для работы непосредственно с базами данных.

В классах моделях инициализируются столбцы таблиц, представляющих данные. Аннотация @Entity (сущность) передает информацию Spring о том, что данный класс хранит поля таблиц. Также каждый класс-модель включает в себя поля с аннотациями @Id и @GeneratedValue, которые автоматически генерируют Id строки согласно порядковому номеру создания объекта.

Дополнительно, в ходе данного курсового проекта использовались ассоциации. Так, с помощью связей «один ко многим» и «многие к одному» и аннотаций @OneToMany и @ManyToOne каждому продукту присвоена категория согласно ее идентификационному номеру. Для формирования пользовательской корзины использовалось отношение «многие ко многим», а также аннотации @JoinTable и @JoinColumn, создающих отдельную таблицу, состоящую из двух столбцов – номер пользователя и номер товара, который он добавил к корзине (рисунок 7).

	! user_id ↕	! product_id ↕
1	10	9
2	10	8
3	10	9

Рисунок 7 – внешний вид таблицы user_product_list

В результате, проект имеет четыре таблицы, хранящих данные о категориях, продуктах, пользователях и добавленных пользователями продуктах в корзину. Структура представлена на рисунке 8.

▼	tables 4
▼	category
	! category_id bigint
	! category_name varchar(255)
▼	product
	! product_id bigint
	! image varchar(255)
	! product_description varchar(255)
	! product_name varchar(255)
	! product_price integer
	! product_unit integer
	! category_id bigint
▼	user_product_list
	! user_id bigint
	! product_id bigint
▼	users
	! user_id bigint
	! active integer
	! email varchar(255)
	! first_name varchar(255)
	! last_name varchar(255)
	! password varchar(255)
	! role varchar(255)

Рисунок 8 – структура таблиц базы данных проекта

2. 2. 4 Разработка клиентской части

Файлы, визуально отображающие веб-приложение, являются HTML-документами. Они хранятся в папке `templates`, путь которой `Project/src/main/resources/templates`. Они подразделены на три категории: директория для файла, отображающего содержание корзины, директория администраторской части, и файлы, видимые всем пользователям.

Важным документом является файл `blocks.html`. В нем прописываются фрагменты, которые впоследствии будут вставлены в каждую HTML-страницу. Фреймворк `Thymeleaf` дает возможность для разработчика избавиться от постоянного копирования кода с помощью специальных атрибутов: `th:fragment` и `th:insert`. Так, в одном документе, в данном случае `blocks.html`, для постоянных частей документа, таких как `<head>` и навигационная панель веб-приложения, прописывается атрибут `th:insert`.

Помимо этого, `Thymeleaf` предоставляет возможность работы с файлами серверной части, отправляя специальные запросы с использованием атрибута `th:href`.

Для входа и выхода пользователей используются атрибуты `Security`, такие как `sec:authorize=""`.

Так как данный курсовой проект не предполагает обширного использования возможностей языков разметки и стилей, в нем используется фреймворк `Bootstrap`, упрощающий создание стилей веб-страниц. Он подключается в хедере файла.

На главной странице веб-приложения располагаются все реализуемые товары. С помощью атрибутов `th:text` и `th:src` в HTML-документ добавляются значения полей из базы данных. Например, атрибут `th:text=""${product.productName}"` возвращает наименование продукта. Помимо списка продукции, на этой же странице предусмотрен список категорий, выбор одной из которых открывает список продукции определенной категории по ее идентификационному номеру. При этом в контроллере происходит поиск по ID и возвращается список.

Вход в аккаунт происходит на странице `login.html`, где расположены формы ввода адреса электронной почты, пароля, а также чек-бокс для запоминания входа. Здесь же используются атрибуты Thymeleaf, такие как `th:action` и `th:if`. Вход и выход отправляют `post`-запрос на сервер.

Веб-страницы регистрации `signup.html` одержит в себе формы имени, фамилии, адреса электронной почты и пароля. После нажатия кнопки «Зарегистрироваться», отправляется запрос, и в базу данных добавляется новый пользователь. При этом, пароль в ней хранится в хэшированном виде.

