МИНИСТЕРСТВО ЭНЕРГЕТИКИ, ПРОМЫШЛЕННОСТИ И СВЯЗИ СТАВРОПОЛЬСКОГО КРАЯ

Государственное бюджетное профессиональное образовательное учреждение «Ставропольский колледж связи

имени Героя Советского Союза В.А. Петрова»

Цикловая комиссия вычислительной техники

**КУРСОВАЯ РАБОТА**

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

на тему:

Разработка элементов информационной системы автоматизации документооборота для юридической компании

Студента Титова Сергея Дмитриевича

Специальность 09.02.07 Информационные

системы и программирование

Курс 4 группа ИП192

Руководитель Еременко М. П.

Оценка выполнения курсовой работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ / Еременко М.П.

Ставрополь, 2023 г.

Содержание

[Введение 3](#_Toc129264626)

[1. Теоретические основы разработки программного обеспечения 4](#_Toc129264627)

[1.1 Этапы жизненного цикла программных систем 4](#_Toc129264628)

[1.2 Функциональное моделирование программного продукта 4](#_Toc129264629)

[1.3 Объектно-ориентированный подход к разработке проекта 5](#_Toc129264630)

[2. Разработка программного продукта 7](#_Toc129264631)

[2.1 Техническое задание на разработку 7](#_Toc129264632)

[2.2 Функциональные диаграммы информационной системы 7](#_Toc129264633)

[2.3 Объектно-ориентированное проектирование системы 10](#_Toc129264634)

[2.3.1 Модели вариантов использования системы 10](#_Toc129264635)

[2.3.2 Диаграмма классов 11](#_Toc129264636)

[2.3.3 Диаграмма деятельности 11](#_Toc129264637)

[2.3.4 Диаграмма последовательности 12](#_Toc129264638)

[2.3.5 Диаграмма кооперации 13](#_Toc129264639)

[2.4 Разработка интерфейса программного продукта 14](#_Toc129264640)

[2.5 Техническая документация программной системы 17](#_Toc129264641)

[Заключение 18](#_Toc129264642)

[Список литературы 20](#_Toc129264643)

[Приложение 1 22](#_Toc129264644)

[Приложение 2 24](#_Toc129264645)

[Приложение 3 26](#_Toc129264646)

Введение

Современный мир невозможен без информационных технологий, которые позволяют эффективно управлять и обрабатывать большие объемы информации. В сфере юриспруденции, где имеется большое количество документов, создание информационной системы для управления документами является необходимостью.

Целью данного проекта является разработка информационной системы для управления документами в юридической фирме, которая позволит упростить процесс работы с документами, сократить время на поиск нужной информации и повысить качество работы с документами в целом.

В результате изучения предметной области проекта были выявлены основные требования к информационной системе для управления документами в юридической фирме. Одним из основных требований является безопасность хранения и обработки конфиденциальной информации.

Для достижения поставленных целей и требований в рамках проекта будут рассмотрены возможные способы реализации задачи.

# 1. Теоретические основы разработки программного обеспечения

# 1.1 Этапы жизненного цикла программных систем

1. Планирование: этот этап включает в себя определение целей и задач проекта, определение объема и сбор требований к системе.
2. Анализ: на этом этапе собранные требования анализируются для определения осуществимости проекта, определяется дизайн системы и создаются функциональные спецификации.
3. Дизайн: на этом этапе разрабатывается архитектура системы и создается детальный проект компонентов системы.
4. Внедрение: на этом этапе происходит фактическое программирование и кодирование системы, а программное обеспечение разрабатывается в соответствии с проектными спецификациями.
5. Тестирование. Этот этап включает в себя тестирование программного обеспечения, чтобы убедиться, что оно соответствует требованиям и функциональным спецификациям, а также выявить любые дефекты или проблемы, которые необходимо устранить.
6. Развертывание: на этом этапе программное обеспечение развертывается в производственной среде и становится доступным для конечных пользователей.
7. Техническое обслуживание: этот заключительный этап включает в себя постоянное техническое обслуживание и поддержку, чтобы гарантировать, что система продолжает удовлетворять потребности своих пользователей, а также устранять любые проблемы или дефекты, которые могут возникнуть.

# 1.2 Функциональное моделирование программного продукта

Функциональный подход к проектированию системы предполагает разбиение системы на составные части и определение того, как они взаимодействуют. Одной из популярных методологий для этого является интегрированное определение функционального моделирования IDEF0, которое используется для моделирования процессов, анализа и проектирования информационных систем. Другой методологией является метод IDEF3, который используется для моделирования и анализа бизнес-процессов. Диаграммы потоков данных DFD — это еще одна методология, используемая для представления потока данных в системе.

Функциональные диаграммы используются для представления функций системы, входов и выходов этих функций и того, как они взаимодействуют друг с другом. Целью элементов функциональной схемы является представление различных функций и их взаимосвязей друг с другом. Основными элементами функциональной схемы являются прямоугольники, стрелки и аннотации. Прямоугольники представляют функции системы, стрелки представляют поток входных и выходных данных между функциями, а аннотации описывают цель или детали каждой функции.

