 ***Федеральное агентство по рыболовству***

***Федеральное государственное бюджетное образовательное учреждение высшего образования***

***«Астраханский государственный технический университет»***

**Система менеджмента качества в области образования, воспитания, науки и инноваций сертифицирована DQS по международному стандарту ISO 9001:2015**

Институт Информационных технологий и коммуникаций

Направление подготовки 09.03.01 Информатика и вычислительная техника Профиль Автоматизированные системы обработки информации и управления Кафедра Автоматизированные системы обработки информации и управления

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

Информационная система формирования процессуальных документов и локальных   
нормативных актов в судебном делопроизводстве

(название темы)

Работа выполнена обучающимся группы ДИНРб-41 Самойлов Дмитрий Сергеевич

(Фамилия Имя Отчество)

Руководитель ВКР

ст. преп. Алтуфьев М. Ю. \_

(ученая степень, ученое звание, Фамилия Имя Отчество)

Консультант по проектной документации ст. преп. Мамлеева А.Р.

(название раздела, ученая степень, ученое звание, Фамилия И.О., подпись)

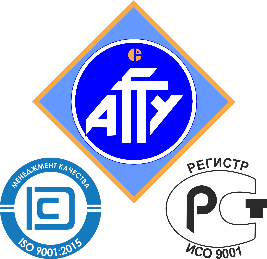
Нормоконтролер ассистент Кравченкова Е.П.

(ученая степень, ученое звание, Фамилия И.О., подпись)

Допущен к защите « \_» 20 г.

Заведующий кафедрой д.т.н, профессор Хоменко Т.В.

Астрахань 2024

***Федеральное агентство по рыболовству Федеральное государственное бюджетное образовательное***

***учреждение высшего образования***

***«Астраханский государственный технический университет»***

**Система менеджмента качества в области образования, воспитания, науки и инноваций сертифицирована DQS по международному стандарту ISO 9001:2015**

УТВЕРЖДАЮ

Заведующий кафедрой Автоматизированные системы

обработки информации и управления д.т.н., профессор Т.В. Хоменко

(подпись)

«\_\_»\_ 20 г.

**З А Д А Н И Е**

на выполнение выпускной квалификационной работы

Обучающемуся

учебной группы ДИНРб-41 института Информационных технологий и коммуникаций

(институт/факультет)

Самойлов Дмитрий Сергеевич

(фамилия, имя, отчество - полностью)

**ТЕМА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

Информационная система формирования процессуальных документов и локальных\_\_\_\_\_\_\_ нормативных актов в судебном делопроизводстве \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема ВКР сформулирована в соответствии с запросом кафедры АСОИУ Астраханского

Государственного Технического Универсистета \_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждена распоряжением директора института Информационных технологий и

коммуникаций (распоряжение от « 09 » декабря 2023 г. № 46)

**РУКОВОДИТЕЛЬ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

Алтуфьев Михаил Юрьевич ст. преп.

(фамилия, имя, отчество – полностью, ученая степень, ученое звание)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утвержден распоряжением директора института Информационных технологий и

коммуникаций (распоряжение от « 09 » декабря 2023 г. № 46)

Представление выпускной

квалификационной работы на кафедру « » 20 г.

Дата защиты « » 20 г.

Целевая установка и исходные данные:

Разработка базы данных и программного продукта для формирования процессуальных документов и локальных нормативных актов в судебном делопроизводстве

|  |  |  |
| --- | --- | --- |
| № п/п | Перечень чертежей, подлежащих разработке | Формат, количество |
| 1 | Роли в системе и варианты использования | А4, 1 |
| 2 | Представление сущностей в системе | А4, 1 |
| 3 | Диаграмма компонентов системы | А4, 1 |
| 4 | Диаграмма активности системы | A4, 2 |
|  |  |  |
|  |  |  |
|  |  |  |

Руководитель выпускной квалификационной работы:

(подпись)

|  |  |  |
| --- | --- | --- |
| № п/п | Содержание расчетно-пояснительной записки (перечень вопросов, подлежащих разработке) | Консультанты |
| 1 | Разработка технической документации | Куркурин Н. Д. |
| 2 | Проектирование компонентов системы | Морозов А. В. |
| 3 | Тестирование программного продукта | Лаптев В. В. |
| 4 | Проектная документация | Алтуфьев М. Ю. |
|  |  |  |
|  |  |  |

Руководитель выпускной квалификационной работы:

(подпись)

**Основная рекомендуемая литература**

1. Белов С.В. Требования к оформлению студенческих работ / С.В. Белов, В.В. Лаптев, А.В. Морозов, В.В. Толасова, А.Р. Мамлеева. – Астрахань, АГТУ, 2022. 104 с.
2. PlantUML Guide [Электронный ресурс] URL: <http://plantuml.com/guide>(дата обращения 10.4.2023).
3. Натан А. WPF 4. Подробное руководство. - Пер. с англ. - СПб.: Символ-Плюс, 2011. - 880 с., ил.

Задание принял к исполнению « » 20 г.

Обучающийся

(подпись)

|  |  |  |
| --- | --- | --- |
| **УТВЕРЖДАЮ** | К заданию на выпускную  квалификационную работу | |
| Заведующий кафедрой  д.т.н., профессор Т.В. Хоменко  (подпись)  «\_ »\_ 20 \_г. |  |

**К А Л Е Н Д А Р Н Ы Й Г РА Ф И К**

подготовки выпускной квалификационной работы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № п/п | Разделы, темы и их  содержание, графический материал | По плану | | По факту | | Отметка руководителя о выполнении |
| Дата | Объем  в % | Дата | Объем  в % |
| 1 | Утверждение темы  проекта и обоснование актуальности | 09.12.2023 | Утверждено | 09.12.2023 | Утверждено |  |
| 2 | Составление и  согласование технического задания на выпускную  квалификационную работу | 23.12.2023 | Согласовано | 23.12.2023 | Согласовано |  |
| 3 | Технический проект (по результатам прохождения производственно-  технологической практики) | 03.02.2024 | 100% | 03.02.2024 | 100% |  |
| 4 | Рабочий проект (по результатам прохождения  преддипломной практики). | 24.05.2024 | 100% | 24.05.2024 | 100% |  |
| 5 | Предварительная защита ВКР | 11.06.2024 | Допущен к защите | 11.06.2024 | Допущен к защите |  |
| 6 | Получение отзыва  руководителя на работу | 17.06.2024 | Получено | 17.06.2024 | Получено |  |
| 7 | Защита ВКР | 21.07.2024 |  | 21.07.2024 |  |  |

Руководитель выпускной квалификационной работы:

(подпись)

Обучающийся:

(подпись)

« » 20 г

**РЕФЕРАТ**

с. 53, рис. 10, табл. 20, прил. 6

**Приложение для настольных компьютеров, локальное, подписание документов, формирование документов, суд, судебное делопроизводство, эффективность, шаблоны документов, шаблоны печатей, пользовательский опыт, доступность, удобство.**

Разработана система для увеличения эффективности формирования и подписания процессуальных документов и локальных нормативных актов. Программный комплекс может быть внедрен в судебный департамент Астраханской области.

Выявлены основные участники бизнес-процессов формирования, формализованы процессы формирования и подписания документов, установлены основные прецеденты использования.

Определены требования к информационной безопасности: проведена классификация системы по классу защищённости, сформирован перечень мер по защите информации, составлены инструкции к организационным мерам защиты информации.

Разработаны инструкции по инсталляции и выполнению программного продукта.

**ANNOTATION**

53 рages, 10 figures, 20 tables, 6 applications.

**Desktop application, local, document signing, document generation, court, court record keeping, efficiency, document templates, stamp templates, user experience, accessibility, convenience.**

A system has been developed to increase the efficiency of the formation and signing of procedural documents and local regulations. The software package can be implemented in the judicial department of the Astrakhan region.

The main participants in the business processes of formation have been identified, the processes of forming and signing documents have been formalized, and the main use cases have been established.

The requirements for information security are defined: the classification of the system by security class is carried out, a list of information protection measures is formed, instructions for organizational information protection measures are compiled.

Instructions have been developed for the installation and execution of the software product.

**СОДЕРЖАНИЕ**

[Введение 9](#_Toc169473664)

[1 Технический проект 11](#_Toc169473665)

[1.1 Анализ предметной области 11](#_Toc169473666)

[1.2 Обзор аналогичных информационных систем 16](#_Toc169473667)

[1.3 Технология обработки информации 17](#_Toc169473668)

[1.3.1 Вариант использования «Вход в систему «Дело»» 17](#_Toc169473669)

[1.3.2 Вариант использования «Вход через оффлайн режим» 17](#_Toc169473670)

[1.3.3 Вариант использования «Создание документа» 18](#_Toc169473671)

[1.3.4 Вариант использования «Установка штампа и подписание документа» 18](#_Toc169473672)

[1.3.5 Вариант использования «Посмотреть шаблоны печатей» 18](#_Toc169473673)

[1.3.6 Вариант использования «Посмотреть шаблоны документов» 18](#_Toc169473674)

[1.3.7 Вариант использования «Создать новый шаблон» 18](#_Toc169473675)

[1.3.8 Вариант использования «Удалить шаблон» 18](#_Toc169473676)

[1.4 Основные сценарии 18](#_Toc169473677)

[1.5 Цель и назначение информационной системы 19](#_Toc169473678)

[1.6 Инфологическая модель данных 20](#_Toc169473679)

[1.7 Входная и выходная информация 21](#_Toc169473680)

[1.8 Требования к программному и аппаратному обеспечению 23](#_Toc169473681)

[1.9 Безопасность информационной системы 23](#_Toc169473682)

[2 Рабочий проект 25](#_Toc169473683)

[2.1 Общие сведения о работе системы 25](#_Toc169473684)

[2.2 Функциональное назначение программного продукта 25](#_Toc169473685)

[2.2 Описание физической архитектуры системы 26](#_Toc169473686)

[2.3 Инсталляция и выполнение программного продукта 26](#_Toc169473687)

[2.4 Даталогическая модель данных 26](#_Toc169473688)

[2.5 Обеспечение целостности данных 27](#_Toc169473689)

[2.6 Использование JSON-объектов 28](#_Toc169473690)

[2.7 Описание программы 28](#_Toc169473691)

[2.8 Руководство пользователя 42](#_Toc169473692)

[2.9 Сообщения системы 47](#_Toc169473693)

[3 Программа и методика испытаний 49](#_Toc169473694)

[Заключение 52](#_Toc169473695)

[Список использованных источников 53](#_Toc169473696)

[Приложение 1](#_Toc169473697) [Диаграмма вариантов использования 54](#_Toc169473698)

[Приложение 2](#_Toc169473699) [Диаграмма активности – «формирование документа» 55](#_Toc169473700)

[Приложение 3](#_Toc169473701) [Диаграмма активности – «подписание документа» 56](#_Toc169473702)

[Приложение 4](#_Toc169473703) [Диаграмма сущность-связь 57](#_Toc169473704)

[Приложение 5](#_Toc169473705) [Техническое задание 58](#_Toc169473706)

[Приложение 6](#_Toc169473707) [Оптический носитель информации 63](#_Toc169473708)

# ВВЕДЕНИЕ

В современном мире, где информационные технологии прочно вошли во все сферы деятельности человека, особое внимание уделяется автоматизации и оптимизации рабочих процессов. Одной из ключевых областей, где требуется высокая точность и соблюдение процессуальных норм, является судебное делопроизводство. Подготовка процессуальных документов и локальных нормативных актов занимает значительную часть времени судебных работников, что выдвигает на первый план задачу разработки программного продукта, направленного на упрощение и ускорение данного процесса.

Судебное делопроизводство, являясь неотъемлемой частью функционирования правового государства, требует постоянного внимания со стороны законодателя и практикующих юристов. Ключевое значение имеет разработка и внедрение эффективных механизмов для формирования, обработки и хранения процессуальных документов и локальных нормативных актов. В условиях постоянно растущего объема информации и стремительного развития информационных технологий, система правосудия сталкивается с необходимостью обновления методов работы. Появляется потребность в создании комплексного информационного решения, которое бы обеспечивало бесперебойность, эффективность и четкость в формировании юридически значимых документов.

Практическая работа посвящена исследованию и разработке решений, ориентированных на автоматизацию процессов создания, редактирования, хранения и подписания процессуальных документов и локальных нормативных актов, что потенциально приводит к минимизации ручного труда, снижению возможности ошибок и повышению общей прозрачности судебных процедур. В этом контексте центральным становится задача разработки программно-аппаратного комплекса, включающего в себя как программное обеспечение для работы с документами, так и методологии их стандартизации и классификации.

