

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерного проектирования
Кафедра проектирования информационно-компьютерных систем
Дисциплина «Объектно-ориентированное проектирование и
программирование»

«К ЗАЩИТЕ ДОПУСТИТЬ»
Руководитель курсового проекта
ассистент
_____ А.Д. Ларькин
____.____.2023

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему:
«ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ КАФЕ»

БГУИР КП 1-58 01 01 018 ПЗ

Выполнил студент группы 210901
Лойко Алеся Игоревна

(подпись студента)
Курсовой проект представлен на
проверку _____.____.2023

(подпись студента)

Минск 2023

РЕФЕРАТ

БГУИР КП 1-58 01 01 018 ПЗ

Караваяев М.И. Веб-приложение для автоматизации работы кафе: пояснительная записка к курсовому проекту / М.И.Караваяев – Минск: БГУИР, 2023. – 31 с.

Пояснительная записка 31 с., 13 рис., 10 источников, 3 приложений

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ КАФЕ

Цель проектирования: создать приложение для автоматизации просмотра меню кафе и удаленного заказа блюд.

Методология проведения работы: в процессе решения поставленных задач использованы принципы системного подхода, парадигма Объектно-ориентированного программирования, шаблон архитектуры системы *MVVM*, принципы *SOLID*, паттерны проектирования программного обеспечения.

Результаты работы: в результате работы был выполнен анализ рынка в сфере сайтов станций технического обслуживания, рассмотрено общетехническое обоснование разработки программного средства; сделан общий анализ программного средства; рассчитаны параметры проектируемого программного средства; спроектировано и реализовано веб-приложение; проведено тестирование веб-приложения; разработана графическая часть проекта.

Программное средство обеспечивает автоматизацию просмотра меню и предоставления возможности заказа еды в цифровой форме.

Структура предполагает разделение ролей администратора и покупателя. Пользователь-покупатель может самостоятельно выбрать блюда, оформить заказ. Пользователь-администратор может редактировать информацию о блюдах.

Область применения результатов: могут быть использованы при цифровизации деятельности кафе.

СОДЕРЖАНИЕ

Перечень условных обозначений, символов и терминов	6
Введение	7
1. Анализ исходных данных на курсовое проектирование	8
1.1 Анализ исходных данных к курсовому проекту	8
1.2 Обоснование и описание выбора языка программирования, средств разработки, используемых технологий и сторонних библиотек	11
2. Проектирование и разработка программного средства	13
2.1 Проектирование архитектуры и описание состояний программного средства	13
2.2 Проектирование и разработка графического интерфейса	14
2.3 Описание и реализация используемых в программном средстве алгоритмов	15
3. Эксплуатация программного средства	16
3.1 Ввод в эксплуатацию и обоснование минимальных технических требований к оборудованию	16
3.2 Руководство по эксплуатации программного средства	17
Заключение	22
Список использованных источников	23
Приложение А (обязательное) Листинг программного кода	24
Приложение Б (обязательное) Справка о проверке курсового проекта на плагиат	30
Приложение В (обязательное) Ведомость документов	31

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ

База данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.

Объектно-ориентированное программирование (в дальнейшем ООП) – парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

Среда разработки – комплекс программных средств, используемый программистами для разработки программного обеспечения типов.

ООП – Объектно-ориентированное программирование;

JVM – (*Java Virtual Machine*) виртуальная машина Java;

PostgreSQL – это свободно распространяемая объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом, написанном на языке C;

Intelij IDEA – интегрированная среда разработки;

Java FX – графический интерфейс пользователя;

ОС – Операционная система;

ПО – Программное обеспечение;

ЯП – Язык программирования.

Использование условных обозначений и терминов является неотъемлемой частью разработки. Термины, представленные в перечне, являются ключевыми для достижения эффективности и функциональности веб-приложения. Понимание этих терминов обеспечивает ясность в коммуникации и успешное выполнение поставленных задач проекта.

ВВЕДЕНИЕ

Пищевая отрасль является неотъемлемой частью жизни нашего общества. Каждому человеку необходимо кушать, кто-то готовит еду самостоятельно из продуктов, покупаемых в магазине, кто-то ходит в кафетерии, рестораны или пиццерии.

Конечно, заведения позволяют комфортно заказывать себе блюда: где-то по залу ходят официанты, где-то меню лежит на каждом столике, где-то есть электронные экраны для заказа. Часто заведения расположены в местах скопления людей – возле метро, в центре города, – и потому людям удобно приходить. Однако часто у человека нет возможности или времени для посещения кафе – для таких случаев рестораны часто организуют доставку еды, которую можно заказать прямо из дома – по телефону или через сайт.

Цель данной курсовой работы заключается в создании веб-сервиса для кафе. Для достижения этой цели предполагается решить следующие задачи:

Разработка пользовательского интерфейса, обеспечивающего удобный доступ и манипулирование данными о блюдах.

Обеспечение безопасности данных и конфиденциальности пользовательской информации при использовании приложения.

Проведение тестирования и оптимизации приложения для обеспечения высокой производительности и стабильной работы.

Детальная постановка задачи включает в себя проектирование архитектуры приложения с использованием выбранных технологий, создание базы данных для хранения финансовых данных, разработку модулей функционала и их интеграцию в рамках единой системы, а также тестирование приложения на различных этапах разработки для обеспечения его корректной работы и соответствия заявленным требованиям. Курсовой проект выполнен самостоятельно, проверен в системе «Антиплагиат».

Процент оригинальности составляет 90,43%. Цитирования обозначены ссылками на публикации, указанными в «Списке использованных источников».

1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

1.1 Анализ исходных данных к курсовому проекту

Веб-сервис обеспечивает возможность оформления заказа пользователям без необходимости очного посещения кафе. Для хороших условий для заказа пользователю необходимо видеть интересующую его информацию: внешний вид (фотографию) блюда, название блюда, цену, состав и размер порции. Также у пользователя должна быть возможность узнать информацию о кафе: расположение (адрес), время работы, рабочий телефон и соцсети. Чтобы покупатель смог оформить заказ на сайте, у него должен быть свой личный кабинет, в нем можно добавлять блюда в избранное или корзину и оформлять заказ.

На данный момент уже существует некое количество веб-сайтов различных ресторанов и кафе.