Правила размещения элементов на функциональной диаграмме заключаются в том, чтобы блоки располагались в логическом порядке: входы слева, а выходы справа. Стрелки должны быть помечены данными, которые они представляют, а аннотации должны быть размещены так, чтобы их было легко читать и понимать. В контексте курсовой работы на тему «Разработка информационной системы управления документами в юридической фирме» функциональная схема будет использоваться для представления различных функций и процессов, связанных с управлением документами в юридической фирме, включая такие задачи, как как создание, просмотр, утверждение, хранение и поиск документов.

# 1.3 Объектно-ориентированный подход к разработке проекта

Принципы объектно-ориентированного представления программных систем:

1. Абстракция: сосредоточение внимания на существенных характеристиках объекта или системы и игнорирование других.
2. Инкапсуляция: объединение данных и функций в единый блок (класс) и ограничение доступа к его внутренней работе.
3. Наследование: механизм, в котором новый класс основан на существующем классе и приобретает свойства и поведение родительского класса.
4. Полиморфизм: способность объекта принимать множество форм, позволяющая одинаково обращаться с разными объектами.

Типы моделей в UML:

1. Структурные модели: описывают статическую структуру системы, такую как диаграммы классов, диаграммы объектов и диаграммы компонентов.
2. Поведенческие модели: описывают динамическое поведение системы, например, диаграммы действий, диаграммы вариантов использования и диаграммы состояний.

Правила создания моделей UML:

1. Использование стандартной нотации UML.
2. Схемы должны быть простыми и легко читаемыми.
3. Использование соответствующие отношения между объектами и классами.
4. Добавление соответствующих меток и аннотаций, чтобы прояснить значение модели.
5. Следование принципа модульности, чтобы сделать модель более управляемой и понятной

# 2. Разработка программного продукта

# 2.1 Техническое задание на разработку

Техническое задание (ТЗ) — это документ, определяющий объем, цели, результаты и требования проекта. Его цель - установить общее понимание между заказчиком и подрядчиком и гарантировать, что проект будет завершен вовремя, в рамках бюджета и к удовлетворению заказчика.

В процессе проектирования информационной системы документооборота в юридической фирме было создано и утверждено техническое задание, обеспечивающее четкое понимание целей и требований проекта. Документ ТЗ включает подробное описание масштаба проекта, функциональности, требований к производительности, сроков и бюджета. Это важный документ, который служит основой для планирования проекта, мониторинга и контроля. Использование ТЗ гарантирует, что все заинтересованные стороны имеют общее понимание проекта, что имеет решающее значение для успеха проекта.

С техническими заданием информационной системы можно ознакомиться в Приложении 1.

* 1. Функциональные диаграммы информационной системы

Контекстная диаграмма – это диаграмма высокого уровня, которая обеспечивает обзор информационной системы, показывая ее взаимодействие с внешними сущностями, такими как пользователи, другие системы или организации. Она помогает определить границы системы и идентифицировать ее входы и выходы. С ней можно ознакомиться на рисунке 1.

Диаграмма IDEF0 – это инструмент функционального моделирования, который использует структурированный подход для описания функций, входов, выходов и элементов управления системы или процесса. Она обеспечивает визуальное представление системы и ее компонентов, позволяя лучше понять, как работает система и как ее можно улучшить.

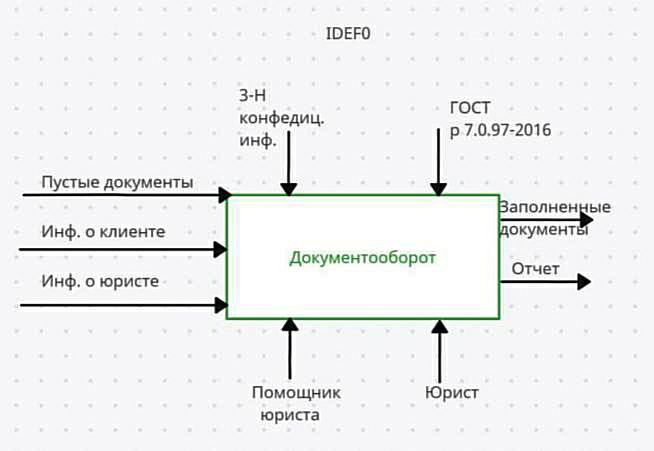


Рис. 1. Контекстная диаграмма информационной системы

Первая декомпозиция диаграммы IDEF0 разбивает систему на основные функции и определяет входы и выходы каждой функции. Это позволяет получить детальное представление о системе и ее компонентах. Данная разбивка изображена на рисунке 2.