Целью учебной практической работы является разработка и анализ программного продукта, предназначенного для формирования процессуальных документов и локальных нормативных актов, используемых в судебном делопроизводстве. Работа включает в себя изучение текущего состояния делопроизводства в судебных органах, анализ существующих аналогов программных средств, проектирование структуры и функциональности разрабатываемого продукта, а также формулирование требований к его качеству и безопасности.

Программный продукт должен обладать рядом характеристик, таких как гибкость настройки под нужды конкретного судебного органа, простота использования, возможность интеграции с базой данных, а также высокий уровень защиты информации от несанкционированного доступа.

В процессе практической работы будет проведена разработка программного продукта, реализующего основные принципы и механизмы подписания и формирования процессуальных документов и локальных нормативных актов в контексте судебной деятельности. Также практическая работа включает в себя тестирование разработанного прототипа с целью выявления его функциональных возможностей и потенциальных ограничений.

# 1 ТЕХНИЧЕСКИЙ ПРОЕКТ

## Анализ предметной области

**1.1.1 Документооборот**

Документооборот (электронный документооборот) - система создания, интерпретации, передачи, приема и архивирования документов в организации с момента их создания или получения до завершения исполнения или отправления. Система электронного документооборота также контролирует выполнение работ с документами и обеспечивает защиту от несанкционированного доступа.

Документооборот — это один из инструментов управления предприятием, служащий для обмена электронной информацией как внутри предприятия (между сотрудниками предприятия), так и с внешними контрагентами.

Существует также такое понятие, как "делопроизводство", которое часто используют как синоним термина "документооборот", однако, их следует разделять. Делопроизводство — это комплекс мероприятий в организации по документационному обеспечению управления. А термином "документооборот", как правило, описывается движение документов в рамках документационного обеспечения управления.

Документооборот в организации может быть как централизованным, так и представленным на уровне конкретных подразделений. В централизованном документообороте рассматривают три основных типа документов: входящие (от внешних контрагентов), исходящие (из подразделений) и внутренние (для рассылки внутри организации).

Выделяют ряд основных этапов документооборота:

* Составление документа (при совершении хозяйственной операции)
* Передача в бухгалтерию
* Проверка принятых документов бухгалтером
* Таксировка, группировка и контировка документа в бухгалтерии
* Сдача на хранение в архив (последняя стадия жизненного цикла документа в рамках документооборота)

Основной коммуникативной проблемой предприятия среднего уровня является проблема, связанная с документооборотом. Документооборот может быть некачественным, недостоверным, неформализованным и т.д. Эти проблемы являются источником потери информации, непроизводительных затрат ресурсов предприятия, которые выражаются в конечном счете в финансовых потерях. Средством регуляции (нормализации) взаимодействия на таком предприятии является оптимизация документооборота и его последующая автоматизация, т.е. создание системы электронного документооборота (или СЭД). При этом часто автоматизация документооборота происходит параллельно с постановкой и автоматизацией учета на предприятии.

Системы автоматизации документооборота вошли в широкий обиход за рубежом раньше, чем у нас, соответственно, сложилась и своя терминология, описывающая различные области электронного документооборота. Существует целый класс продуктов для автоматизации документооборота, так называемые системы электронного документооборота, или СЭД.

**1.1.2 Электронный документооборот (ЭДО)**

Электронный документооборот (ЭДО) — это применение документов в электронном виде без заполнения бумажных носителей. Передача данных осуществляется через особое программное обеспечение, а все образцы и формуляры заполняются на компьютере. Цель ЭДО в том, чтобы обеспечить повышение скорости работы организации в целом, и бухгалтера в частности, удобства ведения расчетов и конфиденциальности.

Электронная цифровая подпись (ЭЦП) является необходимым условием для полноценной реализации защищенного электронного документооборота. Возможности применения ЭЦП и шифрования данных обеспечиваются благодаря интегрированным в систему «ДЕЛО» сертифицированным средствам криптографической защиты информации. Система предоставляет возможность подписывать посредством ЭЦП файлы, хранящиеся в базе данных и отправляемые по электронной почте. При необходимости, документ может быть подписан несколькими сотрудниками, что может использоваться для автоматизации процедур согласования и утверждения документов.

**1.1.3 Система электронного документооборота (СЭД)**

Система электронного документооборота (СЭД) — это сервис, который позволяет организовать работу с электронными документами внутри компании. Он автоматизирует делопроизводство, ускоряет процессы, освобождает сотрудников от рутинных задач. СЭД закрывают все потребности делопроизводства компаний, а функциональность таких систем превышает возможности бумажного документооборота.

Так, в сервисах электронного документооборота:

* налаживается работа с электронными документами на всех этапах их жизненного цикла;
* поиск нужных файлов занимает минимум времени;
* файлы надежно хранятся в едином архиве;
* возможна настройка маршрутизации;
* сокращается время на бумажную работу, корректировку, согласование, подписание документации;
* взаимодействие между сотрудниками автоматизируется: обмениваться документами, выдавать и получать задания, отправлять уведомления можно в один клик;
* действия с документами отмечаются и сохраняются в системе ― за счет этого делопроизводство становится контролируемым, а исполнительность сотрудников повышается.

Благодаря СЭД предприятия оптимизируют административные расходы на внутренний документооборот и хранение документации. При выстроенном безбумажном делопроизводстве процессы в компании ускоряются — документы быстрее проходят все согласования, решения принимаются оперативнее. Это не влияет на качество работы, напротив, использование СЭД делает компанию гибкой, способной адаптироваться к ситуации на рынке.

Примеры популярных СЭД:

* МосЭДО.
* Интра.татар.ру.
* Практика.
* ТЕЗИС.
* Корус.
* Directum.
* Docsvision.
* Дело (EOS).

**1.1.4 СЭД «ДЕЛО»**

СЭД «ДЕЛО» – многопользовательская система, разработанная компанией ЭОС, с полным набором инструментов для автоматизации работы с документами, задачами и процессами. Позволяет оптимизировать документооборот, повысить его эффективность, снизить долю рутинного труда при обработке документов за счет использования технологий и сервисов с элементами искусственного интеллекта.

**1.1.5 Электронная цифровая подпись (ЭЦП)**

Электронная подпись (ЭП), Электронная цифровая подпись (ЭЦП), Цифровая подпись (ЦП) позволяет подтвердить авторство электронного документа (будь то реальное лицо или, например, аккаунт в криптовалютной системе). Подпись связана и с автором, и с самим документом с помощью криптографических методов и не может быть подделана с помощью обычного копирования.

ЭЦП — это реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

Виды электронной подписи:

* Простая электронная подпись (ПЭП)
* Неквалифицированная электронная подпись (НЭП)
* Квалифицированная электронная подпись (КЭП)

**Простая электронная подпись, или ПЭП**

Простая подпись — это знакомые всем коды доступа из СМС, коды на скретч-картах, пары «логин-пароль» в личных кабинетах на сайтах и в электронной почте. Простая подпись создается средствами информационной системы, в которой ее используют, и подтверждает, что электронную подпись создал конкретный человек.

**Неквалифицированная электронная подпись, или НЭП**

Усиленная неквалифицированная электронная подпись (далее — НЭП) создается с помощью программ криптошифрования с использованием закрытого ключа электронной подписи. НЭП идентифицирует личность владельца, а также позволяет проверить, вносили ли в файл изменения после его отправки.

Человек получает в удостоверяющем центре два ключа электронной подписи: закрытый и открытый. Закрытый ключ хранится на специальном ключевом носителе с пин-кодом или в компьютере пользователя — он известен только владельцу и его нужно держать в тайне. С помощью закрытого ключа владелец генерирует электронные подписи, которыми подписывает документы.

Открытый ключ электронной подписи доступен всем, с кем его обладатель ведет ЭДО. Он связан с закрытым ключом и позволяет всем получателям подписанного документа проверить подлинность ЭП.

То, что открытый ключ принадлежит владельцу закрытого ключа, прописывается в сертификате электронной подписи. Сертификат также выдается удостоверяющим центром. Но при использовании НЭП сертификат можно не создавать. Требования к структуре неквалифицированного сертификата не установлены в федеральном законе № 63-ФЗ «Об электронной подписи».

**Квалифицированная электронная подпись или КЭП**

Усиленная квалифицированная электронная подпись — самый регламентированный государством вид подписи. Так же, как и НЭП, она создается с помощью криптографических алгоритмов и базируется на инфраструктуре открытых ключей, но отличается от НЭП в следующем:

Обязательно имеет квалифицированный сертификат в бумажном или электронном виде, структура которого определена приказом ФСБ России № 795 от 27.12.2011.

Программное обеспечение для работы с КЭП сертифицировано ФСБ России.

Выдавать КЭП может только удостоверяющий центр, который аккредитован Минкомсвязи России.

Сертификат электронной подписи — это бумажный или электронный документ, позволяющий проверить подлинность электронной подписи, он подтверждает принадлежность ключа проверки электронной подписи владельцу сертификата. Сертификаты ЭП выпускаются удостоверяющим центром, который занимается выдачей и обслуживанием сертификатов электронной подписи. Термин «сертификат ключа проверки» или «открытый ключ» подразумевает, что этот документ содержит все данные о владельце закрытого ключа электронной подписи, в том числе и данные по удостоверяющему центру, который выпустил подпись. Удостоверяющий центр, выдавший сертификат, отвечает за достоверность данных, которые содержатся в нём.

В сертификате ключа проверки электронной подписи обязательно присутствуют следующие сведения:

* ключ проверки электронной подписи;
* наименование средства электронной подписи;
* наименование удостоверяющего центра, выпустившего сертификат;
* сведения об ограничениях в сфере применения ЭЦП;
* дата начала и окончания срока действия сертификата;
* сведения о владельце электронной подписи, включая:
* ФИО и СНИЛС — для физического лица,
* ИНН физического или юридического лица, в зависимости от типа заявителя,
* наименование и местонахождение организации для владельца — юридического лица, другие сведения;
* информация о точке распространения списка отозванных сертификатов.

Также квалифицированный сертификат может содержать сведения об области своего применения, дополнительную информацию о владельце и издателе сертификата, адрес службы актуальных статусов сертификатов, адрес службы штампов времени и другую информацию.

Квалифицированный сертификат ключа проверки электронной подписи можно получить только в удостоверяющем центре, прошедшем аккредитацию Минкомсвязи.

## Обзор аналогичных информационных систем

В сети интернет представлены программные средства формирования и подписи документов ДокМастер и Directum.

ДокМастер - программа подготовки электронных документов, созданная компаниями ЗАО «Сервер-Центр», ЗАО «Атлас-2».

Особенностями является:

* создание документа производится на основании утвержденного шаблона, содержащего структуру, экранную и печатную форму, инструкцию по заполнению.
* единый шаблон документа (например, налоговой декларации) все абоненты скачивают через интернет из специального хранилища – Репозитория форматов.
* доступ к программе может осуществляться по сертификату ЭЦП. Также ЭЦП можно поставить на любом созданном документе или шаблоне.

Недостатки системы:

* отсутствует возможность подключения к системе «Дело».
* отсутствует возможность создания своих шаблонов документов.

Directum — система управления цифровыми процессами и документами с элементами искусственного интеллекта, в основе которых лежат технологии машинного обучения и компьютерного зрения.

Особенностями решения являются:

* большое количество разного функционала
* позволяет хранить, организовывать и управлять документами, включая возможность создания новых документов по утвержденным шаблонам.
* обеспечивает поддержку электронной подписи, что позволяет безопасно подписывать и утверждать документы в электронном формате.

Недостатки автоматизирующей системы:

* сложность внедрения.
* отсутствует возможность создания своих шаблонов документов.
* Отсутствует возможность подключения к системе «Дело»

## Технология обработки информации

Анализ предметной области показал, что программа рассчитана на одного пользователя. В приложении 1 представлена диаграмма вариантов использования.

**1.3.1 Вариант использования «Вход в систему «Дело»»**

Основной сценарий: пользователь включает программу и имеет возможность войти в систему «Дело» и получить доступ к работе с системой.

**1.3.2 Вариант использования «Вход через оффлайн режим»**

Основной сценарий: пользователь включает программу и имеет возможность войти в систему с ограниченными возможностями не указывая данные к системе «Дело».

