



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: Приложение «Мясная лавка»

---

Студент: Прокофьев Сергей Яковлевич

Группа: ИКБО-03-19

Работа представлена к защите \_\_\_\_\_ (дата) \_\_\_\_\_ / Прокофьев С.Я. /  
(подпись и ф.и.о. студента)

Руководитель: ст. преподаватель Зорина Н.В.

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: Прокофьев Сергей Яковлевич

Группа: ИКБО-03-19

Срок представления к защите: 25.05.2021

Руководитель: Зорина Наталья Валентиновна

**Тема:** Приложение «Мясная лавка»

**Исходные данные:** индивидуальное задание на разработку; документация по Spring Framework и JEE, документация по языку Java ( версия не ниже 8); инструменты и технологии: JDK (не ниже 8), создание Spring MVC web-приложений, Spring ORM, Gradle, YouTrack, gitHub, IntelliJIDEA. Нормативный документ: инструкция по организации и проведению курсового проектирования СМК 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 проведён 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: \_\_\_\_\_ /С.Я. Прокофьев/, «\_\_\_\_\_» \_\_\_\_\_ 2021 г.

## АННОТАЦИЯ

Курсовой проект содержит 28 страниц отчета, 10 иллюстраций, 8 использованных источников литературы.

Целью курсового проекта является разработка приложения на тему «Мясная лавка» с применением технологий Spring Framework (Spring Boot, Spring ORM, Spring Security), Gradle и JDK.

Структура курсовой работы представлена тремя разделами: введением, основной частью и заключением.

Введение содержит общую характеристику курсовой работы: обосновывается актуальность выбранной темы;

В основной части три раздела:

- В разделе «Сбор и анализ требований к программному продукту» описывается анализ предметной области и существующих аналогов интернет-магазинов «Мясная лавка»
- В разделе «Разработка программного продукта» описываются требования заказчика, технологии и структура программного продукта
- Раздел «Тестирование программного продукта» содержит план тестирования и сами тест кейсы.

В разделе «Заключение» дается оценка результатов, достигнутых при разработке приложения.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	6
1. СБОР И АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ .....	8
1.1. Описание предметной области .....	8
1.2. Анализ существующих аналогов .....	8
Выводы к разделу 1 .....	11
2. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА .....	12
2.1. Проектирование программного продукта .....	12
2.2. Выбор средств и технологии ведения разработки .....	12
2.3. Структура программного продукта .....	13
Выводы к разделу 2 .....	17
3. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА .....	18
3.1. Тестовый план .....	18
3.1.1. Test Plan Identifier (идентификатор) .....	18
3.1.2. Цель .....	18
3.1.3. Области, подвергаемые тестированию .....	19
3.1.4. Области, не подвергаемые тестированию .....	19
3.1.5. Тестовая стратегия и подходы .....	19
3.1.6. Критерии .....	20
3.1.7. Ресурсы .....	20
3.1.8. Роли и ответственность .....	20
3.1.9. Метрики .....	20
3.2. Тест кейсы .....	20
Выводы к разделу 3 .....	25
ЗАКЛЮЧЕНИЕ .....	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	27
Приложение .....	28

## **СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ**

ПО — Программное обеспечение;

JDK — Java Development Kit

JRE — Java Runtime Environment

JVM — Java Virtual Machine

MVC — Model View Controller

ORM — Object-Relational Mapping

JPA — Java Persistence API

API — Application programming interface

БД — База данных;

## **ВВЕДЕНИЕ**

Интернет-магазин является посредником между покупателем и складом производителя, для чего достаточно иметь доступ в интернет. Интернет-магазин не нуждается в офисе, складских помещениях, не несет расходов на содержание помещений, владелец магазина может управлять продажами даже из своего дома. Именно поэтому цены в таких магазинах минимальны. Приобретенный товар доставляется потребителю непосредственно со склада производителя. Но ввиду своей специфики такой магазин имеет ряд недостатков по сравнению с традиционным магазином. Рассмотрим преимущества и недостатки подробнее

Основные преимущества онлайн-магазина:

- Возможность просмотра полного ассортимента товара и услуг в любой момент времени;
- Минимальное контактирование с другими людьми
- Доставка на дом в удобное время, что позволяет покупателю сэкономить время при приобретении товара;
- Более низкая стоимость товаров;
- Отсутствие расходов на содержание помещений;
- Автономность работы магазина;
- Для интернет-магазина не требуется получение различных разрешений и лицензий от специальных служб;
- Возможность обратной связи с клиентами в режиме реального времени независимо от их геолокации.

Основными недостатками интернет-магазина являются:

- Риск мошенничества со стороны непроверенного магазина;
- Невозможность потрогать/понюхать товар
- Возможность плохого качества товара

Таким образом, торговля через интернет-магазин является актуальным и выгодным решением как для покупателя, так и для предпринимателя.

На сегодняшний день, Россия занимает 9-е место в мире по объему потребления мяса. Это по-прежнему самый популярный продукт на столе у жителя страны, который, несмотря на колебания в доходах россиян, продолжает пользоваться высоким спросом.