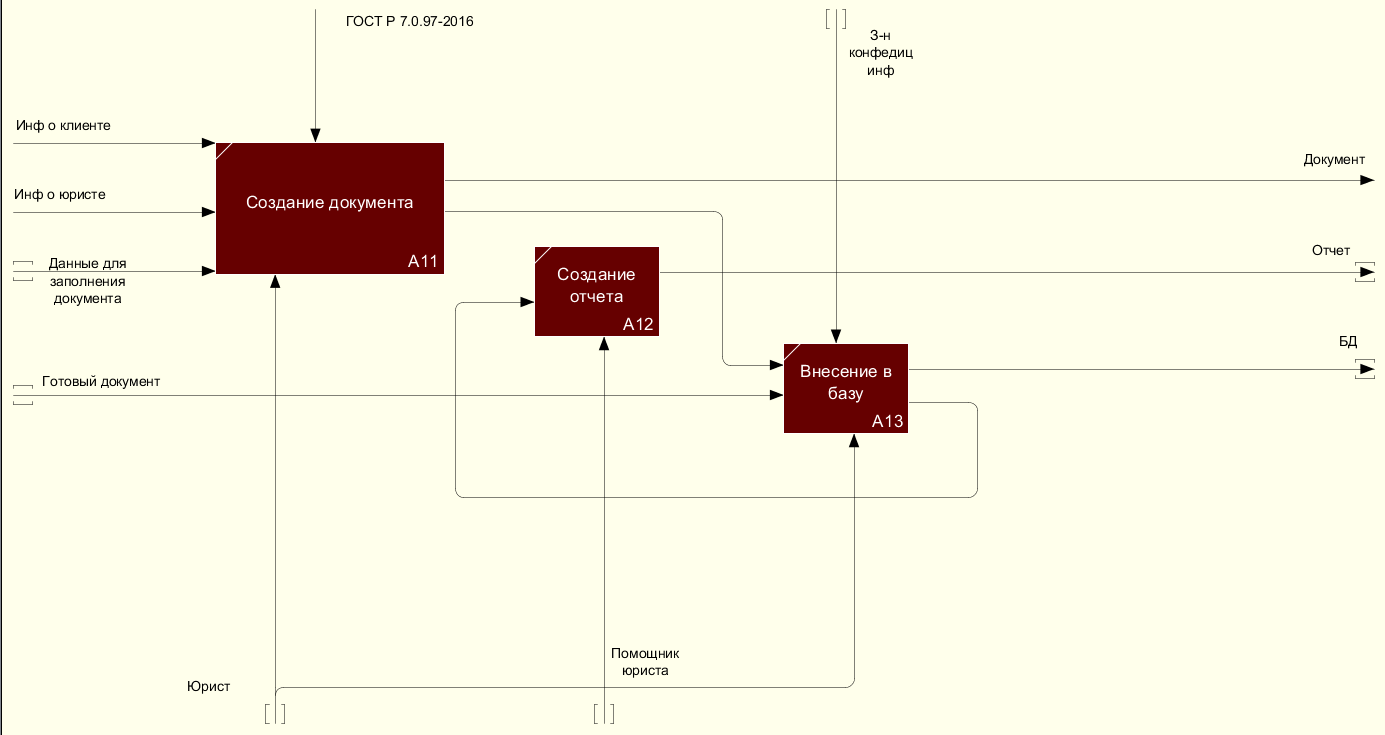


Рис. 2. IDEF0 – первое разбиение.

Вторая декомпозиция диаграммы IDEF0 (рисунок 3) еще больше разбивает каждую основную функцию на более мелкие, более детальные функции, обеспечивая более глубокое понимание системы и ее компонентов. Такой уровень детализации помогает определить области, в которых система может быть улучшена и оптимизирована.