**1.3.3 Вариант использования «Создание документа»**

Основной сценарий: пользователь включает программу, авторизуется через систему «Дело» или же включает оффлайн режим, после чего он имеет возможность создать документ, для этого ему нужно, выбрать шаблон документа, на основе которого будет создаваться новый документ, вставить блоки текста на основе шаблона и нажать создать.

**1.3.4 Вариант использования «Установка штампа и подписание документа»**

Основной сценарий: пользователь включает программу и имеет возможность установить штамп и подписать документ, для этого ему нужно войти в систему через систему дело или через оффлайн режим, выбрать документ, выбрать шаблон штампа, ввести данные, характеризующие подпись и нажать подписать.

**1.3.5 Вариант использования «Посмотреть шаблоны печатей»**

Основной сценарий: пользователь включает программу, авторизуется через систему «Дело» или же включает оффлайн режим, переходит в менеджер шаблонов, после чего он имеет возможность просмотреть шаблоны печатей.

**1.3.6 Вариант использования «Посмотреть шаблоны документов»**

Основной сценарий: пользователь включает программу, авторизуется через систему «Дело» или же включает оффлайн режим, переходит в менеджер шаблонов, после чего он имеет возможность просмотреть шаблоны документов.

**1.3.7 Вариант использования «Создать новый шаблон»**

Основной сценарий: пользователь включает программу, авторизуется через систему «Дело» или же включает оффлайн режим, переходит в менеджер шаблонов, после чего он имеет возможность создать новый шаблон.

**1.3.8 Вариант использования «Удалить шаблон»**

Основной сценарий: пользователь включает программу, авторизуется через систему «Дело» или же включает оффлайн режим, переходит в менеджер шаблонов, после чего он имеет возможность удалить шаблон.

## Основные сценарии

В приложении 2 и 3 представлены диаграммы активностей подписания документа и формирования документа.

Основной сценарий, где пользователю требуется сформировать документ.

1. Пользователь выбирает подходящий шаблон для создания документа.
2. После выбора шаблона пользователь начинает вставлять блоки текста в документ.
3. Проверяется выбран ли шаблон документа. Если нет, процесс возвращается к шагу 2. Если да, процесс переходит к следующему шагу.
4. Пользователь вводит текст в выбранные блоки и нажимает кнопку, чтобы сформировать документ.

Основной сценарий, где пользователю требуется подписать электронный документ.

1. Пользователь выбирает документ, который он хочет подписать.
2. Пользователь вводит необходимые данные для подписи документа.
3. После чего выбирает шаблон печати для подписи.
4. После ввода всех необходимых данных и выбора шаблона печати пользователь нажимает на кнопку сохранить.
5. Проверяется, выбран ли документ. Если нет, процесс возвращается к шагу 2. Если да, процесс переходит к следующему шагу.
6. Проверяется, введены ли данные для подписи. Если нет, процесс возвращается к шагу 3. Если да, документ успешно подписывается.

## Цель и назначение информационной системы

Цель проекта – разработка базы данных и программного продукта для формирования процессуальных документов и локальных нормативных актов в судебном делопроизводстве.

Цель разработки – сокращение времени, потраченного на формирование и подписывание документов сотрудниками областного суда.

Задачами является, создание программного продукта, который предоставит возможности:

* создания шаблонов документов;
* создания шаблонов печати;
* подписывания документов при помощи созданных шаблонов печати;
* формирования документов при помощи созданных шаблонов документов;
* взаимодействие с системой «Дело».

## Инфологическая модель данных

В результате исследования предметной области определены основные сущности базы данных:

* Шаблон документа
* Шаблон печати

Таблица «Шаблон документа» предназначена для хранения информации об шаблонах, которые используются для создания документов. Атрибуты таблицы «Шаблон документа» представлены в табл. 1.1.

Таблица 1.1 – Таблица «Шаблон документа»

| **Параметр** | **Тип** | **Размер** | **Примечание** |
| --- | --- | --- | --- |
| Наименование шаблона | текстовый | 256 | Наименование шаблона |
| Дата создания шаблона | дата | 10 | Дата создания шаблона |
| Название файла шаблона | текстовый | 64 | Название документа, который является шаблоном |
| Начальное ключевое слово | текстовый | 256 | Ключевое слово, от которого начинается область вставки блоков текста |
| Конечное ключевое слово | текстовый | 256 | Ключевое слово, до которого идёт область вставки блоков текста |
| Количество вхождений | числовой | 4 | Количество вхождений промежутка от начального ключевого слова до конечного |

Таблица «Шаблон печати» предназначена для хранения информации об шаблонах, которые используются для создания и установки печати. Атрибуты таблицы «Шаблон печати» представлены в табл. 1.2.

Таблица 1.2 – Таблица «Шаблон печати»

| **Параметр** | **Тип** | **Размер** | **Примечание** |
| --- | --- | --- | --- |
| Наименование шаблона | текстовый | 64 | Наименование шаблона |
| Дата создания шаблона | дата | 10 | Дата создания шаблона |
| Название файла шаблона | текстовый | 64 | Название изображения, которое является шаблоном |
| Координаты точек | json | 64 | Представляет собой json файл, в котором хранятся три точки (1 точка – координаты вставки номера сертификата, 2 – текста владельца, 3 – даты действия) |

Инфологическая модель показала, что в программном продукте присутствуют две сущности. В приложении 2 представлена диаграмма сущность-связь.

## Входная и выходная информация

Для предоставления пользователю возможности пользоваться программой ему нужно подключиться к системе «Дело». Для этого требуются следующие данные:

* Имя
* Пароль

Для возможности создания шаблонов документов, требуются следующие данные:

* Название шаблона
* Дата создания шаблона
* Документ, на основе которого будет создаваться шаблон
* Начальное ключевое слово
* Конечное ключевое слово

Пример того, как будут выглядеть шаблоны документов представлен ниже.

* Содержимое документа
* Начальное ключевое слово:
* Блок №1
* Конечное ключевое слово
* Содержимое документа
* Начальное ключевое слово:
* Блок №2
* Конечное ключевое слово
* Содержимое документа
* Начальное ключевое слово:
* Блок №3
* Конечное ключевое слово.

Для возможности создания шаблонов печати, требуются следующие данные:

* Название шаблона
* Дата создания шаблона
* Изображение, на основе которого будет создаваться шаблон
* Область вставки номера сертификата
* Область вставки текста, содержащего владельца подписи
* Область вставки даты действия подписи

Ниже приведён пример шаблона печати.

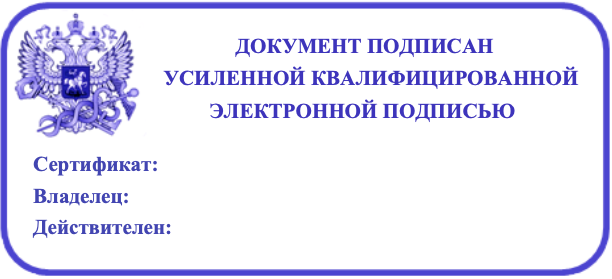


Рисунок 1.1 – пример шаблона печати

Для подписания электронного документа необходимы следующие данные:

* Word документ
* Шаблон печати в электронном виде
* Номер сертификата
* Владелец
* Дата начала действия подписи
* Дата окончания действия подписи

Выходными данными являются:

* Документы сгенерированные по определенному шаблону
* Документ подписанный с помощью ЭЦП
* Шаблоны документов

## Требования к программному и аппаратному обеспечению

*Требования к техническому обеспечению:*

* Процессор не ниже 1,6 ГГц;
* ОЗУ не меньше 1 Гб;

*Требования к программному обеспечению:*

* Операционная система Windows 7 или более поздние.
* Платформа .NET Framework 4.0.
* СЭД «Дело» версии 12.2.1

## Безопасность информационной системы

Поскольку в программе, используется EOS API, и никаких личных данных пользователей и внутренних документов организации не хранится, нет нужды предусматривать методы защиты данной информации, а также хранить данные для входа в систему в зашифрованном виде.

Исходя из способов классификации информационных систем по требованиям защиты информации, предлагаемым Федеральной службой по техническому и экспортному контролю, описанным в методическом документе «Меры защиты информации в государственных информационных системах» [12], разрабатываемую систему следует отнести к информационным системам с уровнем защищённости четвертого класса (К4), который предупреждает, что в результате нарушения одного из свойств безопасности информации возможны незначительные негативные последствия в социальной, политической, международной, экономической, финансовой или иных областях деятельности и (или) информационная система и (или) оператор (обладатель информации) могут выполнять возложенные на них функции с недостаточной эффективностью.

Согласно требованиям ФСТЭК, в информационной системе подлежат реализации следующие меры защиты информации:

* идентификация и аутентификация субъектов доступа и объектов доступа (ИАФ.1, ИАФ.2, ИАФ.3, ИАФ.4);
* управление доступом субъектов к объектам доступа (УПД.1, УПД.2, УПД.3, УПД.4, УПД.5, УПД.6, УПД.10, УПД.11, УПД.13, УПД.14, УПД.15, УПД.16);
* ограничение программной среды (ОПС.1, ОПС.2, ОПС.3);
* защита машинных носителей информации (ЗНИ.1, ЗНИ.2, ЗНИ.5, ЗНИ.8);
* регистрация событий безопасности (РСБ.1, РСБ.2, РСБ.3, РСБ.5, РСБ.6);
* антивирусная защита (АВЗ.1, АВЗ.2);
* контроль (анализ) защищенности информации (АНЗ.1, АНЗ.2, АНЗ.3, АНЗ.4);
* обеспечение целостности информационной системы и информации (ОЦЛ.1, ОЦЛ.3, ОЦЛ.4, ОЦЛ.6);
* обеспечение доступности информации (ОДТ.1, ОДТ.2, ОДТ.3, ОДТ.4, ОДТ.5, ОДТ.7);
* защита среды виртуализации (ЗСВ.1, ЗСВ.2, ЗСВ.3, ЗСВ.4, ЗСВ.6, ЗСВ.7, ЗСВ.8);
* защита технических средств (ЗТС.2, ЗТС.3, ЗТС.4, ЗТС.5);
* защита информационной системы, ее средств, систем связи и передачи данных (ЗИС.3, ЗИС.11, ЗИС.15, ЗИС.20, ЗИС.24, ЗИС.30).

**2 РАБОЧИЙ ПРОЕКТ**

**2.1 Общие сведения о работе системы**

Программный продукт разработан в среде разработки Visual Studio 2022, с использованием технологии Windows Presentation Foundation (WPF) на языке C# 10.

Программный продукт использует шаблон проектирования архитектуры приложения MVVM, что обеспечивает удобное создание и сопровождение ПО.

В качестве СУБД используется SQLite. В базе данных хранится информация о созданных шаблонах печати и шаблонах документов. Для связи СУБД и сервера используется Entity Framework Core v7.0.11. Для реализации MVVM, используются такие библиотеки как CommunityToolkit.MVVM версии 8.2.2 и Microsoft.Xaml.Behaviors.Wpf версии 1.1.17.

Для взаимодействия с файлами word используется библиотека Microsoft.Office. Interop.Word, а для файлов pdf – PdfSharp.

Для реализации внедрения зависимостей (DI) используется библиотека Microsoft.Extensions.Hosting.

**2.2 Функциональное назначение программного продукта**

Разработанный программный продукт предназначен для ускорения и облегчения подписания и формирования деловых документов.

Программа должна предоставлять пользователю возможность установки печати, подписания и формирования процессуальных документов и локальных нормативных актов в судебном делопроизводстве. Программа имеет следующие функциональные возможности:

* авторизация в программу с помощью системы «Дело»;
* вход в программу в оффлайн режиме;
* просмотр всех РК из системы дело доступных пользователю;
* фильтрация РК;
* просмотр всех файлов выбранной РК;
* просмотр шаблонов печатей;
* просмотр шаблонов документов;
* создание шаблонов печатей;
* создание шаблонов документов;
* удаление шаблонов печатей;
* удаление шаблонов документов;
* формирование документа на основе шаблона;
* подписание документа на основе шаблона печати.

**2.2 Описание физической архитектуры системы**

Программный продукт для формирования и подписания процессуальных документов и локальных нормативных актов в контексте в контексте судебной деятельности представляет автоматизированную систему, состоящую из двух компонентов: клиентского приложения и системы «Дело».

Клиентское приложение устанавливает соединение с системой «Дело» с помощью API EOS.

**2.3 Инсталляция и выполнение программного продукта**

Программный продукт содержится в папке Eos\_utility, в файле appsettings.json нужно указать строку подключения к базе «Дело» для дальнейшей работы с ней программы. После запустить eos\_utility.exe.