Именно эти два фактора доказывают актуальность выбранной мною темы "Приложение «Мясная лавка»"

В ходе выполнения курсовой работы должно было быть проведено освоение и углубление профессиональных компетенций, предусмотренных федеральным образовательным стандартом, а именно: ПК-1, в том числе ПК-1.2, ПК-1.12, ПК-1.14.

## **1. СБОР И АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ**

### **1.1. Описание предметной области**

Предметной областью для данной курсовой работы является исследование аналогов интернет-магазинов для собственной реализации приложения «Мясная лавка».

В современном мире, а особенно во время пандемии, люди отдают предпочтение интернет-магазинам. Владельцам содержание интернет-магазина выгоднее, чем обычного магазина, а для покупателей совершать покупки в интернет-магазине намного удобнее и безопасней.

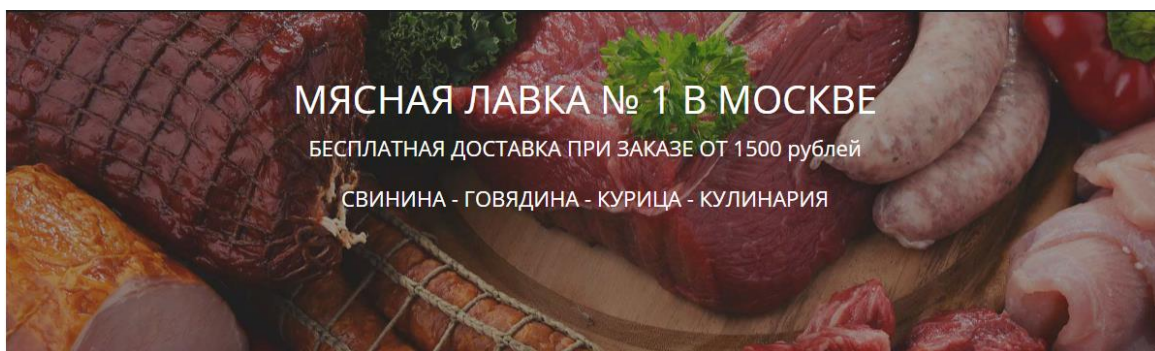
Создав заказ (выбрав желаемые товары), клиент имеет возможность выбрать способ оплаты, время и способ доставки. При оформлении заказа часто есть возможность сообщить дополнительные пожелания продавцу. В интернет-магазинах, рассчитанных на повторные покупки, ведется отслеживание возвратов посетителя и история покупок.

### **1.2. Анализ существующих аналогов**

В настоящее время внутри российского сегмента интернета можно найти множество специализированных на продаже мясной продукции интернет-магазинов. Проанализировав некоторое их количество, я выделил общие черты, которые показывают, что должно быть в моем веб-сервисе.

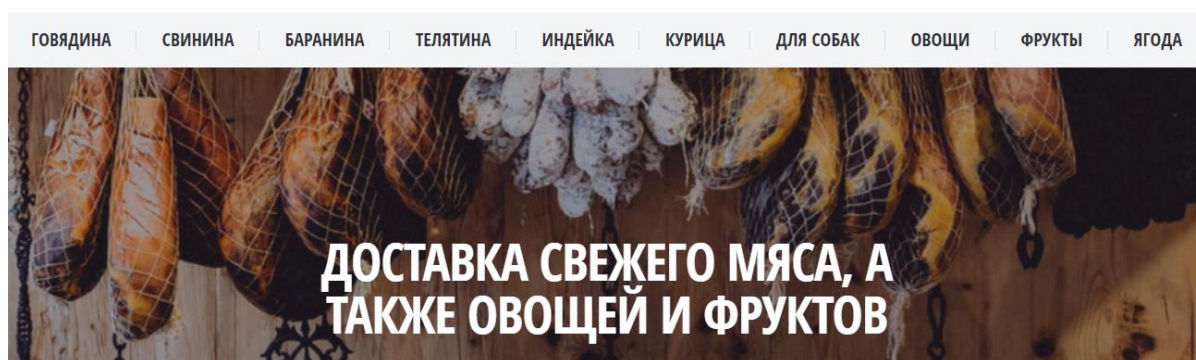
При входе на сайт первым делом бросаются в глаза фотографии мясных изделий и акции, которые предоставляет магазин покупателям (*Рисунок 1.1*). Кликнув на акцию, пользователь сможет ознакомиться с ней подробнее.





*Рисунок 1.1 – Интерфейс интернет-ресурса «myaso-1»<sup>[1]</sup>*

Также на сайтах присутствует панель выбора видов мяса (*Рисунок 1.2*). Благодаря ей пользователь может быстро перемещаться между разными мясными изделиями и сравнивать цены, что позволяет выбрать наиболее подходящее для него мясо.



*Рисунок 1.2 - Интерфейс интернет-ресурса «еситеат»<sup>[2]</sup>*

Перейдем к странице, на которой пользователь может выбрать товары (*Рисунок 1.3*). Здесь клиент может выбрать понравившийся ему товар, положить его в корзину и продолжить поиск товаров. При нажатии на товар можно прочитать подробную информацию о нем.