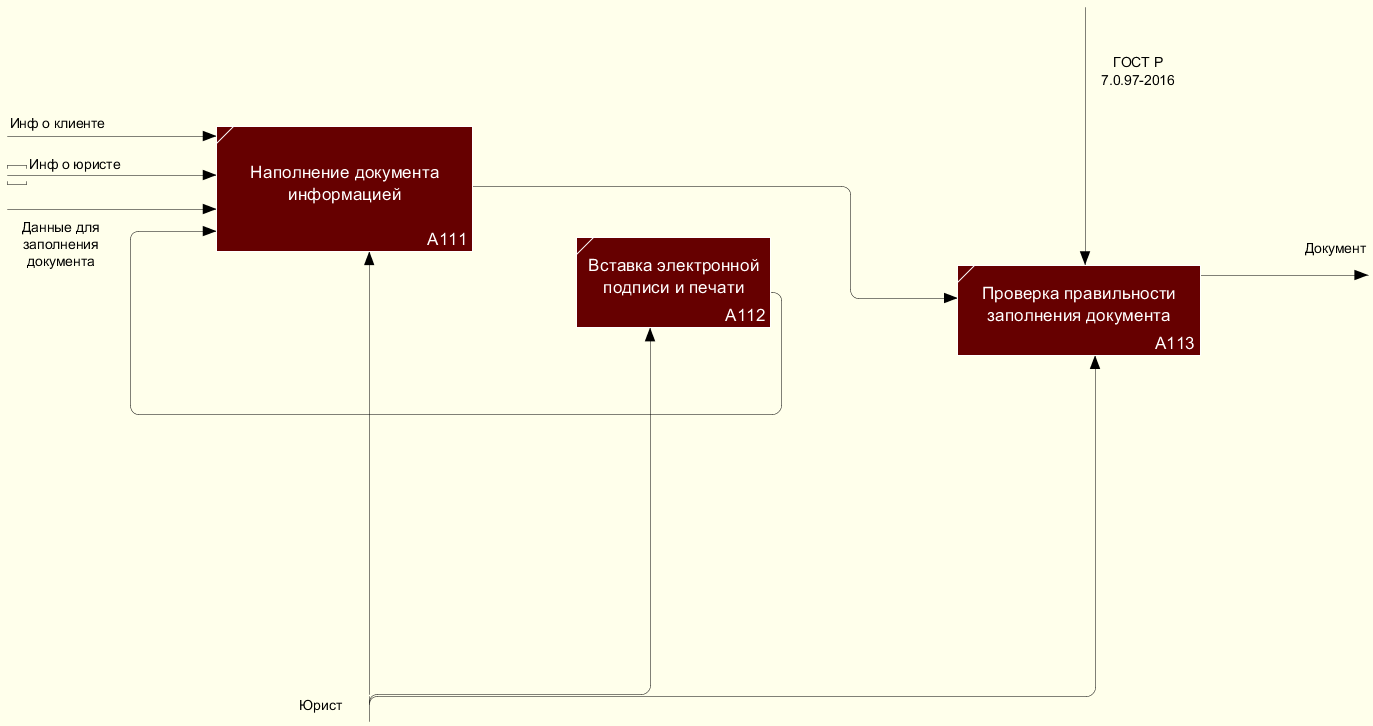


Рис. 3. IDEF0 – второе разбиение (разбиение процесса создания документа).

Диаграмма потока данных (DFD) — это графическое представление потока данных в системе. Она показывает, как информация вводится, обрабатывается и выводится системой или организацией. В DFD используются символы для представления процессов, хранилищ данных, потоков данных и внешних объектов. Они широко используются в системном анализе и проектировании, чтобы помочь понять и документировать сложные системы. DFD могут использоваться для моделирования как ручных, так и автоматизированных систем, и могут применяться для выявления неэффективности или потенциальных областей для улучшения системы. Такая диаграмма для нашей системы изображена на рисунке 4.

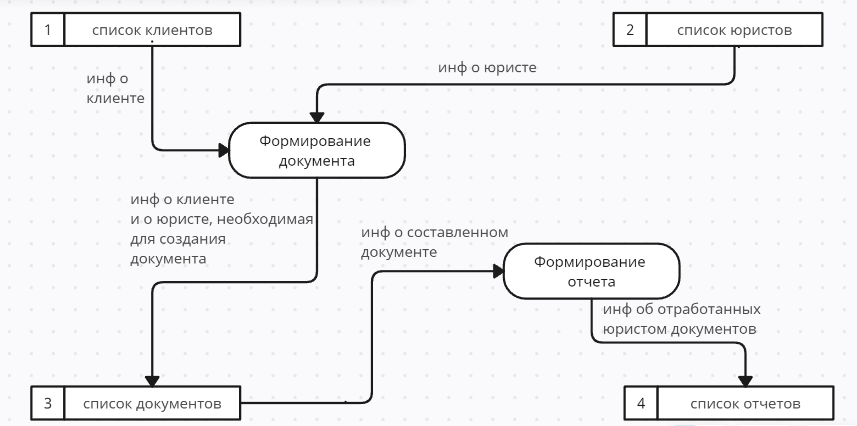


Рис. 4. DFD диаграмма информационной системы.

* 1. Объектно-ориентированное проектирование системы

### 2.3.1 Модели вариантов использования системы

Для данной системы диаграмма вариантов использования изображена на рисунке 5.

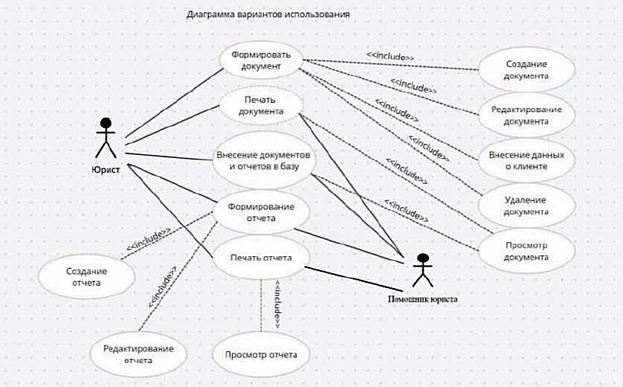


Рис. 5. Диаграмма вариантов использования для информационной системы.

Диаграмма вариантов использования – это тип диаграммы, который показывает различных участников (пользователей), взаимодействующих с системой, и конкретные действия или варианты использования, которые каждый участник может выполнять в системе. Она обеспечивает высокоуровневое представление системы и ее предполагаемой функциональности с точки зрения пользователей.

### 2.3.2 Диаграмма классов

Диаграмма классов – это тип диаграммы UML, которая показывает структуру системы, иллюстрируя классы, атрибуты, операции или методы и отношения между классами. Она показывает объекты в системе, их свойства и операции, которые могут быть выполнены над ними. Нагляднее с диаграммой классов можно ознакомится на рисунке 6.

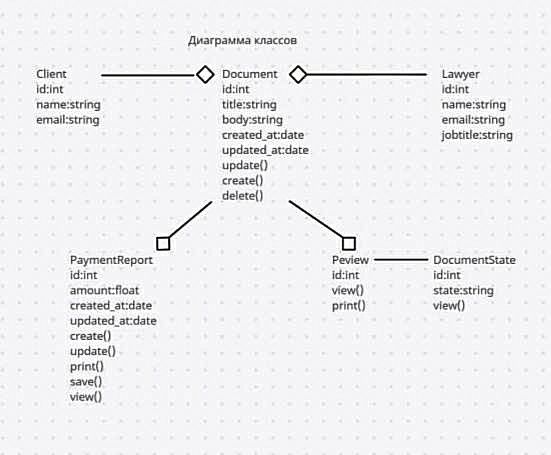


Рис. 6. Диаграмма классов.

