МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по производственной практике**

**ПМ.02. Осуществление интеграции программных модулей**

Студент

Чемоданов Павел Юрьевич

Группа 22П-1

Специальность 09.02.07 Информационные системы и программирование

Руководитель практики от колледжа:

Пентин Николай Сергеевич

Руководитель практики от организации:

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_* Павленко Сергей Михайлович

подпись

УТВЕРЖДАЮ:

Директор

Устюжанинов Вадим Борисович \_\_\_\_\_\_\_\_\_\_\_\_

Наименование организации

ООО "ЭЛЕКТРОН-СОФТ"

\_\_\_\_\_\_\_\_\_\_\_/\_ \_\_\_\_\_Павленко С.М.\_

подпись расшифровка

М. П.

2025 уч. год

**Содержание**

1. Содержание
2. Характеристика объекта практики (юридический адрес, специализация, структура)
3. Описание рабочего места
4. Состав программного и технического обеспечения, имеющегося на предприятии, их назначение.
5. Описание выполненных видов работ
   1. Разработан и обоснован вариант интеграционного решения с помощью графических средств среды разработки.
   2. Результаты верно сохранены в системе контроля версий.
   3. В системе контроля версий выбрана верная версия проекта; - проанализирована его архитектура.
   4. Архитектура доработана для интеграции нового модуля.
   5. Выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости).
   6. Выполнена доработка модуля и дополнительная обработка исключительных ситуаций, в том числе, с созданием классов-исключений (при необходимости).
   7. Определены качественные показатели полученного проекта.
   8. Результат интеграции сохранен в системе контроля версий.
   9. Протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды.
   10. Разработан тестовый сценарий и тестовые пакеты.
   11. Выполнено тестирование интеграции и ручное тестирование.
   12. Выявлены ошибки системных компонент (при наличии).
   13. Заполнены протоколы тестирования.
   14. Продемонстрировано знание стандартов кодирования более чем одного языка программирования.
   15. Выявлены все имеющиеся несоответствия стандартам в предложенном коде.
6. Руководство оператора
7. Заключение.
8. Приложения к отчету: диск со всеми подтверждающими материалами, отчет в электронном виде, презентация для выступления и др. материалы.

# **2. Характеристика объекта практики**

Производственная практика проходила в ООО «Электрон-софт» — организации, специализирующейся на разработке и сопровождении информационных решений для автоматизации бизнес-процессов. Это современное IT-предприятие, осуществляющее внедрение программных решений в различные сферы: от бухгалтерского учёта до документооборота.

ООО «Электрон-софт» располагается по адресу:  
**Кировская область, г. Киров, ул. Ленина, д. 68.**  
Телефоны для связи:

* +7 (833) 271-49-10
* +7 (833) 235-48-11
* +7 (912) 826-96-53

Электронная почта:

* cgbnhn@ya.ru
* uvb.com@gmail.com

Официальный сайт: [electron.ru](http://electron.ru)

## 3. Описание рабочего места



Рисунок 1 – Рабочее место

В связи с дистанционным форматом прохождения практики, моё рабочее место было организовано дома — в комфортной обстановке, способствующей сосредоточенной разработке. Основу составлял личный ноутбук, оснащённый всем необходимым для продуктивной работы.

**Технические характеристики оборудования:**

* **Имя устройства:** DESKTOP-159LLU1
* **Процессор:** Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz, 2594 МГц
* **Оперативная память:** 32 ГБ DDR3
* **Тип системы:** 64-разрядная ОС на архитектуре x64
* **Операционная система:** Windows 10

## 4. Состав программного и технического обеспечения, их назначение

Для реализации проекта использовался следующий набор программных инструментов:

* **Microsoft Visual Studio 2022** — основная среда разработки для написания WPF-приложения на C#. Использовалась для создания интерфейсов, бизнес-логики, тестов и сетевых функций.
* **Microsoft SQL Server Management Studio (SSMS)** — управление и работа с реляционной базой данных MS SQL, включая написание запросов, проектирование схем и выполнение отладки.
* **Git + GitHub** — система контроля версий и облачный репозиторий для отслеживания изменений, откатов и командной работы.
* **Microsoft Office** — создание документации, отчётов, презентаций и таблиц.
* **Антивирусное ПО** — защита среды разработки и данных.

