**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ОТЧЕТ**

**по учебной практике (по профилю специальности)**

по профессиональному модулю ПМ 02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность 09.02.07 «Информационные системы и программирование»

(код и наименование специальности)

Студент(ка) 4 курса 22290907/1097 группы

Моисеев Евгений Васильевич

(ФИО полностью)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «29» сентября 2025 г. по «11» октября 2025 г.

Руководитель практики Хисамутдинова А.С.

(подпись) (расшифровка подписи)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2025

СОГЛАСОВАНО:

Председатель ПЦК

\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_ г.

**ЗАДАНИЕ**

**на учебную практику (по профилю специальности)**

по профессиональному модулю ПМ 02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность 09.02.07 «Информационные системы и программирование»

(код и наименование специальности)

Студент(ка) 4 курса 22290907/1097 группы

Моисеев Евгений Васильевич

(ФИО полностью)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «29» сентября 2025 г. по «11» октября 2025 г.

**Виды работ, обязательные для выполнения** *(переносится из программы соответствующего ПМ):*

1. *Участие в выработке требований к программному обеспечению;*
2. *Стадии проектирования программного обеспечения;*
3. *Разработка модулей программного обеспечения;*
4. *Тестирование программных модулей и их интеграции;*
5. *Разработка программной документации и стандарты кодирования.*

**Индивидуальное задание:** Вариант 11

Задание выдал « 29 » сентября 2025 г. Хисамутдинова А.С.

(подпись)  (Ф.И.О.)

Задание получил « 29 » сентября 2025 г. Моисеев Е.В.

(подпись)  (Ф.И.О.)

**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ДНЕВНИК**

**прохождения учебной практики**

**(по профилю специальности)**

По профессиональному модулю ПМ 02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 «Информационные системы и программирование»\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование специальности)

Студент(ка) 4 курса 22290907/1097 группы

\_\_\_\_\_\_Моисеев Евгений Васильевич\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Место прохождения практики: \_\_ УВЦ, пр. Энгельса, 23\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование и адрес организации)

Период прохождения практики

с «29» сентября 2025 г. по «11» октября 2025 г.

Руководитель практики Хисамутдинова А.С.

(подпись) (расшифровка подписи)

Санкт-Петербург

2025

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Виды выполненных работ и заданий по программе практики | Подпись руководителя практики |
| **1** | **2** | **3** |
| 29.09.25 | Анализ предметной области  Анализ методов решения |  |
| 30.09.25 | Анализ и выбор средств  Исследовательские работы. Обоснование принципиальной возможности решения задачи |  |
| 01.10.25 | UML. Проектирование диаграммы вариантов использования и последовательности |  |
| 02.10.25 | Формирование основного алгоритма системы и детализация в виде алгоритма одной из ее функций  Моделирование структуры ПО |  |
| 03.10.25 | Создание базы данных и ERD модели с применением case-средств. Выполнение резервного копирования БД и восстановления |  |
| 04.10.25 | Проектирование интерфейса пользователя  Разработка модели данных |  |
| 06.10.25 | Разработка программного продукта. Создание функциональности ПО |  |
| 07.10.25 | Модульное тестирование |  |
| 08.10.25 | Создание тестовых случаев |  |
| 09.10.25 | Отладка программных модулей |  |
| 10.10.25 | Разработка документа «Руководство системного программиста» в соответствии с ЕСПД |  |
| 11.10.25 | Подготовка отчета |  |

Содержание

[Введение 6](#_Toc210984493)

[1 Задание №1 «Выработка требований к ПО» 7](#_Toc210984494)

[2 Задание № 2 «Моделирование системы средствами UML, составление алгоритмов и спецификации» 11](#_Toc210984495)

[3 Задание № 3 «Разработка БД» 16](#_Toc210984496)

[4 Задание № 4 «Разработка программного продукта» 29](#_Toc210984497)

[Задание № 5 «Тестирование программного продукта» 44](#_Toc210984498)

[5 Задание №6 «Работа с GitHub» 51](#_Toc210984499)

[ПРИЛОЖЕНИЕ А 52](#_Toc210984500)

[ПРИЛОЖЕНИЕ Б 61](#_Toc210984501)

[ПРИЛОЖЕНИЕ В 62](#_Toc210984502)

Введение

Цель учебной практики: получить опыт создания сложных интегрированных систем, участвуя во всех стадиях и этапах жизненного цикла разработки ПО. В прохождении практики происходит так же подготовка к демонстрационному экзамену. Идёт участие в во всех этапах разработки приложения, среди которых: разработка БД, составление документации, разработка приложения, а в частности его модулей, проведение автоматизированного и ручного тестирований.

Задачами учебной практики являются: обучение разработке кода программного продукта на основе готовой спецификации на уровне модуля;

обучение использованию инструментальных средств на этапе разработки программного продукта; обучение проводить тестирование программного модуля по определенному сценарию.

Прохождение учебной практики позволяет углубить знания в своей профессиональной компетенции, а также отработать навыки создания типовых информационных систем.

# Задание №1 «Выработка требований к ПО»

Цель работы - провести анализ предметной области, методов решения. Проанализировать и выбрать технические и программные средства разработки. Разработать техническое задание в соответствие с ГОСТ 19.201-78.

Анализ методов решения:

Первый и основной вопрос, который требует наибольшего внимания это выбор основы технологии, которая будет реализовывать БД. Выбор по своей сути стоит между SQL БД и NoSQL БД.

SQL реализация будет заключаться в разработки большого количества отношений и их связей. NoSQL реализация может иметь множество других форм реализации, но наиболее интересная и подходящей будет документо-ориентированная, позволяющая организовывать свою структуру файлов-документов и их связей.

Выбор будет сделан в пользу SQL (то есть реляционной модели), поскольку пользование в системе предполагает большое количество не регулярно изменчивых по структуре данных, то их хранение наиболее оптимально будет организовать в отношениях SQL баз данных.

Преимущество SQL баз данных заключаются в хранении предсказуемых, структурированных данных, где важны надежность, точность и соблюдение целостности, в то время как преимущества документо-ориентированных БД это возможность хранения и обработки типизированных, но при этом иногда различающихся по структуре данных.

Поскольку разработанная система не предполагает регулярного обновления приложения и форматов хранимых данных, то и преимущества документо-ориентированных БД несущественны для данного проекта. Таким образом решение будет реализовано средствами SQL.

Обзор средств программирования:

Перед реализацией информационной системы следует сделать два основных выбора: СУБД и технологии для реализации приложения.

СУБД существует великое множество, но выбор СУБД ограничим 4-мя популярными: MySQL, PostgreSQL, Microsoft SQL Server, SQLite.

Выбор будет сделан в пользу PostgreSQL, поскольку:

* Microsoft SQL Server это платный лицензионный продукт, который обладает высоким разнообразием функционала, который в текущем проекте не востребован.
* SQLite не подойдёт, поскольку требуется работа с сервером и скорее всего подразумевается многопользовательский доступ, который SQLite не обеспечивает.
* В свою очередь выбор между PostgreSQL и MySQL почти не играет роли, но PostgreSQL имеет свободную лицензию MIT, в то время как MySQL распространяется под двойной лицензией и GPL/commercial, что означает уверенность в том, что ПО не станет платным. Кроме того PostgreSQL универсален и с его помощью можно настроить БД под почти любые задачи. Такой выбор подтверждается статистикой показанной ниже. (рисунок 1)

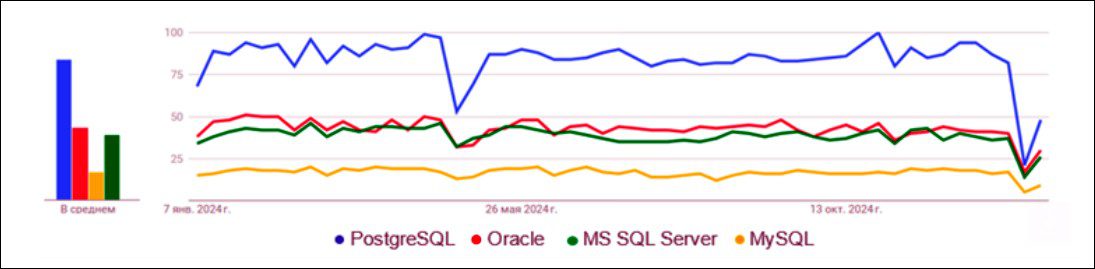


Рисунок 1 - Динамика популярности СУБД в России в 2024 г., Google Trends.

Выбор технологии реализации приложения состоит из выбора языка программирования, а также выбора фреймворка для работы с базой данных.

Возможные языки программирования: C++, C#, python, Java.

C++ — это компилируемый, статически типизированный язык программирования общего назначения, разработанный как расширение языка C, сочетающий низкоуровневый контроль с поддержкой объектно-ориентированного и обобщенного программирования.

C# — это компилируемый статически типизированный язык программирования, поддерживает автоматическую сборку мусора, полиморфизм и обобщённые типы. Ключевые особенности — его универсальность для разработки веб-приложений (ASP.NET), игр (Unity), десктопных и мобильных приложений, а также тесная интеграция с платформой .NET.

Python – это высокоуровневый, интерпретируемый язык программирования общего назначения с простым синтаксисом, поддерживающий несколько парадигм программирования (ООП, процедурное, функциональное).

Java – это многоплатформенный (кроссплатформенный), объектно-ориентированный язык программирования общего назначения, который позволяет создавать приложения для различных платформ, таких как серверы, компьютеры и мобильные устройства.

Среди этих языков наибольшим преимуществом обладает C#, поскольку работа с C# почти всегда подразумевает работу с платформой .Net, на которой доступны к применению широкий спектр библиотек для работы с SQL БД, в частности ORM библиотек. Именно ORM становится главным фактором выбора средств разработки и c++ не подходит для этих целей, поскольку сам язык не ориентирован в своей основе на ООП. Python в свою очередь ориентирован на ООП, но зачастую имеет проблемы с производительностью, которые в свою очередь решают тем, что прибегают к работе с более низкоуровневыми программирование на том же C++. Java в свою очередь почти полностью аналогичен по своей философии языку программирования C#, однако выбор в пользу C# сделан, поскольку на C# присутствует Entity Framework – фреймворк для реализации ORM, и хотя на Java тоже есть свои ORM фреймворки, Entity Framework разрабатывается непосредственно той же организацией, которой и разрабатывается весь язык C#, что означает наиболее приспособленную работу Entity Framework с средствами языка C#, по сравнению с фреймворками на Java, разрабатываемыми сторонними разработчиками.

Разработанное техническое задание представлено в приложении А

# Задание № 2 «Моделирование системы средствами UML, составление алгоритмов и спецификации»

Цель работы: разработать диаграмму вариантов использования (прецедентов) информационной системы. Определить основных актеров, варианты их использования и типы взаимодействия.

Разработать диаграмму активности для определенного процесса Вашей информационной системы. Если процесс выходит достаточно коротким, то необходимо разработать ещё одну диаграмму активности для другого процесса.

Разработать диаграмму последовательности любого другого процесса информационной системы.

Проанализировать техническое задание, составить краткую спецификацию разрабатываемого модуля выделить входные и выходные данные; сформировать основной алгоритм в виде блок-схемы в соответствии с техническим заданием. Детализировать в виде алгоритма одну из функций.

Выполнение работы:

Ниже представлены диаграмма прецендентов использования системы (рисунок 2), диаграмма активности работы прачечной (рисунок 3) и Диаграмма последовательности заказа материалов Технологом (рисунок 4).

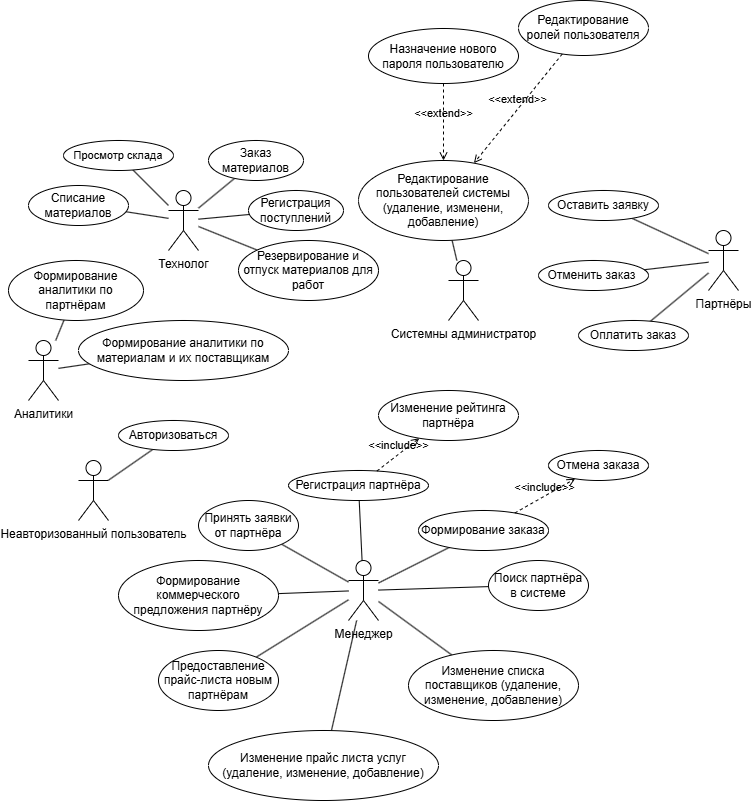


Рисунок – UML диаграмма прецендентов использования системы.