**2.4 Даталогическая модель данных**

Физическая модель данных реализована в СУБД SQLite.

Схема данных представлена в Приложении 2.

Ниже представлены таблицы баз данных с указанием имени поля, типа и назначения. Первичные ключи будут отмечены как PK (Primary Key)

Таблица «Шаблон печати» предназначена для хранения информации о пользователях системы.

Атрибуты таблицы «Шаблон печати» представлены в таблице 2.1.

Таблица 2.1 – Таблица «Шаблон печати»

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип** | **Примечание** |
| Id | uuid | Id таблицы, PK |
| Name | text | Название шаблона |
| FileName | text | Название файла, представляющий шаблон |

Продолжение таблицы 2.1

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип** | **Примечание** |
| PointsJson | json | Представляет собой json файл, в котором хранятся три точки (1 точка – координаты вставки номера сертификата, 2 – текста владельца, 3 – даты действия) |
| Created | dateTime | Дата создания записи |

Таблица «Шаблон документа» предназначена для хранения информации о тикетах, используемых для авторизации пользователей в системе.

Атрибуты таблицы «Шаблон документа» представлены в таблице 2.2.

Таблица 2.2 – Таблица «Шаблон документа»

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип** | **Примечание** |
| Id | uuid | Id таблицы, PK |
| Name | text | Название шаблона |
| FileName | text | Название файла, представляющий шаблон |
| Created | dateTime | Дата создания записи |
| StartKeyWord | text | Ключевое слово, от которого начинается область вставки блоков текста |
| EndKeyWord | text | Ключевое слово, до которого идёт область вставки блоков текста |
| CountRange | int | Кол-во вхождений |

**2.5 Обеспечение целостности данных**

Ограничения PRIMARY KEY и UNIQUE установлены на следующих полях в таблицах:

* поле «Id» в таблице «SealPattern»;
* поле «Id» в таблице «DocPattern».

Ограничения UNIQUE установлены на следующих полях в таблицах:

* поле «Name» в таблице «SealPattern»;
* поле «Name» в таблице «DocPattern».

**2.6 Использование JSON-объектов**

JSON-объектом является поле PointsJson в таблице «SealPattern», которое используется для координат точек, в которые нужно вставить текст на печати, с помощью которых подписываются документы.

**2.7 Описание программы**

В автоматизированной системе вся информация хранится в таблицах в базе данных, взаимодействие с которой осуществляется посредством класса **EFPatternRepository**, который наследует класс DbContext из EF, а также реализует такие интерфейсы как ISealPatternRepository и IDocPatternRepository. Описание данного класса приведено в таблице 2.3.

Таблица 2.3 – Описание класса EFPatternRepository

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| SealPatterns | private DbSet<SealPattern> | Справочник шаблонов печатей |
| DocPatterns | private DbSet<DocPattern> | Справочник шаблонов документов |
| **Метод** | | **Назначение** |
| public DatabaseContext (DbContextOptions<DatabaseContext> options) : base(options) | | Конструктор с параметрами |
| public void AddSealPattern(SealPattern imagePattern) | | Метод добавления нового шаблона печати |
| public IList<SealPattern> GetSealPatterns() | | Метод получения списка всех шаблонов печати |
| public SealPattern GetSealPatternByName(string name) | | Метод получения шаблона печати по названию |
| public List<string> GetSealPatternsNames() | | Метод получения списка названий всех шаблонов печати |
| public void RemoveSealPattern(Guid id) | | Метод удаления шаблона печати по идентификатору |
| public void AddDocPattern(DocPattern docPattern) | | Метод добавления нового шаблона документа |
| public List<DocPattern> GetDocPatterns() | | Метод получения списка всех шаблонов документов |

Продолжение таблицы 2.3

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| public DocPattern GetDocPatternById(Guid id) | Метод получения шаблона документа по идентификатору |
| public void RemoveDocPatternById(Guid removePatternId) | Метод удаления шаблона документа по идентификатору |
| public List<string> GetDocPatternsNames() | Метод получения списка названий всех шаблонов документов |
| DocPattern GetDocPatternByName(string name) | Метод получения шаблона документа по наименованию |

В таблице 2.4 приведено описание класса модели **SealPattern**, который представляет сущность шаблона печати в базе данных.

Таблица 2.4 – Описание класса SealPattern

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| Id | public Guid | Поле идентификатора |
| Name | public string | Поле наименования шаблона печати |
| FileName | public string | Поле наименования изображения, представляющего шаблон печати |
| Created | public DateTime | Поле даты создания шаблона |
| PointsJson | public string | Поле хранящее json точек |

В таблице 2.5 приведено описание класса модели **DocPattern**, который представляет сущность шаблона документа в базе данных.

Таблица 2.5 – Описание класса DocPattern

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| Id | public Guid | Поле идентификатора |
| Name | public string | Поле наименования шаблона документа |
| FileName | public string | Поле наименования документа, представляющего шаблон |
| Created | public DateTime | Поле даты создания шаблона |
| StartKeyWord | public string | Поле начального ключевого слова |

Продолжение таблицы 2.5

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| EndKeyWord | public string | Поле конечного ключевого слова |
| CountRange | public int | Поле количества вхождений |

В таблице 2.6 приведено описание класса **FileManager**, который упрощает работу с файловой системой для удобного создания шаблонов.

Таблица 2.6 – Описание класса FileManager

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| \_currDirPath | private string | Поле текущей директории |
| \_files | private string[] | Поле, содержащее массив файлов текущей директории |
| **Метод** | | **Назначение** |
| public FileManager(string dirPath) | | Конструктор с параметром |
| void Update() | | Обновляет массив файлов |
| public string[] GetFiles() | | Возвращает массив файлов |
| public bool ImportFile(string filePath, string fileName) | | Импортирует файл |
| public string GetFile(string FileName) | | Возвращает путь файла по названию |

В таблице 2.7 приведено описание класса **PdfFile**, который упрощает работу с документами формата pdf.

Таблица 2.7 – Описание класса PdfFile

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| \_PDFNewDoc | private PdfSharp.Pdf.PdfDocument | Поле pdf документа |
| **Метод** | | **Назначение** |
| public PdfFile(string filePath) | | Конструктор с параметром |
| public void Open(string filePath) | | Открывает документ по заданному пути |
| public void InsertPicture(string picturePath, double pictureWidth, double pictureHeight) | | Вставляет изображение в документ с заданным размером |
| public void Save(string filePath) | | Сохраняет pdf документ |

Продолжение таблицы 2.7

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| public void Close() | Закрывает открытый на данный момент документ |
| public void Dispose() | Реализует интерфейс IDisposable, для корректного освобождения объекта из памяти |

В таблице 2.8 приведено описание класса **WordFile**, который упрощает работу с документами формата doc и docx.

Таблица 2.8 – Описание класса WordFile

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| \_application | private Word.Application | Поле, представляющее приложение word |
| Document | public Word.Document | Поле, представляющее документ word |
| \_filePath | private string | Поле пути открытого документа |
| \_isOpen | private bool | Поле, показывающее открыт файл или нет |
| **Метод** | | **Назначение** |
| public WordFile(string filePath) | | Конструктор с параметром |
| public WordFile(string filePath, string copyPath) | | Конструктор с параметрами, для открытия копии файла |
| public bool OpenCopy(string filePath, string copyPath) | | Открывает копию документа |
| public bool Open(string filePath) | | Открывает документ по указанному пути |
| public int InsertTextInRange(string startWord, string endWord, string Text) | | Вставляет текст в определённый промежуток, от начального слова до конечного, если встречается не раз, то многократно |

Продолжение таблицы 2.8

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| public int RemoveTextInRange(string startWord, string endWord) | Удаляет текст из определённого промежутка, от начального слова до конечного, если встречается не раз, то многократно |
| public bool InsertPicture(string picturePath, double pictureWidth, double pictureHeight) | Вставляет изображение в документ с заданным размером |
| public void Save() | Сохраняет открытый на текущий момент документ |
| public bool IsOpen() | Возвращает логическое значение, открыт файл или нет |
| public void Close() | Закрывает открытый на данный момент документ |
| public void Dispose() | Реализует интерфейс IDisposable, для корректного освобождения объекта из памяти |

В таблице 2.9 приведено описание класса **RcFile**, который представляет объект коллекции ListBox в окне RcFiles для вывода файлов, прикрепленных к регистрационной карточке.

Таблица 2.9 – Описание класса RcFile

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| name | private string | Поле наименования файла |
| image | private ImageSource | Поле пути файла |
| **Метод** | | **Назначение** |
| public GetName() | | Возвращает наименование файла |
| public SetName(string name) | | Устанавливает наименование файла |
| public GetImage() | | Возвращает путь файла |
| public SetImage(ImageSource imageSource) | | Устанавливает путь файла |

В таблице 2.10 приведено описание класса **DataItem**, который представляет объект коллекции ItemsControl в окне TextBlocks для вывода полей для ввода текста.

Таблица 2.10 – Описание класса DataItem

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| text | private string | Поле текста блока |
| label | private string | Поле обозначения блока |
| **Метод** | | **Назначение** |
| public GetText() | | Возвращает текст блока |
| public SetText (string text) | | Устанавливает текст блока |
| public GetLabel() | | Возвращает обозначение блока |
| public SetLabel(string label) | | Устанавливает обозначение блока |

В таблице 2.11 приведено описание класса **Rectangle**, который представляет объект для отрисовывания прямоугольника в окне CreateSealPattern для создания шаблона печати.

Таблица 2.11 – Описание класса Rectangle

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| point1 | public Point | Поле первой точки |
| point2 | public Point | Поле второй точки |
| size | public Point | Поле размера прямоугольника, первая координата – длина, вторая - ширина |
| **Метод** | | **Назначение** |
| public Rectangle() | | Конструктор по умолчанию |
| public Rectangle(Point point1, Point point2) | | Конструктор с параметрами |
| public void CalculateSize() | | Подсчитывает размер прямоугольника |

Ниже приведено описание класса **WordToXpsConverter**, который используется для конвертации документа word в документ Xps, для дальнейшего вывода его в DocumentViewer.

Метод - public static async System.Threading.Tasks.Task ConvertToXps(string inputFilePath, string outputFilePath), назначение - Конвертирует файлы docx в формат xps.

В таблице 2.12 приведено описание класса **MainViewModel**, который выступает посредником между моделями и окном MainWindow.

Таблица 2.12 – Описание класса MainViewModel

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| username | [ObservableProperty]  private string | Поле имени пользователя |
| password | [ObservableProperty]  private string | Поле пароля пользователя |
| **Метод** | | **Назначение** |
| [RelayCommand]  public void ClickLinkOfflineMode() | | Команда, которая включает оффлайн режим |
| [RelayCommand]  public void ClickAuthorizeButton() | | Команда, которая авторизует пользователя через систему «Дело» |

В таблице 2.13 приведено описание класса **ChooseRcViewModel**, который выступает посредником между моделями и окном ChooseRcWindow.

Таблица 2.13 – Описание класса ChooseRcViewModel

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| EApiHead | private readonly dynamic | Устанавливает связь с БД системы «Дело» и хранит информацию о ней;  подключает пользователя в БД ДЕЛО и хранит информации об этом пользователе;  создаёт объекты остальных типов. |
| resultSet | private dynamic | Осуществляет отбор и хранения результатов этого  отбора |
| currentRcType | [ObservableProperty]  private string | Представляет текущий тип РК |
| currentRc | [ObservableProperty]  private int | Представляет индекс текущей РК |
| dateTimeSince | [ObservableProperty]  private DateTime | Представляет начальную дату для фильтра РК |

Продолжение таблицы 2.13

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| dateTimeUntil | [ObservableProperty]  private DateTime | Представляет конечную дату для фильтра РК |
| enabledDate | [ObservableProperty]  private bool | Представляет логическое значение включен ли фильтр РК по дате |
| RcTypesDictionary | private static Dictionary<string, RcTypes> | Словарь, хранящий названия типов РК, сопоставленный их типу |
| typesList | [ObservableProperty]  private List<string> | Представляет список названий типов РК |
| rcArray | [ObservableProperty]  private ObservableCollection<string> | Представляет список всех найденных РК |
| **Метод** | | **Назначение** |
| public ChooseRcViewModel(dynamic eApiHead) | | Конструктор с параметром |
| partial void OnCurrentRcTypeChanged(string value) | | Обновляет список РК при смене типа |
| partial void OnDateTimeSinceChanged(DateTime value) | | Обновляет список РК при смене начальной даты фильтра |
| partial void OnDateTimeUntilChanged(DateTime value) | | Обновляет список РК при смене конечной даты фильтра |
| partial void OnEnabledDateChanged(bool value) | | Обновляет список РК при смене переключателя по фильтру даты |
| [RelayCommand]  private void OnMouseDoubleClick() | | Открывает список прикреплённых файлов при двойном нажатии на РК |
| private void GiveDocs() | | Обновляет результат отбора РК |
| void fillList(dynamic src) | | Заполняет список РК |

В таблице 2.14 приведено описание класса **RcFilesViewModel**, который выступает посредником между моделями и окном RcFilesWindow.