		
<u><b>Рибай</b></u> <u><b>Wagyu (Prime) A4</b></u>	<u><b>Стейк Фланк</b></u> <u><b>Праймбиф (Prime)</b></u>	<u><b>Стейк Портерхаус</b></u> <u><b>Праймбиф (Prime)</b></u>
Япония А4, кукурузный откорм. Охлажденное мясо.	Воронеж, кукурузный откорм. Охлажденное мясо.	Воронеж, кукурузный откорм. Охлажденное мясо.
<b>24 270 руб. / 1 кг.</b>	<b>1 290 руб. / 1 кг.</b>	<b>3 990 руб. / 1 кг.</b>
Средний вес: 1.5-2.0 кг	Средний вес: 0.9-1.8 кг	Средний вес: 0.7-1.0 кг
 <a href="#">В корзину</a>	 <a href="#">В корзину</a>	 <a href="#">В корзину</a>

Рисунок 1.3 - Интерфейс интернет-ресурса «Steak home»<sup>[3]</sup>

В правом верхнем углу обычно можно увидеть кнопку корзины, в которой лежат выбранные ранее товары. Перейдя на нее (Рисунок 1.4) можно увидеть итоговую цену с учетом скидок (при их наличии). Здесь же можно удалить товары, которые пользователь передумал покупать.

## КОРЗИНА

[История заказов](#)

	Оссобуко из телятины <b>690 Р/кг</b>	— 0.6 кг +	<b>414 Р</b>	×
	Каре ягнёнка 8 рёбер отборные <b>2 245 Р/кг</b> <input type="checkbox"/> Замариновать для гриля + 60 Р/кг	— 0.4 кг +	<b>898 Р</b>	×
	Шницель из телячьего филе <b>1 490 Р/кг</b> <input type="checkbox"/> Сделать без соли <input type="checkbox"/> Сделать без специй	— 0.75 кг +	<b>1 117.50 Р</b>	×

Товаров 4 шт / 2.75 кг

Промокод ▾

Доставка Доставка курьером **350 Р**  
в пределах МКАД, Москва

Оплата Наличными

**ИТОГО 4 729.50 Р**

**ОФОРМИТЬ ЗАКАЗ**

Нажимая кнопку, вы даете согласие на обработку ваших персональных данных. Политика конфиденциальности

Рисунок 1.4 - Интерфейс интернет-ресурса «primeteat»<sup>[4]</sup>

## **Выводы к разделу 1**

Проведя анализ, можно сделать вывод, что основной функционал приложения «Мясная лавка» должен заключаться в возможности выбора продукции и оформления заказа онлайн.

Учитывая особенности рассмотренных интернет-ресурсов. Можно выделить ряд основных требований к разрабатываемому проекту:

1. Должна быть функция добавления товаров в корзину;
2. Должны быть функции работы с корзиной;
3. Должны быть функции расчета скидок
4. Должны быть функции регистрации и авторизации клиента;
5. Должна быть навигация и понятный интерфейс.

## **2. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА**

### **2.1. Проектирование программного продукта**

После формирования списка общих требований в анализе предметной области необходимо сформулировать список требований заказчика, которые будут отражать его видение проекта и необходимость тестирования функционала.

Список основных требований заказчика (функции, которые должны быть в ПО и быть протестированы):

1. Добавление продукта в корзину;
2. Удаление продукта из корзины;
3. Получение информации о конкретном продукте в магазине;
4. Оформление заказа;
5. Регистрация и авторизации пользователей в системе;

Данные функции могут быть разбиты на подфункции, например, тестирование регистрации и авторизация, т.к. они требуют от программного обеспечения проверять данные, введенные вручную.