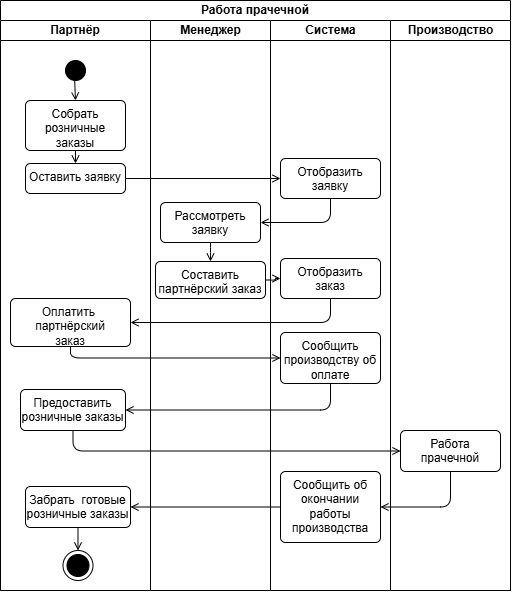


Рисунок – Диаграмма активности работы прачечной

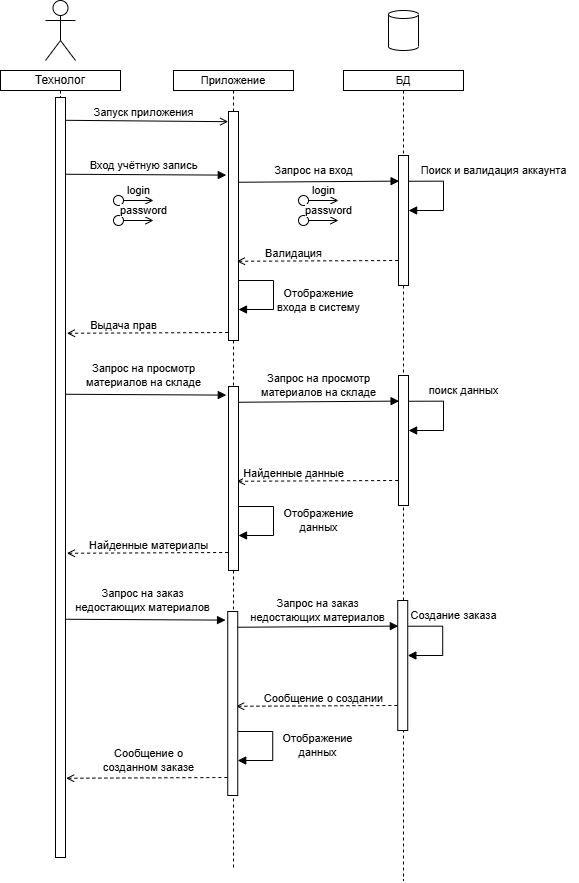


Рисунок – Диаграмма последовательности заказа материалов Технологом

Описание алгоритма: Для обеспечения безопасности хранимых паролей их необходимо шифровать, для этого будет использоваться функция HashPassword(), представленная на блок-схеме ниже. (рисунок 5)

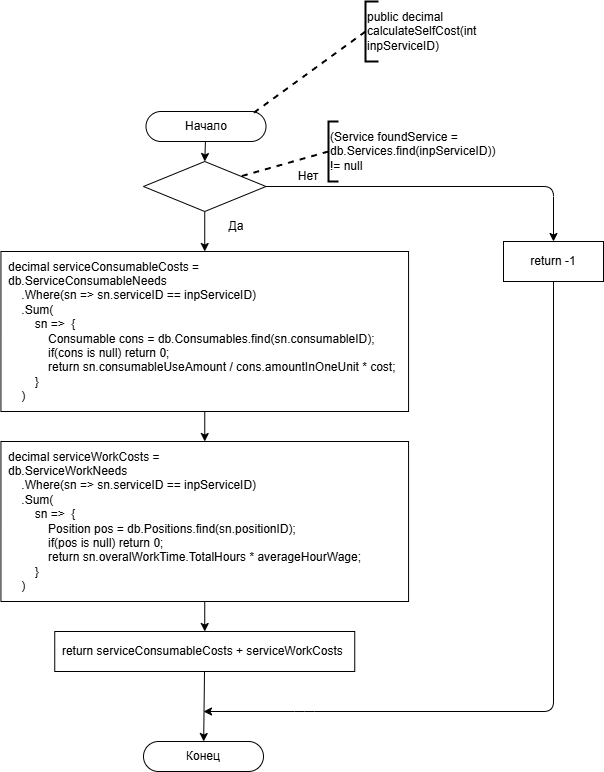


Рисунок – Блок-Схема алгоритма хэширования пароля

# Задание № 3 «Разработка БД»

Цель работы - на основе описания предметной области необходимо создать базу данных в выбранной СУБД для разрабатываемой системы.

Обязательна 3 нормальная форма с обеспечением ссылочной целостности. При разработке базы данных обратить внимание на согласованную схему именования, создать необходимые первичные и внешние ключи.

Воспроизвести все сущности предметной области, создать таблицы, поля с подходящими типами данных и связи, непосредственно относящиеся к разрабатываемой подсистеме и ее функционалу.

Разработать ER-диаграмму. ER-диаграмма должна быть представлена в формате PDF и содержать таблицы, связи между ними, атрибуты и ключи.

Заказчик системы не предоставил файлы с данными для переноса в новую систему. Поэтому необходимо подготовить файлы с данными для импорта и загрузить в разработанную базу данных (форматы csv или xml).

Напишите SQL-скрипт генерации таблиц БД. Использовать конструктор/генерацию по ER нельзя.

Выполните резервное копирование БД, сохраните полученные результаты (создать скрипт БД).

К полученной БД составьте таблицы словаря данных.

Выполнение работы:

Ниже представлена ER диаграмма БД. (рисунок 6)

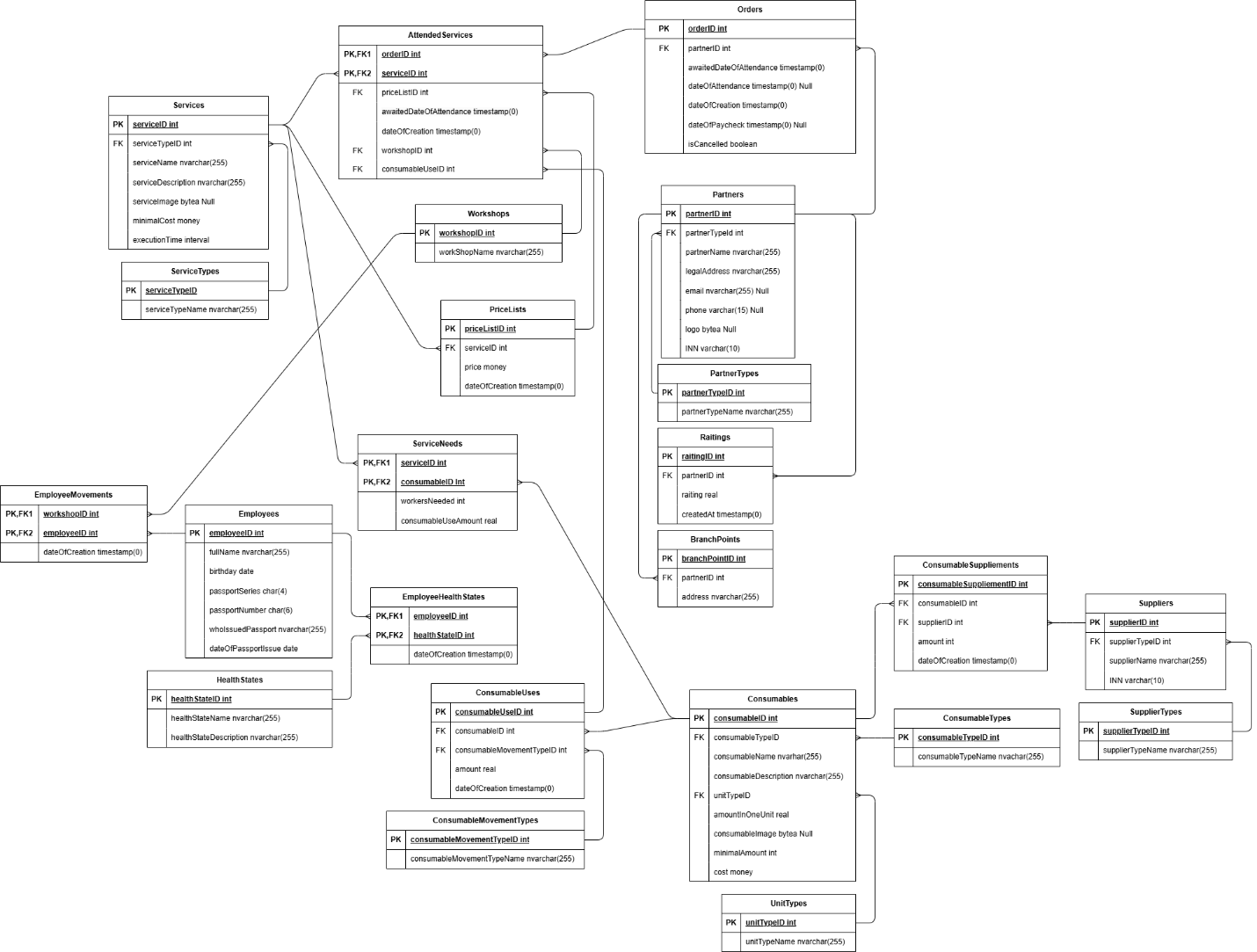


Рисунок 6 –ER диаграмма БД

Словари данных представлены ниже. (рисунок 7-рисунок 27)