Таблица 2.14 – Описание класса RcFilesViewModel

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| EApiHead | private readonly dynamic | Устанавливает связь с БД системы «Дело» и хранит информацию о ней;  подключает пользователя в БД ДЕЛО и хранит информации об этом пользователе;  создаёт объекты остальных типов. |
| resultSet | private readonly dynamic | Осуществляет отбор и хранения результатов этого  отбора |
| selectedRcIndex | private readonly int | Индекс выбранной РК |
| patternsFileManager | private FileManager | Упрощает работу с файлами шаблонов |
| sealPatternRepository | private ISealPatternRepository | Осуществляет возможность взаимодействия со справочником шаблонов печатей |
| docPatternRepository | private IDocPatternRepository | Осуществляет возможность взаимодействия со справочником шаблонов документов |
| toggleButtonValue | [ObservableProperty]  private bool | Представляет переключатель режима функционала – печать или документ |
| docPatterns | [ObservableProperty]  private ObservableCollection<string> | Представляет список шаблонов документов |

Продолжение таблицы 2.14

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| files | [ObservableProperty]  private ObservableCollection<RcFile> | Представляет список прикреплённых файлов текущей РК |
| sealPatterns | [ObservableProperty]  private ObservableCollection<string> | Представляет список шаблонов печатей |
| filePath | [ObservableProperty]  private string | Путь до выбранного файла |
| due | [ObservableProperty]  private string | Подразделение из системы «Дело» |
| sinceDate | [ObservableProperty]  private DateTime | Дата начала действия подписи |
| untilDate | [ObservableProperty]  private DateTime | Дата окончания действия подписи |
| certificate | [ObservableProperty]  private string | Номер сертификата |
| owner | [ObservableProperty]  private string | Владелец подписи |
| selectedSealPattern | [ObservableProperty]  private string | Выбранный шаблон печати |
| selectedDocPattern | [ObservableProperty]  private string | Выбранный шаблон документа |
| docsVisibility | [ObservableProperty]  private Visibility | Видимость функционала создания документа |
| sealsVisibility | [ObservableProperty]  private Visibility | Видимость функционала установки печати и подписи |
| **Метод** | | **Назначение** |
| public RcFilesViewModel() | | Конструктор по умолчанию |
| public RcFilesViewModel(dynamic oHead, dynamic resultSet, int number) | | Конструктор с параметрами |
| [RelayCommand]  private void ClickOpenBrowserDialog() | | Открывает диалоговое окно выбора папки |
| partial void OnToggleButtonValueChanged(bool value) | | Меняет режим функционала при смене переключателя |

Продолжение таблицы 2.14

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| [RelayCommand]  private void OnClickButtonInsertText() | Открывает окно с вставкой текстовых блоков в шаблон документа |
| [RelayCommand]  private void OnClickButtonDocPatternManager() | Открывает окно с менеджером шаблонов документов |
| [RelayCommand]  private void OnClickButtonChooseFile() | Открывает диалоговое окно выбора файла |
| [RelayCommand]  private void OnClickButtonChooseDue() | Открывает диалоговое окно выбора подразделения |
| [RelayCommand]  private void OnClickButtonSetSeal() | Устанавливает печать на выбранный документ и подписывает его |
| [RelayCommand]  private void OnClickButtonOpenSealPatternManager() | Открывает окно с менеджером шаблонов печатей |
| private void FillList() | Заполняет список прикреплённых файлов текущей РК |
| private void ChangePicture() | Берёт за основу шаблон изображения и устанавливает в нужные места данные, указанные пользователем |
| private void WordEDS() | Устанавливает печать на документ с расширениями doc или docx |
| private void PdfEDS() | Устанавливает печать на документ с расширением pfd |

Продолжение таблицы 2.14

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| public void save\_file\_wf(dynamic oHead, int action, int? aISN\_REF\_FILE, int? aISN\_RC, int? aKind\_Doc, string aDescription, string aCategory, int? aSecur, int? aLockFlag, string aFileAccessDues, string aFileName) | Прикрепляет документ к выбранной РК |
| dynamic check\_null(dynamic prop, string date\_mask) | Проверяет дату на null |

В таблице 2.15 приведено описание класса **PatternManagerViewModel<T>**, который выступает посредником между моделями и окном PatternManagerWindow.

Таблица 2.15 – Описание класса PatternManagerViewModel<T>

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| patterns | [ObservableProperty]  ObservableCollection<T> | Представляет список шаблонов |
| **Метод** | | **Назначение** |
| public PatternManagerViewModel(ObservableCollection<T> patterns) | | Конструктор с параметром |
| [RelayCommand]  private void AddPattern() | | Добавляет шаблон |
| [RelayCommand]  private void RemovePattern(T value) | | Удаляет шаблон |
| void OnWindowCreateDocPatternClose(Object sender, EventArgs eventArgs) | | Обновляет список шаблонов, когда закрывается окно создания шаблонов документов |
| void OnWindowCreateSealPatternClose(Object sender, EventArgs eventArgs) | | Обновляет список шаблонов, когда закрывается окно создания шаблонов печатей |

В таблице 2.16 приведено описание класса **CreateSealPatternViewModel,** который выступает посредником между моделями и окном CreateSealPatternWindow.

Таблица 2.16 – Описание класса CreateSealPatternViewModel

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Назначение** | |
| patternRepository | private ISealPatternRepository | Осуществляет возможность взаимодействия со справочником шаблонов печатей | |
| patternName | [ObservableProperty]  private string | Название шаблона | |
| patternPath | [ObservableProperty]  private string | Путь до изображения представляющее шаблон | |
| mousePosX | [ObservableProperty]  private string | Координата X позиции мышки на изображении | |
| mousePosY | [ObservableProperty]  private string | Координата Y позиции мышки на изображении | |
| imageHeight | [ObservableProperty]  private double | Высота изображения | |
| imageWidth | [ObservableProperty]  private double | Ширина изображения | |
| rectangles | [ObservableProperty]  private ObservableCollection<Rectangle> | Список прямоугольников для вставки информации | |
| **Метод** | | | **Назначение** |
| [RelayCommand]  private void OnClickOpenFileDialog() | | | Открывает диалоговое окно выбора файла png или jpg |
| [RelayCommand]  private void OnClickBackRectangle() | | | Убирает последнюю выделенную зону |
| [RelayCommand]  private void OnClickSavePattern(Object window) | | | Сохраняет текущий шаблон |
| public void OnLeftButtonDown(object s, MouseButtonEventArgs e) | | | Создаёт новую зону и сохраняет данные положения мыши на изображении |

Продолжение таблицы 2.14

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| public void OnMouseMove(object s, MouseEventArgs e) | Обновляет данные положения мыши на изображении и пересчитывает размер прямоугольника |

В таблице 2.17 приведено описание класса **CreateDocPatternViewModel,** который выступает посредником между моделями и окном CreateDocPatternWindow.

Таблица 2.17 – Описание класса CreateDocPatternViewModel

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Назначение** |
| doc | [ObservableProperty]  private FixedDocumentSequence | Представляет выбранный документ |
| filePath | [ObservableProperty]  private string | Путь к выбранному файлу |
| patternName | [ObservableProperty]  private string | Название шаблона |
| startKeyWord | [ObservableProperty]  private string | Начальное ключевое слово |
| endKeyWord | [ObservableProperty]  private string | Конечное ключевое слово |
| **Метод** | | **Назначение** |
| [RelayCommand]  private void OnClickCreatePattern(Object window) | | Сохраняет текущий шаблон |
| [RelayCommand]  private async Task OnClickButtonChooseFile() | | Открывает диалоговое окно для выбора docx или doc файла |

**2.8 Руководство пользователя**

После запуска программы на экран выводится окно с авторизацией через систему «Дело», либо же войти в оффлайн режим. Вид окна представлен на   
рисунке 2.1.

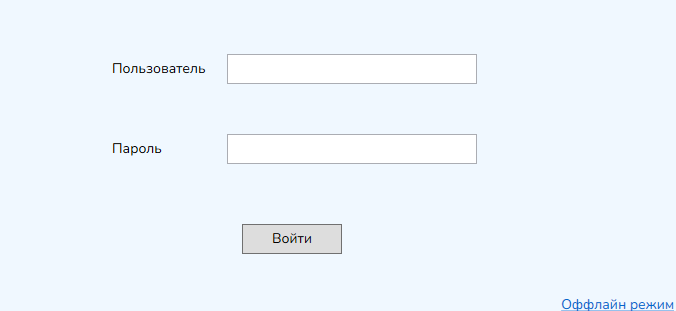


Рисунок 2.1 – Окно авторизации

После авторизации через систему «Дело», пользователю на экран выводится окно с со всеми доступными ему РК, вид которого представлен на рисунке 2.2, где он может отфильтровать РК по типу и дате.

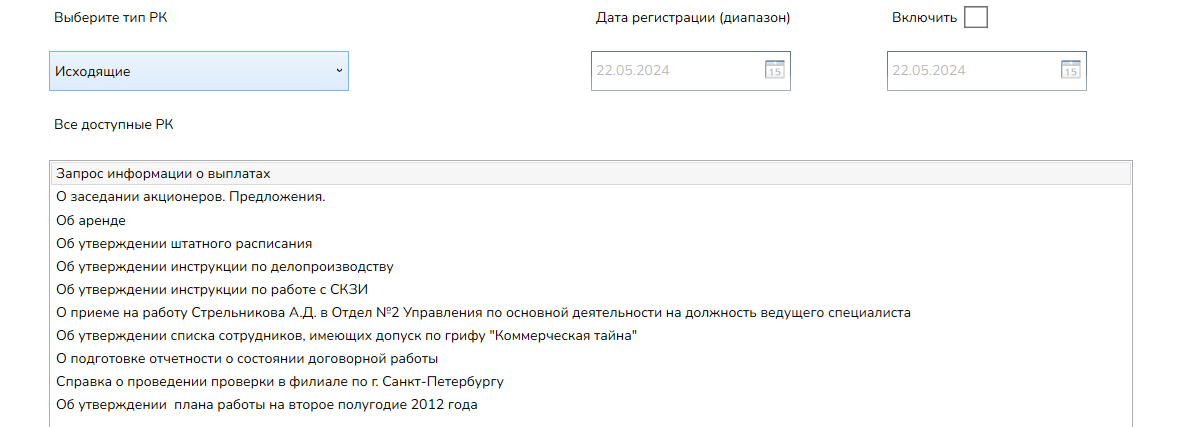


Рисунок 2.2 – Окно со всеми РК

При двойном нажатии на РК, открывается окно со всеми прикреплёнными файлами данной РК. В данном окне есть возможность создать новый документ на основе шаблона, а также подписать документ на основе шаблона.

Данное окно представлено на рисунке 2.3.

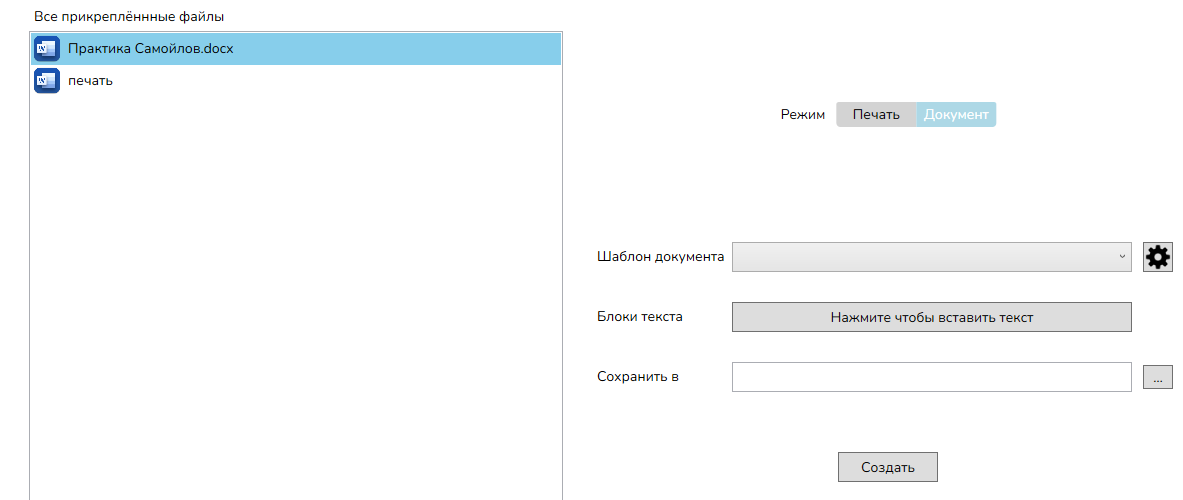


Рисунок 2.3 – Окно со всеми прикреплёнными к РК файлами

Для того чтобы создать документ на основе шаблона, нужно выбрать шаблон документа, после чего нажать на кнопку «Нажмите чтобы вставить текст», после чего откроется новое окно с шаблоном и текстовыми блоками, вид которого представлен на рисунке 2.4.