### **2.2. Выбор средств и технологии ведения разработки**

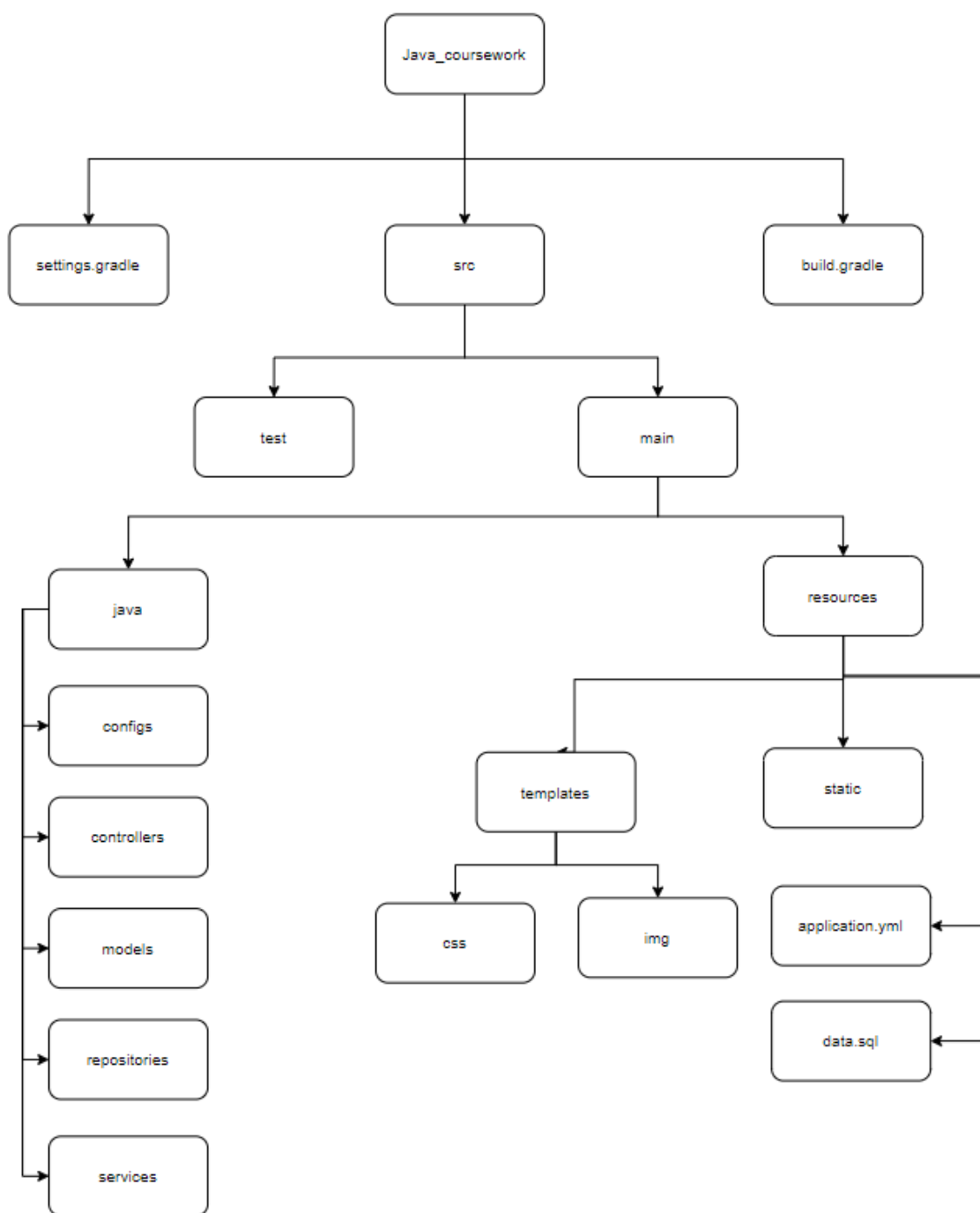
В курсовом проектировании использованы следующие технологии и веб-средства:

1. JDK — это реализация спецификации платформы Java, включающая в себя компилятор и библиотеки классов. Он позволяет разработчикам создавать программы, которые могут выполняться и запускаться посредством JVM и JRE;
2. Spring MVC — веб-фреймворк Spring<sup>[5]</sup>, позволяет создавать веб-сайты, поддерживает контроллеры и REST контроллеры в Spring Boot приложениях.

3. Spring ORM — это технология для отображения объектов в бд, ну т.е. чтобы представить наш джава-объект в виде строки таблицы.
4. Spring Data JPA — это спецификация Java для доступа, управления и хранения данных между классами или объектами Java и реляционной бд.
5. GitHub — сервис онлайн-хостинга репозитория, обладающий всей функциональностью управления исходным кодом и функциями распределённого контроля версий.
6. PostgreSQL — Это свободно распространяемая объектно-реляционная система управления базами данных, самая передовая из открытых систем управления бд в мире и реальная альтернатива коммерческим бд.
7. JUnit — библиотека для модульного тестирования программ Java.
8. Gradle<sup>[6]</sup> — система автоматизации сборки с открытым исходным кодом, построенная на основе принципов Apache Maven и Apache Ant.
9. IntelliJ IDEA — умный редактор кода. Он обладает интуитивно понятным и простым интерфейсом для взаимодействия. Ее позиционируют как самую умную среду разработки для Java с поддержкой всех последних технологий и фреймворков.

### **2.3. Структура программного продукта**

Проект состоит из 2 модулей: модуль main с кодом приложения и модуль test с тестированием приложения (Рисунок 2.1).



*Рисунок 2.1 - Структура программного продукта*

В директории `resources` находятся `data.sql` (для первичного заполнения бд), `application.yml` с настройками приложения и еще две директории `static` с `html` файлами и `templates` с изображениями (в директории `img`) и файлами `css` (в директории `css`).

Директории `java` содержит в себе файлы реализации серверной части приложения.