### 2.3.3 Диаграмма деятельности

Диаграмма деятельности показывает поток работ или действий, которые происходят в системе, процессе или варианте использования. Она используется для моделирования поведения системы или процесса и обычно показывает последовательность действий, точки принятия решений и разветвляющиеся пути, которые возникают на основе определенных условий. Для нашей системы диаграмма деятельности изображена на рисунок 7.

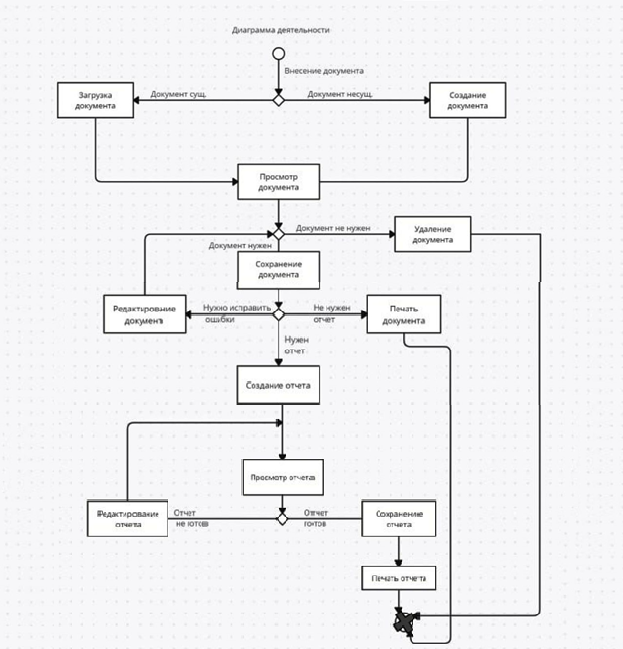


Рис. 7. Диаграмма деятельности.

### 2.3.4 Диаграмма последовательности

Диаграмма последовательности – это тип диаграммы взаимодействия, которая показывает последовательность взаимодействия между объектами или компонентами системы во времени. Она демонстрирует, как происходит обмен сообщениями между объектами и в каком порядке, чтобы достичь конкретной задачи или сценария использования. Она полезна для моделирования и визуализации потока управления в системе, а также для выявления потенциальных проблем и возможностей для оптимизации. В нашем случае диаграмма последовательности выглядит так, как показано на рисунке 8.



Рис. 8. Диаграмма последовательности

### 2.3.5 Диаграмма кооперации

Диаграмма кооперации (рисунок 9) — это тип диаграммы в UML, которая показывает взаимодействие между объектами или компонентами в системе для выполнения конкретной задачи или достижения определенной цели. Она фокусируется на сообщениях или методах, которые передаются между объектами или компонентами, и на том, как они работают вместе для достижения желаемого результата.

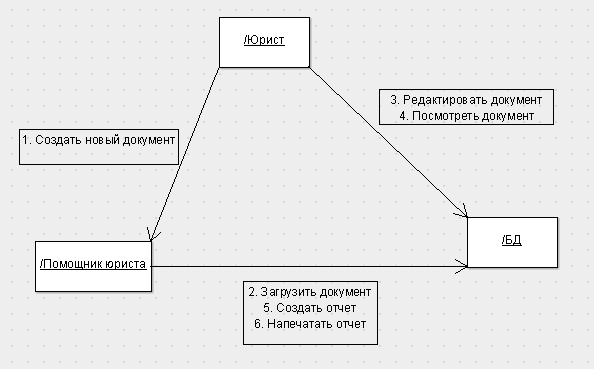


Рис. 9. Диаграмма кооперации

### 2.4 Разработка интерфейса программного продукта

Для создания программного интерфейса и информационной системы важно разработать базу данных, поскольку она служит основой для хранения и организации данных. В базе данных хранится вся информация, необходимая для работы системы, например, учетные записи пользователей, информация о клиентах, шаблоны документов и готовые документы.

Разработка базы данных включает в себя несколько этапов, в том числе проектирование схемы базы данных, создание таблиц и связей, а также заполнение базы данных данными. Разработка схемы включает в себя определение типов данных, которые необходимо хранить, взаимосвязей между данными и того, как данные будут организованы в таблицы. После разработки схемы создаются таблицы и устанавливаются связи между ними.

База данных «Lawyer\_BD» служит основой информационной системы, предоставляя необходимые данные для правильного функционирования программного интерфейса и других компонентов системы. Для наглядности на рисунке 10 изображена схема базы данных в программе MS SQL Management Studio.