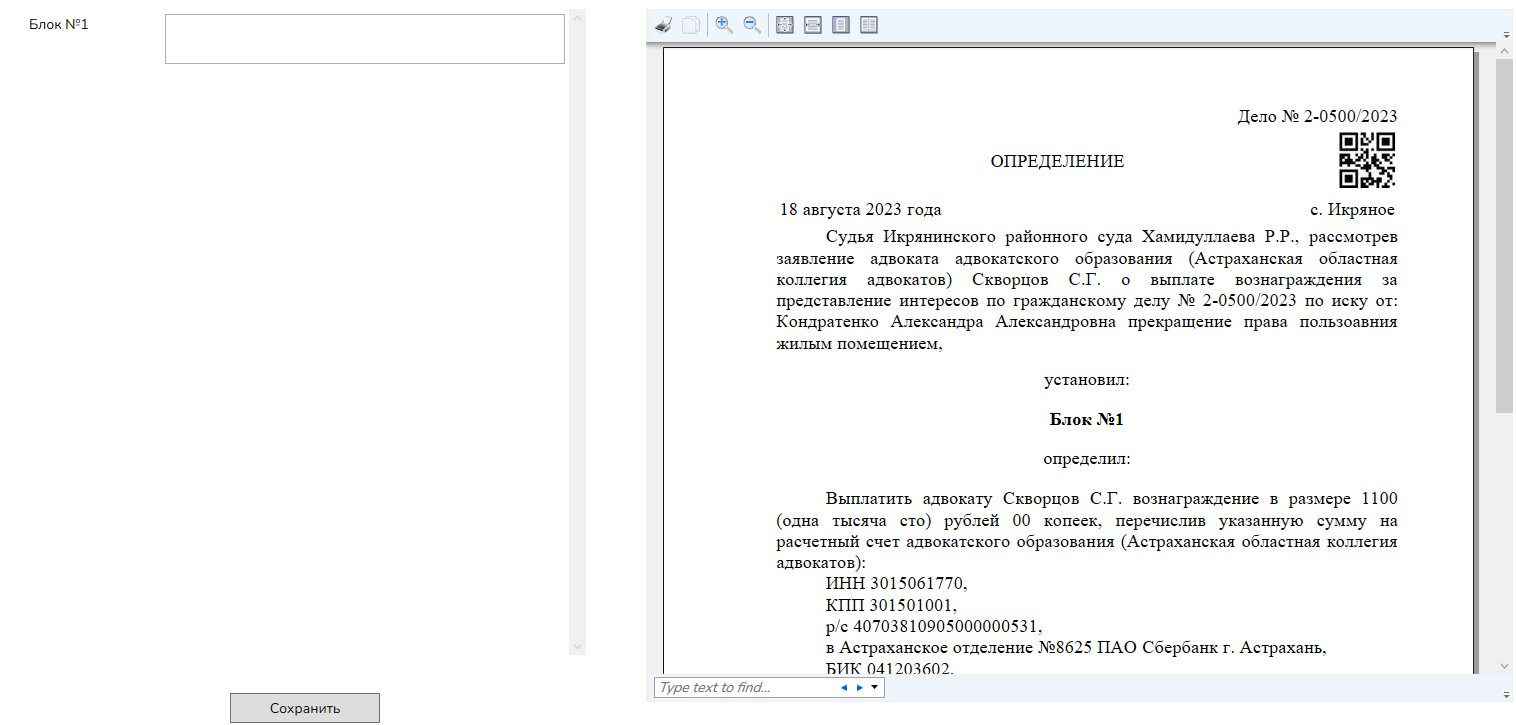


Рисунок 2.4 – Окно со всеми прикреплёнными к РК файлами

После чего требуется ввести во все доступные блоки текста нужную информацию и нажать сохранить. Окно автоматические закроется, после чего в окне, которое представлено на рисунке 2.3 нажать кнопку «Создать», создастся новый файл по шаблону в который будет вставлены блоки текста которые были указаны. Созданный файл автоматически отправится в базу данных системы «Дело» и прикрепится к текущей РК.

В окне, которое представлено на рисунке 2.3, можно открыть менеджер шаблонов документов нажав на шестерёнку рядом со списком шаблонов. Вследствие чего откроется новое окно менеджер шаблонов, в котором указаны имеющиеся шаблоны, а также можно создать новый или удалить имеющийся. Окно с менеджером шаблонов представлено на рисунке 2.5.

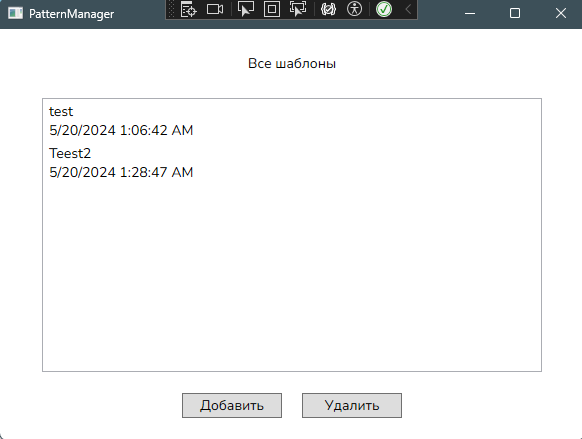


Рисунок 2.5 – Окно с менеджером шаблонов

При нажатии на кнопку создать открывается новое окно создания шаблона, вид окна представлен на рисунке 2.6.

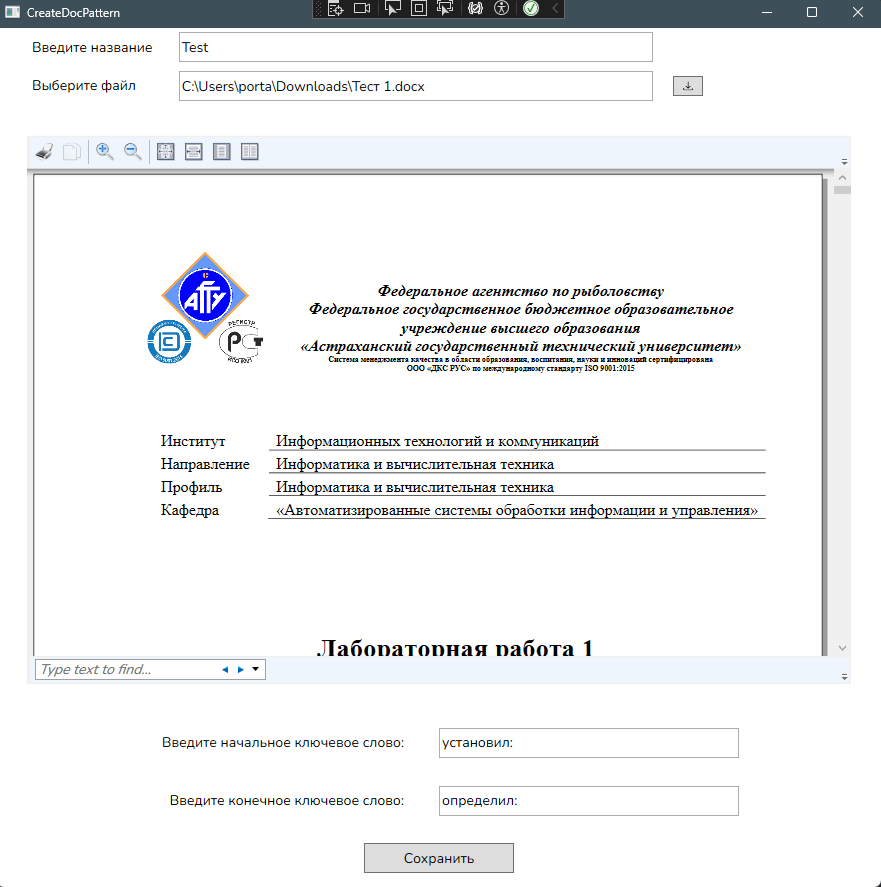


Рисунок 2.6 – Окно с созданием шаблона документа

В данном окне требуется ввести название шаблона, указать документ как основу шаблона, а также указать начальные и конечные ключевые слова, после чего нажать кнопку сохранить.

В окне, которое представлено на рисунке 2.3, можно переключить режим на печать, где открывается возможность установить печать на выбранный документ, предварительно выбрав шаблон печати и указав всю необходимую информацию для её установки, а также подписания документа. После чего требуется нажать на кнопку «Установить», создастся новый файл, в который будет вставлено изображение печати, а также подписано цифровой подписью. Созданный файл автоматически отправится в базу данных системы «Дело» и прикрепится к текущей РК.

Вид окна с режимом печати представлен на рисунке 2.7.

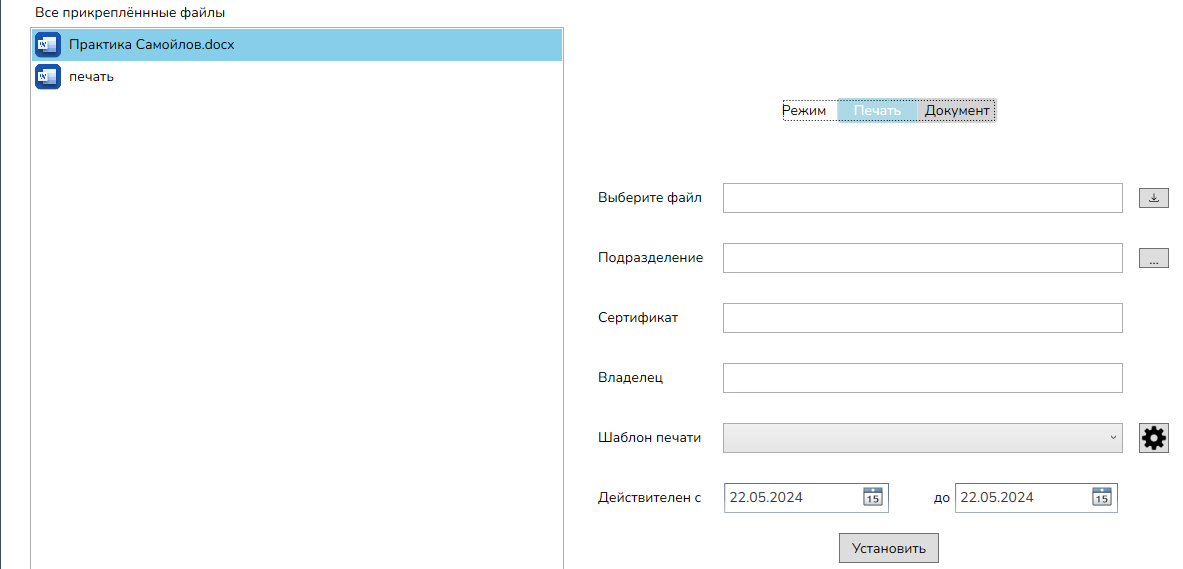


Рисунок 2.7 – Окно с режимом установки печати

В данном режиме также можно открыть менеджер шаблонов, в котором будут указываться все шаблоны печатей (см. рис. 2.5), при нажатии на кнопку создать, откроется новое окно с созданием нового шаблона печати, вид которого представлен на рисунке 2.8.

