

Содержание

Список используемых сокращений.....	5
Введение.....	6
1. Общие сведения.....	7
1.1. Обозначение и наименование программы.....	7
1.2. Программное обеспечение, необходимое для функционирования программы.....	7
1.3. Языки программирования, на которых написана программа	8
2. Функциональное назначение	9
3. Описание логической структуры.....	10
3.1. Анализ предметной области	10
3.1.1. Описание предметной области	10
3.1.2. Анализ аналогов разрабатываемого приложения.....	11
3.2. Методология разработки сервиса.....	13
3.2.1. Выбор языков разработки	13
3.2.2. Выбор инструментов для разработки интерфейса приложения	14
3.2.3. Выбор среды для разработки приложения и БД.....	14
3.3. Реализация сервиса	15
3.3.1. Стадии и этапы разработки.....	15
3.3.2. Структурная схема сервиса.....	15
3.3.3. UML-диаграмма структуры БД	16
3.3.4. Разработка серверной части.....	17
3.3.5. UML-диаграммы классов экранов.....	23
3.3.6. Разработка графического интерфейса.....	29
3.3.7. Диаграммы деятельности	37

3.3.8. Выводы к разделу	40
4. Руководство по использованию сервиса	41
5. Вызов и загрузка.....	43
Заключение	44
Список использованных источников	45

Список используемых сокращений

СЭД – Система электронного документооборота

ОС – Операционная система

ПК – Персональный компьютер

JAR – Java Archive

JRE – Java Runtime Environment

CSS – Cascading Style Sheets

ГИП – Графический интерфейс пользователя

API – Application programming interface

SQL - Structured Query Language

ЭДО – Электронный документооборот

МЦ – Медицинский центр

БД – База данных

IDE – Integrated Development Environment

Введение

Система электронного документооборота – неотъемлемая часть современных компаний, в какой сфере деятельности они не осуществляли бы деятельность.

На сегодняшний день все реже, и реже предпочтение отдается бумажным носителям, поскольку такой вид носителя информации крайне неудобен при наличии больших потоков документов.

Уже напечатанный документ нельзя отредактировать или в максимально сжатые сроки распространить среди подчиненных или отправить руководству. Кроме того, простой лист бумаги, например, с жалобой на условия труда очень просто скрыть, убрать в «долгий ящик» или вовсе проигнорировать. Также часто имеют место случаи, когда какой-либо бумажный документ передается не сотруднику, которому положено передать этот документ, а «в обход» установленной внутри организации должностной иерархии.

Таким образом, внедрение СЭД имеет ряд преимуществ:

1. сокращение расходов на печать и ксерокопирование документов;
2. сокращение сроков согласования;
3. обеспечение прозрачности процесса согласования;
4. освобождение рабочего пространства от накопившихся бумаг;
5. быстрая отправка копий документов;
6. возможность одновременного обращения к документу любого количества уполномоченных пользователей.

Итак, целью курсовой работы было избрано создание системы оборота электронных документов с ограничениями по времени их рассмотрения и правилами пересылки, так как нынешние проблемы бумажного документооборота говорят об актуальности данной тематики.

В ходе выполнения курсовой работы планируется освоить компетенции ПК-13 (готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности).

1. Общие сведения

В ходе выполнения курсовой работы был разработан программный продукт, позволяющий осуществлять документооборот внутри предприятия с учетом сроков рассмотрения документов.

Хранение документов организации реализовано на основе 2-ух серверов на основе nginx: web-сервера Debian 10.3 и сервера баз данных Debian 10.3, mysql Ver 15.1 Distrib 10.3.22-MariaD, который выполняет функции файлового хранилища и хранилища данных пользователей системы.

1.1. Обозначение и наименование программы

Для данного программного продукта было избрано название «*DocTracer*». Это название со всей полнотой отражает суть и назначение программы: «*doc*» – сокращение от «*document*» (англ. «документ»), а «*tracer*» можно перевести как «маячок», «трассировщик» или «следящее устройство». Таким образом, название приложения «*DocTracer*» можно интерпретировать как «следящее устройство для документов».

1.2. Программное обеспечение, необходимое для функционирования программы

Для корректной работы приложения необходимо установить сам программный продукт, поставляемый в виде .jar исполняемого архива, также нужно установить JRE, среду исполнения Java-программ.

Для функционирования программы на персональный компьютер сотрудника должна быть установлена ОС Windows (рекомендуемая версия – Windows 10), кроме того, для загрузки программного продукта требуется наличие не менее 10.5 Мб свободной памяти, наличие графического адаптера (видеокарты).