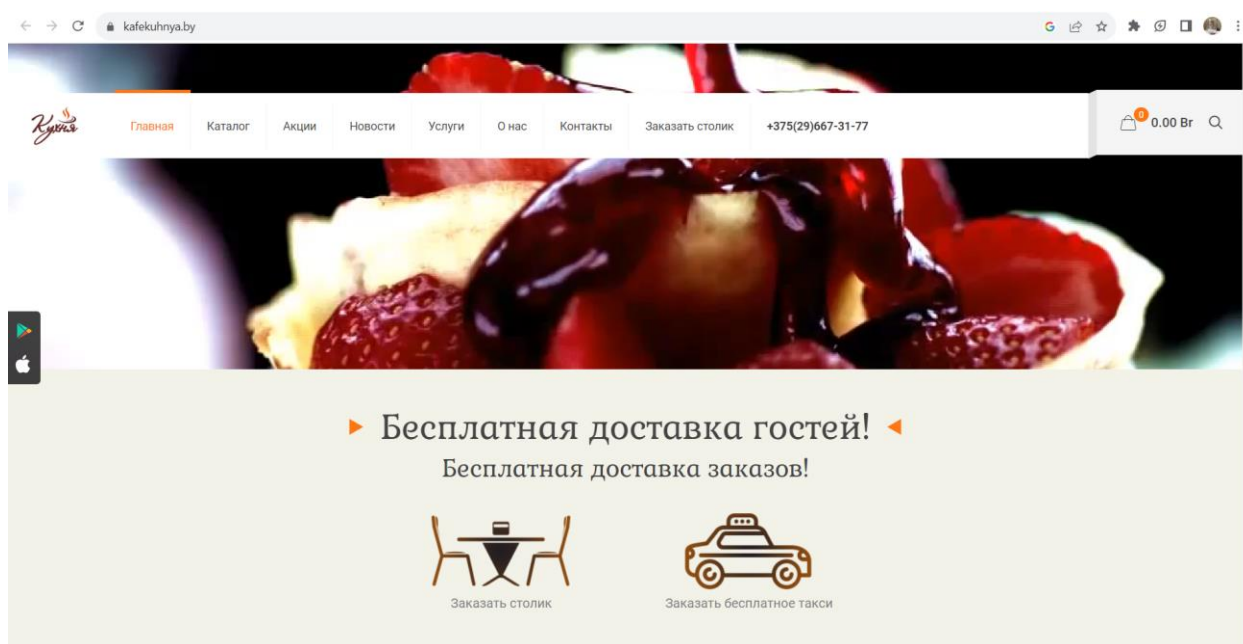


Рисунок 1.1 – Сайт кафе «Кухня»

Например, на сайте кафе «Кухня» из отличительных особенностей имеется сразу баланс на счете посетителя. Это удобно, т.к. пользователю не нужно заходить в личный кабинет и вспоминать свой пароль, но для сайта кафе «Огнище» такое решение не подойдет, т.к. на нем есть возможность зайти как администратор.

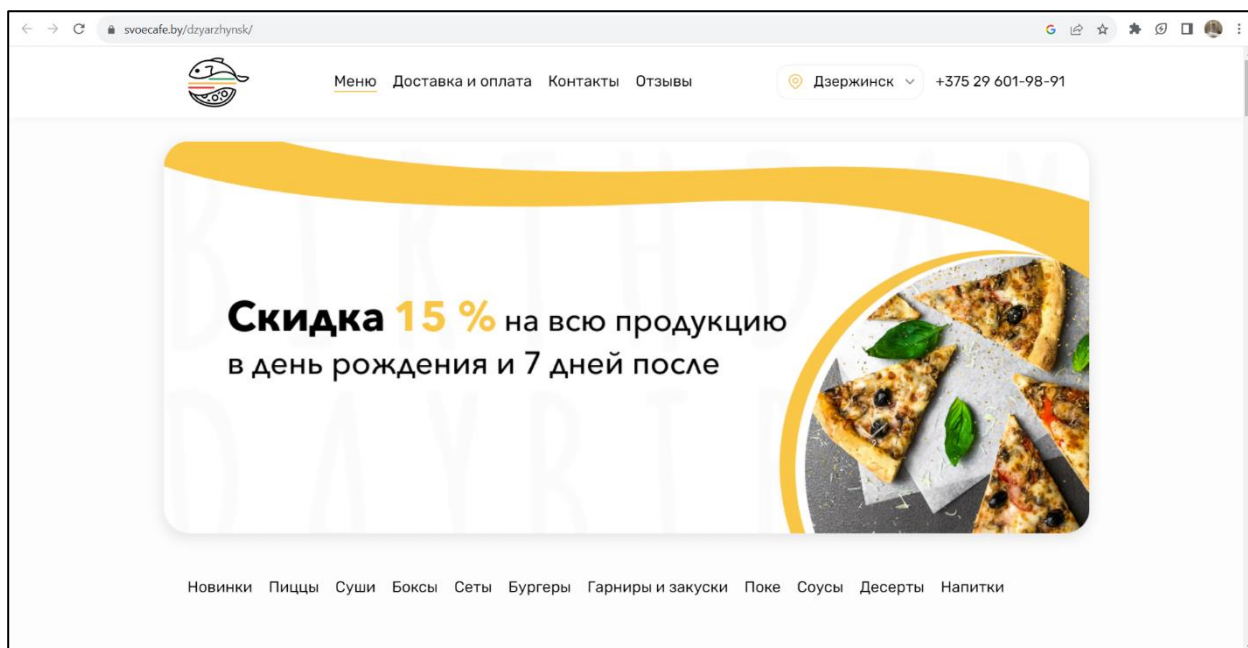


Рисунок 1.2 – Сайт кафе «Свое кафе»

На большинстве сайтов кафетериев и ресторанов, в том числе и на сайте «Свое кафе», навигация по сайту находится в виде полосы в верхней части экрана. На сайте «Огнище» она будет расположена так же, поскольку такое расположение является привычным для большинства пользователей.