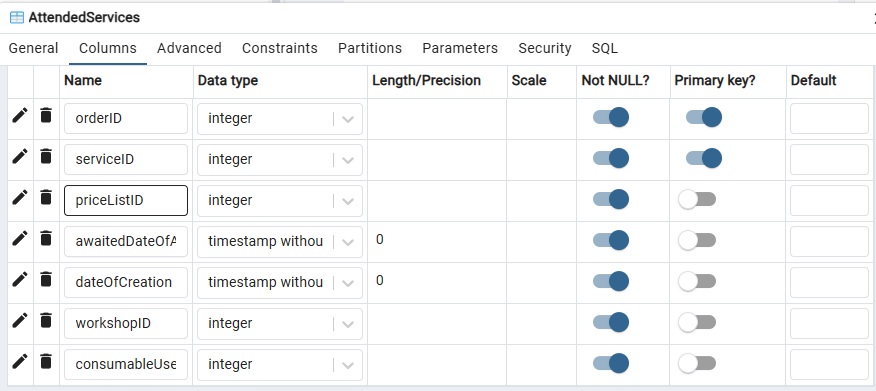


Рисунок 7 – Словарь данных таблицы AttendedServices

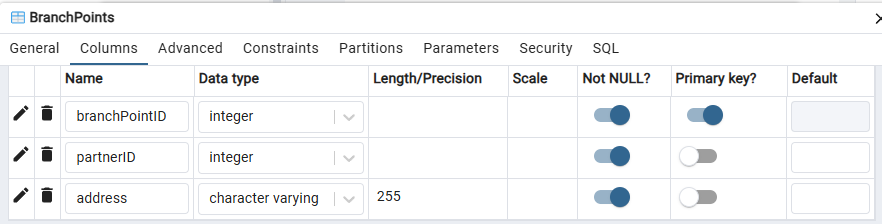


Рисунок 8 – Словарь данных таблицы BranchPoints

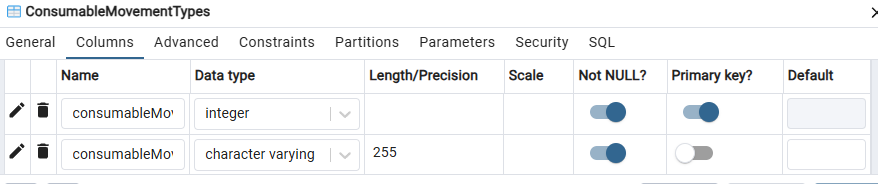


Рисунок 9 – Словарь данных таблицы ConsumableMovementTypes

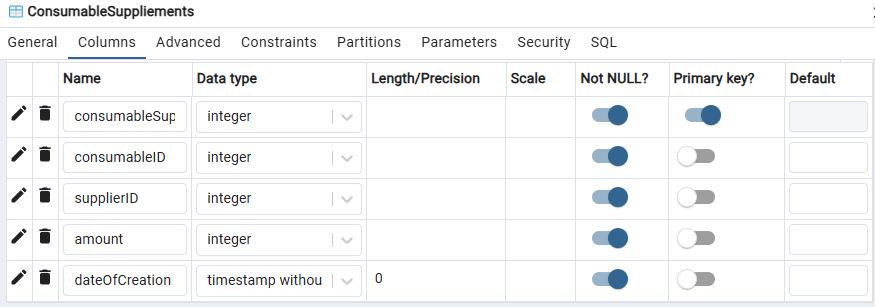


Рисунок 10 – Словарь данных таблицы ConsumableSuppliements

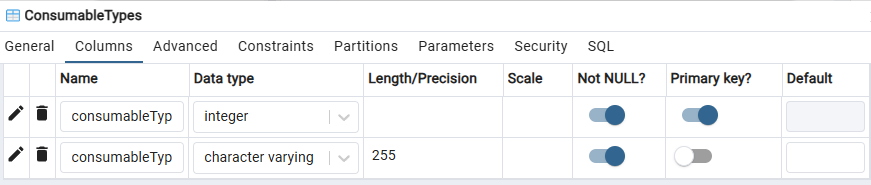


Рисунок 11 – Словарь данных таблицы ConsumableTypes

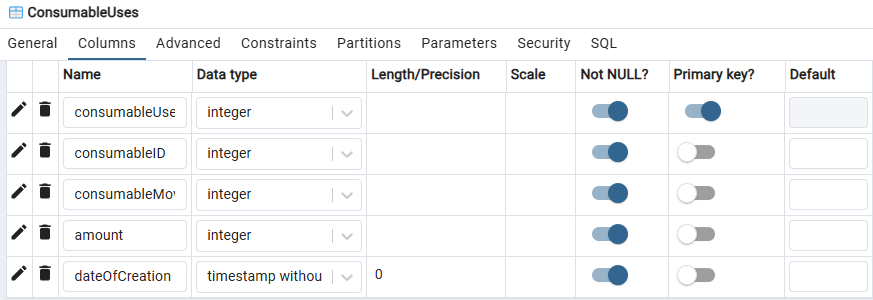


Рисунок 12 – Словарь данных таблицы ConsumbleUses

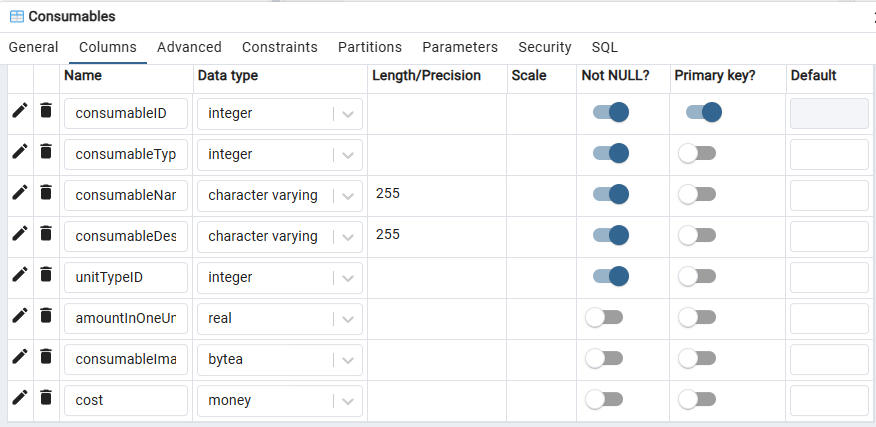


Рисунок 13 – Словарь данных таблицы Consumables

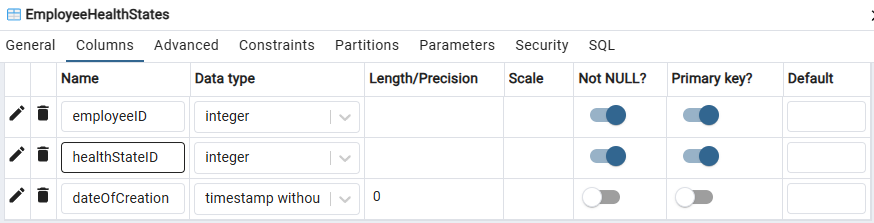


Рисунок 14 – Словарь данных таблицы EmployeeHealthStates

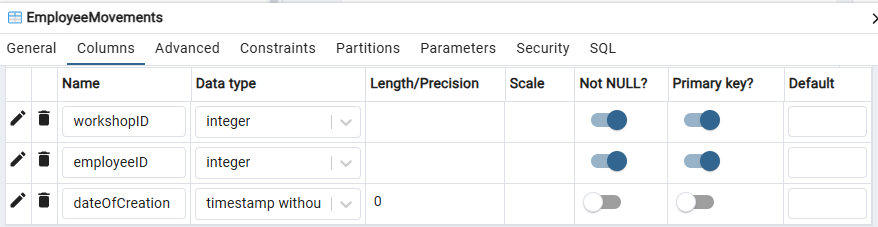


Рисунок 15 EmployeeMovements

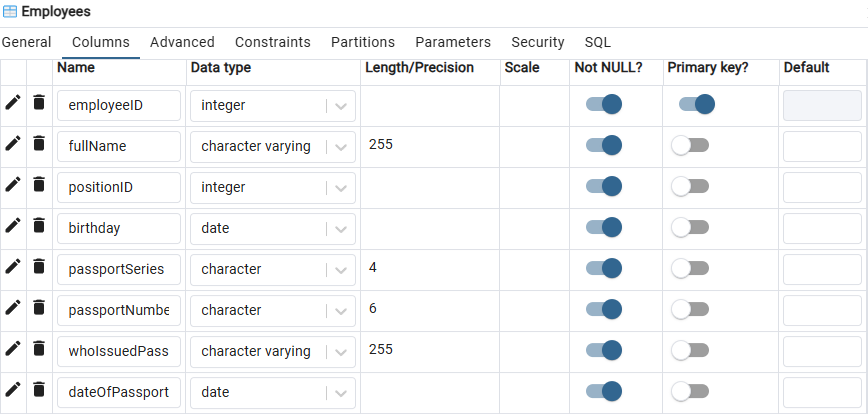


Рисунок 16 – Словарь данных таблицы Employees

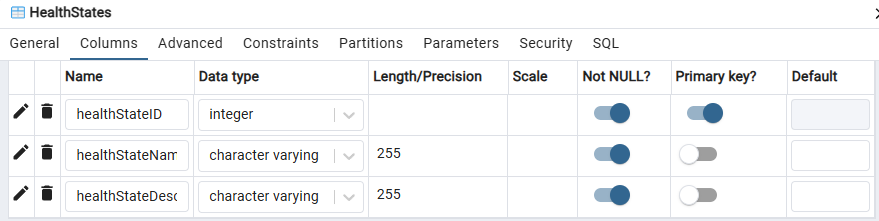


Рисунок 17 – Словарь данных таблицы HealthStates

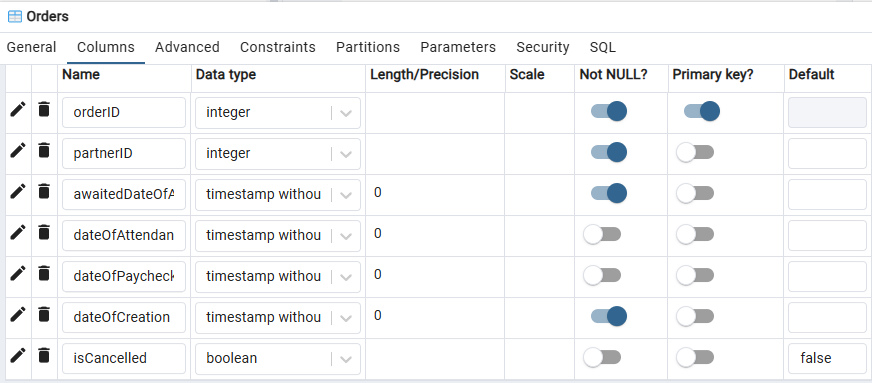


Рисунок 18 – Словарь данных таблицы Orders

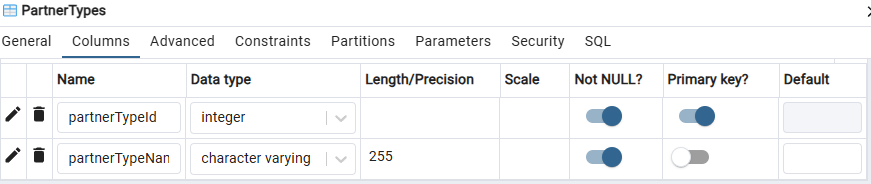


Рисунок 19 – Словарь данных таблицы PartnerTypes

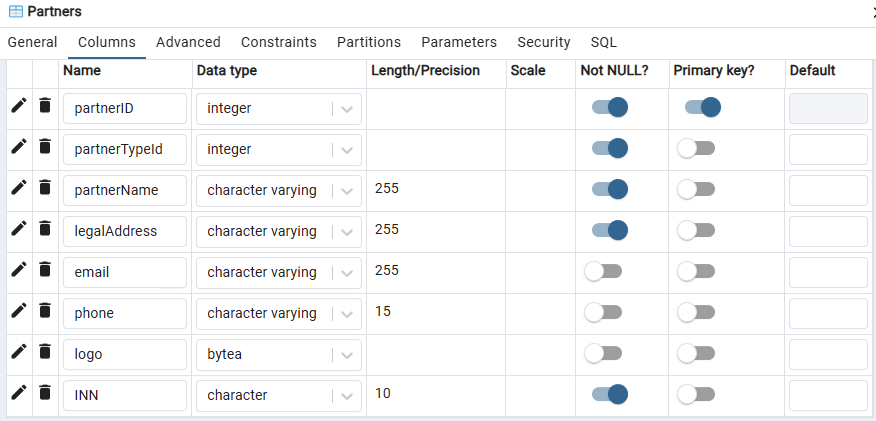


Рисунок 20 – Словарь данных таблицы Partners

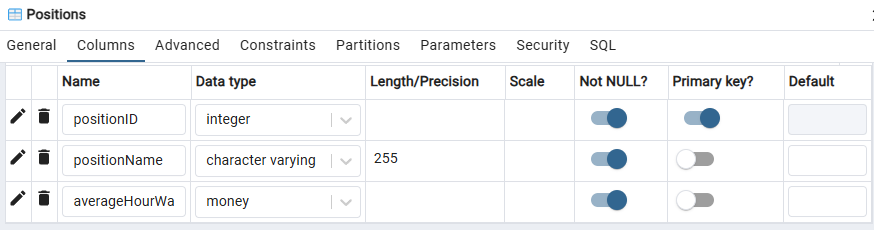


Рисунок 21– Словарь данных таблицы Positions

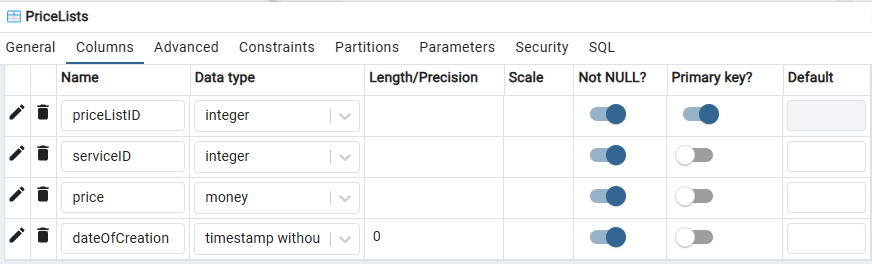


Рисунок 22 – Словарь данных таблицы PriceLists

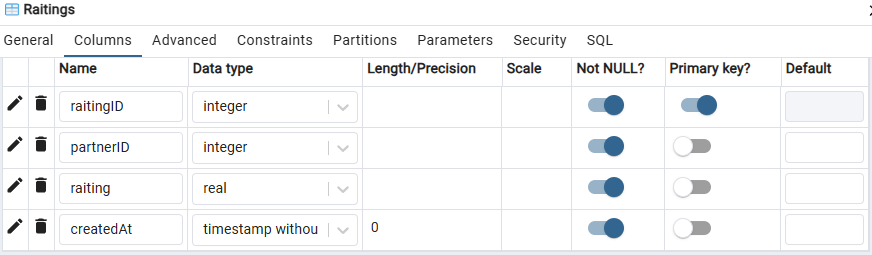


Рисунок 23 – Словарь данных таблицы Raitings

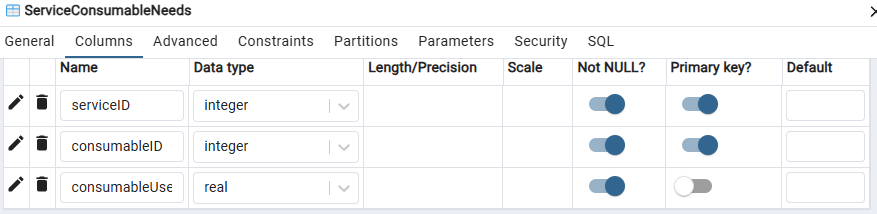


Рисунок 24 – Словарь данных таблицы Raitings ServiceConsumableNeeds

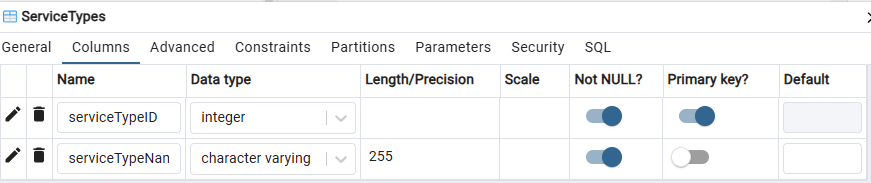


Рисунок 25 – Словарь данных таблицы ServiceTypes

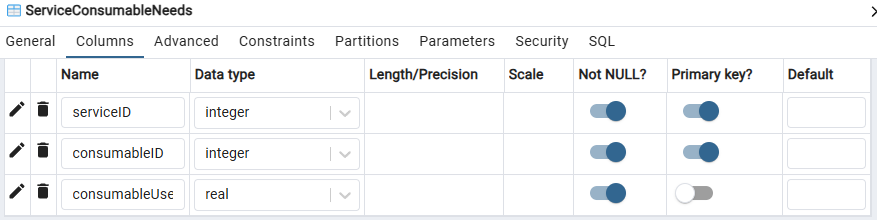


Рисунок 26 – Словарь данных таблицы ServiceConsumableNeeds

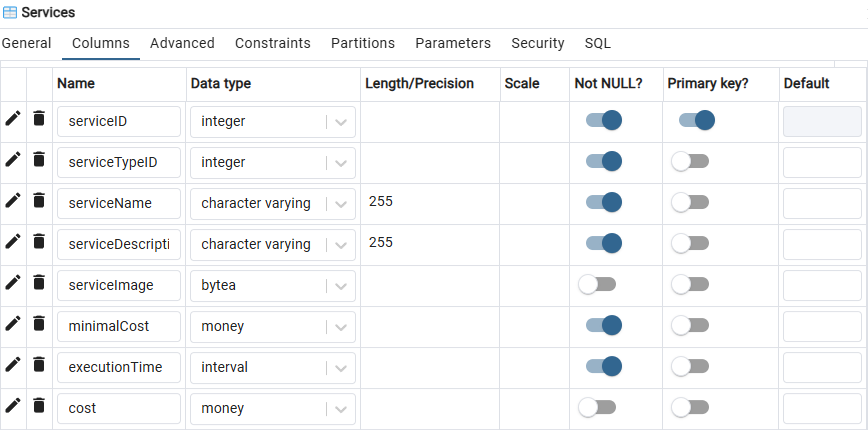


Рисунок 27 – Словарь данных таблицы Services

Файлы для импорта находятся в репозитории, в папке импорт данных, как представлено на рисунке ниже. (рисунок 28

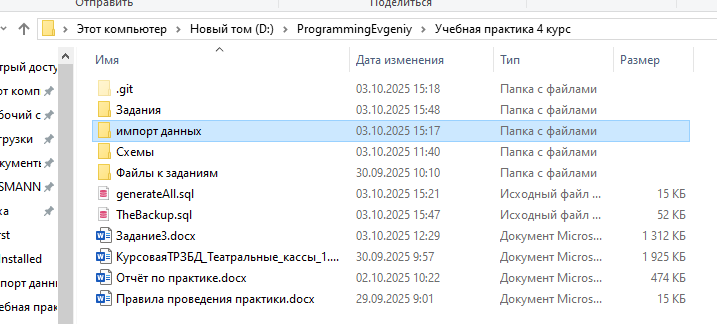


Рисунок 28 – Папка импорт данных

В импорте данных находятся 4 папки, как на представлено на рисунке ниже. (рисунок 29) При загрузке данных в БД нужно будет идти по порядку папок, чтоб соблюдать ограничения внешних ключей.

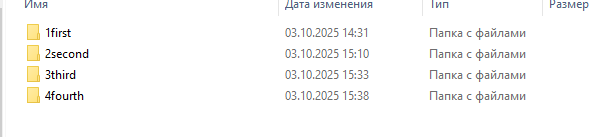


Рисунок 29 – Папки для импорта

Соответствующие данные для импорта названы так же, как и таблицы, что представлено на рисунке ниже. (рисунок 30)

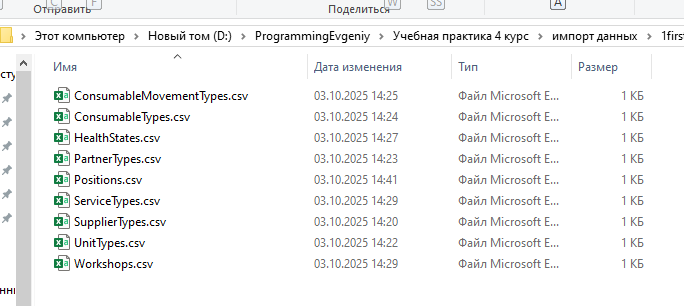


Рисунок 30 – csv файлы для импорта

Пример того, каким является содержание таких файлов импорта представлен на рисунке ниже. (рисунок 31)

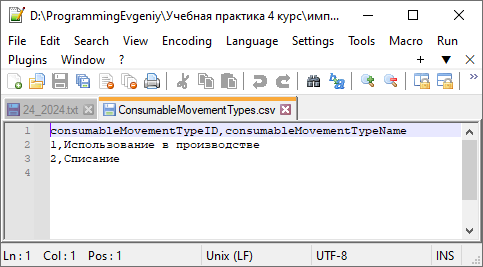


Рисунок 31 - Пример содержания файла импорта

Для импорта данных был использован клиент для управления БД – PG admin 4, как представлено на рисунке ниже. (рисунок 32)