После входа изменяется навигационная панель веб-приложения: удаляются кнопки входа и регистрации, вместо них появляется кнопка выхода. Кроме того, если пользователь является администратором, добавляется кнопка «Администрирование», нажатие на которую открывает веб-страницу `~/admin/index`, где перечислен функционал. Он включает в себя управление категориями и продуктами, добавление пользователей, а также вывод списка пользователей и персонала. Все эти данные доступны благодаря использованию атрибута `th:text`, который, как говорилось ранее, имеет доступ к полям баз данных. Также, атрибут `th:each` позволяет выводить данные каждой строки таблицы, избавляя от написания циклов.

После нажатия кнопки «Добавить в корзину» около товара с главной страницы, пользователь перенаправляется на страницу `~/account/cart`, где представлен список товаров, добавленных пользователем. Данные для корзины берутся из таблицы `user_product_list`, которая описывалась в предыдущей части.

Администраторские возможности более расширены. Здесь можно добавлять и удалять данные из таблиц баз данных путем отправления `post`-запросов на сервер. Добавление информации о единицах работает с помощью атрибута `th:value`. Интересно отметить загрузку изображений: файл загружается в папку `static/images`, а данные о нем в таблице СУБД хранятся как текст, т.е. путь к файлу.

Выводы к разделу 2

Разработка веб-приложения «Экомаркет» включает в себя несколько важнейших этапов. Первым из них является добавление необходимых зависимостей для сборки и конфигурация проекта, например добавление настроек безопасности и загрузки файлов. На этом же этапе происходит настройка функционала Spring ORM – подключение к фреймворку Hibernate в файле `application.properties`, позволяющая удобно взаимодействовать с СУБД.

Следующим этапом является написание непосредственно программного кода на языке Java по модели MVC, создание классов-моделей, контроллеров, сервисов и репозиториев.

В конце разрабатывается внешний вид веб-приложения с использованием языка разметки HTML и интеграцией Thymeleaf для взаимодействия с кодом и таблицами базы данных.

Подразделение работы на этапы значительно упрощает создание программного продукта, облегчает понимание работы Spring, улучшает эффективность разработки.

В конце имеется продукт, готовый к использованию. Соответственно, остается лишь протестировать возможности описанного функционала, а именно регистрацию и вход пользователей, добавление продукции и категорий, добавление пользователей и отображения. Процесс тестирования веб-приложения «Экомаркет» будет описан в следующей главе.

3. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

3. 1. Тестовый план

3. 1. 1 Test Plan Identifier (идентификатор)

Тестовый план нумеруется как «id_1», уровень тестового плана обозначается как «plan_level_1», уровень программного обеспечения – «websoftware_1». Данный план является мастер-планом, или «Master Plan».

Номер ревизии плана: 1.

Контактная информация главного составителя плана тестирования:

ФИО: Подгорная Ольга Александровна

Email: podornaya.o.a@edu.mirea.ru

Номер телефона: +7 (977) 473-02-69

3. 1. 2 Цель

Целью плана является создание условий тестирования функций программного обеспечения «Веб-приложение «Экомаркет»», улучшение работоспособности системы посредством уменьшения вероятности отказа работы основных модулей программы.

Тестирование объектов необходимо для повышения удовлетворения пользователей веб-сайта от использования данного веб-приложения, а также уменьшение количества ошибок программного обеспечения в процессе эксплуатации [10].

3. 1. 3 Области, подвергаемые тестированию

- 1) Регистрация и авторизация обычных пользователей в системе;
- 2) Добавление, удаление и отображение категорий товаров;
- 3) Добавление, удаление и отображение самих товаров;
- 4) Добавление, удаление и отображение пользователей

3. 1. 4 Области, не подвергаемые тестированию

- 1) Сетевая безопасность сервера;

- 2) Бизнес-аналитика приложения;
- 3) Функция выхода из аккаунта;

3. 1. 5 Тестовая стратегия и подходы

Безопасность будет тестироваться методом «белого ящика», остальное будет проверяться методом «серого ящика».