- В директории configs находится конфигурация приложения
- Директория controllers содержит в себе четыре контроллера для работы с корзиной (Рисунок 2.2), регистрацией, товаром и остальных страниц

```

@Controller
public class BasketController {

    @Autowired
    private BasketService basketService;

    @GetMapping("/basket")
    @ResponseBody
    public ModelAndView getBasket(){
        ModelAndView mav = new ModelAndView( "basket");
        List<Meat> meats = basketService.getSetMeat(basketService.getMeats());
        meats.sort(Comparator.comparing(Meat::getId));
        mav.addObject( "meat", meats);
        mav.addObject( "num", basketService.getCounts(meats, basketService.getMeats()));
        mav.addObject( "price", basketService.getPrice(basketService.getMeats()));
        mav.addObject( "discount", (int) Math.round(basketService.getStock(basketService.getMeats())));
        return mav;
    }

    @GetMapping("/basket/delete/{name}")
    public String deleteBasket(@PathVariable String name){
        basketService.deleteMeat(name);
        return "redirect:/basket";
    }

    @GetMapping("/basket/buy")
    public String deleteBasket(){
        basketService.buy();
        return "redirect:/";
    }
}

```

*Рисунок 2.2 - Листинг BasketController.java*

- В директории models находятся сущности товара(Рисунок 2.3), пользователя и его роли. Они связаны с помощью аннотации ManyToMany.

```

@Entity
@Table(name = "meat")
@Getter
@Setter
public class Meat {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    @Column(name = "sort")
    String sort;
    @Column(name = "name")
    String name;
    @Column(name = "image")
    String image;
    @Column(name = "weight")
    int weight;
    @Column(name = "price")
    int price;
    @Column(name = "description")
    String description;
    @Column(name = "quantity")
    int quantity;
    @ManyToMany(mappedBy = "meats")
    private Set<User> baskets=new HashSet<>();
}

```

*Рисунок 2.3 - Листинг Meat.java*

- В repositories лежат два репозитория, для работы с базой данных
- В директории services находятся три сервиса для работы с корзиной, товаром (Рисунок 2.4) и пользователем

```

@Service
@RequiredArgsConstructor
public class MeatService {

    @Autowired
    private MeatRepository meatRepository;

    @Transactional(readOnly = true)
    public List<Meat> getBySort(String sort) {
        List<Meat> meats = meatRepository.findAllBySort(sort);
        meats.sort(Comparator.comparing(Meat::getId));
        return meats;
    }

    @Transactional(readOnly = true)
    public Meat getByName(List<Meat> meats, String name){
        for (Meat meat : meats) {
            if (meat.getName().equals(name))
                return meat;
        }
        return null;
    }
}

```

*Рисунок 2.4 - Листинг MeatService.java*



## Выводы к разделу 2

В результате анализа требований заказчика к программному обеспечению были сделаны выводы о необходимых техниках тестирования, которые позволят выявить слабые места ПО, а также продемонстрировать корректность его работы.

Была составлена структура проекта, определена связь между классами (Рисунок 2.5) в серверной части приложения и составлена схема бд (Рисунок 2.6), для оптимального хранения информации