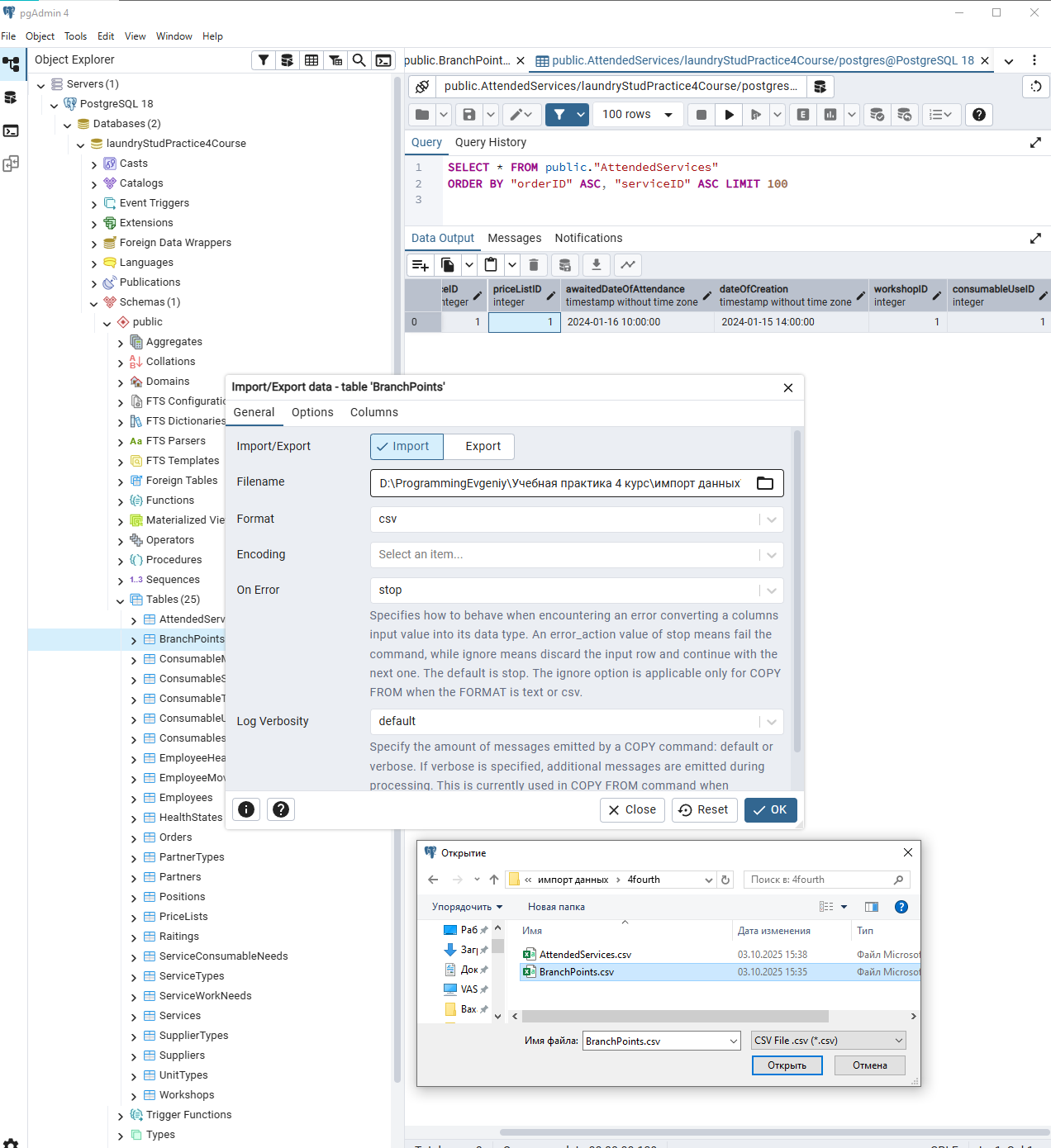


Рисунок 32 – Пример импорта данных через PG admin 4

Чтобы убедиться в корректности импорта данных следует просмотреть записи некоторых таблиц, как представлено на рисунках ниже. (рисунок 33-рисунок 36)

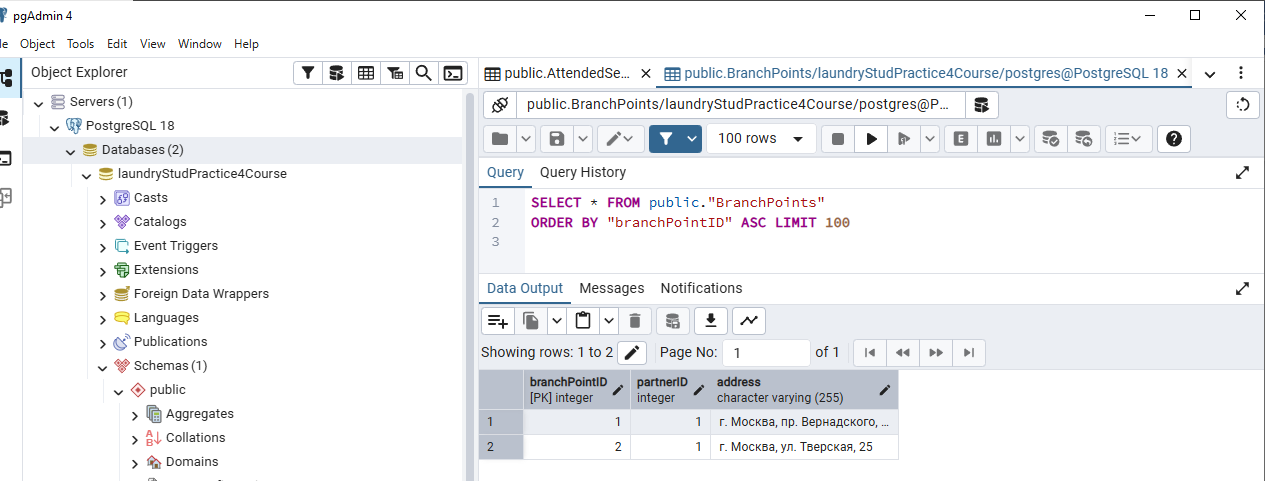


Рисунок 33 – Просмотр содержания таблицы с запросом

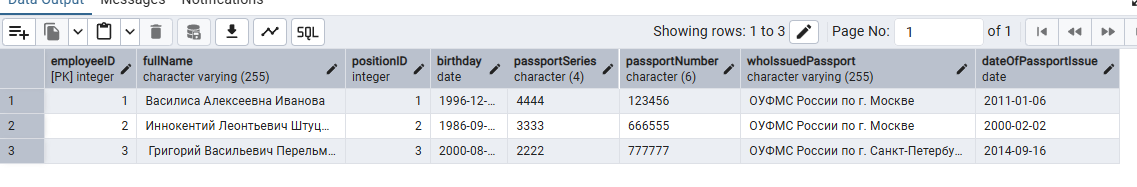


Рисунок 34 – Пример заполненных данных 1

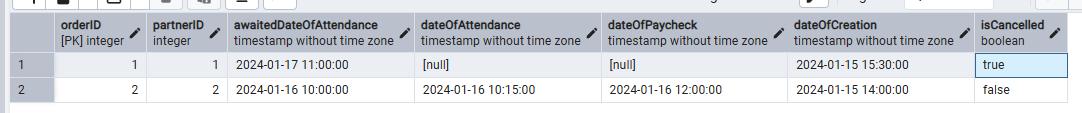


Рисунок 35 – Пример заполненных данных 2

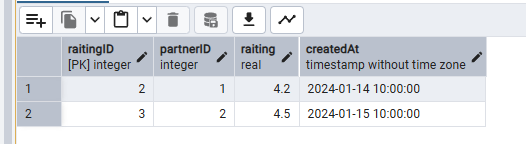


Рисунок 36 – Пример заполненных данных 3

Кроме того следует создать backup БД, как это было сделано и результат сохранения backup БД представлен на рисунках ниже. (рисунок 37-рисунок 38)

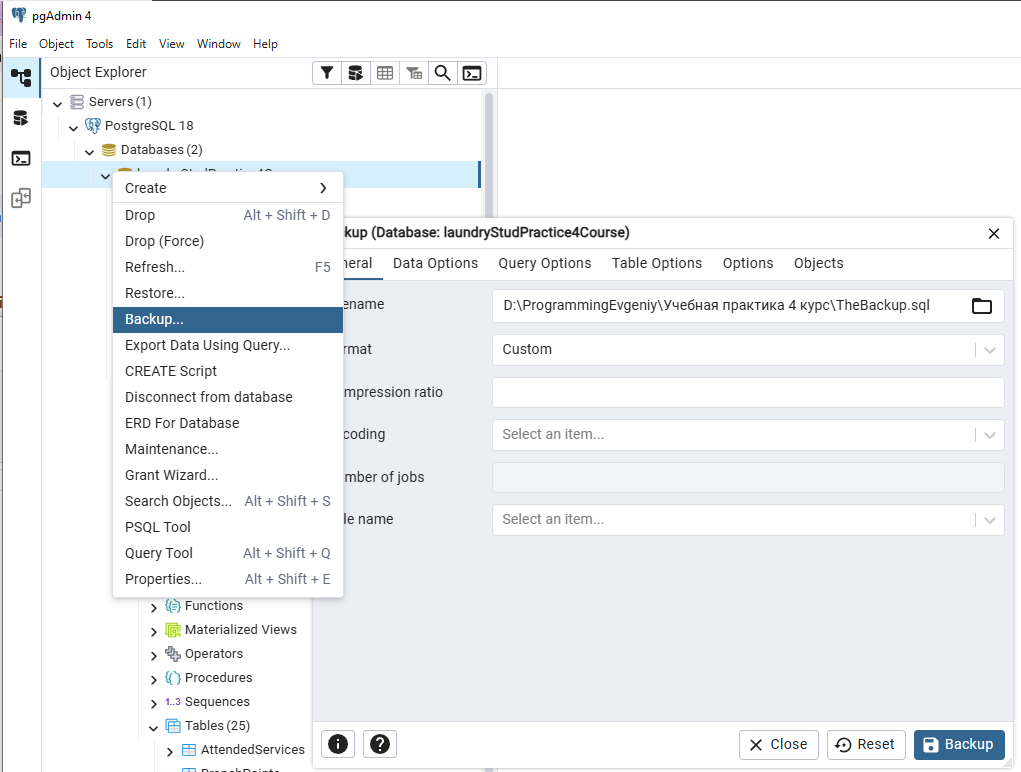


Рисунок 37 – Создание backup через PG admin 4

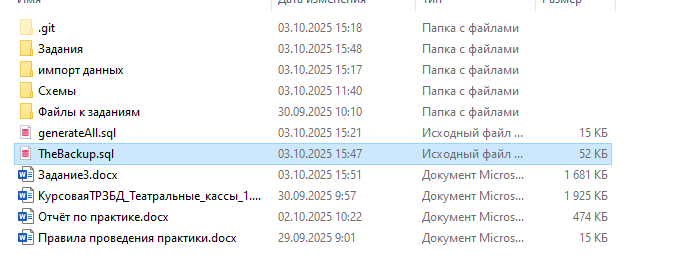


Рисунок 38 – Файл, результат backup

# Задание № 4 «Разработка программного продукта»

Цель работы - на основе разработанного ТЗ, диаграмм и БД разработать программный продукт, который будет включать все необходимые функции и соответствовать руководству по стилю. Приложение должно корректно работать и не должно происходить аварийного завершения работы.

Первым делом создаётся новый проект в VisualStudio Приложение WPF (Майкрософт), как представлено на рисунке ниже. (рисунок 39)

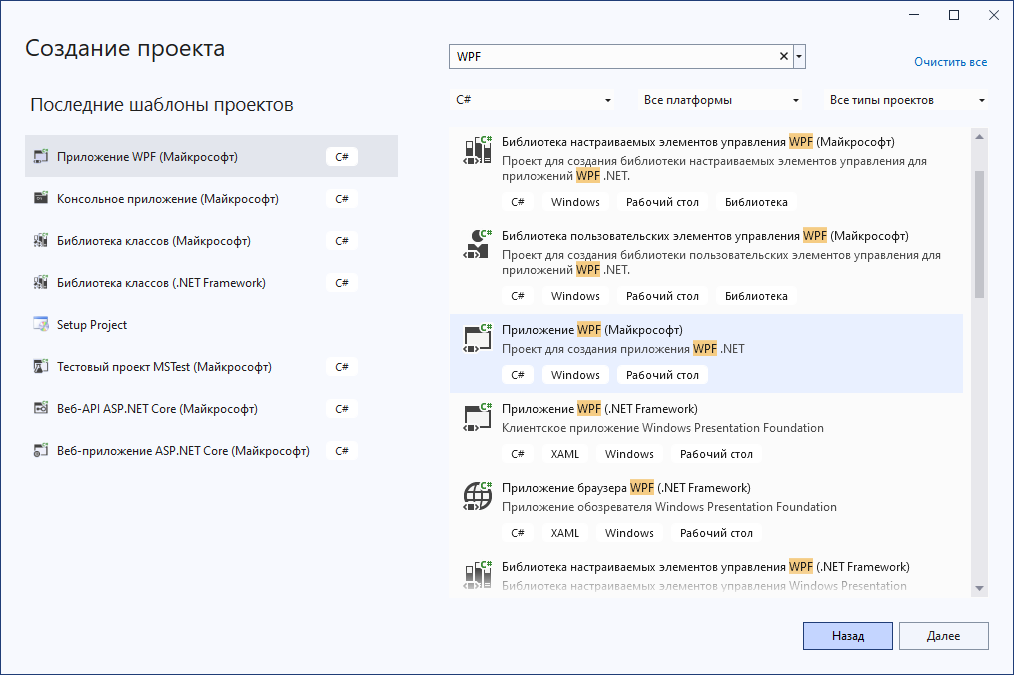


Рисунок 39 – Создание проекта приложения

Далее в Средства > Диспетчер пакетов NuGet > управление пакетами NuGet для решения скачиваем все нужные пакеты, как на представлено на рисунке ниже.(рисунок 40)

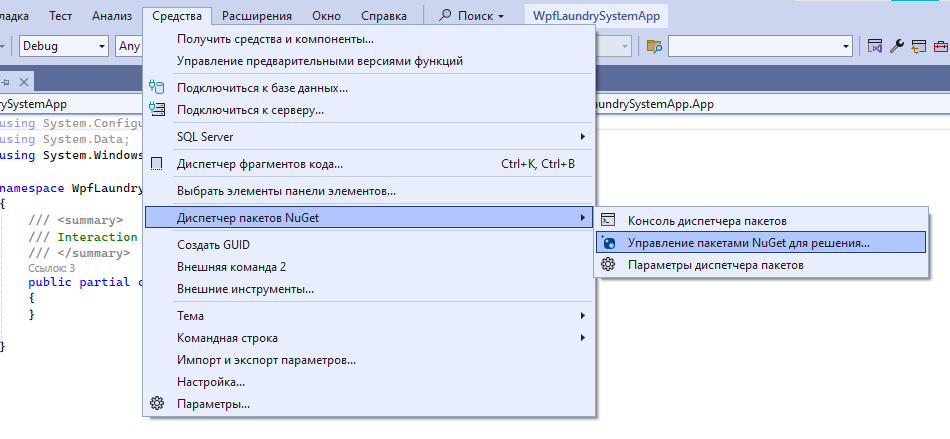
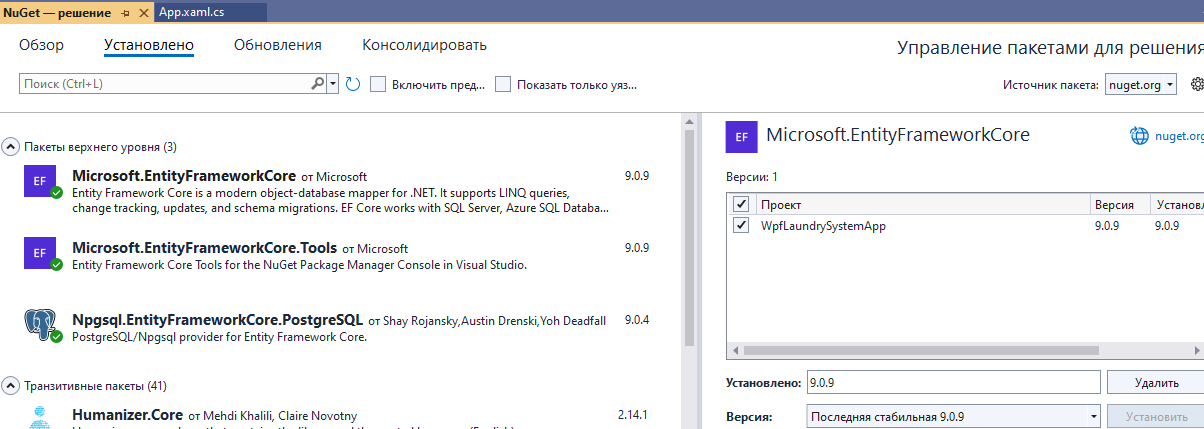