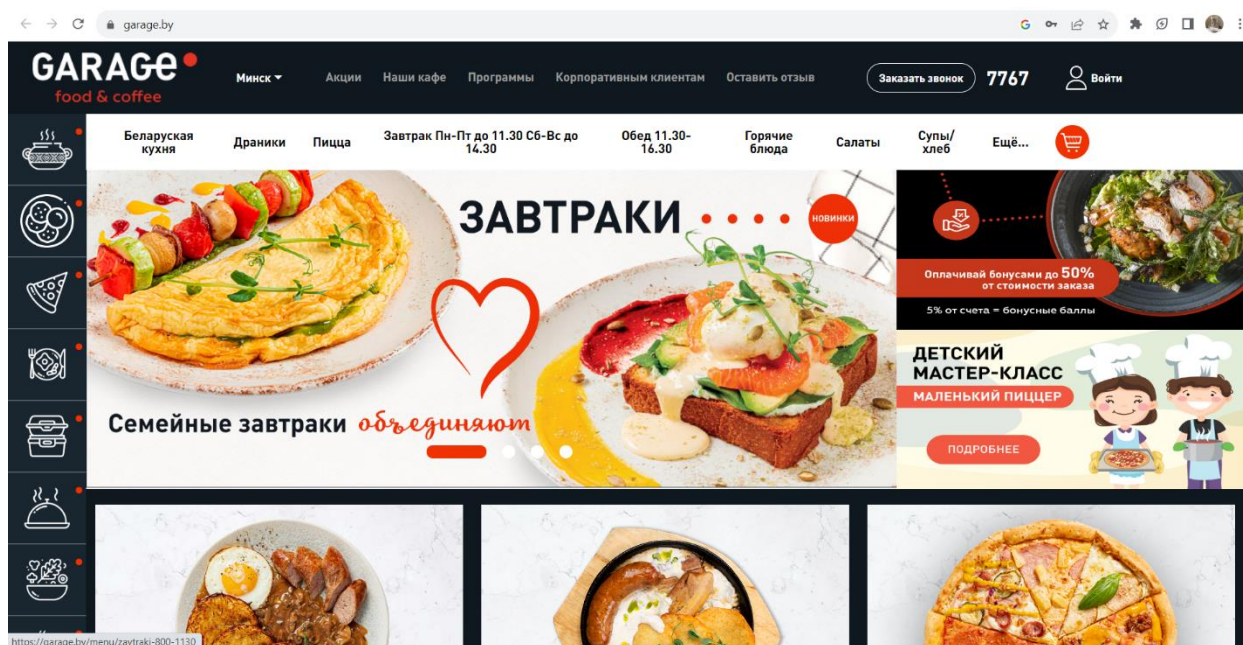


Рисунок 1.3 – Сайт кафе «Garage»

Посетитель, заходя на сайт кафе «Гараж» сразу видит акции и элементы меню. Цель такого решения – побуждать покупателя сразу делать заказ, не

задумываясь о том, где он будет его есть, т.е. такое решение подходит для заказа еды на вынос. Для кафе «Огнище» же это необязательно, т.к. индивидуальный интерьер кафе является его преимуществом и более предусмотрительно будет, чтобы посетитель на главной странице видел именно фотографии кафе.

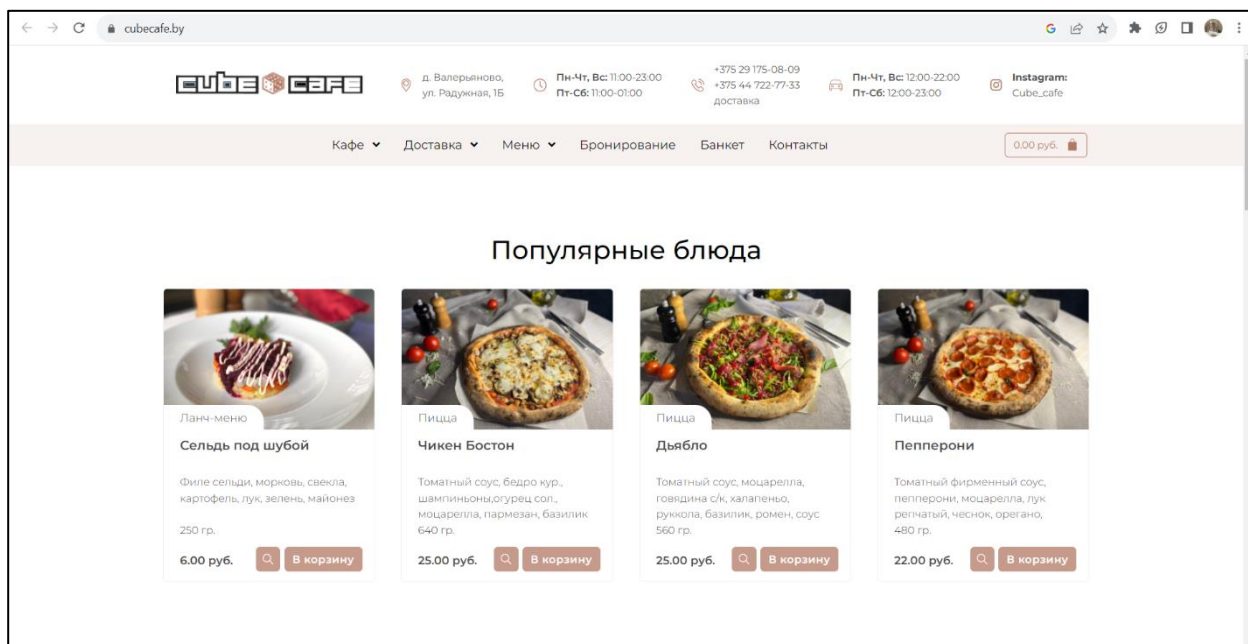


Рисунок 1.4 – Сайт кафе «CubeCafe»

На сайте кафетерия *CubeCafe* решена проблема поиска основной информации, т.к. она представлена в верхней строке сайта на каждой его странице. Данная особенность будет реализована и на сайте кафе «Огнище».

У большинства из сайтов можно выделить общие преимущества и недостатки. Из преимуществ:

- с момента открытия сайта перед пользователем сразу появляется меню, благодаря чему ему не нужно искать его;
- карточки с блюдами выглядят негромоздко и лаконично, на них только главная информация о блюде: изображение, название.

Из недостатков:

- сложно найти основную информацию, т.к. информация о времени работы, о местоположении и контактах разбросана по разным страницам сайта;
- поскольку перед покупателем на первой же странице открывается меню, пользователь не знает, как выглядит само заведение;
- сложно найти кнопку «Войти в аккаунт», т.к. она часто такого же цвета, как и остальные элементы на сайте.

Веб-сервис, который будет создан в процессе данной курсовой работы, будет обладать всеми перечисленными преимуществами, и не будет обладать некоторыми недостатками. Например, сложность поиска основной информации решается размещением её на верхней панели, чтобы она дублировалась на каждой странице, а плохая видимость кнопки «Войти» решается выделением её ярким цветом.