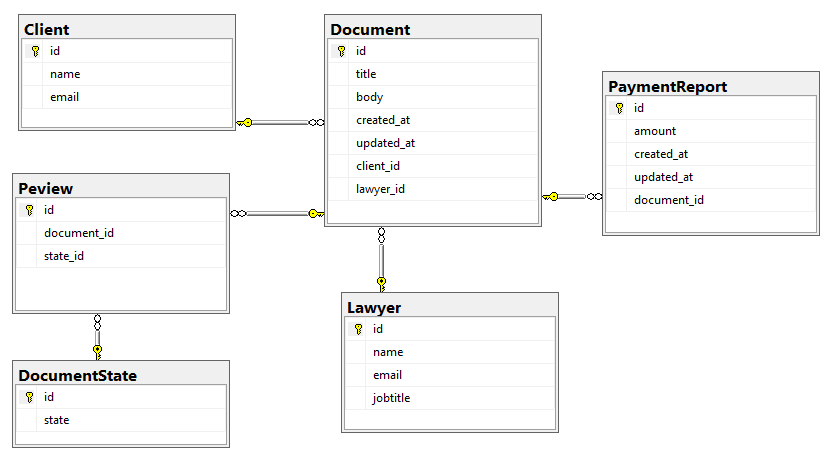
**

Рис. 10. Схема базы данных «Lawyer\_BD»

Интерфейс приложения – это часть программной системы, с которой взаимодействуют пользователи. Он предоставляет пользователям визуальный и интерактивный способ ввода данных, просмотра информации и управления функциональностью системы. Дизайн интерфейса важен, потому что он напрямую влияет на взаимодействие с пользователем, а хорошо продуманный интерфейс может сделать систему более интуитивно понятной и удобной для пользователя. На рисунке 11 можно увидеть первоначальное окно «Вход в систему». После авторизации, в зависимости от введённого логина и пароля, можно войти в аккаунт Юриста (рисунок 12) либо в аккаунт Помощника юриста (рисунок 13). С исходным кодом продукта можно ознакомиться в Приложении 2.

C# - это язык программирования, разработанный Microsoft, который широко используется для разработки приложений в среде .NET. C# имеет ряд особенностей, которые делают его хорошим выбором для разработки интерфейса информационной системы, в том числе:

1. Объектно-ориентированный: предназначен для поддержки создания сложных программных систем.
2. Простота в освоении: относительно простой язык для изучения.
3. Кроссплатформенность: можно использовать для разработки приложений, которые могут работать на разных платформах, что делает его универсальным языком для разработки интерфейсов.
4. Интегрированная среда разработки (IDE): используется в сочетании с Microsoft Visual Studio, мощной IDE, которая предоставляет ряд инструментов и функций для разработки и отладки приложений.

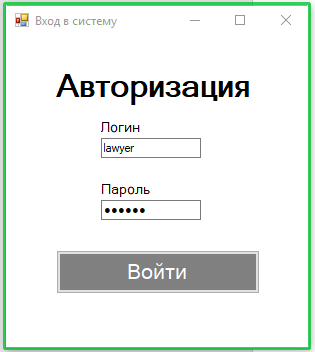


Рис. 11. Окно «Вход в систему»

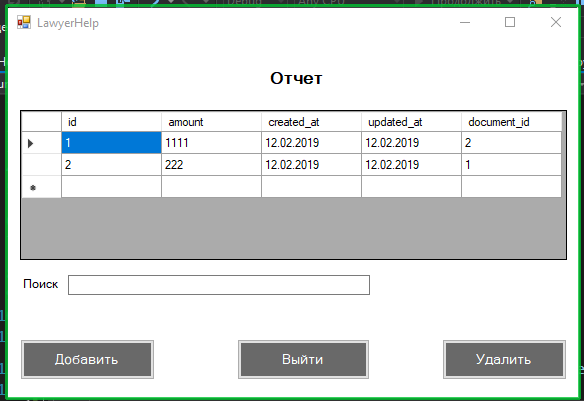


Рис. 12. Аккаунт «Помощника юриста».

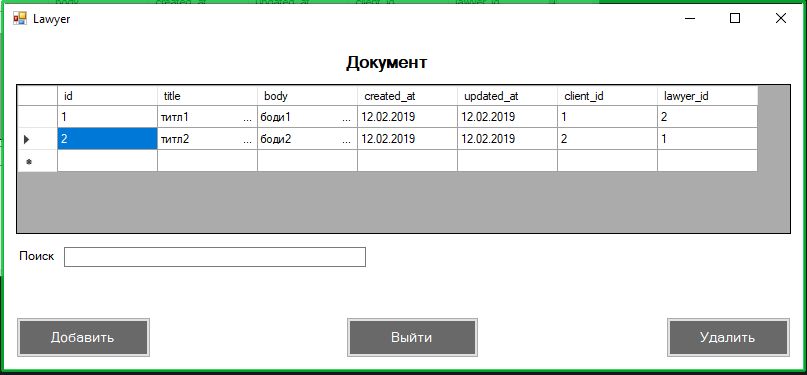


Рис. 13. Аккаунт «Юриста».

### 2.5 Техническая документация программной системы

Руководство пользователя — это документ, содержащий инструкции по использованию программной системы или приложения. Его цель — помочь пользователям понять систему и ее функциональные возможности, а также направить их в выполнении конкретных задач или операций в системе.

При подготовке Руководства пользователя следует руководствоваться нормативными документами, такими как руководства по стилю и стандартные рабочие процедуры, чтобы обеспечить последовательность и ясность представления информации. Кроме того, Руководство пользователя должно соответствовать всем применимым законам или правилам, связанным с системой программного обеспечения и ее использованием.

Следует отметить, что в процессе разработки системы документооборота, также было создано Руководство пользователя, дающее рекомендации по использованию системы.

С Руководством пользователя информационной системы можно ознакомиться в Приложении 3.

### Заключение

В результате курсовой работы была разработана информационная система для совершенствования процесса документооборота в юридической фирме, позволяющая обеспечить централизованный и эффективный способ хранения, управления и отслеживания документов, а также автоматизировать определенные рутинные задачи, такие как создание, распространение и утверждение документов.