Рисунок 40 – Открытие управления пакетами NuGet

Это пакеты EntityFramework.Core EntityFramework.Tools и NpgSQL. EntityFramework.PostgreSQL



Затем в Средства > Диспетчер пакетов NuGet > консоль диспетчера пакетов, как на рисунке ниже. (рисунок 41)

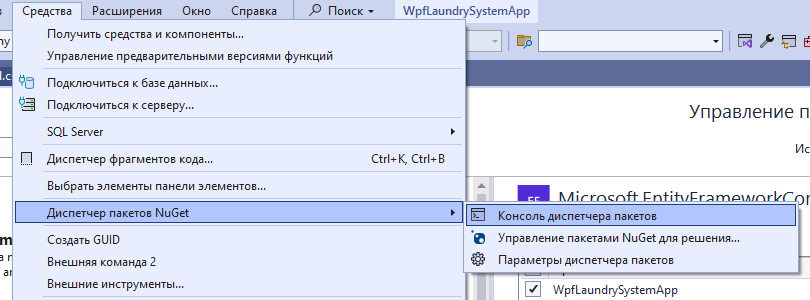


Рисунок 41 – Открытие консоли диспетчера пакетов

Вводим следующую команду для создания модели БД

dotnet ef dbcontext scaffold "Host=localhost;Port=5432;Database=laundryStudPractice4Course;Username=postgres;Password=POSTGREmoiseiev;SSL Mode=Prefer" Npgsql.EntityFrameworkCore.PostgreSQL -o Models -c ApplicationDbContext --data-annotations –force

Результатом команды стали файлы классов моделей сущностей, как на рисунке ниже. (рисунок 42)

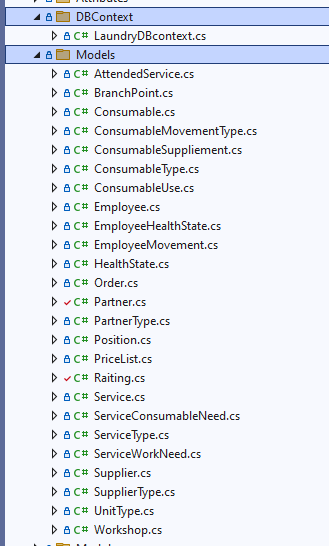


Рисунок 42 – Модели сущностей БД

В данной работе было решено подойти к разработке универсального класса интерфейса, поэтому создаём модуль DynamicTableCreator, суть которого будет в том, что с его помощью и использованием рефлексии C#, а также метаданных EntityFramework автогенерировать интерфейс (код модуля DynamicTableCreator можно просмотреть в репозитории гитхаб)

Для работы DynamicTableCreator классы моделей БД необходимо разметить специальносозданными атрибутами. Для этого были разработаны эти самые атрибуты. Их код представлен на рисунке ниже. (рисунок 43)

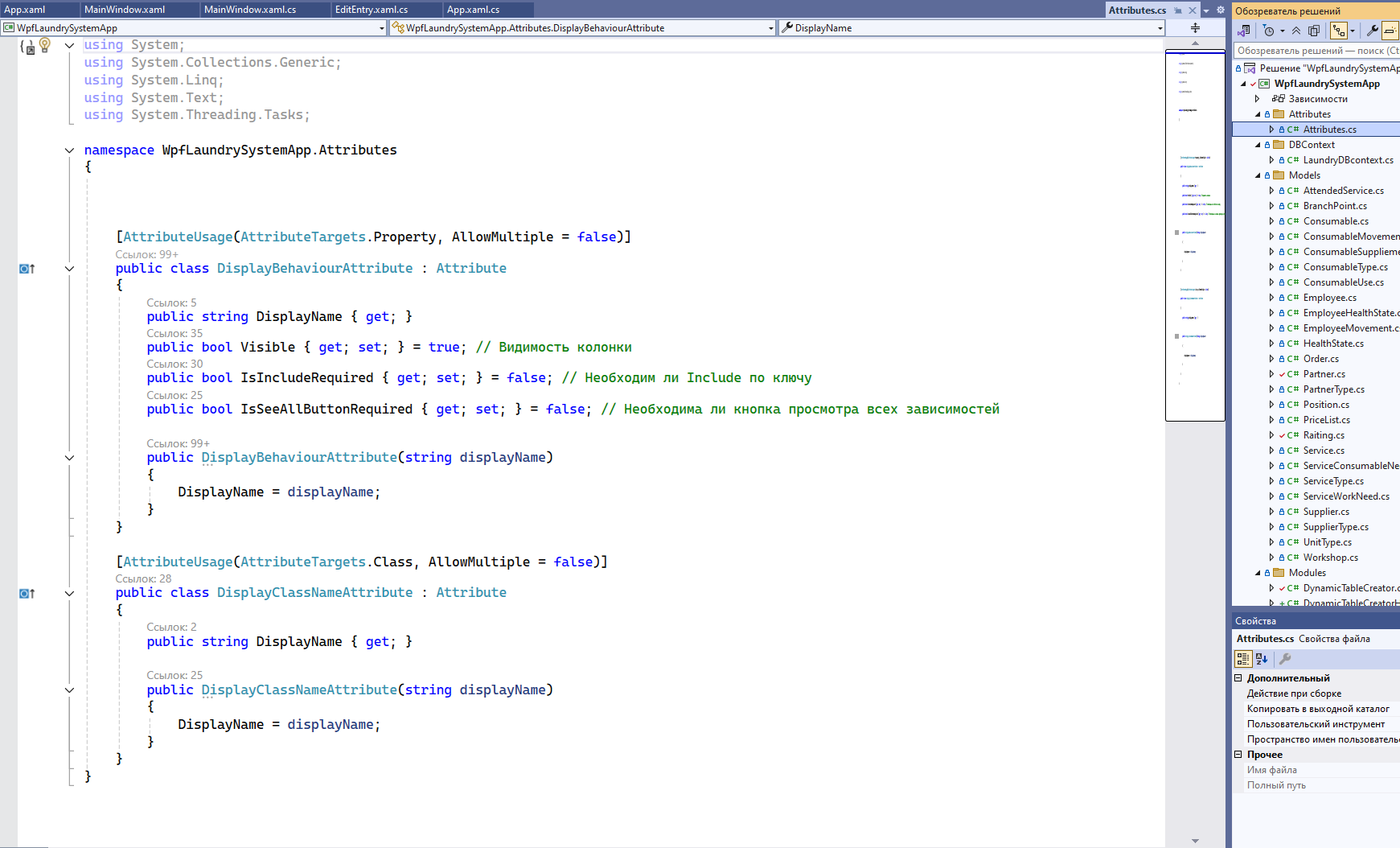


Рисунок 43 – Код собственных атрибутов

Пример разметки таким атрибутами на модели таблицы Partners, представлен на рисунке ниже. (рисунок 44)



Рисунок 44 – Пример разметки модели Partner

Далее создадим два окна, через которые будут происходить взаимодействия с БД: DisplayTable и EditEntry (их код также в репозитории). Следует настроить стиль в соответствии с ТЗ, для этого в App.xaml в Resources добавим необходимые настройки (весь код app.xaml также в репозитории)

По итогу получается следующее приложение:

По умолчанию запускается таблица партнёры. (рисунок 45)

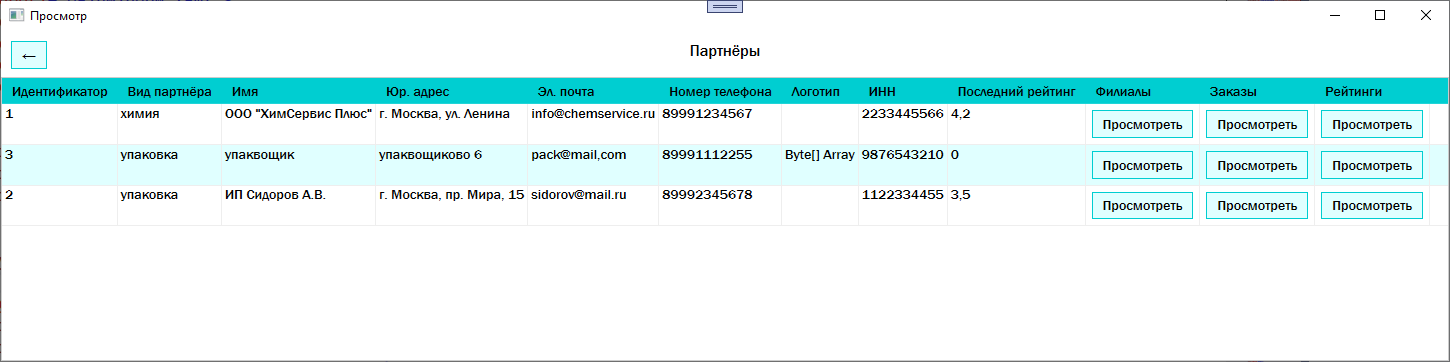


Рисунок 45 – Просмотр таблицы Партнёры

При нажатии ПКМ появляется контекстное меню и в нём 3 опции. (рисунок 46)

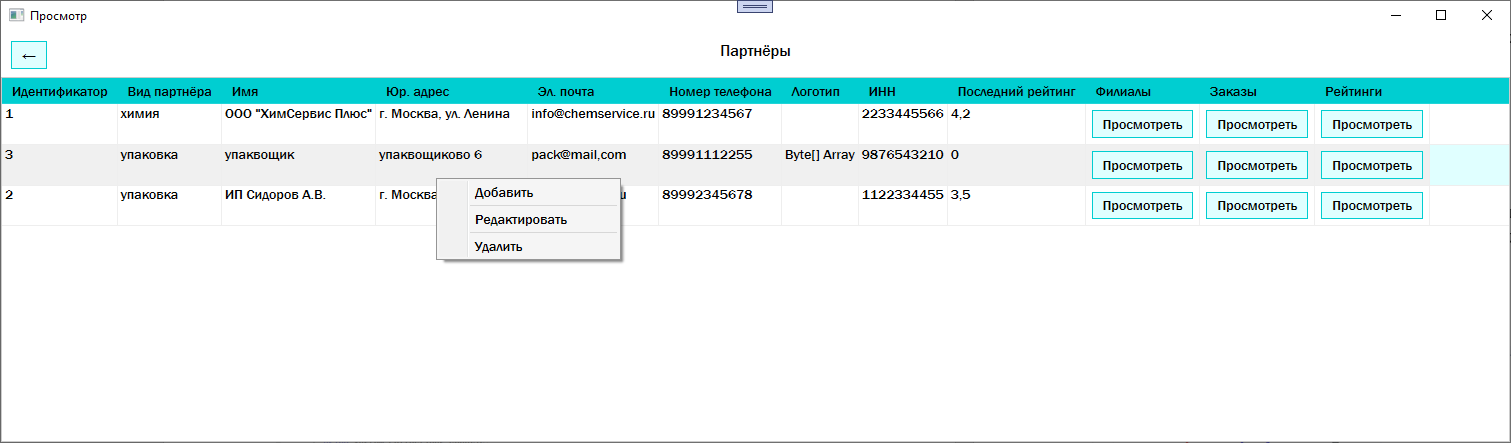


Рисунок 46 – Появление контекстного меню

Пункт добавление открывает новое окно с автоматически сгенерированными полями ввода. (рисунок 47)

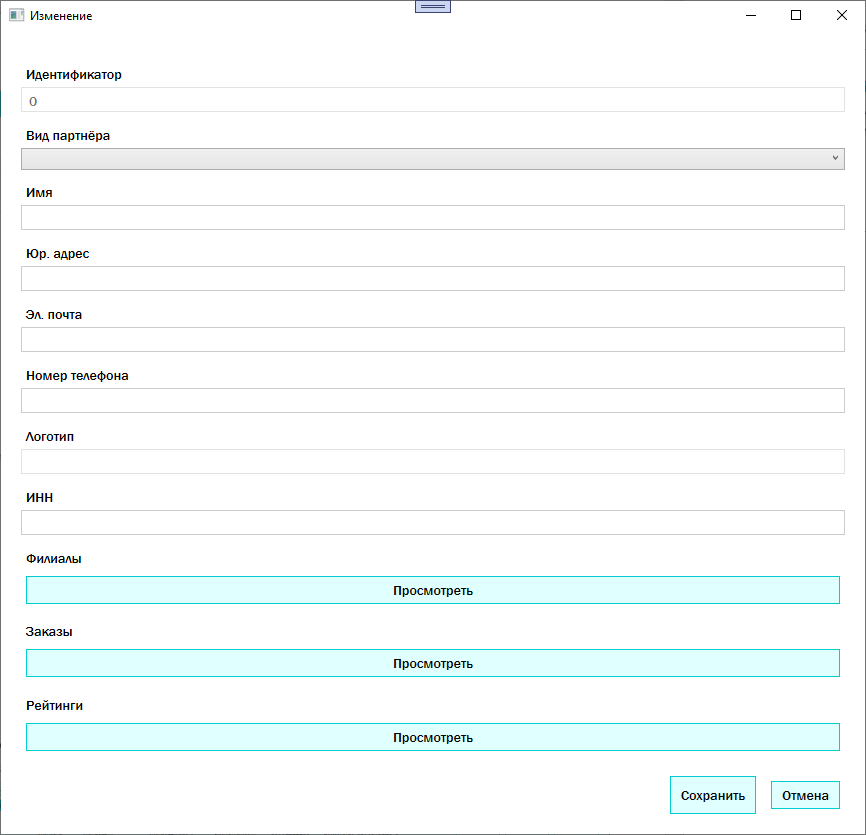


Рисунок 47 – Окно добавления записи

А вместо внешних ключей выпадающие списки (идентификатор и логотип вручную редактировать невозможно). (рисунок 48)

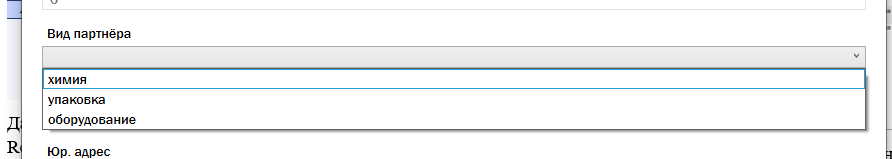


Рисунок 48 - Пример выпадающейго списка вместо внешних ключей

В окне возможно ввести данные, а затем сохранить. Пользователь получит сообщение о сохранении, а затем в таблице отобразиться новая запись. Введённые данные и результат работы программы представлены на рисунках ниже. (рисунок 49-рисунок 51)