Уровни тестирования:

- 1) Модульное тестирование

Данный вид тестирования проводится разработчиком и одобряется руководителем, после чего осуществляется системное/интеграционное тестирование.

- 2) Системное/интеграционное тестирование

Проводится менеджером тестирования. Системное/интеграционное тестирование происходит после исправления критических дефектов приложения. Далее следует приемочное тестирование.

- 3) Приемочное тестирование

Проводится конечными пользователями с помощью менеджера тестирования.

3. 1. 6 Критерии

Необходимо, чтобы пользователи с ролью «Пользователь» зарегистрировались в веб-приложении «Экомаркет» и добавили некоторые продукты в корзину. Пользователи с ролью «Администратор» должны добавить новые категории, товары для них и пользователей.

После этого осуществляется проверка количества успешно загруженных товаров и категорий на веб-сайт, а для пользователей – количество успешно добавленных товаров в корзину, причем данные изменения должны корректно отображаться в веб-браузере.

Для каждой тестируемой метрики ее показатели «выполнение тест-кейсов» и «успешное прохождение тест-кейсов» должны быть более 60 процентов и менее 80 процентов соответственно. В этом случае тестирование продолжается.

3. 1. 7 Ресурсы

- 1) Операционная система: macOS Big Sur версии 11.2
- 2) Установлена СУБД PostgreSQL версии 13.1

3. 1. 8 Роли и ответственность

Разработчик является ответственным лицом за выполнение модульного тестирования.

Менеджер тестирования ответственен за выполнение системного/интеграционного тестирования.

Конечные пользователи ответственны за приемочное тестирование.

3. 1. 9 Метрики

- 1) Покрытие требований тест-кейсами – процентное отношения количества покрытых тест-кейсами требований к общему количеству требований;
- 2) Успешное прохождение тест-кейсов – процентное отношение количества успешно выполненных тест-кейсов к общему количеству выполненных тест-кейсов;
- 3) Выполнение тест-кейсов – процентное отношение количества выполненных тест-кейсов к количеству тест-кейсов, запланированных к выполнению.

3. 2. Тест кейсы

В таблице 1 представлен сформированный тест-кейс для тестирования регистрации пользователя в системе. На рисунке 9 представлен внешний вид формы регистрации нового пользователя.

Таблица 1 – Тест-кейс регистрации на веб-сайте

ID: 1	Название: Регистрация на веб-сайте		
R_1	Подгорная О. А.	Высокий приоритет	Модуль: Форма регистрации
<p>Приготовления на начало тестирования:</p> <ol style="list-style-type: none"> 1. Перейти на веб-страницу проекта 2. Нажать на кнопку «Регистрация» навигационной панели 			
<p>Шаги:</p> <ol style="list-style-type: none"> 1. Нажать на поля ввода и ввести туда корректные требуемые данные (имя, фамилию, адрес электронный почты, пароль) 2. Нажать на кнопку «Зарегистрироваться» 		<p>Ожидаемый результат:</p> <ol style="list-style-type: none"> 1. Сообщение об ошибке в случае неправильного ввода адреса электронной почты. 2. В таблицу «users» используемой базы данных добавляется запись о новом пользователе, пароль хранится в хэшированном виде, значение “role” равно «USER» 	
Статус: «Разработано»		Дата выполнения: 22.05.2021	Ссылка на дефекты:

Sign Up

Имя

Фамилия

Электронная почта

Пароль

Рисунок 9 – Форма регистрации нового пользователя

В таблице 2 представлен сформированный тест-кейс добавления, удаления и отображения категорий на веб-странице. На рисунке 10 представлены форма категории и ее отображение в администраторском.