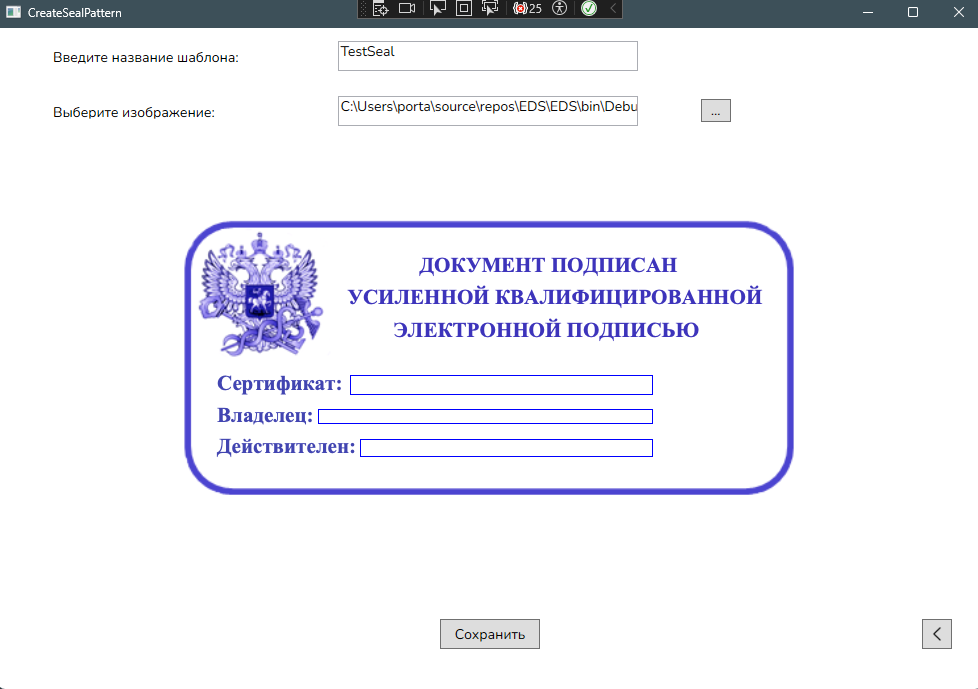


Рисунок 2.8 – Окно с созданием шаблона печати

В данном окне требуется указать название шаблона, указать изображение, которое является шаблоном и на изображении поочерёдно выделить три зоны вставки данных. Есть возможность отмены последнего выделения, нажав на кнопку «<«. После того как все условия будут выполнены нужно нажать на кнопку «Сохранить» и шаблон сохранится.

Также в окне, представленном на рисунке 2.1, есть возможность включить оффлайн режим, при нажатии на кнопку «оффлайн режим», откроется новое окно, в котором представлен функционал без взаимодействия с системой «Дело». Вид данного окна представлен на рисунке 2.9.

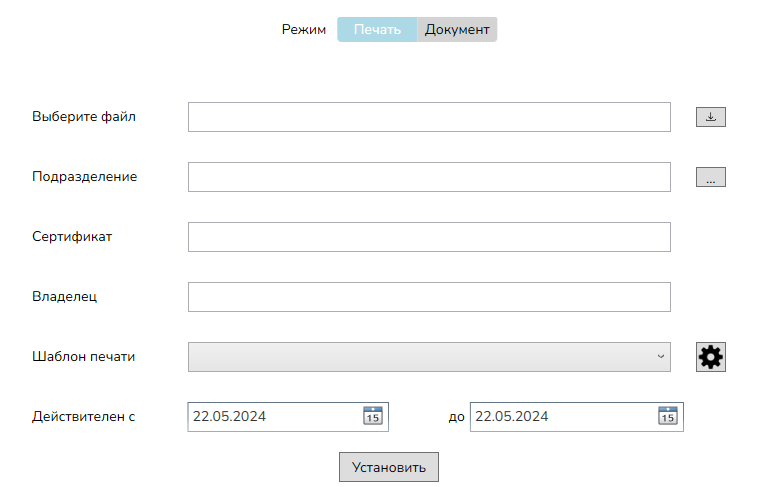


Рисунок 2.9 – Окно оффлайн режима

В оффлайн режим все созданные и подписанные документы не отправляются в систему «Дело», а сохраняются локально на ПК пользователя.

**2.9 Сообщения системы**

В таблице 2.18 приведены сообщения системы.

Таблица 2.18 – Сообщения системы

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Сообщение** | **Причина возникновения, способ**  **устранения** |
| 1 | Поле «Пользователь» не может быть пустым | Пользователь не ввел логин при авторизации |
| 2 | Поле «Пароль не может быть пустым | Пользователь не ввел пароль при авторизации |
| 3 | Не удалось установить соединение с БД ДЕЛО. | Пользователь ввёл не верные или не существующие данные. |
| 4 | Заданный диапазон не найден! | Пользователь ввёл ключевые слова которых нету в документе, при создании шаблона документа. |

Продолжение таблицы 2.18

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Сообщение** | **Причина возникновения, способ**  **устранения** |
| 5 | Шаблон успешно добавлен | Шаблон успешно был создан. |
| 6 | Нужно указать название шаблона | Пользователь не указал название создаваемого шаблона. |
| 7 | Нужно указать путь до файла | Пользователь не указал путь до файла. |
| 8 | Нужно выделить все три поля | Пользователь выделил не все три поля для вставки текста, при создании шаблона печати. |
| 9 | Выберите шаблон | Пользователь не выбрал шаблон из списка шаблонов. |
| 10 | Дата начала действия подписи должна быть меньше её окончания | Пользователь указал некорректные даты действия подписи. |
| 11 | Поддерживаются только форматы docx и pdf | Пользователь указал не предусмотренные расширения файлов. |
| 12 | Не удалось найти файл штампа | Файл шаблона был утерян или повреждён, требуется пересоздать его. |

**3 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ**

1. Запустить программу, убедиться, что появилось окно с авторизацией, авторизироваться через систему «Дело».
2. Убедиться, что открылось окно со всем доступными РК, поменять фильтры и убедиться в том, что выдаётся корректный результат.
3. Двойным нажатием левой кнопки мыши на любую РК, убедиться, что открылось окно со всеми прикреплёнными файлами.
4. Нажать на переключатель режимов печати и документов, убедиться в том, что всё корректно переключается.
5. В режиме документов нажать на кнопку настроек и убедиться, что открылось окно с менеджером шаблонов.
6. В менеджере шаблонов нажать на кнопку «Создать», убедиться, что открылось окно с созданием шаблона документа.
7. Выбрать файл, указать название шаблона, указать ключевые слова, и нажать кнопку «Сохранить». В менеджере шаблонов убедиться, что появился новый шаблон с таким же названием и датой создания.
8. Выбрать созданный шаблон, нажать на кнопку «Удалить», убедиться в том, что шаблон был удалён.
9. Повторить шаги 6 и 7.
10. В окне со всем прикреплёнными файлами выбрать шаблон, после чего нажать на кнопку «Нажмите чтобы вставить текст», убедиться в том, что открылось окно с вставкой текста в блоки, а также что вывелся шаблон.
11. Вставить текст в блоки и нажать кнопку «Сохранить».
12. После в окне со всем прикреплёнными файлами нажать кнопку «Создать». Убедиться, что создался новый файл и был прикреплён к текущей РК.
13. Поменять режим на печать, после чего нажать кнопку настроек, убедиться, что открылось окно с менеджером шаблонов.
14. В менеджере шаблонов нажать на кнопку «Создать», убедиться, что открылось окно с созданием шаблона печати.
15. Ввести название шаблона, указать изображение, после чего поочерёдно выделяем три зоны вставки текста, убеждаемся в корректном положении, и нажимаем кнопку «Сохранить». В менеджере шаблонов убедиться, что появился новый шаблон с таким же названием и датой создания.
16. Выбрать созданный шаблон, нажать на кнопку «Удалить», убедиться в том, что шаблон был удалён.
17. Повторить шаги 14 и 15.
18. В окне со всем прикреплёнными файлами выбрать шаблон, после чего ввести все нужные данные для произведения операции подписания и штампирования. Нажать на кнопку «Установить», убедиться в том, что создался новый файл и был прикреплён к текущей РК.
19. Закрыть все окна.
20. Запустить программу, убедиться, что появилось окно с авторизацией.
21. Нажать на кнопку «оффлайн режим», убедиться, что открылось окно оффлайн режима.
22. Нажать на переключатель режимов печати и документов, убедиться в том, что всё корректно переключается.
23. В режиме документов нажать на кнопку настроек и убедиться, что открылось окно с менеджером шаблонов.
24. В менеджере шаблонов нажать на кнопку «Создать», убедиться, что открылось окно с созданием шаблона документа.
25. Выбрать файл, указать название шаблона, указать ключевые слова, и нажать кнопку «Сохранить». В менеджере шаблонов убедиться, что появился новый шаблон с таким же названием и датой создания.
26. Выбрать созданный шаблон, нажать на кнопку «Удалить», убедиться в том, что шаблон был удалён.
27. Повторить шаги 6 и 7.
28. В окне со всем прикреплёнными файлами выбрать шаблон, после чего нажать на кнопку «Нажмите чтобы вставить текст», убедиться в том, что открылось окно с вставкой текста в блоки, а также что вывелся шаблон.
29. Вставить текст в блоки и нажать кнопку «Сохранить».
30. После в окне со всем прикреплёнными файлами, выбрать место куда будет сохранён документ и нажать кнопку «Создать». Убедиться, что создался новый файл и был сохранён в указанное место.
31. Поменять режим на печать, после чего нажать кнопку настроек, убедиться, что открылось окно с менеджером шаблонов.
32. В менеджере шаблонов нажать на кнопку «Создать», убедиться, что открылось окно с созданием шаблона печати.
33. Ввести название шаблона, указать изображение, после чего поочерёдно выделяем три зоны вставки текста, убеждаемся в корректном положении, и нажимаем кнопку «Сохранить». В менеджере шаблонов убедиться, что появился новый шаблон с таким же названием и датой создания.
34. Выбрать созданный шаблон, нажать на кнопку «Удалить», убедиться в том, что шаблон был удалён.
35. Повторить шаги 14 и 15.
36. В окне со всем прикреплёнными файлами выбрать шаблон, после чего ввести все нужные данные для произведения операции подписания и штампирования, а также выбрать место куда будет сохранён документ. Нажать на кнопку «Установить», убедиться в том, что создался новый файл и был сохранён в указанное место.

**ЗАКЛЮЧЕНИЕ**

В результате выполнения преддипломной практики были решены следующие задачи:

* определены ключевые сущности в системе;
* изучено API системы «Дело» – EOS API;
* изучена архитектура проектирования MVVM;
* изучены фреймворки создания MVVM приложений, а также библиотеки работы с документами Word и PDF.
* проектирование дизайна системы;
* проектирование и разработка основных интерфейсов системы.

Разработанный программный продукт позволяет упростить формирование и подписание процессуальных документов и локальных нормативных актов в судебном делопроизводстве.

Интерфейсы разработаны в соответствии с требованиями эргономики рабочего места пользователя информационной системы и интуитивно понятны.

В процессе выполнения преддипломной практики были получены практические навыки в среде PlantUML online и языка нотации UML, практические навыки в среде Visual Stuido 2022, проектировании структуры базы данных и ее реализации в СУБД SQLite, разработка клиентского приложения на WPF.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Хоменко Т.В., Белов С.В., Куркурин Н.Д., Мамлеева А.Р. Методические указания по подготовке к оформлению выпускной квалификационной работы бакалавра. – Астрахань, АГТУ, 2022. 51 с.
2. PlantUML в двух словах [Электронный ресурс] URL: <https://plantuml.com/ru/> Дата обращения: 15.05.2023
3. Программирование на С# для начинающих. Основные сведения / Алексей Васильев. – Москва: Эксмо, 2018. – 592 с. – (Российский компьютерный бестселлер).
4. Программирование на С# для начинающих. Особенности языка / Алексей Васильев. – Москва: Эксмо, 2019. – 528 с. – (Российский компьютерный бестселлер).
5. Холл Г.М., Адаптивный код на C#. Проектирование классов и интерфейсов, шаблоны и принципы SOLID – М.: Вильямс, 2015. – 432 с.: ил.
6. Фаулер М., Райс Д., Фоммел М., Хайет Э., Ми Р., Стаффорд Р., Шаблоны корпоративных приложений – М.: Вильямс, 2016. – 544 с.: ил
7. Справочник ЭЦП. Какие бывают виды электронной подписи. [Электронный ресурс] URL: <http://elektronnayapodpis.ru/wiki/kakie-byvayut-vidy-elektronnoy-podpisi> (дата обращения 16.11.2023).
8. Wikipedia. Digital signature. [Электронный ресурс] URL: <https://en.wikipedia.org/wiki/Digital_signature> (дата обращения 15.11.2023)
9. C# и Windows Forms | Первое приложение в Visual Studio [Электронный ресурс] URL: <https://metanit.com/sharp/windowsforms/1.1.php> (дата обращения 10.12.2023).
10. Роберт Мартин, Чистый код: создание, анализ и рефакторинг. – СПб.: Питер, 2021. – 464 с.: ил.
11. Гамма Э., Хэлм Р., Джонсон Р., Влиссидес Дж., Паттерны объектно-ориентированного проектирования. – СПб.: Питер, 2023. – 448 с.: ил. – (Серия «Библиотека программиста»).
12. Натан А. WPF 4. Подробное руководство. - Пер. с англ. - СПб.: Символ-Плюс, 2011. - 880 с., ил.