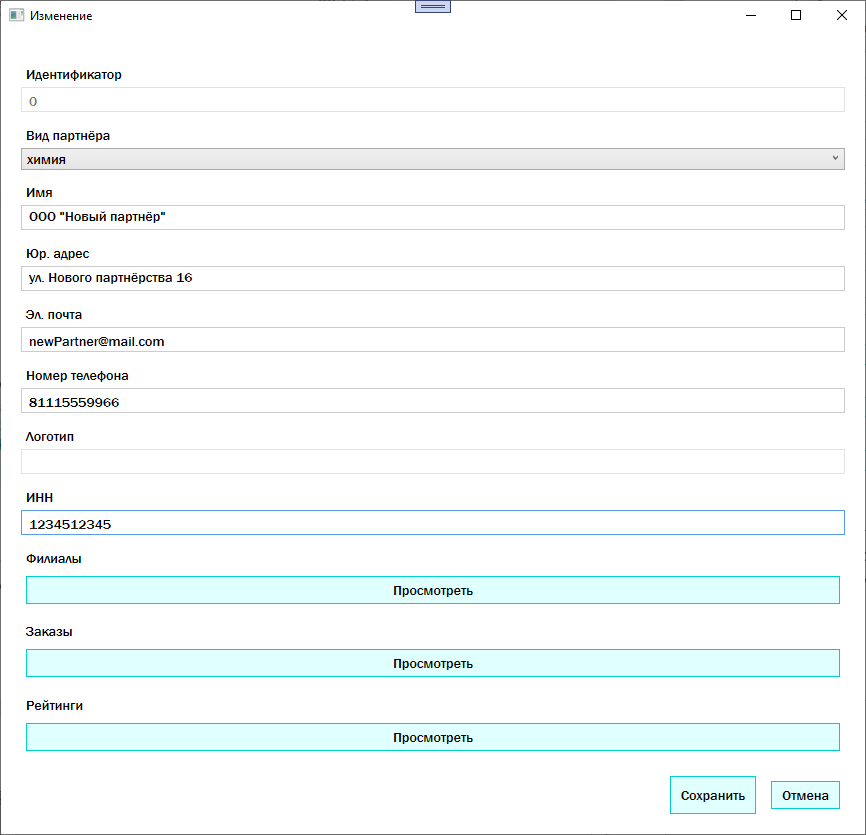


Рисунок 49 – Окно добавления партнёра с введёнными данными

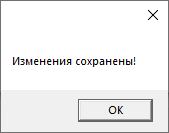


Рисунок 50 – Сообщение об успешном изменении данных

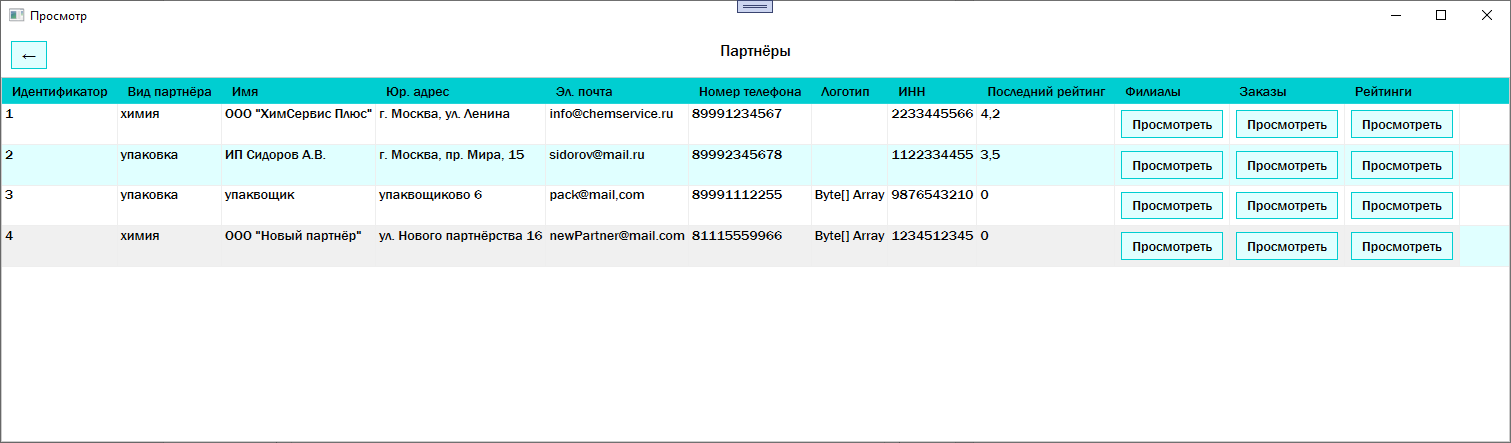


Рисунок 51– Результат добавление нового партнёра в таблицу

В приложении предусмотрено сообщение об ошибке пользователю, если не выбрать вид партнёра при сохранении, то будет выведено следующее сообщение, представленное на рисунке ниже. (рисунок 52)

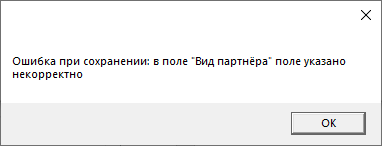


Рисунок 52 – Сообщение об ошибке при невыбранном поле

После нажатия ок пользователь сможет продолжить редактировать и всё же выбрать вид партнёра.

Редактирование происходит аналогично, только открывается окно с уже заполненными полями выбранной записи.

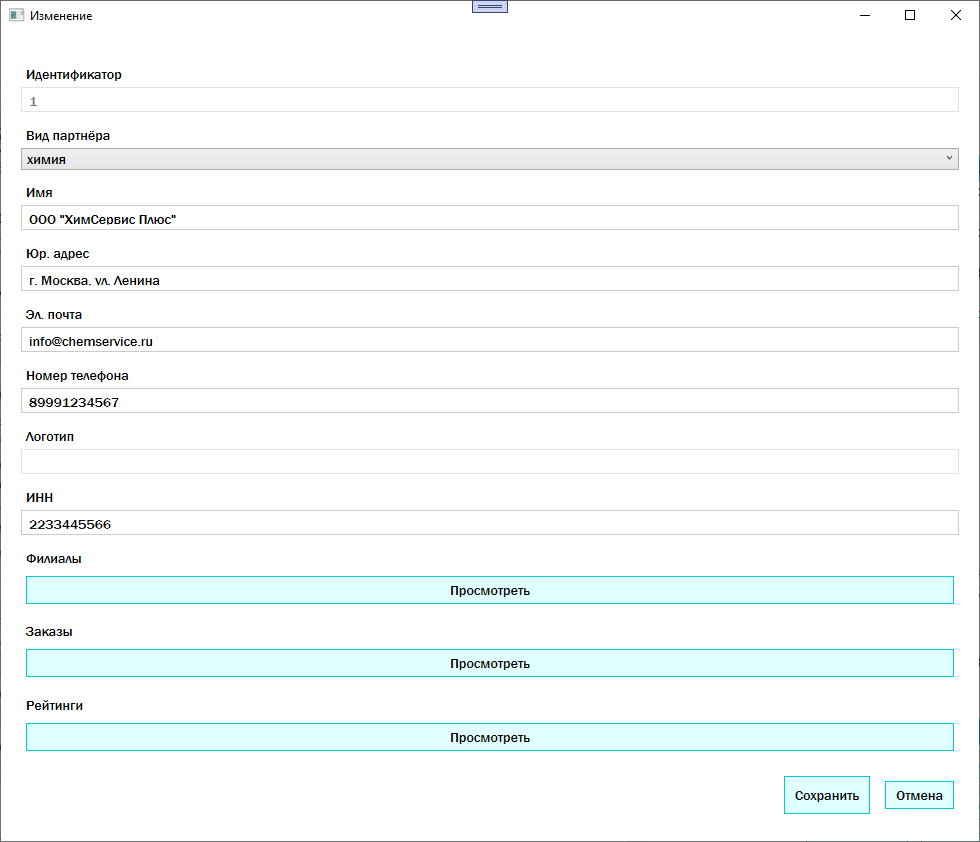


Рисунок 53 – Редактирование записи

Кроме того, если при изменении или добавлении пользователь нажмёт кнопку отмены, тогда партнёр не будет создан (или изменён).

Пункт удаление, соответственно, удаляет запись.

У партнёра есть три столбца с и коллекциями, что представлено на рисунке ниже. (рисунок 54)



Рисунок 54 – 3 столбца с коллекциями

Это их филиалы, их заказы и их рейтинги (история рейтингов)

Нажатие на кнопуку любой из них откроет советующую таблицу с записями, относящимися к выбранному партнёру. (рисунок 55)

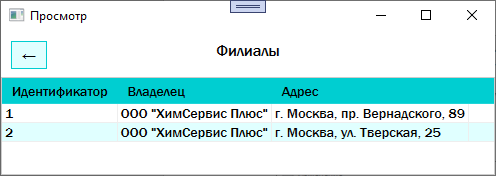


Рисунок 55 – Таблица филиалы ООО ХимСервис плюс

В этих таблицах абсолютно аналогично можно вызвать контекстное меню. (рисунок 56)

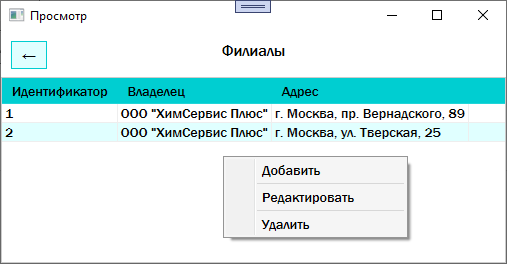


Рисунок 56 – Вызов контекстного меню в другой таблице

Которое вызывает соответствующее авто генерируемое окно изменений. (рисунок 57)



Рисунок 57 – Авто генерируемое окно изменения для другой таблицы

Чтоб вернуться на предыдущую таблицу можно нажать кнопку со стрелочкой влево. (рисунок 58)



Рисунок 58 – Кнопка со стрелочкой влево

У партнёра также отображается колонка с актуальным, то есть самым новым рейтингом. (рисунок 59)

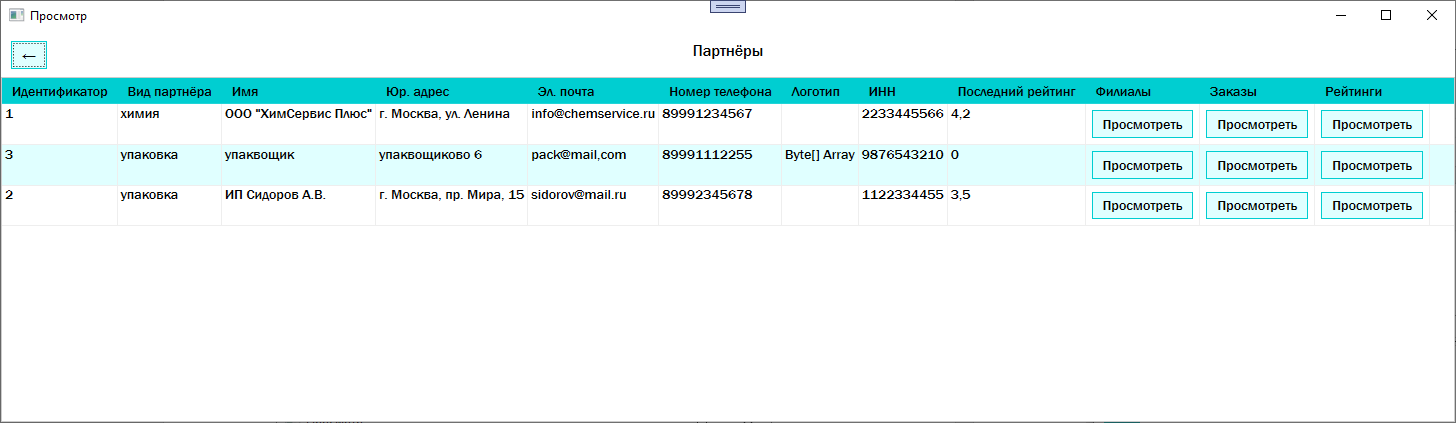


Рисунок 59 – Отображение актуального рейтинга партнёра

Остальные таблицы, которые можно просмотреть, представлены на рисунках ниже. (рисунок 61-рисунок 64)

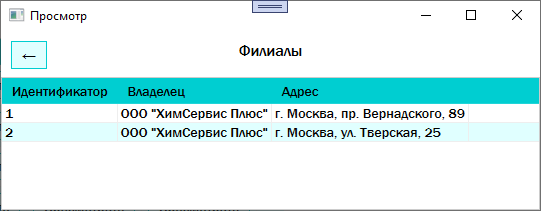


Рисунок 60 – Окно с записями филиалов

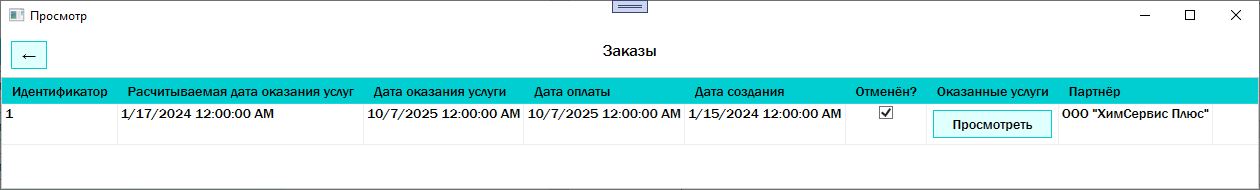


Рисунок 61 – Окно с записями заказов

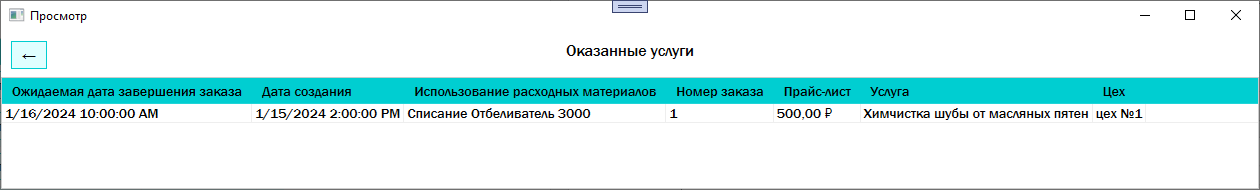


Рисунок 62 – Окно с записями оказанных услуг

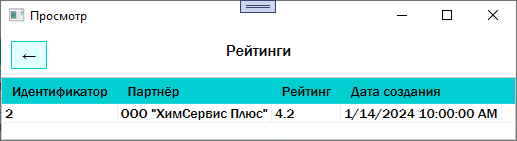


Рисунок 63 – Окно с записями рейтингов

Модули представлены на рисунках ниже. (рисунок 64-рисунок 65)

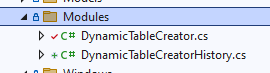


Рисунок 64 – Два модуля работы с динамическим генератором таблиц

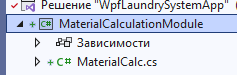


Рисунок 65 – Модуль расчёта значений материала

Модуль MaterialCalc – сделан отдельным модулем, согласно ТЗ практики.

# Задание № 5 «Тестирование программного продукта»

Цель работы - необходимо разработать позитивные и негативные тест-кейсы (не менее 3 каждого) для тестирования программных модулей. К ним необходимо разработать юнит-тесты.