Таблица 2 – Тест-кейс добавления, удаления и отображения категорий

ID: 2	Название: Управление продуктами		
R_2	Подгорная О. А.	Высокий приоритет	Модуль: Добавление, удаление, отображение категорий на веб- странице
Приготовления на начало тестирования: 1. Авторизоваться как администратор 2. Нажать на кнопку «Администрирование» навигационной панели 3. Нажать на кнопку «Управление категориями»			
Шаги:		Ожидаемый результат:	
1. Нажать на поле ввода и ввести туда корректные требуемые		1. Отображение категории в разделе «Управление	

данные (наименование категории)	категориями» после ее добавления	
2. Нажать на кнопку «Добавить»	2. Отображение категории на главной странице после ее добавления	
3. Перейти на главную страницу и проверить отображение категории	3. В таблицу «categories» используемой базы данных добавляется запись о новой категории	
4. Вернуться в раздел «Управление категориями» и нажать кнопку «Удалить».	4. После удаления категории стирается запись о ней в БД	
Статус: «Разработано»	Дата выполнения: 22.05.2021	Ссылка на дефекты:

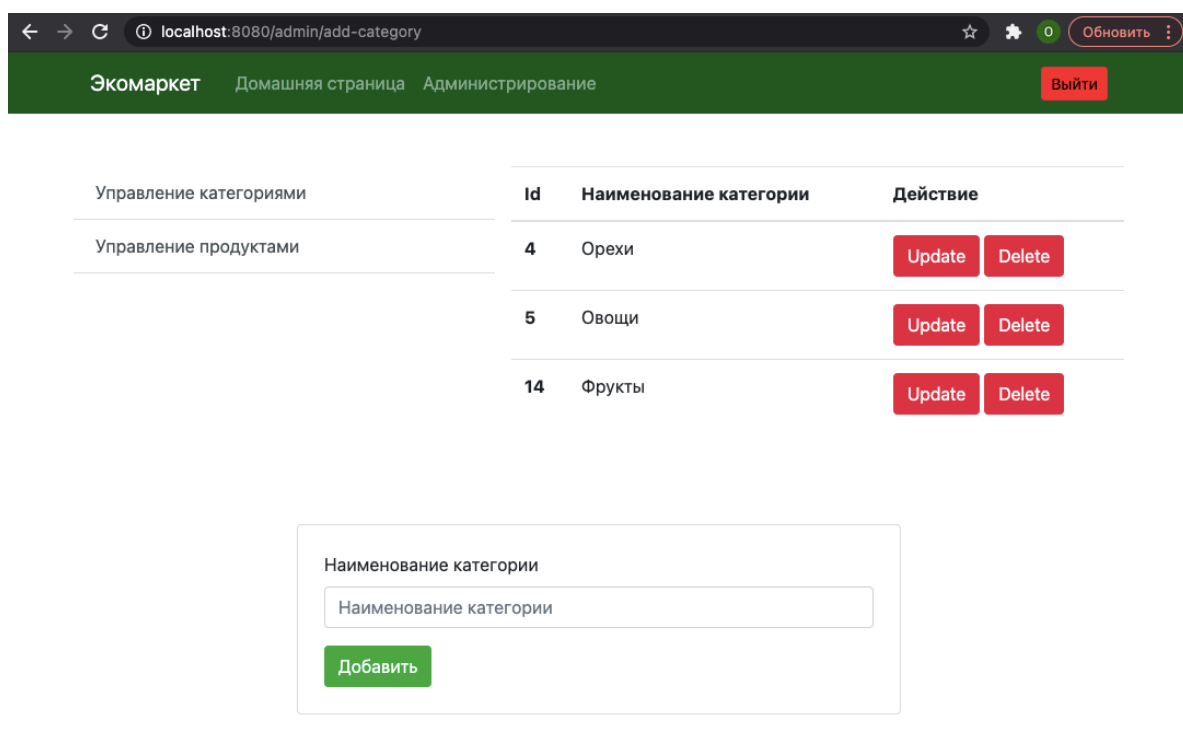


Рисунок 10 – раздел «Управление категориями»

В таблице 3 представлен сформированный тест-кейс добавления, удаления и отображения товаров на веб-странице. На рисунках 11 и 12 представлен раздел «Управление продуктами».

