

Министерство образования Российской Федерации

Пензенский государственный университет

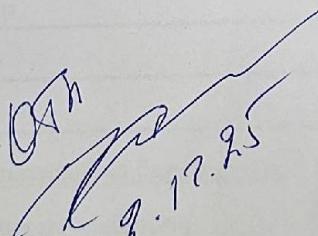
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по курсу «Разработка клиент-серверных приложений»
на тему «Сетевое клиент-серверное приложение обеспечивающее
выполнение удаленного задания»

Выполнил:

студент группы 22ВВП1

Демин М.С.



A handwritten signature in black ink, appearing to read 'Демин М.С.' followed by a date.

Проверил:

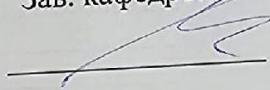
к.т.н. Зинкин С. А.

Пенза 2025

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники
Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ


М.А. Митрохин

«15» 09 2025 г.

ЗАДАНИЕ
на курсовое проектирование по курсу
Разработка клиент-серверных приложений

Студенту Ремизу Мансму Сергеевичу Группа 22ВВП_1
Тема проекта Сетевое клиент-серверное приложение
обеспечивающее выполнение удаленного задания

Исходные данные (технические требования) на проектирование

Сеть Microsoft TCP/IP

Стек протоколов TCP/IP

Протоколы TCP и UDP

Интерфейс WinSock2

На удаленном сервере размещена база данных

Дополнительные условия по указанию

преподавателя: клиент-серверная система взаимодействия
поставщиков и клиентов.

Объем работы по курсу

1. Расчетная часть

Анализ граф-схемы алгоритма

Алгоритм работы
сервера

2. Графическая часть

UML-диаграммы

ГСА в Visio

3. Экспериментальная часть

Отладка сетевого приложения на сети

Отладка программного обеспечения и формирование контента
сервера

Срок выполнения проекта по разделам

1 По графику

2

3

4

5

6

7

8

Дата выдачи задания "15" _09_ 20

Руководитель

Дата защиты проекта По графику

Задание получил

Зинкин С. А.

Студент

" 15 " 09 2025

Дзенин Н. С. !

Содержание

| | |
|---|----|
| Введение..... | 3 |
| 1. Постановка задачи | 4 |
| 2. Выбор решения..... | 5 |
| 3. Теоретическая часть задания | 6 |
| 4. Описание алгоритма работы | 7 |
| 4.1 Описание протокола передачи | 8 |
| 5. Отладка и тестирование | 11 |
| 6. Описание программы (БД)..... | 12 |
| 6.1 Архитектура доступа к данным..... | 12 |
| 6.2 Структура базы данных..... | 12 |
| 6.3 Инициализация базы данных..... | 14 |
| 6.4 Репозиторий: слой доступа к данным..... | 15 |
| 6.5 Сервисы..... | 16 |
| 6.6 Управление соединениями и транзакциями..... | 18 |
| 6.7 Оптимизация работы с БД | 19 |
| 6.8 Обработка ошибок и восстановление | 20 |
| 6.9 Производительность и масштабируемость | 22 |
| 7. Руководство пользователя..... | 24 |
| Заключение | 26 |
| Список используемых источников..... | 27 |
| Приложение А | 28 |
| Приложение Б..... | 50 |
| Приложение Б1 | 50 |
| Приложение Б2 | 51 |
| Приложение Б3 | 52 |
| Приложение Б4 | 53 |
| Приложение Б5 | 53 |
| Приложение В | 54 |

Введение

Архитектура "Клиент-сервер" представляет собой фундаментальную модель распределения задач в вычислительных системах, где сервер предоставляет ресурсы и услуги, а клиенты их потребляют. В современной разработке программного обеспечения эта архитектура остается ключевой для создания масштабируемых и надежных приложений.

В рамках данной курсовой работы была разработана многоуровневая клиент-серверная система на платформе .NET 8, моделирующая взаимодействие между поставщиками товаров и клиентами. Система реализует комплексный функционал, включающий управление товарами, систему рейтингов, обмен сообщениями и интеллектуальное формирование корзины покупок на основе персональных предпочтений клиентов.

Особенностью проекта является гибкая модульная архитектура, изолированные слои данных и бизнес-логики, а также реализация уникального алгоритма формирования корзины, учитывающего индивидуальные оценки клиентов.

Заключение

В процессе выполнения курсовой работы была разработана и реализована многоуровневая клиент-серверная система взаимодействия поставщиков и клиентов. Были получены и закреплены следующие практические навыки: программирования на C# с использованием современных возможностей .NET 8, проектирования многослойной архитектуры приложений, реализации REST-подобного API на основе HttpListener, работы с реляционными базами данных (SQLite) через паттерн Repository, разработки консольного интерфейса с интерактивными меню. Были детально изучены принципы построения бизнес-логики с учётом персонализированных данных (система рейтингов и алгоритм формирования корзины), а также организация асинхронного сетевого взаимодействия между клиентом и сервером. Система была полностью завершена, протестирована и готова к использованию. Также изучены принципы инверсии зависимостей, внедрения сервисов и разделения ответственности между слоями приложения, что важно для создания поддерживаемого и масштабируемого кода.

Список используемых источников

1. Microsoft Documentation. .NET 8 Documentation –
<https://docs.microsoft.com/en-us/dotnet/>.
2. SQLite Documentation [Электронный ресурс].
– <https://www.sqlite.org/docs.html>.
3. RFC 7230-7237: HTTP/1.1 Specification –<https://tools.ietf.org/html/rfc7230>.
4. Metanit. Учебные материалы по C# и .NET –<https://metanit.com/sharp/>.
5. Martin Fowler. Patterns of Enterprise Application Architecture. – Addison-Wesley, 2002. – 560 с.
6. Jeffrey Richter. CLR via C#. – Microsoft Press, 2012. – 896 с.
7. Microsoft Learn. ASP.NET Core Documentation –
<https://learn.microsoft.com/en-us/aspnet/core/>.
8. Stack Overflow. Сообщество разработчиков –<https://stackoverflow.com/>.
9. GitHub. Исходный код проектов .NET –<https://github.com/dotnet>.
10. Newtonsoft.Json Documentation – <https://www.newtonsoft.com/json>.