Выполнение работы:

В решение добавляется Тестовый проект MSTest (Майкрософт), как представлено на рисуках ниже. (рисунок 66-рисунок 67)

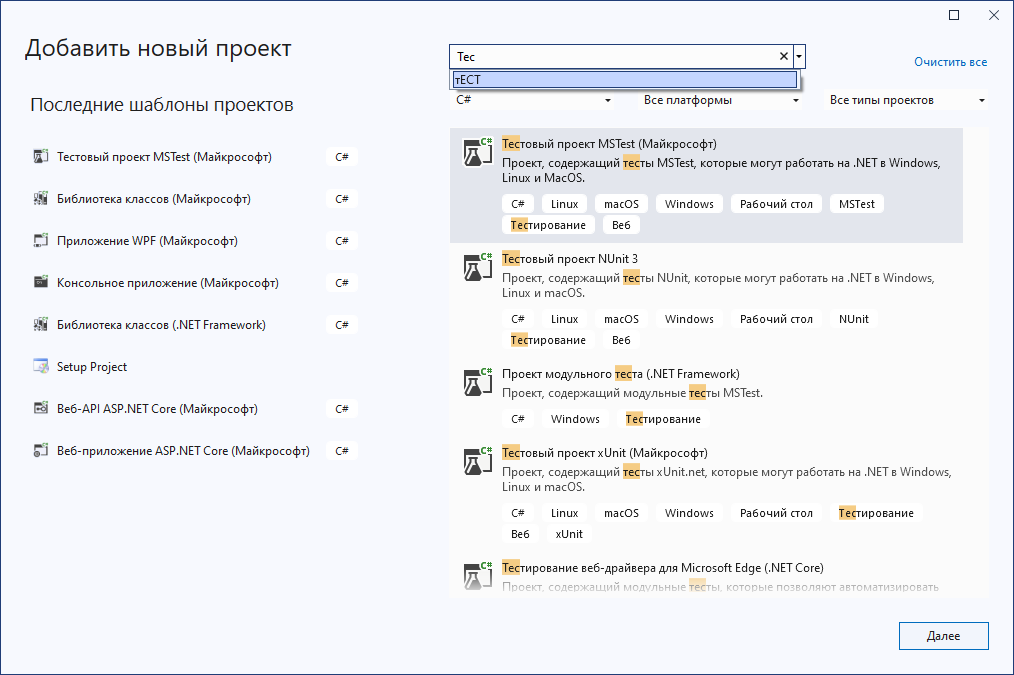


Рисунок 66 – Добавление проекта MSTest

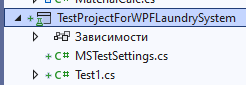


Рисунок 67 – Добавленный проект в решении

В зависимостях добавляем ссылку на проект, как представлено на рисунке ниже. (рисунок 68)

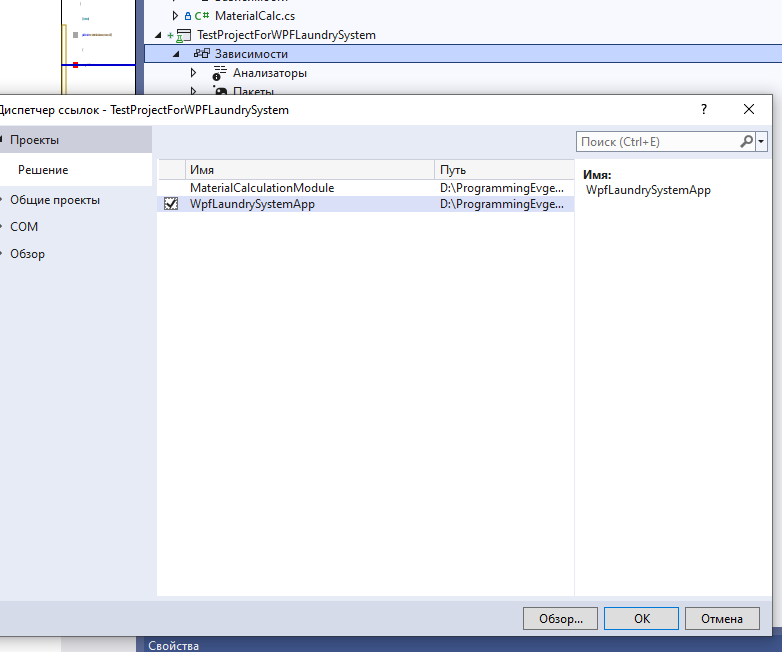


Рисунок 68 – Добавление зависимостей

В настройках конфиге тестового проекта поменяем целевую платформу на net8.0-Уwindows, как представлено на рисунке ниже.(рисунок 69)

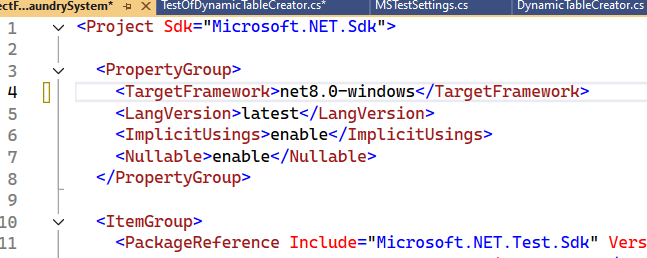


Рисунок 69 – Изменение целевой платформы проекта тестов

Пишем код тестов (код тестов представлен в репозитоии). После написания запускаем тесты. Результат прохождения тестов представлен на рисунке ниже. (Рисунок 70)

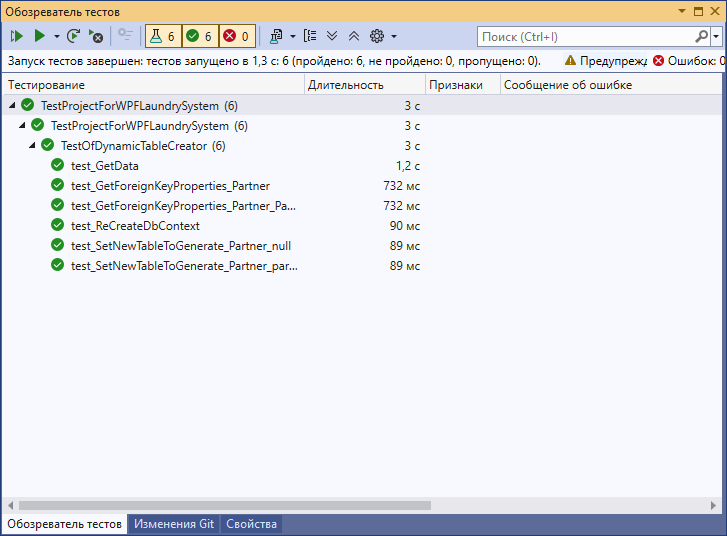


Рисунок 70 – Результат прохождения тестов

Кроме автоматизированного тестирования для программы также предусмотрены тест-кейсы, представленные в таблицах ниже. (таблица 1-таблица 6)

Таблица 1 – Тест 01 Открытие формы редактирования партнёра (OK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ПКП по партнёру | * Строка с партнёрам выделилась * Появилось контекстное меню с опциями: добавить, редактировать, удалить |
| 1. Нажать кнопку «Редактировать» | * Открывается окно редактирования * Данные из ранее выделенной строки партнёра введены в соответствующие поля нового окна |

Таблица 2 – Тест 02 Открытие формы добавления партнёра (OK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ПКП по партнёру | * Появилось контекстное меню с опциями: добавить, редактировать, удалить |
| 1. Нажать кнопку «Добавить» | * Открывается окно редактирования * Присутствуют пустые автоматически сгенерированные поля (все пустые, кроме идентификатора, он с значением 0) |

Таблица 3 – Тест 03 Открытие формы просмотра истории рейтинга партнёра (OK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ЛКМ по кнопке «Просмотреть» в столбце «Рейтинги» в строке с партнёром «ООО “ХимСервис Плюс”» | * Заголовок по центру сменился на «Рейтинги» * Строки с партнёрами пропали, и вместо них отображаются строки рейтингов * У строк рейтингов в столбце Партнёр указан «ООО “ХимСервис Плюс”» |

Таблица 4 – Тест 04 Попытка перейти назад до открытия других форм (NOK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ЛКМ по кнопке с символом «←» | * Кнопка подсветилась при наведении * После нажатия ничего не произошло, строки и заголовок окна не сменились |

Таблица 5 – Тест 05 Попытка удалить объект, на который ссылаются другие объекты (NOK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ПКП по строке с партнёром «ООО “ХимСервис Плюс”» | * Строка с партнёром выделилась * Появилось контекстное меню с опциями: добавить, редактировать, удалить |
| 1. Нажать кнопку «удалить» | * Выводиться сообщение об ошибке удаления * В сообщении сказано: Произошла ошибка удаления. Перед удаление объекта убедитесь, что у него нет зависимых (В таблице зависимые объекты можно посмотреть по кнопке "Просмотреть"), |

Таблица 6 - Тест 06 Попытка добавить объект, у которого не выбраны внешние ключи (NOK)

|  |  |
| --- | --- |
| Действия | Ожидаемый результат |
| 1. Запустить приложение | * Открывается окно «Партнёры», где в строках таблицы отображается информация по партнёрам |
| 1. Нажать ПКП по партнёру | * Появилось контекстное меню с опциями: добавить, редактировать, удалить |
| 1. Нажать кнопку «Добавить» | * Открывается окно редактирования * Присутствуют пустые автоматически сгенерированные поля (все пустые, кроме идентификатора, он с значением 0) |
| 1. Нажать кнопку «Сохранить» | * Выведется сообщение:   Ошибка при сохранен в поле “Вид партнёра” поле указано некорректно |

# Задание №6 «Работа с GitHub»

Цель работы - необходимо загрузить все практические результаты в отдельный репозиторий с названием, совпадающим с названием проекта. Использовать только Git Bash.

Выполнение работы:

Для того, чтоб открыть Git Bash в паке следует нажать ПКМ > open Git Bash here.

Далее командной строке написать:

git init

git add \*

git commit Комментарий коммита

Результатом стане созданный репозиторий с первым коммитом, как представлено на рисунке ниже. (рисунок 71)

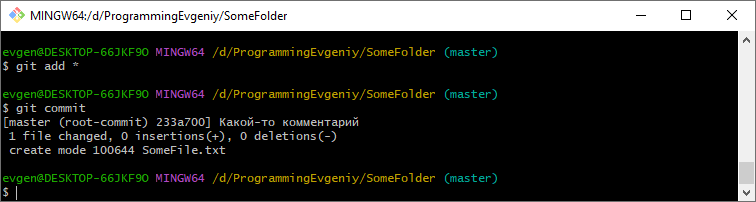


Рисунок 71 – Созданный репозиторий в GitBash

На сайте github необходимо создать. Репозиторий, как представлено на рисунке ниже. (рисунок 72)

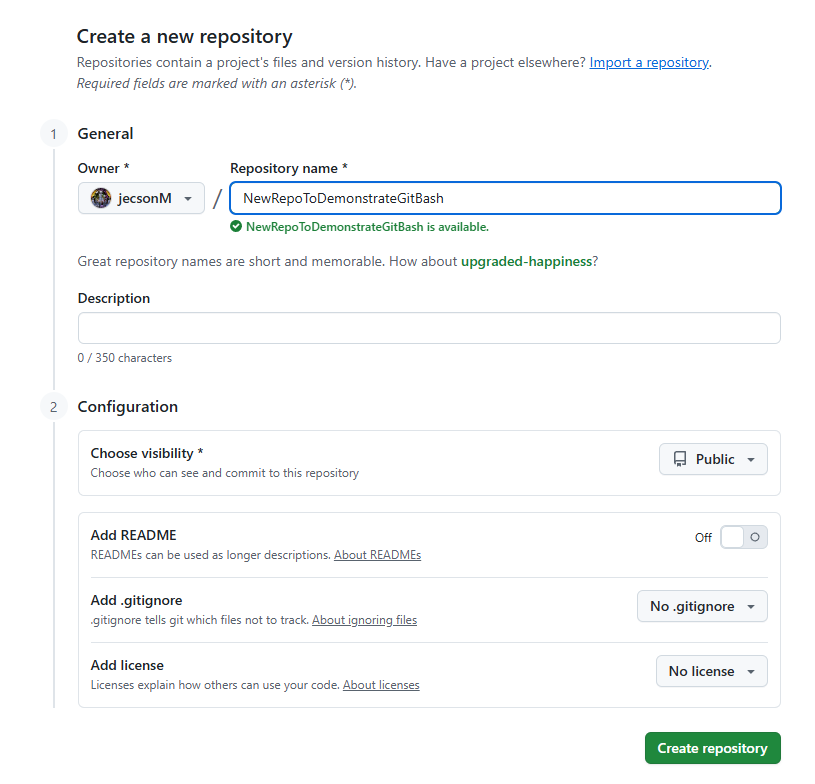


Рисунок 72 – Создание репозитория на GitHub

В gitBash прописать команды (ссылка на репозиторий зависит от ссылки созданного репозитория, в данном случае https://github.com/jecsonM/My-studPractice4Course.git):

git remote add origin https://github.com/jecsonM/My-studPractice4Course.git

git push -u origin master

Так результаты создания такого репозитория представлены на рисунках ниже.