В системе будут реализованы роли администратора и пользователя. У администратора есть контроль над данными, хранящимися в базе данных сайта, он может добавлять, удалять и редактировать информацию о блюдах. У пользователя есть возможность регистрироваться в системе, просматривать страницы сайта, добавлять блюда в избранное или в корзину и делать заказ.

1.2 Обоснование и описание выбора языка программирования, средств разработки, используемых технологий и сторонних библиотек

Для реализации веб-сервиса будет использоваться объектно-ориентированная парадигма программирования. Как объектно-ориентированный язык был выбран язык *Java*. Для работы с базой данных будет использован язык *MySQL*. Вёрстка сайта будет реализована в *IDE Visual Studio Code*, для остальной работы будет использована *IntelliJ Idea*.

Java - один из наиболее популярных языков программирования, который остается актуальным в течение многих лет.

Преимущества языка *Java*:

- независимость кода от системы, в которой он запускается;
- надежность. *Java* является строго типизированным языком;
- сравнительно простой синтаксис;
- мультифункциональность.

MySQL – свободная реляционная система управления базами данных. *MySQL* предназначена для хранения данных в удобном для работы виде, их обработки, структурирования и предоставления доступа к ним.

У *MySQL* также есть преимущества:

- удобный доступ к базам данных;
- корректное распределенное хранение данных на сервере;
- быстрый поиск нужной информации в базе с помощью языка *SQL*;
- идентификация и обработка отдельных данных, их преобразование и отправка;
- множественный доступ к базе с разных устройств – например, из браузеров нескольких пользователей;

- выдача разных прав доступа различным пользователям;
- обеспечение безопасности данных: их защита, шифрование и контроль доступа;
- возможность контролировать версии базы данных: делать бэкапы, обновлять базу или откатывать назад
- контроль состояния базы данных.

IntelliJ IDEA предоставляет отличные возможности для разработки на *Java*, включая поддержку *Spring Boot* и *MySQL*.

Выбор языка программирования *Java* обоснован не только его популярностью, но и широким сообществом разработчиков, что обеспечивает доступ к обширным библиотекам и фреймворкам. Это способствует более эффективной разработке и поддержке веб-сервиса.

Использование *MySQL* как системы управления базами данных обеспечивает надежное хранение информации и эффективное выполнение запросов на основе языка *SQL*. Это особенно важно для веб-сервисов, требующих эффективного взаимодействия с большим объемом данных. Благодаря своей открытой природе, *MySQL* предоставляет разработчикам гибкость в управлении данными и легко интегрируется с приложениями, написанными на *Java*.

IntelliJ IDEA, с поддержкой *Spring Boot*, дополняет выбранный стек технологий, обеспечивая продуктивную разработку, отладку и профилирование *Java*-приложений. Её интегрированная среда разработки предоставляет широкий спектр инструментов для повышения эффективности работы разработчиков, включая автоматическую проверку кода, рефакторинг, и поддержку тестирования.

2 ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ И ОПИСАНИЕ СОСТОЯНИЙ ПРОГРАММНОГО СРЕДСТВА

2.1 Проектирование архитектуры и описание состояний программного средства

Архитектура платформы веб-приложения – это структура системы, которая определяет уровни абстракции и то, как различные компоненты приложения взаимодействуют друг с другом. Она обеспечивает масштабируемость, гибкость и эффективность веб-приложения, а также позволяет легко поддерживать систему и вносить в неё изменения.

Архитектура веб-приложения для автоматизации работы кафе может основываться на классической архитектурной модели *MVC (Model-View-Controller)*.

Model-View-Controller (MVC, «Модель-представление-поведение», «Модель-представление-контроллер») — архитектура программного обеспечения, в которой модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента так, что модификация одного из компонентов оказывает минимальное воздействие на остальные.

MVC состоит из трех компонент: *View* (представление, пользовательский интерфейс), *Model* (модель, ваша бизнес-логика) и *Controller* (контроллер, содержит логику на изменение модели при определенных действиях пользователя, реализует *Use Case*).

Основная идея этого паттерна в том, что и контроллер и представление зависят от модели, но модель никак не зависит от этих двух компонент. Это как раз и позволяет разрабатывать и тестировать модель, ничего не зная о представлении и контроллерах. В идеале контроллер так же ничего не должен знать о представлении (хотя на практике это не всегда так), и в идеале для одного представления можно переключать контроллеры, а также один и тот же контроллер можно использовать для разных представлений (так, например, контроллер может зависеть от пользователя, который вошел в систему).

Пользователь видит представление, на нем же производит какие-то действия, эти действия представление перенаправляет контроллеру и подписывается на изменение данных модели, контроллер в свою очередь

производит определенные действия над моделью данных, представление получает последнее состояние модели и отображает ее пользователю.

Состояния *MVC* включают:

1 инициализацию – приложение инициализируется и загружаются необходимые компоненты *MVC*;

2 обработку запросов – контроллер принимает запросы от пользователей и передаёт их в модель для обработки;

3 обновление модели – модель получает данные от контроллера и обрабатывает их, выполняя необходимые операции с данными;

4 обновление представления – модель обновляет представление с помощью оповещений или коллбэков, чтобы отобразить новые данные для пользователя;

5 взаимодействие с пользователем – пользователь взаимодействует с представлением, выполняя действия, которые требуют изменений в модели или перерисовки представления.

Таким образом, архитектура *MVC* подходит для реализации её в приложении для автоматизации работы кафе «Огнище», создаваемом в данной курсовой работе.

2.2 Проектирование и разработка графического интерфейса

Графический интерфейс – это способ взаимодействия приложения с пользователем, реализованный с помощью визуальных элементов управления: кнопок, окон, полей ввода и т.д. Графический интерфейс сайта называется *frontend*, а процесс создания интерфейса – *frontend*-разработкой.

Проектирование и разработка графического пользовательского интерфейса включает несколько шагов:

- 1 Исследование и анализ требований пользователя – понимание потребностей пользователей и цели приложения, определение основных функциональных и дизайнерских требований.
- 2 Создание структуры интерфейса – определение основных компонентов, таких как меню, панели инструментов, кнопки, поля ввода и т.д.
- 3 Проектирование визуального вида – определение стиля и внешнего вида интерфейса, выбор цветовой схемы, шрифтов, иконок, изображений.
- 4 Создание прототипа – разработка прототипа интерфейса, который визуализирует компоненты и их расположение.