# ПРИЛОЖЕНИЕ 1

# Диаграмма вариантов использования

(Use Case Diagram)

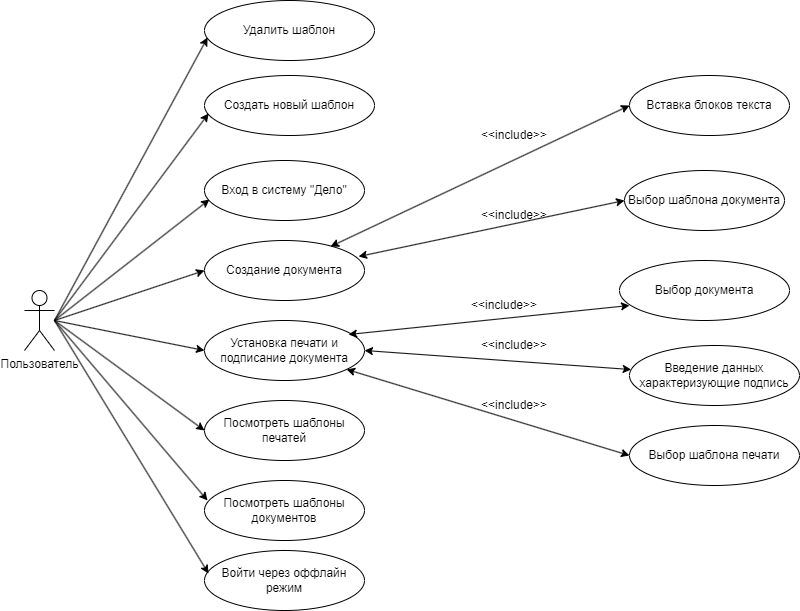


Рисунок 1 – диаграмма вариантов использования

# ПРИЛОЖЕНИЕ 2

# Диаграмма активности – «формирование документа»

(Activity Diagram)

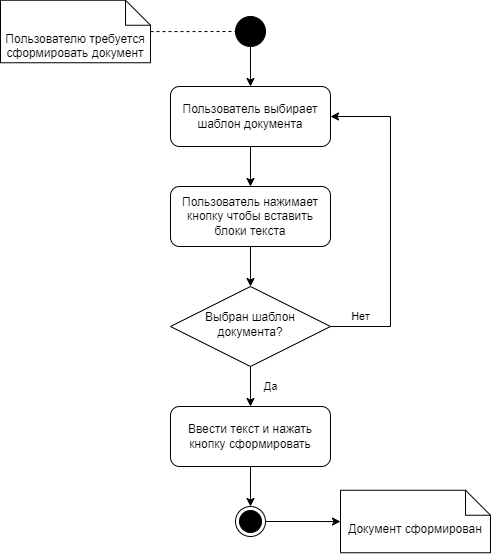


Рисунок 2 – диаграмма активности – «формирование документа»

# ПРИЛОЖЕНИЕ 3

# Диаграмма активности – «подписание документа»

(Activity Diagram)

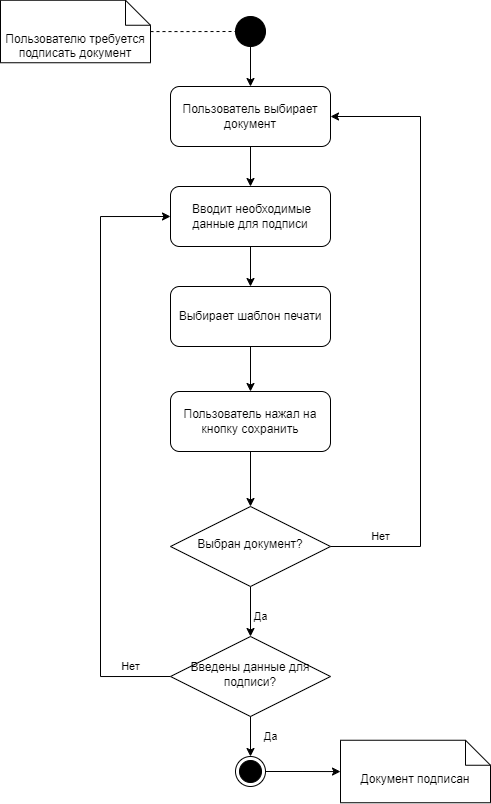


Рисунок 3 – диаграмма активности – «подписание документа»

# ПРИЛОЖЕНИЕ 4

# Диаграмма сущность-связь

(ER Diagram)

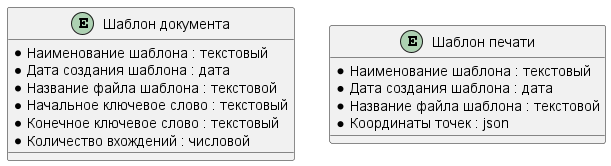


Рисунок 4 – диаграмма сущность-связь

# ПРИЛОЖЕНИЕ 5

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

**на разработку десктопного приложения для цифровых подписей, в рамках выпускной квалификационной работы**

Направление 090301 – Информатика и вычислительная техника

Исполнитель: студент гр. ДИНРб-41 **Самойлов Д. С.**

1. **Назначение, цели и задачи разработки**

**Цель разработки** – упрощение и ускорение процесса создания и подписывания документов сотрудниками областного суда.

**Назначение разработки:**

* Предоставить легкое и удобное приложение, позволяющее
* сокращение времени, затрачиваемого на создание или изменение документов сотрудниками суда;
* увеличение эффективного времени работы сотрудников суда;
* упрощение и ускорение процесса подписывания электронных документов.

**Основные задачи,** решаемые разработчиком в процессе выполнения выпускной квалификационной работы:

* анализ предметной области;
* формализация полученных данных;
* проектирование архитектуры будущей системы;
* разработка программного продукта в соответствии с требованиями;
* интегрирование разработанного продукта;
* документирование проекта в соответствии с установленными требованиями.

1. **Характер разработки:** выпускная квалификационная работа.
2. **Основания для разработки**

* Учебный план направления 09.03.01 «Информатика и вычислительная техника» 2020 года набора.
* Выпускная квалификационная работа, защита которой состоится летом 2024 года.
* Разработчик: студент группы ДИНРб-41 Самойлов Дмитрий Сергеевич.

Пользователи: работники департамента областного суда Астраханской области.

1. **Плановые сроки выполнения** – летний семестр 2023/24учебного года:

Начало «13» февраля 2024 г.

Окончание «15» июня 2024 г.

1. **Требования к проектируемой системе**
   1. **Требования к функциональным характеристикам**

Проектируемая система должна соответствовать следующим основным требованиям:

* предоставляет пользователю возможность выбрать электронный документ в проводнике или из системы «Дело»;
* предоставляет возможность подписать документы таких популярных форматов как doc, docx, pdf;
* предоставляет возможность создавать шаблоны документов;
* предоставляет возможность сгенерировать документ по определённому шаблону;
* предоставляет возможность ввести информацию с клавиатуры характеризующую подпись, например:
* Номер сертификата;
* Владельца;
* Дата начала действительности подписи;
* Дата окончания действительности подписи.
* предоставляет возможность просмотреть справку о программе и её создателе;
* предоставляет возможность сохранить часто вводимые данные и последний сгенерированный документ;
  1. **Требования к эксплуатационным характеристикам**

Программа не должна аварийно завершаться при любых действиях пользователя.

Время реакции программы на действия пользователя не должно превышать 2 секунд.

* 1. **Требования к программному обеспечению:**

Средства разработки: текстовый редактор Visual Studio 2022, платформа Windows Presentation Foundation, фреймворк .NET Core 6, язык C#.

Операционная система: Windows 7 и более поздние.

1. **Стадии и этапы разработки**
   1. **Эскизный проект (ЭП)**

* Анализ предметной области.
* Разработка структуры программного комплекса.
* Подготовка проектной документации.
  1. **Технический проект (ТП)**
* Разработка интерфейсов для просмотра.
* Разработка плана интеграций с другими частями системы.
* Подготовка пояснительной записки.
  1. **Рабочий проект (РП)**
* Программная реализация.
* Тестирование и отладка программы.
* Подготовка программной и эксплуатационной документации.
  1. **Эксплуатация (Э)**

Описание и анализ результатов проведенной работы.

1. **Требования к документированию проекта**

К защите выпускной квалификационной работы проекта должны быть представлены следующие документы:

* Пояснительная записка к ВКР:
* Презентация доклада.
* Программа, презентация и пояснительная записка к выпускной квалификационной работе на оптическом носителе.

Требования к структуре документов определены соответствующими стандартами ЕСПД.

Требования к оформлению определены соответствующими методическими указаниями.

1. **Порядок контроля и приемки**

Контроль выполнения выпускной квалификационной работы проводится руководителем поэтапно в соответствии с утвержденным графиком выполнения проекта.

На завершающем этапе руководитель осуществляет нормоконтроль представленной исполнителем документации и принимает решение о допуске (недопуске) проекта к защите.

Защита выпускной квалификационной работы проводится комиссией в составе не менее двух человек, включая руководителя проекта.

В процессе защиты проекта исполнитель представляет документацию, делает краткое сообщение по теме разработки и демонстрирует ее программную реализацию.

При выставлении оценки учитывается:

* степень соответствия представленной разработки требованиям технического задания;
* качество программной реализации, документации и доклада по теме проекта;
* соблюдение исполнителем графика выполнения выпускной квалификационной работы.

1. **Литература**
2. Хоменко Т.В., Белов С.В., Куркурин Н.Д., Мамлеева А.Р. Методические указания по подготовке к оформлению выпускной квалификационной работы бакалавра. – Астрахань, АГТУ, 2022. 51 с.
3. PlantUML в двух словах [Электронный ресурс] URL: <https://plantuml.com/ru/> Дата обращения: 15.05.2023
4. Программирование на С# для начинающих. Основные сведения / Алексей Васильев. – Москва: Эксмо, 2018. – 592 с. – (Российский компьютерный бестселлер).
5. Программирование на С# для начинающих. Особенности языка / Алексей Васильев. – Москва: Эксмо, 2019. – 528 с. – (Российский компьютерный бестселлер).
6. Холл Г.М., Адаптивный код на C#. Проектирование классов и интерфейсов, шаблоны и принципы SOLID – М.: Вильямс, 2015. – 432 с.: ил.
7. Фаулер М., Райс Д., Фоммел М., Хайет Э., Ми Р., Стаффорд Р., Шаблоны корпоративных приложений – М.: Вильямс, 2016. – 544 с.: ил
8. Справочник ЭЦП. Какие бывают виды электронной подписи. [Электронный ресурс] URL: <http://elektronnayapodpis.ru/wiki/kakie-byvayut-vidy-elektronnoy-podpisi> (дата обращения 16.11.2023).
9. Wikipedia. Digital signature. [Электронный ресурс] URL: <https://en.wikipedia.org/wiki/Digital_signature> (дата обращения 15.11.2023)
10. C# и Windows Forms | Первое приложение в Visual Studio [Электронный ресурс] URL: <https://metanit.com/sharp/windowsforms/1.1.php> (дата обращения 10.12.2023).
11. Роберт Мартин, Чистый код: создание, анализ и рефакторинг. – СПб.: Питер, 2021. – 464 с.: ил.
12. Гамма Э., Хэлм Р., Джонсон Р., Влиссидес Дж., Паттерны объектно-ориентированного проектирования. – СПб.: Питер, 2023. – 448 с.: ил. – (Серия «Библиотека программиста»).
13. Натан А. WPF 4. Подробное руководство. - Пер. с англ. - СПб.: Символ-Плюс, 2011. - 880 с., ил.

# ПРИЛОЖЕНИЕ 6

# Оптический носитель информации