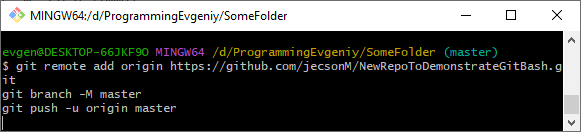


Рисунок 73 – Ввод команд на добавление origin репозиторию

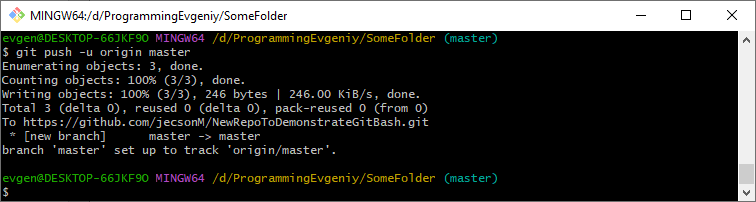


Рисунок 74 – результат отправки репозитория на Github

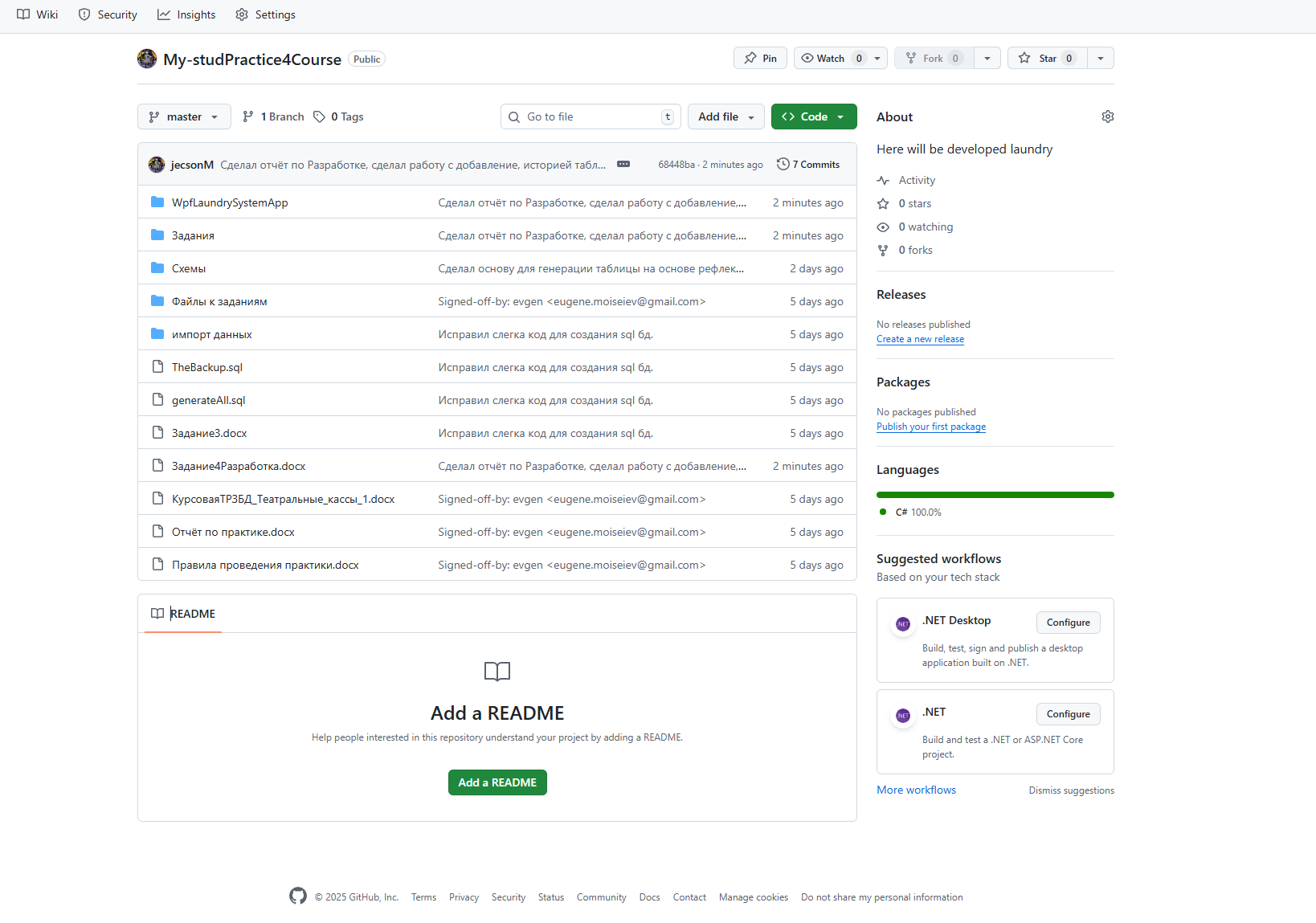


Рисунок 75 – репозиторий GitHub с данными

Заключение

В заключение можно отметить, что прохождение учебной практики позволило не только закрепить полученные теоретические знания, но и применить их на практике в процессе разработки настольного приложения на языке C#. В ходе работы были реализованы все основные этапы создания программного продукта, включая проектирование базы данных в MS SQL Server, разработку архитектуры приложения и создание графического интерфейса с использованием технологии WinForms. Полученный опыт способствовал более глубокому пониманию принципов построения и функционирования программных систем, а также особенностей их тестирования и отладки.

Все цели и задачи, поставленные перед началом практики, были успешно выполнены. Работа над проектом позволила освоить современные подходы к разработке программных решений, научила планировать собственную деятельность, контролировать результаты и принимать ответственные технические решения. Практика также способствовала развитию внимательности, аккуратности и умения доводить начатое до конца.

Таким образом, учебная практика стала важным этапом профессионального становления, помогла сформировать целостное представление о процессе разработки программного обеспечения и заложила основу для дальнейшего развития в сфере программирования и проектирования информационных систем.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Троелсен, Э. Язык программирования C# 9 и платформа .NET 5 / Э. Троелсен, Ф. Джепикс. — 8-е изд. — Москва: Вильямс, 2022. — 1056 с.
2. Шилдт, Г. Полный справочник по C# / Г. Шилдт. — 10-е изд. — Москва: Вильямс, 2021. — 912 с.
3. Нагаев, А. WinForms. Разработка оконных приложений на C# / А. Нагаев. — Санкт-Петербург: БХВ-Петербург, 2020. — 368 с.
4. Головач, И. Программирование на C#. Учебное пособие / И. Головач. — Москва: Издательство Юрайт, 2021. — 432 с.
5. Уильямс, М. Microsoft SQL Server 2019. Руководство администратора / М. Уильямс. — Москва: Вильямс, 2020. — 784 с.
6. Хори, К. SQL Server. Проектирование и разработка баз данных / К. Хори. — Санкт-Петербург: Питер, 2019. — 624 с.
7. ГОСТ 19.401-78. Текст программы. Общие требования к содержанию и оформлению. — Москва: Издательство стандартов, 1978. — 12 с.
8. ГОСТ 19.402-78. Описание программы. Общие требования к содержанию и оформлению. — Москва: Издательство стандартов, 1978. — 10 с.
9. ГОСТ 19.505-79. Руководство оператора. Общие требования к содержанию и оформлению. — Москва: Издательство стандартов, 1979. — 8 с.
10. ГОСТ 19.301-79. Программа и методика испытаний. Общие требования. — Москва: Издательство стандартов, 1979. — 10 с.

ПРИЛОЖЕНИЕ А

(обязательное)

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Санкт-Петербургский государственный политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Институт среднего профессионального образования

Информационная система для производственно-сервисного предприятия “Чистая Планета”

**Техническое задание**

Листов \_\_7\_\_

ВЫПОЛНИЛ

Студент группы 22290907/1097

\_\_\_\_\_\_\_\_\_\_ Моисеев Е.В.

\_\_\_\_.\_\_\_\_\_\_\_\_\_. 2025

2025

1. **ВВЕДЕНИЕ**
   1. Полное наименование программной разработки: Информационная система для производственно-сервисного предприятия «Чистая Планета».
   2. Информационная система для производственно-сервисного предприятия «Чистая Планета» представляет из себя защищённую базу данных и интегрированное с базой данных приложение.
   3. Программа предназначена для доступа в компании сотрудникам с целью реализации их бизнес-задач.
2. **ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ**

Разработка ведётся на основании задания к курсовому проекту по профессиональному модулю ПМ 02 «Осуществление интеграции программных модулей» и утверждена Председателем предметно-цикловой комиссии отделения информационных технологий ФГАОУ ВО «СПбПУ» Института среднего профессионального образования.

1. **НАЗНАЧЕНИЕ РАЗРАБОТКИ**
   1. Основное назначение программного продукта: оптимизация деятельности компании-заказчика.
   2. Эксплуатационное назначение программного продукта: программа предназначена для узкого круга пользователей - сотрудников компании “Чистая планета”.
2. **ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ**
   1. Требования к функциональным характеристикам:

* ИС должна осуществлять хранение следующих сущностей: услуги и прайс-лист, склад и расходные материалы, производство (стирка и чистка), сотрудники, партнеры (пункты приема-выдачи);
* Данные о партнёре включают в себя: тип (розничный пункт, корпоративный клиент), наименование, юридический адрес, ИНН, ФИО руководителя, контактные данные (телефон, email), логотип, рейтинг, адреса пунктов, история оказанных услуг для последующего расчета бонусов или скидок;
* ИС должна реализовывать бонусную систему, при которой партнёры получают комиссию от стоимости заказов;
* Все заказы в ИС должны осуществляться в рублях;
* ИС должна реализовывать роль менеджера с соответствующему этой роли функционалу: поиск и регистрация новых партнеров, изменение рейтинга партнера, ведение истории изменений рейтинга, прием заявок от партнеров, формирование для них коммерческого предложения, формирование предложения партнёру с указанием срока и стоимости оказываемых услуг, предоставляет прайс-лист новым партнёрам;
* Заказ формируется менеджером, оформленный заказ включает в себя все согласованные услуги и даты выполнения к каждой услуге, стоимость к каждой услуге;
* ИС должна обеспечивать возможность отмены заказа до внесения предоплаты в течение 3-ёх дней;
* Система должна обеспечить автоматическую отмену заказа, если он не оплачен в течение 3-ёх дней и уведомление об отмене партнёру;
* ИС хранит следующие данные о сотруднике: дата рождения, паспортные данные, банковские реквизиты, должность, состояние здоровья (для допуска к работе с химическими веществами);
* ИС должна обеспечивать для производственного персонала разграничение допуска к работе с конкретным оборудованием;
* ИС должна обеспечивать формирование статистики для аналитиков о перемещении сотрудников при использовании их карт сотрудников с системой турникетов;
* Пользователям с ролью Технолога должен быть обеспечен доступ к заказам материалов (расходных средств);
* ИС хранит следующую информацию о материалах (расходных средствах): тип (моющее средство, растворитель, отбеливатель, пакеты, теги-ярлыки), наименование, поставщик, количество в упаковке, единица измерения (литры, кг, штуки), описание, изображение, стоимость, количество на складе, минимальный запас, история изменений количества материалов на складе;
* ИС должна организовывать работу склада и реализовывать следующие функции при работе с материалами: регистрация поступлений материалов, резервирование материалов для выполнения заказов, отпуск и списание материалов со склада в производство, прием готовых заказов от производства и их выдача партнерам, получение информации о текущих остатках и движении товарно-материальных ценностей по складу;
* ИС должна хранить следующие сведения о поставщике: тип (химия, упаковка, оборудование), наименование, ИНН, история поставок материалов;
* ИС должна обеспечивать доступ менеджерам по изменению списка поставщиков (добавление, удаление, изменение);
* ИС хранит следующие сведения об услуге: код услуги, тип (стирка, химчистка, ремонт), наименование, описание, изображение (например, иконка), минимальная стоимость для партнера, норма времени на выполнение, себестоимость (расчетная), номер цеха, количество персонала на выполнение, необходимые материалы для оказания услуги (нормы расхода);
  1. Требования к надежности:
* Использование лицензированного программного обеспечения;
* Проверка программы на наличие вирусов;
* ИС должна реализовывать обработку исключительных ситуаций в приложении;
* ИС должна уведомлять пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, предупреждать о неотвратимых операциях. Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления;
* Организация бесперебойного питания.
  1. Требования к составу и параметрам технических средств

Для нормального функционирования данной информационной системы необходим компьютер, клавиатура, мышь и следующие технические средства:

- процессор Intel или другой совместимый с архитектурой х86, 64 битный;

- объем свободной оперативной памяти ~2 Гб;

- объем необходимой памяти на жестком диске ~150Гб;

- стандартный VGA-монитор или совместимый;

- стандартная клавиатура;

- манипулятор «мышь».

* 1. Требования к информационной и программной совместимости

Для полноценного функционирования данной системы необходимо наличие операционной системы новее Microsoft Windows 10 или совместимой. Язык интерфейса – русский.

* 1. Для работы системы требуются пакеты .Net Framework.
  2. Требования к маркировке и упаковке

Программа должна поставляться в виде: исходного кода приложения, исполняемых файлов, файлов скрипта БД и прочих графических, текстовых файлов.

* 1. Требования к транспортировке и хранению

Необходимо загрузить исходный код метода расчета материала в отдельный репозиторий с названием, совпадающим с названием проекта. Все результаты должны быть переданы путем загрузки файлов на предоставленный репозиторий системы контроля версий.

* 1. Специальные требования

Теоретическая часть включает подробное описание работы с приложением, разработанные на этапе проектирования программной системы. Практическая часть включает разработку и реализацию программных модулей программного продукта с использованием среды программирования.

1. **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

Технико-экономические показатели не рассчитываются.

1. **СТАДИИ И ЭТАПЫ РАЗРАБОТКИ**

В таблице А.1 представлены стадии и этапы разработки.

Таблица А.1 – Стадии и этапы разработки

| **Содержание стадии** | **Содержание этапа** | **Срок 2025 г.** | | **Форма отчетности** |
| --- | --- | --- | --- | --- |
| **начало** | **конец** |
| Техническое задание | Составление технического задания | 29.09.2025 | 30.09.2025 | Техническое задание |
| Эскизный проект | Проектирование программы | 01.10.2025 | 03.10.2025 | UML-диаграммы |
| Рабочий проект | Разработка спецификаций | 04.10.2025 | 06.10.2025 | Спецификации отдельных компонентов |
| Составление программы | 06.10.2025 | 08.10.2025 | Программная документация |
| Приёмо-сдаточные испытания | 09.10.2025 | | Протокол испытаний |
| Приёмка | Защита результата учебной практики | 10.10.2025 | | Оценка за учебную практику |

1. **ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ**
   1. Порядок контроля

Контроль выполнения должен осуществляться руководителем учебной практики (преподавателем) в соответствие с п.7.

* 1. Порядок приемки

Приемка должна осуществляться с участием руководителя после проведения приемо-сдаточных испытаний. В результате защиты учебной практики должна быть выставлена оценка за курсовое проектирование.

ПРИЛОЖЕНИЕ Б

(обязательное)

**Руководство по стилю**

Все экранные формы пользовательского интерфейса должны иметь заголовок. Кроме того, на главной форме должен быть установлен логотип (представлен в ресурсах). Логотип не искажать: не менять изображение, пропорции изображения, цвет.

Для приложения должна быть установлена иконка (дана в ресурсах), если это реализуемо в рамках платформы.

Использовать шрифт Franklin Gothic Medium.

В качестве основного фона используется белый цвет, дополнительного фона используется цвет #E0FFFF. Для акцентирования внимания пользователя на целевом действии интерфейса используется цвет #00CED1.

В таблице Б.1 представлены примеры исползуемых цветов.

Таблица Б.1 – Примеры импользуемых цветов

|  |  |  |
| --- | --- | --- |
| Основной фон | Дополнительный фон | Акцентирование внимания |
|  | #E0FFFF | #00CED1 |
|  |  |  |

**АТТЕСТАЦИОННЫЙ ЛИСТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)**

По профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность 09.02.07 «Информационные системы и программирование»\_\_\_\_\_\_\_

(код и наименование специальности)

Студент(ка) 4 курса 22290907/1097 группы

Моисеев Евгений Васильевич

(Фамилия, имя, отчество)

Место прохождения практики: \_\_\_\_\_ УВЦ, пр. Энгельса, 23\_\_\_\_\_\_\_\_\_\_\_

(наименование и адрес организации)

Период прохождения практики

с «29» сентября 2025 г. по «11» октября 2025 г.

Виды и качество выполнения работ

|  |  |  |
| --- | --- | --- |
| Виды выполненных работ обучающимся  во время практики | Объем работ, час. | Качество выполнения работ (оценка по пятибалльной системе) |
| Участие в выработке требований к программному обеспечению | 9 |  |
| Стадии проектирования программного обеспечения | 20 |  |
| Разработка модулей программного обеспечения | 22 |  |
| Тестирование программных модулей и их интеграции | 12 |  |
| Разработка программной документации и стандарты кодирования | 9 |  |

Характеристика учебной/профессиональной деятельности обучающегося во время учебной практики (по профилю специальности):

Общие и профессиональные компетенции, предусмотренные программой практики, освоены **/** не освоены.

(нужное подчеркнуть)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики Хисамутдинова А.С. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)