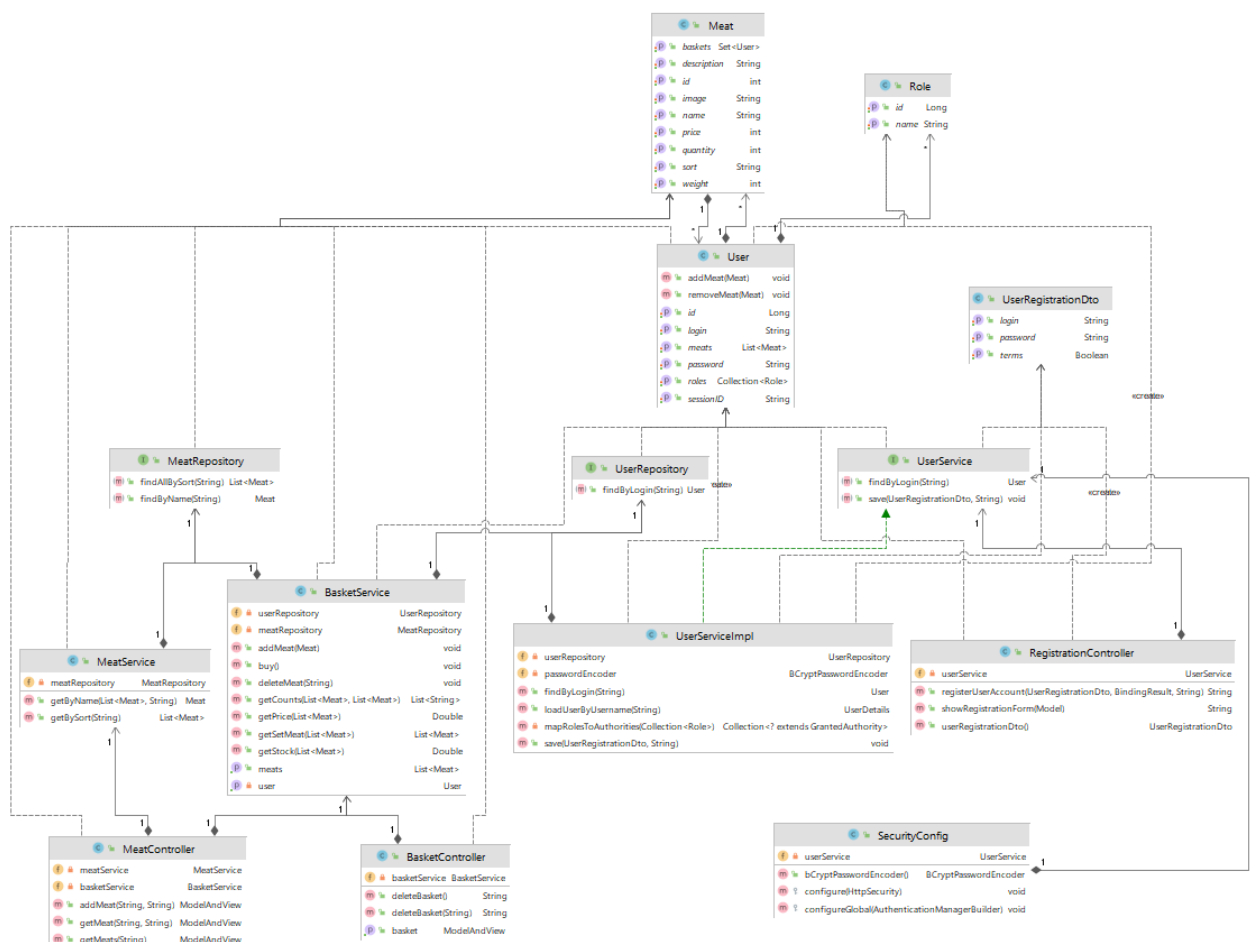


Рисунок 2.5 - Диаграмма классов

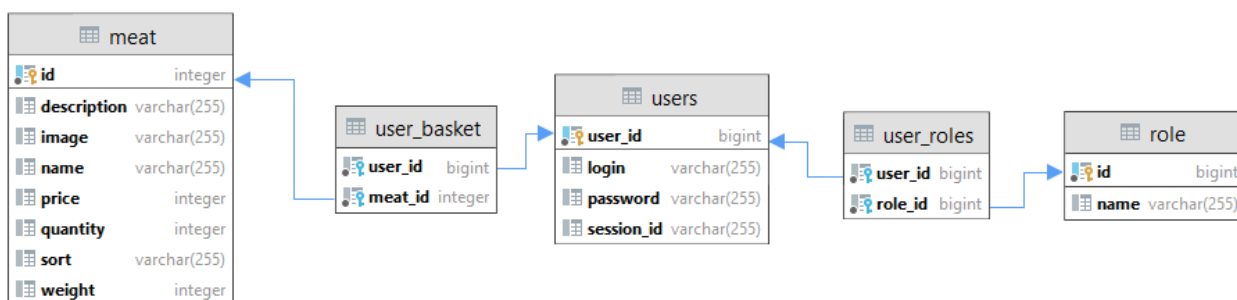


Рисунок 2.6 - Схема БД

Были проанализированы техники тестирования, и из них были выбрана техника «формирование чек-листов или тест-кейсов» для тестирования взаимодействия пользователя с системой веб-приложения.

### 3. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

В этом разделе описывается тестирование программного продукта.

#### 3.1. Тестовый план

Тестовый план составляется по стандарту «IEEE 829» с удалением нескольких пунктов из-за ограниченности проводимого тестирования.

##### 3.1.1. Test Plan Identifier (идентификатор)

Тестовый план нумеруется как «id\_1», уровень тестового плана - «plan\_level\_1», уровень программного обеспечения - «websoftware\_1». План является мастер-планом («Master Plan»).

Номер ревизии плана: 1.

Контактная информация главного составителя плана тестирования:

ФИО: Прокофьев Сергей Яковлевич.

Email: serg.prok477@gmail.com

##### 3.1.2. Цель

Целью плана является создание условий тестирования методов API «веб-приложения «Мясная лавка»», улучшение работоспособности системы, посредством проверки корректности работы функционала серверной части приложения.

Тестирование объектов тестирования необходимо для проверки корректности взаимодействия пользователя и приложения, уменьшение количества ошибок ПО в процессе эксплуатации.

### **3.1.3. Области, подвергаемые тестированию**

1. Регистрация пользователей в системе;
2. Авторизации пользователей в системе
3. Добавление продукта в корзину;
4. Получение информации о конкретном продукте в магазине;
5. Удаление продукта из корзины;
6. Оформление заказа;

### **3.1.4. Области, не подвергаемые тестированию**

1. Сетевая безопасность сервера;
2. Бизнес-аналитика приложения;
3. Функция выхода из аккаунта;
4. Функция получения скидки;

### **3.1.5. Тестовая стратегия и подходы**

Тестирование безопасности будет проходить методом «белого ящика». Остальное будет проверяться методом «серого ящика» (или «чёрного ящика», если доступа к ПО нет).

Уровни тестирования:

#### **1) Функциональное тестирование**

Проводится разработчиком и одобряется руководителем. После одобрения данного вида тестирования, осуществляется приемочное тестирование.