- 5 Разработка интерфейса – реализация компонентов интерфейса, их размещение на сайте и настройка.
- 6 Тестирование и отладка – проверка интерфейса на работоспособность и правильность отображения данных.
- 7 Улучшение интерфейса – оптимизация интерфейса для повышения интерфейса на основании обратной связи.

Пользовательский интерфейс веб-сайта кафе «Огнище» состоит из следующих страниц: главная страница, меню, доставка и оплата, акции, контакты, отзывы.

На каждой странице расположены элементы управления: кнопки, ссылки на другие страницы, полоса прокрутки, поля ввода данных и т.д.

2.3 Описание и реализация используемых в программном средстве алгоритмов

В разрабатываемом веб-приложении используются следующие основные алгоритмы:

Алгоритмы работы с базой данных:

- построение запросов с фильтрацией, сортировкой информации о блюдах на стороне СУБД;
- транзакционная обработка для обеспечения целостности данных.

Алгоритмы регистрации и авторизации:

- авторизация по логину и паролю с проверкой в базе данных;
- регистрация нового аккаунта с сохранением новых логина и пароля в базе данных.

Реализованные алгоритмы обеспечивают необходимый функционал и выполнение процессов функционирования кафе. При необходимости возможно расширение и доработка алгоритмов.

Таким образом, разрабатываемые алгоритмы обеспечивают качественную работу приложения.

3 ЭКСПЛУАТАЦИЯ ПРОГРАММНОГО СРЕДСТВА

3.1 Ввод в эксплуатацию и обоснование минимальных технических требований к оборудованию

Для запуска веб-приложения у клиента должны быть учтены следующие минимальные технические требования:

- компьютер, для базового использования достаточно среднего уровня процессора, например *AMD Ryzen 3*. Для больших объемов лучше выбрать более мощный процессор, такой как *Intel Core i5/i7* или аналогичный от *AMD*, оперативная память (*RAM*) рекомендуется не менее 4-8 Гб оперативной памяти для комфортного использования приложения. Хранилище (Жесткий диск/*SSD*) объемом от 128 - 256 Гб или *SSD* для операционной системы, приложений и хранения данных.

- операционная система, подойдут все основные операционные системы (*Windows, macOS, Linux*).

- интернет-соединение, причем чем стабильнее и быстрее ваше интернет-соединение, тем лучше. Особенно это актуально при работе с большим объемом данных в режиме реального времени.

- браузер: *Google Chrome, Microsoft Edge, Firefox*.

- экран с разрешением, достаточным для комфортного использования интерфейса приложения.

- базовая графическая карта встроенная в процессор или дискретная видеокарта, которые обеспечат плавное отображение интерфейса приложения.

Важно учесть, что эти требования могут меняться в зависимости от того, как используются функциональные особенности веб-приложения. Если приложение работает с большими объемами данных, то при большом количестве изучаемых тем и слов, а также при выполнении сложных вычислительных операций потребуется более мощное оборудование для комфортного использования.

3.2 Руководство по эксплуатации программного средства

При входе на сайт отображается главная страница, представленная на рисунке 3.1.

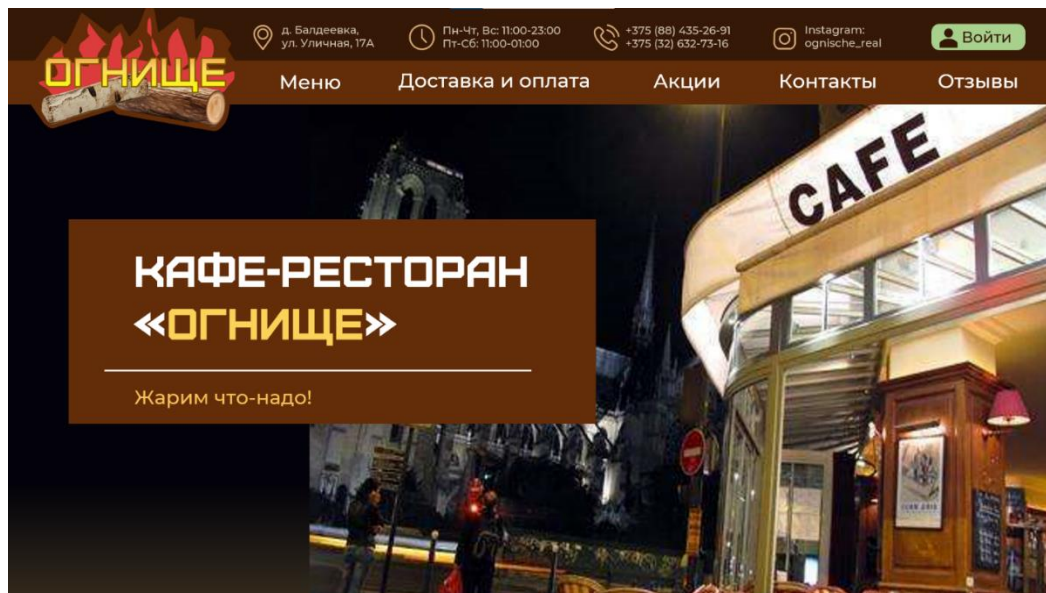


Рисунок 3.1 – Главная страница сайта кафе «Огнище»

На верхней строке сайта расположена самая актуальная информация, а именно: адрес кафе, расписание и время работы, телефоны для справок и социальные сети. Также на этой строке расположена кнопка «Войти», выделенная ярким цветом.

При нажатии на кнопку «Войти» открывается окно входа, представленное на рисунке 3.2.