## 5. Описание выполненных видов работ

## 5.1 Разработан и обоснован вариант интеграционного решения с помощью графических средств среды разработки

Целью практики стала разработка модуля **архива** для корпоративной системы «Единое окно». Архив предназначен для долговременного хранения документов, связанных с заявками на подключение, включая технические условия, прилагаемые файлы и договоры. Разработка велась на платформе **C# и WPF** с использованием **MS SQL Server** в качестве СУБД.

Разработка началась с проектирования архитектуры модуля, построения ER-диаграммы и определения ключевых классов. Были реализованы:

* Document.cs — модель представления документа с учётом типа, даты, расширения, идентификаторов и метаданных;
* SqlService.cs — сервисный класс для взаимодействия с MS SQL Server, включая загрузку, сохранение и выборку файлов;
* FileTransferService.cs и HttpListener.cs — модули для передачи и приёма документов по локальной сети;
* LoginWindow.xaml(.cs) и MainWindow.xaml(.cs) — пользовательский интерфейс с авторизацией, фильтрацией, просмотром и восстановлением документов.

В рамках модуля реализована следующая логика:

* Документы по заявке (включая заявку, договор, техусловия) хранятся **бессрочно**;
* Все остальные материалы — **в течение 5 лет** с момента загрузки;
* Загрузка документов возможна **из разных модулей** системы, при наличии соответствующих прав доступа;
* Получение информации из архива возможно только при наличии **SQL-роли**, определяющей уровень доступа.

В системе предусмотрено разграничение доступа:

* Только авторизованные пользователи с определёнными правами могут загружать, просматривать или восстанавливать документы;
* Каждое действие фиксируется, а удаление доступно только роли «Архивариус»;
* Интерфейс WPF адаптируется под роль пользователя.

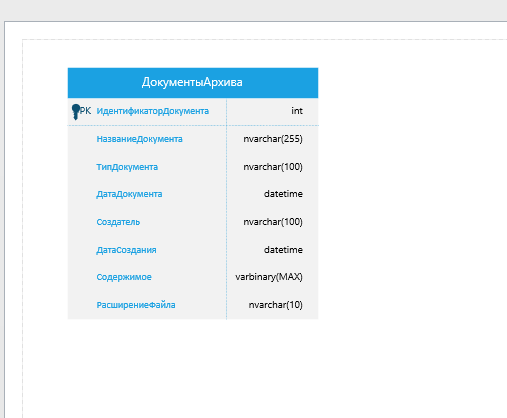


Рисунок 2 - ERD диаграмма

## 5.2 Сохранение в системе контроля версий

Каждый этап разработки сохранялся в репозитории GitHub: <https://github.com/kaminomi1/02>

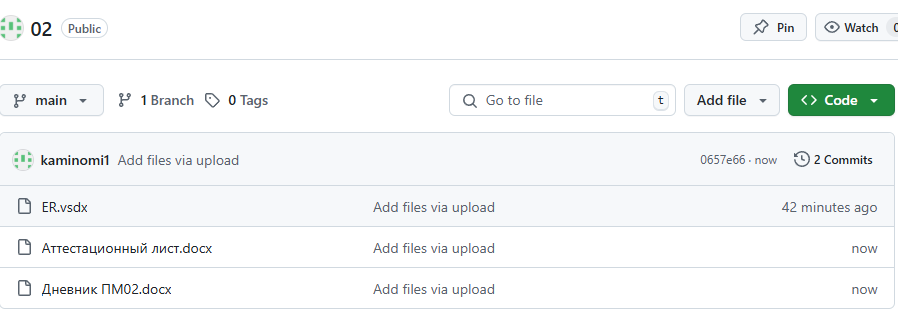


Рисунок 3 – Сохранение в Git

## 5.3 Анализ архитектуры и выбор версии проекта

Используя Git, была выбрана стабильная ветка проекта, содержащая актуальную архитектуру приложения. Проект построен по принципам модульности и разделения ответственности — каждая подсистема обособлена, с чётко определённым API.