#### **2) Приемочное тестирование**

Проводится конечными пользователями (студентами и преподавателями) с помощью менеджера тестирования.

### 3.1.6. Критерии

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

### 3.1.7. Ресурсы

- Операционная система: Linux x64.
- Установленная СУБД PostgreSQL

### 3.1.8. Роли и ответственность

Программист ответственен за выполнение функционального тестирования.

Конечные пользователи ответственны за приёмочное тестирование.

### 3.1.9. Метрики

- Покрытие требований тест-кейсами – процентное отношение количества покрытых тест-кейсами требований к общему количеству требований.
- Успешное прохождение тест-кейсов – процентное отношение количества успешно выполненных тест-кейсов к общему количеству выполненных тест-кейсов.
- Выполнение тест-кейсов – процентное отношение количества выполненных тест-кейсов к количеству тест-кейсов, запланированных к выполнению.

## 3.2. Тест кейсы

В таблице 3.1 представлен сформированный тест-кейс для тестирования регистрации пользователя в системе.

Таблица 3.1 – Тест-кейс для тестирования регистрации на сайте

ID: 1	Название: Регистрация на сайте
-------	--------------------------------

R_1	Прокофьев С.Я.	Высокий приоритет	Модуль: Форма регистрации
Приготовления на начало тестирования: <ol style="list-style-type: none"> <li>1. Перейти на сайт проекта и нажать на кнопку «Войти» в верхней панели.</li> <li>2. Нажать на ссылку «Зарегистрируйся»</li> </ol>			
Шаги: <ol style="list-style-type: none"> <li>1. Нажать на поля ввода и ввести туда корректные требуемые данные (логин, пароль).</li> <li>2. Нажать на кнопку «Зарегистрироваться».</li> </ol>		Ожидаемый результат: <ol style="list-style-type: none"> <li>1. Появление текста «Этот логин занят» при вводе уже существующего логина.</li> <li>2. Если логин свободен (иначе вернуться к пункту 1 и заменить неверные данные), в БД добавляется запись в таблицу «users» с ролью «USER».</li> </ol>	
Статус «Разработано»		Дата выполнения: 22.05.2021	Ссылка на дефекты:

В таблице 3.2 представлен сформированный тест-кейс для тестирования авторизации пользователя в системе.

Таблица 3.2 – Тест-кейс для тестирования авторизации на сайте

ID: 2	Название: Авторизация на сайте		
R_2	Прокофьев С.Я.	Высокий приоритет	Модуль: Форма входа
Приготовления на начало тестирования: <ol style="list-style-type: none"> <li>1. Зарегистрировать свой аккаунт.</li> <li>2. Нажать на ссылку «Вход»</li> </ol>			

Шаги:  1. Нажать на поля ввода и ввести туда корректные требуемые данные (логин, пароль).  2. Нажать на кнопку «Вход».	Ожидаемый результат:  1. Появление уведомления «Неправильный логин или пароль.» при вводе неверных данных.  2. Если логин и пароль верны (иначе вернуться к пункту 1 и заменить неверные данные), пользователь входит в свой аккаунт и перенаправляется на главную страницу.	
Статус «Разработано»	Дата выполнения: 22.05.2021	Ссылка на дефекты:

В таблице 3.3 представлен сформированный тест-кейс для тестирования добавления продукта в корзину;

Таблица 3.3 – Тест-кейс для тестирования добавления продукта в корзину

ID: 3	Название: Добавление продукта в корзину		
R_3	Прокофьев С.Я.	Высокий приоритет	Модуль: Каталог продуктов
Приготовления на начало тестирования:  1. Авторизоваться в учетную запись клиента.  2. Нажать на кнопку «Лавка» в верхней панели.			
Шаги:  Нажать на кнопку «Купить» под понравившимся товаром	Ожидаемый результат:  При нажатии на кнопку, происходит поиск товара, получение авторизованного пользователя и дальнейшее добавление товара в корзину		

	пользователя. После этого появляется текст «Добавлен в корзину» под кнопкой.	
Статус «Разработано»	Дата выполнения: 22.05.2021	Ссылка на дефекты:

В таблице 3.4 представлен сформированный тест-кейс для тестирования получения информации о конкретном продукте в магазине.

Таблица 3.4 – Тест-кейс для тестирования получения информации о конкретном продукте в магазине

ID: 4	Название: Получения информации о конкретном продукте		
R_4	Прокофьев С.Я.	Высокий приоритет	Модуль: Каталог продуктов
Приготовления на начало тестирования: 1. Нажать на кнопку «Лавка» в верхней панели.			
Шаги: Нажать на картинку товара		Ожидаемый результат: При нажатии на картинку, происходит поиск товара и перенаправление пользователя на страницу с подробной информацией о нем	
Статус «Разработано»	Дата выполнения: 22.05.2021	Ссылка на дефекты:	

В таблице 3.5 представлен сформированный тест-кейс для тестирования удаления продукта из корзины.