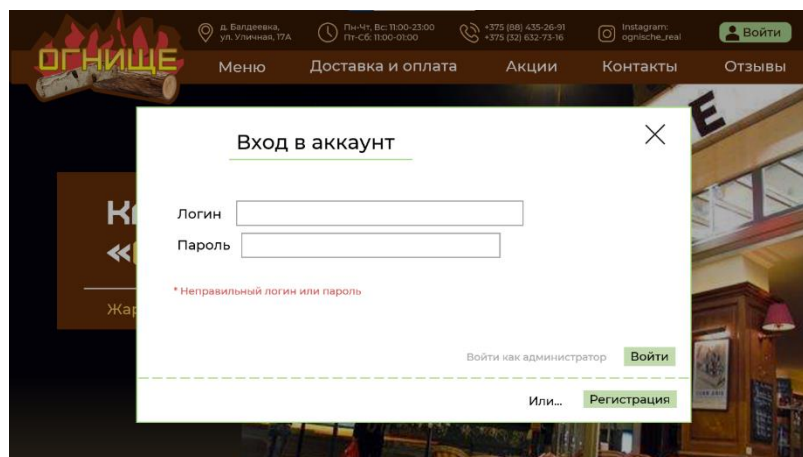
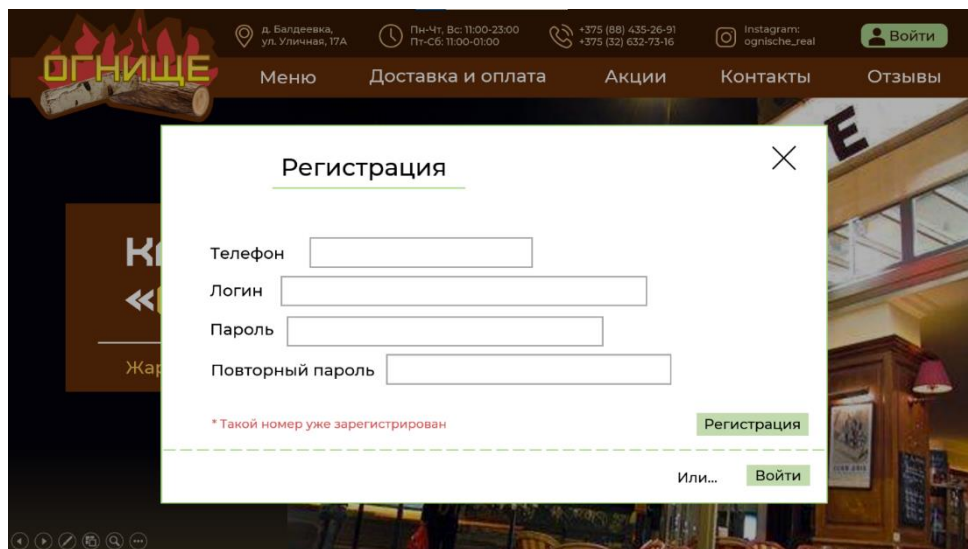


Рисунок 3.2 – Авторизация пользователя или администратора

В окошке авторизации имеются поля для ввода логина и пароля. При неправильно введенных данных появляется надпись красного цвета «Неправильный логин или пароль».

На нижней строке расположена кнопка регистрации. При нажатии она перенаправляет на окошко регистрации, представленное на рисунке 3.3.



The screenshot shows a registration modal window titled "Регистрация" with a close button (X) in the top right corner. The form contains four input fields: "Телефон", "Логин", "Пароль", and "Повторный пароль". Below the "Пароль" field, there is a red error message: "*Такой номер уже зарегистрирован". At the bottom right of the form is a green button labeled "Регистрация". Below the form, there is a link "Или..." followed by a green button labeled "Войти". The background of the website is visible, showing a header with the "ОГНИЩЕ" logo, contact information, and navigation links: "Меню", "Доставка и оплата", "Акции", "Контакты", and "Отзывы".

Рисунок 3.3 – Регистрация на сайте кафе «Огнище»

В окошке регистрации расположены поля для ввода телефона, логина, пароля и подтверждения пароля. При вводе данных, которые уже занесены в базу, высвечивается надпись «Такой номер уже зарегистрирован». Также имеется кнопка для перехода обратно в окно ввода.

Во второй строке сайта справа от логотипа расположены кнопки навигации по сайту. Страница «Меню» представлена на рисунке 3.4.



Рисунок 3.4 – Меню сайта кафе «Огнище»

На странице «Меню» расположены карточки блюд. На каждой карточке есть фотография блюда, название, ингредиенты, цена и кнопка «В корзину». При нажатии на карточку открывается окошко блюда с более подробным описанием, оно расположено на рисунке 3.5.

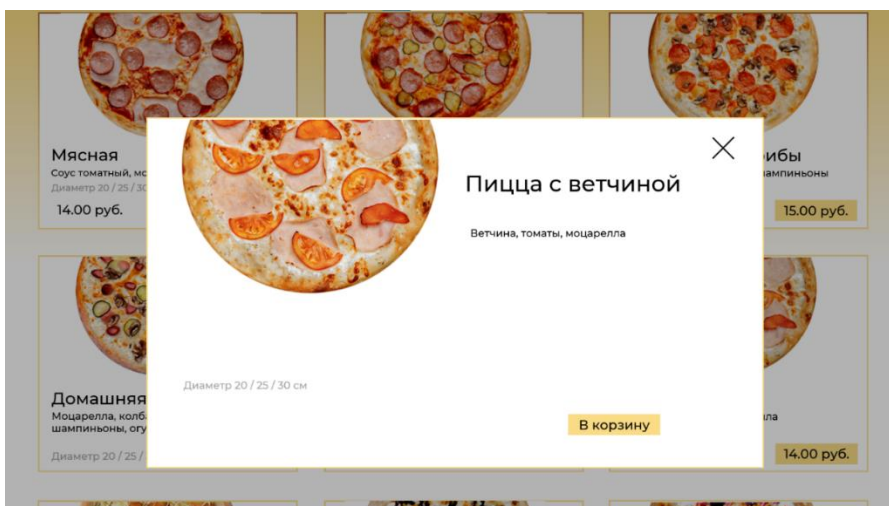


Рисунок 3.5 – Просмотр блюда сайта кафе «Огнище»

Далее по строке навигации расположена ссылка на страницу «Доставка и оплата», представленная на рисунке 3.6. На этой странице есть информация о времени доставки и стоимости доставки. Также цветовым блоком выделена информация о телефонах, по которым совершаются заказы.