## 5.4 Архитектура доработана для интеграции нового модуля

Для внедрения модуля архива была переработана архитектура приложения: добавлены соответствующие таблицы в БД, расширен класс SqlService.cs, переработан интерфейс MainWindow.xaml с учётом взаимодействия с архивом, а также обеспечена совместимость с существующими модулями.

## 5.5 Форматирование данных, транспортные протоколы и постобработка

Документы в архиве сохраняются в виде бинарных файлов с использованием типа данных VARBINARY(MAX), что позволяет хранить любой формат (PDF, DOCX, JPG и т.д.). Метаданные (дата загрузки, тип документа, автор, идентификатор заявки) хранятся отдельно и используются для быстрого поиска и фильтрации.

При передаче данных между клиентом и сервером используется собственный локальный протокол, построенный на базе TCP-соединений. Функциональность реализована в классах FileTransferService.cs и HttpListener.cs. Форматы сообщений стандартизированы: каждая операция сопровождается заголовком с типом действия upload, уникальным идентификатором сессии и размером передаваемого блока.

Постобработка включает:

* Проверку допустимого расширения и размера файла;
* Верификацию пользователя по роли и правам;
* Журналирование действия с результатом (успешно/ошибка);
* Очистку временных файлов после завершения загрузки.

## 5.6 Обработка исключений

Для повышения устойчивости модуля реализована система обработки ошибок. В код добавлены конструкции try-catch, а также созданы пользовательские классы-исключения для обработки типовых проблем, таких как отсутствие подключения к базе данных, попытка загрузки недопустимого файла или нарушение прав доступа.

## 5.7 Определение качественных показателей проекта

Оценка модуля архива проводилась по следующим ключевым метрикам качества:

| **Показатель** | **Оценка** | **Комментарий** |
| --- | --- | --- |
| **Функциональность** | Высокая | Реализованы все заявленные функции по ТЗ |
| **Надёжность** | Высокая | Модуль стабильно работает при нормальной нагрузке |
| **Производительность** | Хорошая | Быстрая загрузка и восстановление документов |
| **Безопасность** | Повышенная | Внедрены SQL-роли, ограничение действий, логирование |
| **Удобство использования** | Среднее | Интерфейс минималистичный, требует обучения |
| **Сопровождаемость** | Высокая | Чистый код, разбитый по логическим классам, с комментариями |
| **Масштабируемость** | Возможна | Архитектура позволяет легко расширить модуль |

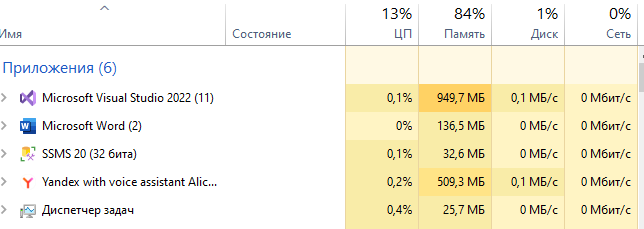


Рисунок 4 - Мониторинг производительности из диспетчера задач

## 5.8 Сохранение результата в системе контроля версий

Разработка проекта велась с использованием системы контроля версий Git.  
Основной репозиторий содержит:

* Скрипты создания и наполнения базы данных (.sql);
* Отчёт
* ERD-диаграмму

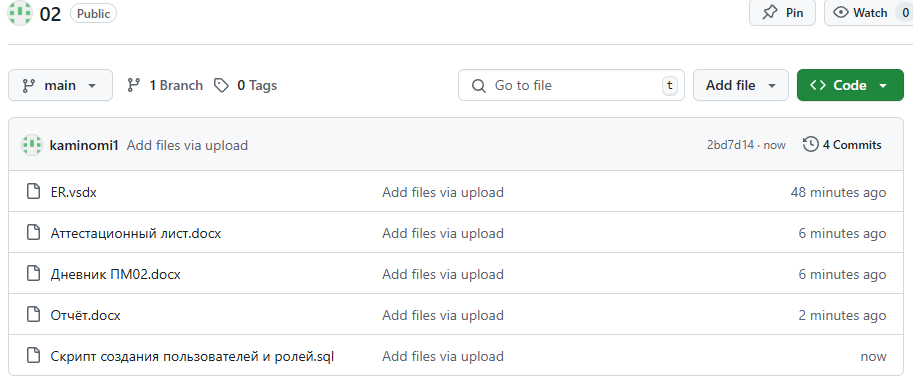


Рисунок 5 – Сохранение в Git

## 5.9 Тестирование и отладка проекта

Отладка производилась с использованием встроенных средств среды Visual Studio 2022. Устанавливались точки останова (breakpoints), отслеживались значения переменных и выполнение методов в реальном времени.

Также применялись внешние инструменты:

* **SQL Server Profiler** — для анализа производительности и корректности SQL-запросов;
* **SSMS** — для ручного анализа данных и логов;

## 5.10 Разработка тестового сценария и тестовых пакетов

Были разработаны тест-кейсы, включающие как стандартные, так и граничные ситуации:

**Примеры сценариев:**

* Загрузка файла допустимого формата и объёма;
* Попытка загрузки без авторизации;
* Просмотр документа с ролью «Чтение»;
* Удаление документа с ролью «Архивариус»;
* Восстановление ранее архивированного файла.

Тестовые пакеты включали как реальные, так и искусственные документы. Все тесты документированы.

## 5.11 Интеграционное и ручное тестирование

Выполнено ручное тестирование всех функций через WPF-интерфейс. Особое внимание уделено:

* Корректной работе ролевой модели;
* Обновлению интерфейса после операций;
* Реакции на некорректные действия пользователя.

Интеграционное тестирование проверяло работу всего модуля в связке с другими частями системы. Подтверждена корректность передачи данных, сессий и управления правами доступа.

## 5.12 Обнаруженные ошибки

Во время тестирования были зафиксированы следующие ошибки:

* При загрузке файла с кириллическим названием возникала ошибка кодировки (исправлено путём применения Encoding.UTF8);
* Неправильное отображение даты при разной локали ОС;
* Отсутствие подтверждающего окна при удалении (доработано);
* Возможность дважды отправить один и тот же документ (добавлена проверка уникальности).

Все баги были устранены до финальной сборки.

## 5.13 Заполнены протоколы тестирования

|  |  |
| --- | --- |
| **Название проекта** | «Модуль архива» |
| **Номер версии** | 1.0 |
| **Имя тестера** | Чемоданов Павел |
| **Даты тестирования** | 02.07.2025 |

|  |  |
| --- | --- |
| **TestCase #** | 1 |
| **Приоритет теста** | Высокий |
| **Название тестирования/**  **Имя** | Загрузка файла допустимого формата и объёма |
| **Резюме испытания** | Проверить возможность загрузки PDF-документа объёмом до 5 МБ пользователем с ролью «Специалист отдела». |
| **Шаги тестирования** | * Запустить приложение «Модуль архива» * Войти под пользователем с ролью «Специалист отдела» * Перейти в раздел «Архив» * Нажать кнопку «Загрузить» * Выбрать PDF-файл объёмом 3.4 МБ * Подтвердить загрузку |
| **Данные тестирования** | * Имя файла: ТУ\_жилой\_дом.pdf * Размер: 3.4 МБ * Расширение: .pdf * Автор: Петров П. П. |
| **Ожидаемый результат** | Файл загружен, данные отобразились в таблице, запись появилась в журнале. |
| **Фактический результат** | Соответствует ожидаемому. |
| **Предпосылки** | Пользователь авторизован, имеет права INSERT |
| **Постусловия** | Документ загружен в архив. |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Тест проверяет базовый сценарий загрузки архива. |