Фрукты

Наименование

Яблоки

Описание

Краснодарские яблоки

Цена

150

Количество

50

Изображение

Выберите файл

яблоки.jpeg

Добавить

Рисунок 11 – форма добавления продукта

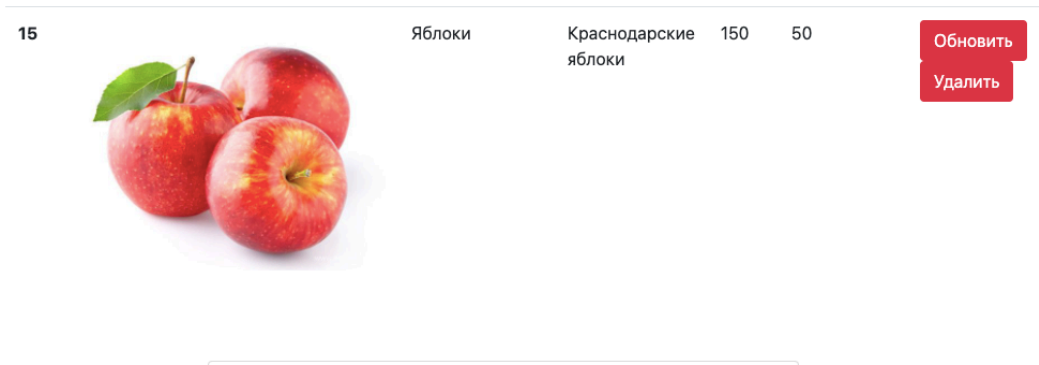


Рисунок 12 – отображение ранее добавленного товара в разделе «Управление продуктами»

Таблица 3 – Тест-кейс добавления, удаления и отображения товаров

ID: 3	Название: Управление продуктами		
R_3	Подгорная О. А.	Высокий приоритет	Модуль: Добавление,

			удаление, отображение товаров на веб- странице
Приготовления на начало тестирования: <ol style="list-style-type: none"> 1. Авторизоваться как администратор 2. Нажать на кнопку «Администрирование» навигационной панели 3. Нажать на кнопку «Управление продуктами» 			
Шаги: <ol style="list-style-type: none"> 1. Нажать на поля ввода и ввести туда корректные требуемые данные (выбрать категорию, наименование, описание, цену, количество, файл-изображение) 2. Нажать на кнопку «Добавить» 3. Перейти на главную страницу и проверить отображение товара 4. Вернуться в раздел «Управление продуктами» и нажать кнопку «Удалить». 		Ожидаемый результат: <ol style="list-style-type: none"> 1. Отображение товара в разделе «Управление продуктами» после его добавления 2. Отображение товара на главной странице после его добавления 3. В таблицу «products» используемой базы данных добавляется запись о новом продукте 4. Изображение сохраняется в директории «/resources/static/images» 5. После удаления товара стирается запись о нем в БД 	
Статус: «Разработано»		Дата выполнения: 22.05.2021	Ссылка на дефекты:

В таблице 4 представлен сформированный тест-кейс добавления, удаления и отображения пользователей на веб-странице. На рисунке 13 представлен внешний вид списка пользователей.

Таблица 4 – Тест-кейс добавления, удаления и отображения товаров

ID: 4	Название: Управление пользователями		
R_4	Подгорная О. А.	Высокий приоритет	Модуль: Добавление, удаление, отображение пользователей на веб-странице
<p>Приготовления на начало тестирования:</p> <ol style="list-style-type: none"> 1. Авторизоваться как администратор 2. Нажать на кнопку «Администрирование» навигационной панели 			
<p>Шаги:</p> <ol style="list-style-type: none"> 1. Нажать на кнопку «Добавить пользователя» 2. Нажать на поля ввода и ввести необходимую информацию (имя, фамилия, адрес электронной почты, пароль, выбрать роль) 3. Нажать на кнопку «Регистрация» 4. Перейти в раздел «Список пользователей и персонала» 5. Нажать кнопку «Удалить» рядом с новым пользователем 		<p>Ожидаемый результат:</p> <ol style="list-style-type: none"> 1. Отображение пользователя в разделе «Список пользователей и персонала» после его регистрации 2. В таблицу «users» используемой базы данных добавляется запись о новом пользователе 3. После удаления пользователя стирается запись о нем в БД 	
Статус: «Разработано»		Дата выполнения: 22.05.2021	Ссылка на дефекты:

Управление категориями					
Управление продуктами					
Добавить пользователя					
Список пользователей и персонала					
Id	Имя	Фамилия	Почта	Роль	Действие
2	Admin	Admin	admin@adm.com	ADMIN	Удалить
10	Test	Test	test@test.ru	USER	Удалить
11	Test1	Test1	test1@test1.ru	USER	Удалить
12	Admin1	Admin1	admin1@adm.com	ADMIN	Удалить

Рисунок 13 – Список пользователей и персонала

Выводы к разделу 3

В данном разделе был составлен тестовый план в соответствии со стандартом «IEEE 829», который позволяет повысить отказоустойчивость и стабильность тестируемого программного продукта. Помимо этого, были продемонстрированы результаты отображения функционала веб-приложения внутри веб-браузера.

ЗАКЛЮЧЕНИЕ

В ходе данной курсовой работы было разработано веб-приложение «Экомаркет» с использованием технологий Spring Framework, Java 14, Spring ORM и других.

Для реализации задачи создания программного продукта был проведен анализ предметной области. Кроме того, был проведен анализ различных технологий разработки, сравнивая которые был сделан выбор инструментария для написания программного кода.

Также был разработан план тестирования для повышения качества работы веб-сервиса и составлены тестовые.

В процессе выполнения курсового проекта были приобретены следующие компетенции:

- ОПК-3 – готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов.
- ПК-13 – готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности.
- ПК-20 – способность оценивать временную и емкостную сложность программного обеспечения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1) Новости ВКонтакте [Электронный ресурс] – Режим доступа: <https://vk.com/press/research-eco-Z> (Дата обращения: 27.04.2021)
- 2) JET BRAINS – Essential tools for software developers and teams [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/help/idea/spring-support.html> (Дата обращения: 27.04.2021)
- 3) Spring.io – Language support [Электронный ресурс] – Режим доступа: <https://docs.spring.io/spring-framework/docs/current/reference/html/languages.html> (Дата обращения: 27.04.2021)
- 4) Gradle – Gradle User Manual [Электронный ресурс] – Режим доступа: https://docs.gradle.org/current/userguide/what_is_gradle.html (Дата обращения: 27.04.2021)
- 5) JavaInUse – developed to help programmers learn and share knowledge [Электронный ресурс] – Режим доступа: <https://www.javainuse.com/spring/spring-design-patterns> (Дата обращения: 21.05.2021)
- 6) YouTrack – Инструмент управления проектами, созданный специально для Agile-команд [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/ru-ru/youtrack/> (Дата обращения: 22.05.2021)
- 7) The Java EE 6 Tutorial [Электронный ресурс] – Режим доступа: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html> (Дата обращения: 22.05.2021)
- 8) Википедия – Свободная энциклопедия [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Hibernate_\(библиотека\)](https://ru.wikipedia.org/wiki/Hibernate_(библиотека)) (Дата обращения: 22.05.2021)
- 9) Владимир Драч – Сравнение современных СУБД [Электронный ресурс] – Режим доступа: <https://drach.pro/blog/hi-tech/item/145-db-comparison> (Дата обращения: 22.05.2021)

- 10) Информационное обеспечение управления качеством [Электронный ресурс]: учебник / А.Г. Схиртладзе [и др.] ; под ред. В.П. Мельникова. — Электрон. дан. — Пенза: ПензГТУ, 2016. — 398 с [Дата обращения 22.05.21].