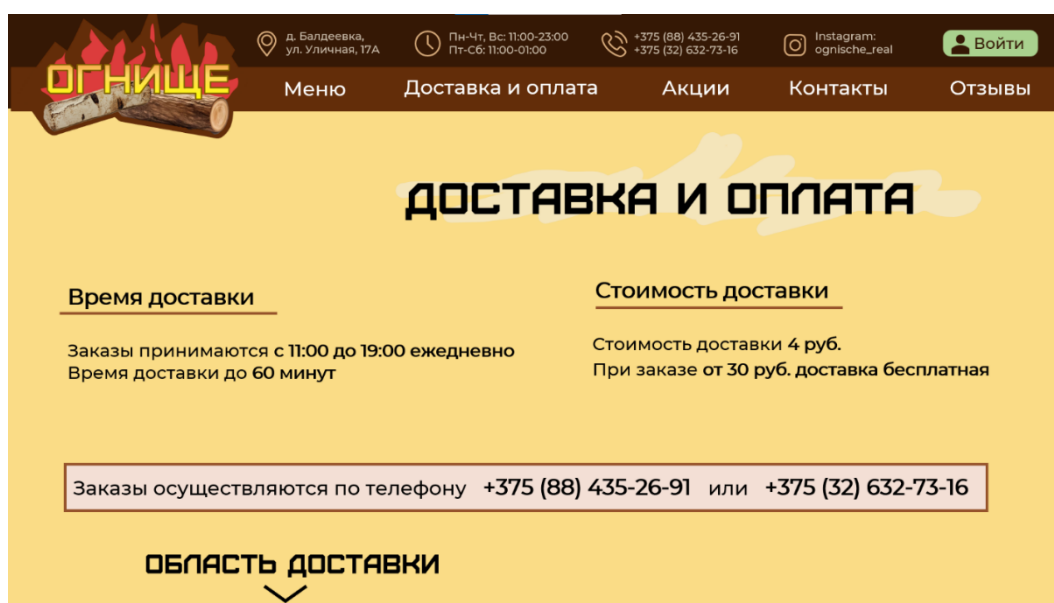


Рисунок 3.6 – Доставка и оплата сайта кафе «Огнище»

Также на сайте имеется страница «Акции», расположенная на рисунке 3.7. На ней расположены рекламные баннеры с информацией об актуальных акциях.

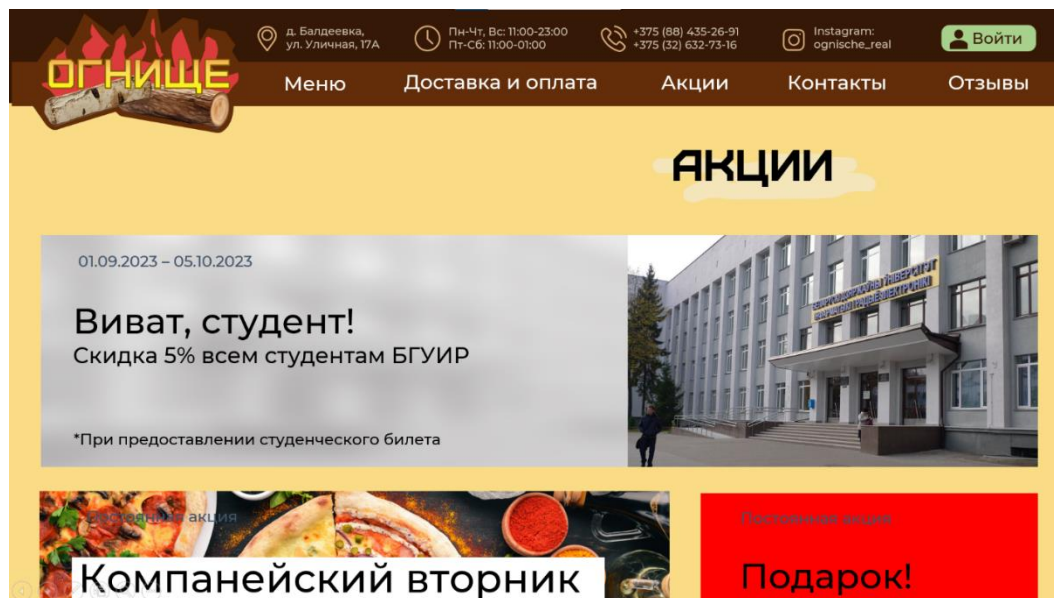


Рисунок 3.7 – Акции сайта кафе «Огнище»

Следующая страница сайта – «Контакты», она представлена на рисунке 3.8. На ней пользователи могут посмотреть адрес и время работы кафе. Также на ней имеется карта, чтобы сориентироваться, и фотография кафе.

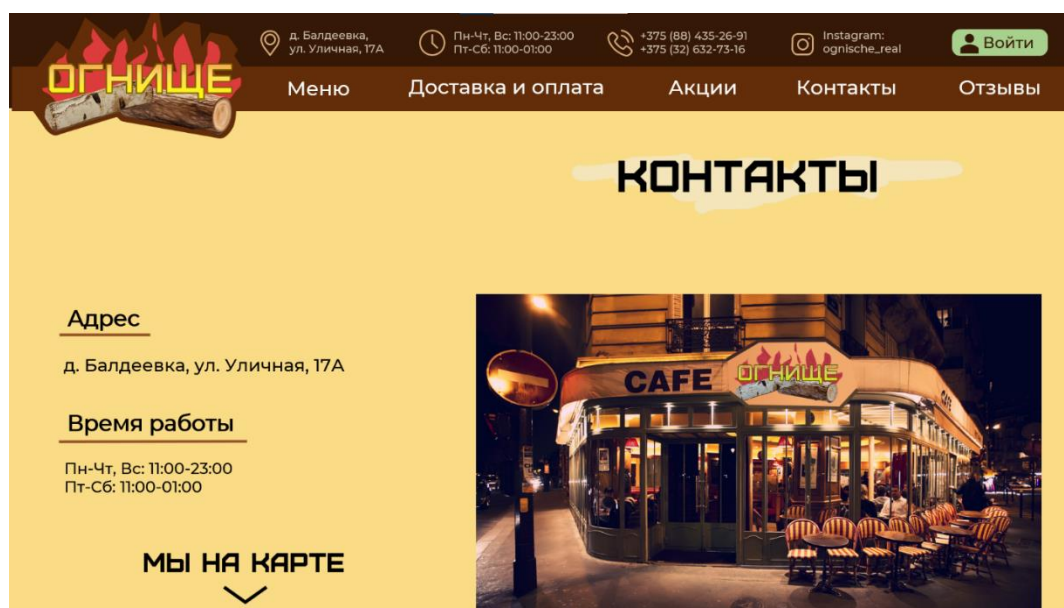


Рисунок 3.8 – Контакты сайта кафе «Огнище»

И последняя страница сайта – «Отзывы». На ней пользователи могут оставлять комментарии о качестве еды и обслуживания. Она представлена на рисунке 3.9.