|  |  |
| --- | --- |
| **TestCase #** | 2 |
| **Приоритет теста** | Критический |
| **Название тестирования/**  **Имя** | Попытка загрузки без авторизации |
| **Резюме испытания** | Проверка невозможности загрузки файла без предварительного входа в систему. |
| * **Шаги тестирования** | * Запустить приложение * Не входить под пользователем * Перейти к разделу «Архив» * Попробовать нажать «Загрузить» |
| **Данные тестирования** | * Файл: test.pdf * Объём: 2 МБ |
| **Ожидаемый результат** | Кнопка «Загрузить» недоступна или появляется сообщение об ошибке доступа |
| **Фактический результат** | Соответствует ожидаемому. |
| **Предпосылки** | Пользователь не авторизован |
| **Постусловия** | Никакие действия не выполнены |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Доступ к архиву надёжно ограничен |

|  |  |
| --- | --- |
| **TestCase #** | 3 |
| **Приоритет теста** | Средний |
| **Название тестирования/**  **Имя** | Проверка, что пользователь с минимальными правами может только просматривать документы |
| **Резюме испытания** | Проверка невозможности загрузки файла без предварительного входа в систему. |
| * **Шаги тестирования** | * Войти под пользователем с ролью Чтение * Перейти в раздел «Архив» * Выбрать файл из таблицы * Открыть его двойным щелчком * Проверить, что кнопки «Удалить» и «Изменить» неактивны |
| **Данные тестирования** | * Пользователь: readonly\_user * Файл: 1.pdf |
| **Ожидаемый результат** | Файл открылся в режиме просмотра, другие действия недоступны |
| **Фактический результат** | Соответствует ожидаемому. |
| **Предпосылки** | Пользователь авторизован с правами SELECT |
| **Постусловия** | Файл просмотрен |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Правильно работает система разграничения доступа |

|  |  |
| --- | --- |
| **TestCase #** | 4 |
| **Приоритет теста** | Высокий |
| **Название тестирования/**  **Имя** | Удаление документа с ролью «Архивариус» |
| **Резюме испытания** | Проверка, что только пользователь с ролью Архивариус может удалять документы |
| * **Шаги тестирования** | * Войти под ролью «Архивариус» * Перейти в раздел «Архив» * Выбрать документ * Нажать «Удалить» * Подтвердить удаление |
| **Данные тестирования** | * Документ: Договор\_подключение.pdf * Дата загрузки: 02.07.2025 |
| **Ожидаемый результат** | Документ удалён, в журнале появилась запись об удалении |
| **Фактический результат** | Соответствует ожидаемому. |
| **Предпосылки** | Пользователь с правами DELETE |
| **Постусловия** | Документ удалён из архива |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Система корректно выполняет удаление и журналирование |

|  |  |
| --- | --- |
| **TestCase #** | 5 |
| **Приоритет теста** | Средний |
| **Название тестирования/**  **Имя** | Восстановление архивного документа |
| **Резюме испытания** | Проверка восстановления ранее удалённого документа обратно в активный список |
| * **Шаги тестирования** | * Войти под ролью «Архивариус» * Перейти в архив * Найти ранее удалённый документ * Нажать «Восстановить» * Подтвердить действие |
| **Данные тестирования** | * Документ: Проект\_№2345.docx * Статус: В архиве |
| **Ожидаемый результат** | Документ перемещён в список активных, состояние обновлено |
| **Фактический результат** | Соответствует ожидаемому. |
| **Предпосылки** | Документ ранее был удалён |
| **Постусловия** | Документ восстановлен |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Механизм восстановления работает корректно |

## 5.14 Стандарты кодирования и межъязыковая практика

Проект соответствует стандартам кодирования **C#**:

* PascalCase — для имён классов, методов;
* camelCase — для переменных и аргументов;
* XML-документирование через /// <summary>.