В результате применения данной информационной системы будет повышена производительность, уменьшено количество ошибок и улучшена совместная работа членов команды.

Недостатки проекта:

1. Отсутствие четких целей: проект может не иметь четких и четко определенных целей, что может затруднить определение того, достигает ли система своих целей.
2. Неадекватный пользовательский ввод: проект мог не получить достаточного вклада от пользователей системы управления документами, таких как юристы, помощники юристов и административный персонал, что может привести к тому, что система не будет соответствовать их потребностям.
3. Недостаточное тестирование системы. Возможно, проект не провел достаточного тестирования системы управления документами для выявления и устранения любых ошибок, уязвимостей в системе безопасности или других проблем.
4. Ограниченная масштабируемость: проект мог не учитывать потенциальный рост и расширение юридической фирмы, что может ограничить масштабируемость системы управления документами.

Возможные пути дальнейшего улучшения:

1. Четкое определение цели. Команда проекта должна четко определить цели системы управления документами, в том числе то, что она должна выполнять, и как будет измеряться успех.
2. Вовлечение пользователей в процесс разработки. Команда проекта должна вовлекать пользователей системы управления документами в процесс разработки, в том числе собирать их мнения и отзывы, чтобы убедиться, что система отвечает их потребностям и удобна для пользователя.
3. Проведение тщательного тестирования. Команда проекта должна провести тщательное тестирование системы управления документами перед ее развертыванием, включая функциональное тестирование, тестирование производительности и тестирование безопасности, чтобы убедиться, что она надежна, безопасна и работает должным образом.
4. Планирование масштабируемости: проектная группа должна планировать потенциальный рост и расширение юридической фирмы, разрабатывая масштабируемую и адаптируемую к изменяющимся потребностям и требованиям систему управления документами. Это может включать использование облачных решений или внедрение других технологий, обеспечивающих простоту масштабирования.
5. Обеспечение обучения и поддержки: команда проекта должна обеспечить надлежащее обучение и поддержку пользователей системы управления документами, чтобы убедиться, что они знают, как использовать ее эффективно и действенно. Это может включать создание руководств пользователя, проведение учебных занятий и предложение постоянной поддержки и помощи.

Также стоит отметить, что разработка системы управления документами является важным шагом на пути к обеспечению эффективного и действенного управления документами в юридической фирме. Хорошо продуманная система управления документами может значительно повысить общую производительность и конкурентоспособность юридической фирмы.

Список литературы

1. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование: учебное пособие для спо / С. В. Белугина. — 3-е изд., стер. — Санкт-Петербург: Лань, 2022. — 312 с. — ISBN 978-5-8114-9817-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/200390>
2. [Зверева В. П.](http://academia-moscow.ru/authors/detail/193626/), [Назаров А. В.](http://academia-moscow.ru/authors/detail/82672/) Сопровождение и обслуживание программного обеспечения компьютерных систем. Учебник, -М.: Издательский центр «Академия», 2018. - 256с.
3. Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для спо / Т. М. Зубкова. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 252 с. — ISBN 978-5-8114-9556-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/200462>
4. Макарова, Н.В. Основы программирования: учебник / Макарова Н.В., Нилова Ю.Н., Зеленина С.Б., Лебедева Е.В. — Москва: КноРус, 2021. — 451 с. — ISBN 978-5-406-03394-4. — URL: https://book.ru/book/936582
5. Попов, А.А. Эргономика пользовательских интерфейсов в информационных системах: учебное пособие / Попов А.А. — Москва: КноРус, 2021. — 304 с. — ISBN 978-5-406-07634-7. — URL: <https://book.ru/book/938669>
6. [Федорова Г.Н.](http://academia-moscow.ru/authors/detail/46229/) [Разработка модулей программного обеспечения для компьютерных систем](http://academia-moscow.ru/catalogue/4831/345763/). Учебник, -М.: Издательский центр «Академия», 2019. - 384с.
7. Руденко, О.Г. Системный анализ и проектирование информационных систем: учебное пособие / Руденко О.Г., Кожухова С.В. – Москва: ИНФРА-М, 2021. – 216 с.
8. Грэхем, И. Основы разработки программного обеспечения: учебник / Грэхем И., Спенсер Д., Блэквелл Дж. – Москва: ДМК Пресс, 2021. – 512 с.
9. Котеров, Д.Г. PHP 7 в подлиннике: учебное пособие / Котеров Д.Г. – Москва: ДМК Пресс, 2021. – 1024 с.
10. Борисов, А.И. Базы данных: учебное пособие / Борисов А.И., Булычев Д.Ю., Никитин Е.А. – Москва: Издательство Юрайт, 2021. – 352 с.

# Приложение 1

Техническое задание на разработку программного продукта

«Автоматизация документооборота»

1. Введение

В данном документе изложены технические требования к разработке информационной системы автоматизации управления документами для юридической фирмы. Цель информационной системы – упростить и упорядочить процесс управления документами, сделав его более эффективным и результативным. Система будет разработана с учетом конкретных потребностей и требований юридической фирмы и будет иметь удобный интерфейс для простоты использования.