Рисунок 3.9 – Отзывы сайта кафе «Огнище»

Всего на сайте 6 страниц: «Главная», «Доставка и оплата», «Акции», «Контакты», «Меню» и «Отзывы».

ЗАКЛЮЧЕНИЕ

В рамках курсового проекта было разработано веб-приложение, предоставляющее удобный и понятный пользователям доступ к меню и основной информации о кафе «Огнище». Веб-сайт позволяет администратору настраивать список блюд, добавлять их, редактировать их название и описание, и позволяет пользователю посмотреть адрес кафе, номера для заказа по телефону, меню, актуальные акции и отзывы.

Пользователи могут войти в свой личный кабинет, добавлять блюда в избранное или в корзину и оформлять заказ. Если же у пользователя аккаунт еще не создан, его можно создать в окошке «Регистрация».

Разработанное веб-приложение является необходимым сервисом для кафе для оформления заказов онлайн, без непосредственного посещения заведения. Оно предоставляет пользователю доступ к необходимой информации, исходя из которой пользователь сможет выбрать желаемое блюдо, возможность оформить заказ через корзину на сайте. Также оно позволяет администратору сайта беспрепятственно управлять карточками блюд в меню.

При решении задач были изучены теоретически методы предъявления и анализа требований к программному обеспечению, пользовательскому интерфейсу. Получены практические навыки по разработке интерфейса, а также получены ценные знания и улучшены навыки в программировании.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- [1] Антиплагиат – обнаружение заимствований [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://antiplagiat.ru/private/>
- [2] Клиент-серверная архитектура применительно к БД [Электронный Ресурс]. – Электронные данные. – Режим доступа: <https://studfile.net/preview/1523607/page:2/>
- [3] *Spring Framework* [Электронный ресурс]. – Электронные данные. – доступа: <http://spring-projects.ru/projects/spring-framework/> .
- [4] Многопользовательские СУБД [Электронный ресурс] – Режим доступа: <https://intuit.ru/studies/courses/508/364/lecture/8643?page=2>
- [5] Диаграмма классов (Электронный ресурс). – Электронные данные. – Режим доступа: https://flexberry.github.io/ru/gpg_class-diagram.html
- [6] *UML* диаграмма классов [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habr.com/rulpost/511798/>
- [7] Диаграмма состояний [Электронный ресурс]. – Электронные данные. – Режим доступа: https://flexberry.github.io/rulfd_statechart-diagram.html
- [8] Блок-схемы (Электронный ресурс). – Электронные данные. – Режим доступа: <https://clck.ru/Rr5hc>
- [9] Справочное руководство по составлению алгоритмов [Электронный ресурс]. – Электронные данные. – Режим доступа: http://koi.tspu.ru/vav/vav_umk_inflalgoritmiz/algorithm8.htm
- [10] Архитектура «клиент-сервер» [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://studfile.net/preview/3640855/page:3/>

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг программного кода

OgnishceApplication.java

```
package org.example.ognishce;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class OgnishceApplication {
    public static void main(String[] args) {
        SpringApplication.run(OgnishceApplication.class, args);
    }
}
```

Akcii.java

```
package org.example.ognishce.models;
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
@Data
@AllArgsConstructor
@NoArgsConstructor
```



```

@Entity

public class Akcii {

    @Id

    @GeneratedValue(strategy = GenerationType.AUTO)

    private Integer id;

    private String img_src;

}

```

Bluda.java

```

package org.example.ognishce.models;

import jakarta.persistence.*;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity

@AllArgsConstructor
@NoArgsConstructor

@Data

public class Bluda {

    @Id

    @GeneratedValue(strategy = GenerationType.AUTO)

    private Integer id;

    private String img_src;

    private String name;

    @Column(name = "ingredients", columnDefinition = "text")

    private String ingredients;

```

```
@Column(name = "opisanie", columnDefinition = "text")
private String opisanie;
private Double cena;
}
```

Izbrannoe.java

```
package org.example.ognishce.models;
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.Entity;

public class Izbrannoe {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;
    private Integer user_id;
    private Integer burglary_id;
    private Integer deserty_id;
    private Integer garniry_id;
    private Integer napitki_id;
    private Integer pizza_id;
    private Integer salaty_id;
```

```
private Integer shaurma_id;  
private Integer sousy_id;  
}
```

Korzina.java

```
package org.example.ognishce.models;  
import jakarta.persistence.*;  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
import lombok.NoArgsConstructor;  
@Data  
@AllArgsConstructor  
@NoArgsConstructor  
@Entity  
public class Korzina {  
    @Id  
    @GeneratedValue(strategy = GenerationType.AUTO)  
    private Integer id;  
    private Integer user_id;  
    private Integer burglary_id;  
    private Integer deserty_id;  
    private Integer garniry_id;  
    private Integer napitki_id;  
    private Integer pizza_id;  
    private Integer salaty_id;  
    private Integer shaurma_id;
```

```
private Integer sousy_id;  
private Integer number;  
}
```

Otzyvy.java

```
package org.example.ognishce.models;  
import jakarta.persistence.*;  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
import lombok.NoArgsConstructor;  
@Data  
@AllArgsConstructor  
@NoArgsConstructor  
@Entity  
public class Otzyvy {  
    @Id  
    @GeneratedValue(strategy = GenerationType.AUTO)  
    private Integer id;  
    private String login_polz;  
    private String text;  
    private String data;  
    private Integer zvezdy;  
}
```

Users.java

```
package org.example.ognishce.models;

import jakarta.persistence.*;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor

public class Users {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;
    private String login;
    private String password;
    private String phone;
}
```

ПРИЛОЖЕНИЕ Б

(обязательное)

Справка о проверке курсового проекта на плагиат

УДАЛЕННЫЕ ДОКУМЕНТЫ | < 1/1 >

ПЕРЕМЕСТИТЬ | УДАЛИТЬ | ИСТОРИЯ ОТЧЕТОВ

<input type="checkbox"/>	Название	Дата загрузки	Оригинальность	
<input type="checkbox"/>	PDF Пояснительная записка А. Лойко	19 Дек 2023 11:26	90,43%	ПОСМОТРЕТЬ РЕЗУЛЬТАТЫ

Ссылаетесь на свои уже опубликованные работы или труды других авторов? Подключите распознавание цитирований для повышения точности анализа.

КУПИТЬ ПРОВЕРКИ

Рисунок Б.1 – Справка о проверке курсового проекта на антиплагиат

ПРИЛОЖЕНИЕ В
(обязательное)
Ведомость документов