SQL-запросы оформлены по стандартам T-SQL: ключевые слова в верхнем регистре, логические блоки структурированы, используются параметры (@Param) для предотвращения SQL-инъекций.

Также использовались JSON-сообщения в транспортном протоколе — знание структуры и форматирования данных в этом формате тоже применялось.

**5.15 Выявление несоответствий стандартам и устранение**

На этапе финального анализа были обнаружены:

* Повторяющийся код в блоках обработки ошибок (вынесено в отдельные методы);
* Жёстко закодированная строка подключения (перенесена в конфигурационный файл с шифрованием);
* Отсутствие логирования для некоторых критичных операций (добавлено логирование в DocumentLog);
* Несогласованное наименование переменных (приведено к единому стилю).

## ****6. Руководство оператора****

## ****6.1. Вход в приложение****

1. Запустить ярлык "Модуль архива"
2. Ввести логин и пароль SQL-пользователя
3. Программа подключится к MSSQL и загрузит интерфейс в соответствии с ролью

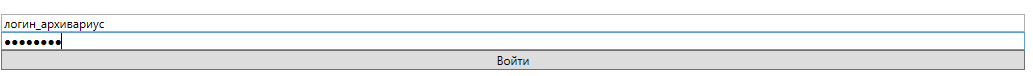


Рисунок 6 – Авторизация пользователя

## ****6.2. Поиск документа****

1. Перейти в раздел "Архив"
2. Ввести параметры поиска (ФИО, дата, тип документа)
3. Нажать кнопку "Поиск"
4. Двойным щелчком открыть документ

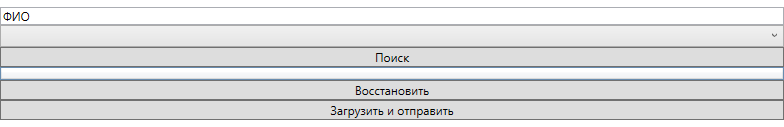


Рисунок 7 – Меню

## ****6.3. Восстановление документа****

1. Выбрать документ в архиве
2. Нажать кнопку "Восстановить"
3. Подтвердить действие в появившемся окне
4. Документ будет перемещён обратно в активные

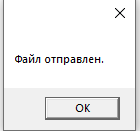


Рисунок 8 – Файл отправлен

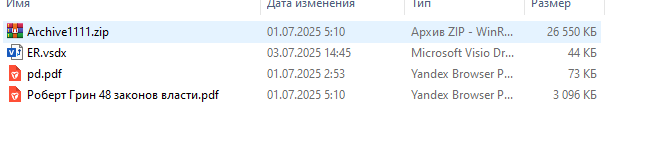


Рисунок 9 – Файл перемещён в папку

**7. Заключение**

В рамках производственной практики в ООО «Электрон-софт» мной был разработан модуль архива для системы «Единое окно». Целью стало создание надёжного инструмента для хранения, загрузки, восстановления и защиты документов, связанных с заявками на подключение.

Разработка велась на базе WPF с использованием языка C# и MS SQL Server. Были реализованы: авторизация, разграничение прав доступа (Архивариус, Специалист отдела, Чтение), обработка файлов, механизм восстановления документов и журналирование действий. Также обеспечена безопасность — через SQL-аутентификацию, валидацию данных и шифрование конфигурации.

Модуль прошёл тестирование, выявленные ошибки устранены. Работа велась с использованием Git, что позволило отслеживать изменения и контролировать версионность проекта.

Практика позволила применить на практике знания по архитектуре приложений, работе с базами данных, интерфейсам WPF и системам безопасности. Полученные навыки станут основой для дальнейшего развития в сфере программной инженерии.

## 8. Приложения

* Исходный код проекта на GitHub
* Электронный отчёт
* Презентация для защиты
* ER-диаграмма базы данных
* Скриншоты интерфейса и тестирования

Ссылка на репозиторий: <https://github.com/kaminomi1/02>