1. Основания для разработки

Система будет разработана с использованием современных методологий разработки программного обеспечения и языков программирования. Система будет основана на архитектуре клиент-сервер с интерфейсом. Система будет разработана для работы на операционной системе Windows.

1. Назначение

Назначение информационной системы – предоставить платформу для эффективного управления юридическими документами для юридической фирмы. Система будет обеспечивать такие функции, как создание, редактирование, хранение, извлечение и совместное использование документов. Система также позволит управлять рабочими процессами документов и утверждать окончательные документы юристами.

1. Требования к разработке
   1. Функциональные требования к производительности:

* Система должна иметь возможность создавать, изменять и удалять документы по запросу юриста или помощника юриста.
* Система должна предоставлять интерфейс для юристов и помощников юристов для отслеживания и управления статусом документов в рабочем процессе.
* Система должна предоставлять инструменты для утверждения окончательных документов и отправки их клиентам.
* Система должна позволять помощникам юристов загружать и скачивать документы для работы юристов.
* Система должна позволять системному администратору создавать, изменять и удалять учетные записи пользователей.
* Система должна предоставлять инструменты для управления конфигурациями и настройками системы.
  1. Требования к надежности:
* Система должна быть доступна 24 часа в сутки 7 дней в неделю.
* Система должна быть способна одновременно работать с несколькими пользователями без сбоев и простоев.
* Система должна иметь функцию резервного копирования и восстановления для предотвращения потери данных.
* Система должна иметь механизм автоматического обнаружения и восстановления ошибок.
* Система должна быть способна предоставлять точную и надежную информацию юристам и помощникам юристов.

5. Требования к программной документации

Требования к программной документации для информационной системы следующие:

* Документ спецификации системных требований
* Документ технического задания на проектирование
* Техническое руководство. Руководство пользователя
* Документация по плану тестирования и тестовым случаям
* Документация по исходному коду
* Руководство по сопровождению

# Приложение 2

Исходный программный код информационной системы

«Автоматизация документооборота»

Окно авторизации

using System;

using System.Windows.Forms;

namespace Document\_Lawyer

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void buttonAvtor\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "lawyer" && textBox2.Text == "lawyer")

{

Lawyer Lawyer = new Lawyer();

Lawyer.Show();

this.Hide();

}

else

{

if (textBox1.Text == "lawyer1" && textBox2.Text == "lawyer1")

{

LawyerHelp LawyerHelp = new LawyerHelp();

LawyerHelp.Show();

this.Hide();

}

else

{

MessageBox.Show("Неверный логин или пароль!");

}

}

}

}

}

Окно Юриста

using System;

using System.Windows.Forms;

namespace Document\_Lawyer

{

public partial class Lawyer : Form

{

public Lawyer()

{

InitializeComponent();

}

private void Lawyer\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "lawyer\_BDDataSet1.Document". При необходимости она может быть перемещена или удалена.

this.documentTableAdapter.Fill(this.lawyer\_BDDataSet1.Document);

}

private void button1\_Click(object sender, EventArgs e)

{

Form MainForm = Application.OpenForms[0];

MainForm.Show();

this.Close();

}

}

}

Окно помощника юриста

using System;

using System.Windows.Forms;

namespace Document\_Lawyer

{

public partial class LawyerHelp : Form

{

public LawyerHelp()

{

InitializeComponent();

}

private void LawyerHelp\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "lawyer\_BDDataSet2.PaymentReport". При необходимости она может быть перемещена или удалена.

this.paymentReportTableAdapter.Fill(this.lawyer\_BDDataSet2.PaymentReport);

}

private void button1\_Click(object sender, EventArgs e)

{

Form MainForm = Application.OpenForms[0];

MainForm.Show();

this.Close();

}

}

}

# Приложение 3

Руководство пользователя

информационной системы «Автоматизация документооборота»

1. Назначение программы:

Программа автоматизации документооборота предназначена для облегчения управления документами в юридической фирме. Программа позволяет юристам и помощникам юристов создавать, изменять, хранить и управлять юридическими документами, а также предоставляет инструменты для отслеживания документооборота. Программа также генерирует отчеты о работе на основе документов, хранящихся в системе.

1. Условия выполнения программы:

Для работы программы требуется компьютер с доступом в Интернет. Доступ к программе осуществляется через интерфейс и может быть использован на любом устройстве. Для доступа к различным функциям программы требуется учетная запись пользователя с соответствующими правами.

1. Выполнение программы:

Чтобы выполнить программу, выполните следующие действия:

* Откройте программу.
* Введите свой логин и пароль для доступа к программе.
* После входа в систему появится панель управления программой, предоставляющая доступ к различным функциям, таким как создание и изменение документов, управление документооборотом и создание отчетов о работе.
* Чтобы создать или изменить документ, выберите соответствующий тип документа и используйте текстовый редактор программы для ввода или редактирования содержимого документа.
* Чтобы управлять документооборотом, используйте приборную панель программы для отслеживания статуса документа, назначения задач и отправки уведомлений соответствующим сторонам.
* Для создания отчетов о результатах работы выберите соответствующий тип отчета и укажите необходимые фильтры для уточнения данных отчета.
* Чтобы выйти из программы, нажмите кнопку «Выход».

1. Сообщение оператору:

Если у вас возникли какие-либо технические проблемы или вопросы о возможностях или функциональности программы, обратитесь за помощью к системному администратору или в службу технической поддержки.