Таблица 3.5 – Тест-кейс для тестирования удаления продукта из корзины

ID: 5	Название: Удаление продукта		
R_5	Прокофьев С.Я.	Высокий приоритет	Модуль: Корзина
Приготовления на начало тестирования: <ol style="list-style-type: none"> <li>1. Авторизоваться в учетную запись клиента.</li> <li>2. Добавить в корзину товар.</li> <li>3. Нажать на кнопку «Корзина» в верхней панели.</li> </ol>			
Шаги: Нажать на крестик перед товаром		Ожидаемый результат: При нажатии на кнопку, происходит поиск товара, получение авторизованного пользователя и дальнейшее удаление товара из корзины.	
Статус «Разработано»		Дата выполнения: 22.05.2021	Ссылка на дефекты:

В таблице 3.6 представлен сформированный тест-кейс для тестирования оформления заказа.

Таблица 3.6 – Тест-кейс для тестирования оформления заказа

ID: 6	Название: Оформление заказа		
R_6	Прокофьев С.Я.	Высокий приоритет	Модуль: Корзина
Приготовления на начало тестирования: <ol style="list-style-type: none"> <li>1. Авторизоваться в учетную запись клиента.</li> <li>2. Добавить в корзину товар.</li> <li>3. Нажать на кнопку «Корзина» в верхней панели.</li> </ol>			



<p>Шаги:</p> <p>Нажать на кнопку купить</p>	<p>Ожидаемый результат:</p> <p>При нажатии на кнопку происходит получение авторизованного пользователя и удаление всех товаров из корзины.</p> <p>Появляется уведомление «Покупка успешна» и перенаправляет пользователя на главную страницу</p>	
<p>Статус «Разработано»</p>	<p>Дата выполнения:</p> <p>22.05.2021</p>	<p>Ссылка на дефекты:</p>

### **Выводы к разделу 3**

В данном разделе был составлен тестовый план в соответствии со стандартом «IEEE 829», который позволит повысить отказоустойчивость и стабильность тестируемого программного продукта.

## ЗАКЛЮЧЕНИЕ

В результате выполнения данного курсового проекта была достигнута основная поставленная цель - разработка веб-приложения на тему «Мясная лавка». На основании требований был подготовлен веб-сайт с необходимым согласно теме контентом.

Приложение устанавливает связь с базой данных, считывает информацию и наполняет веб-страницы. Оно способно изменять таблицы и добавлять в них новые строки.

Были протестированы требования заказчика, изучены техники тестирования и выбраны те их них, которые позволяют провести наиболее качественное и быстрое тестирование веб-системы.

В процессе разработки курсовой работы были приобретены следующие компетенции:

- ПК-1 – способен выполнять разработку и интеграцию программных модулей и компонент, верификацию выпусков программных продуктов информационных систем;
- ПК-1.1 – знание методов и средств сборки модулей и компонентов программного обеспечения при создании информационных систем
- ПК-1.12 – умение применять методы и средства создания программных интерфейсов информационных систем;
- ПК-1.14 – владение разработкой процедур сборки модулей и компонент программного обеспечения при внедрении информационных систем;

Все требования к курсовой работе, указанные в техническом задании, реализованы в полной мере и работают без ошибок. Исходя из выполнения всех тезисов, указанных в начале, можно считать работу выполненной успешно.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Интернет-магазин Myaso-1 [Электронный ресурс]: Режим доступа: <https://myaso-1.ru/>, свободный (Дата последнего обращения: 17.04.2021)
2. Интернет-магазин Эку [Электронный ресурс]: Режим доступа: <https://ecumeat.ru/>, свободный (Дата последнего обращения: 17.04.2021)
3. Интернет-магазин Steak home [Электронный ресурс]: Режим доступа: <https://steakhome.ru/>, свободный (Дата последнего обращения: 17.04.2021)
4. Интернет-магазин Prime meat [Электронный ресурс]: Режим доступа: <https://primemeat.ru/>, свободный (Дата последнего обращения: 17.04.2021)
5. Документация Spring [Электронный ресурс]: Режим доступа: <https://spring.io/>, свободный (Дата последнего обращения: 14.05.2021)
6. Документация Gradle [Электронный ресурс]: Режим доступа: <https://gradle.org/>, свободный (Дата последнего обращения: 13.05.2021)
7. Докука, О. Практика реактивного программирования в Spring 5 / О. Докука, И. Лозинский. — Москва : ДМК Пресс, 2019. — 508 с. — ISBN 978-5-97060-747-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131708> (дата обращения: 01.05.2021). — Режим доступа: для авториз. пользователей.
8. Алпатов, А. Н. Тестирование и отладка программного обеспечения : методические указания / А. Н. Алпатов. — Москва : РТУ МИРЭА, 2020. — 40 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/167578> (дата обращения: 15.05.2021). — Режим доступа: для авториз. пользователей.

## ПРИЛОЖЕНИЕ

Исходный код данного веб-приложения, включающий в себя все файлы, является слишком объемным, прикреплять его в качестве приложения неуместно, можно ознакомиться с ним перейдя по ссылке:  
[https://github.com/Keeper477/Java\\_coursework](https://github.com/Keeper477/Java_coursework)