1.3. Языки программирования, на которых написана программа

Программа написана на высокоуровневом, объектно-ориентированном языке программирования Java с использованием платформы JavaFX, разработанной для создания программ с насыщенным графическим интерфейсом.

Для создания более эстетичного и привлекательного интерфейса, а также для повышения понятности ГИП использовался формальный язык описания внешнего вида CSS (каскадные таблицы стилей).

Приложение использует собственное API для взаимодействия с серверами. Для написания API был выбран скриптовый язык общего назначения PHP.

Кроме того, для создания, модификации и управления базой данных, содержащей данные зарегистрированных в системе пользователей, использовался декларативный язык программирования SQL.

3. Описание логической структуры

3.1. Анализ предметной области

3.1.1. Описание предметной области

Очевидно, системы электронного документооборота предназначены для работы с документами. В таком случае, прежде всего, определим, что такое традиционный бумажный документ.

Документ – письменный акт, установленной формы, составленный компетентными организациями, должностными лицами или гражданами для удостоверения или изложения фактов, имеющих юридическое значение, а также для подтверждения прав и обязанностей [1].

Соответственно, электронный документ можно определить как набор данных, записанных в электронном виде, при условии, что существует признанная участниками ЭДО или компетентными органами процедура преобразования этих данных в традиционный документ.

Таким образом, СЭД – автоматизированная система, сопровождающая опирающийся на человеко-читаемые документы процесс управления работой иерархической организации, при этом использоваться могут как электронные документы, так и образы документов (электронные формы бумажных документов, например, электронная скан-копия).

Итак, перейдем непосредственно к рассмотрению предметной области, для которой ведется разработка СЭД. В рамках курсовой работы была рассмотрена организация хозяйственно-эксплуатационной службы медицинского центра (далее МЦ). Ниже, на рисунке 1, приведена организационная схема данной структурной единицы МЦ с учетом информационных потоков.

Из схемы видно, что в организации есть информационные потоки, которые следует устранить путем ограничения круга адресатов в СЭД. Также следует устранить устную форму обращения между руководителями и их подчиненными. Так, если сотруднику требуется, например, заменить

3. Описание логической структуры

3.1. Анализ предметной области

3.1.1. Описание предметной области

Очевидно, системы электронного документооборота предназначены для работы с документами. В таком случае, прежде всего, определим, что такое традиционный бумажный документ.

Документ – письменный акт, установленной формы, составленный компетентными организациями, должностными лицами или гражданами для удостоверения или изложения фактов, имеющих юридическое значение, а также для подтверждения прав и обязанностей [1].

Соответственно, электронный документ можно определить как набор данных, записанных в электронном виде, при условии, что существует признанная участниками ЭДО или компетентными органами процедура преобразования этих данных в традиционный документ.

Таким образом, СЭД – автоматизированная система, сопровождающая опирающийся на человеко-читаемые документы процесс управления работой иерархической организации, при этом использоваться могут как электронные документы, так и образы документов (электронные формы бумажных документов, например, электронная скан-копия).

Итак, перейдем непосредственно к рассмотрению предметной области, для которой ведется разработка СЭД. В рамках курсовой работы была рассмотрена организация хозяйственно-эксплуатационной службы медицинского центра (далее МЦ). Ниже, на рисунке 1, приведена организационная схема данной структурной единицы МЦ с учетом информационных потоков.

Из схемы видно, что в организации есть информационные потоки, которые следует устранить путем ограничения круга адресатов в СЭД. Также следует устранить устную форму обращения между руководителями и их подчиненными. Так, если сотруднику требуется, например, заменить

2. Функциональное назначение

Программный продукт «DocTracer» предназначен для осуществления электронного документооборота внутри какой-либо компании, предприятия или любой другой организации.

Приложению требуется обеспечить следующие возможности:

- регистрация пользователя;
- авторизация уже зарегистрированного пользователя в системе;
- редактирование персональной информации о пользователе;
- удаление сотрудника вместе со всеми его подчиненными из системы (упразднение структурной единицы организации);
- загрузка документа в систему с последующим выбором адресата и самой отправкой загруженного документа назначенному сотруднику;
- удаление отправителем загруженного им же файла;
- скачивание полученного через СЭД файла;
- получение в любой момент актуального списка отправленных и полученных файлов;
- получение актуальной информации о подчиненных, которые просрочили ответ на документ, поступивший ему, в свою очередь, от его подчиненных.

Таким образом, DocTracer призван обеспечить прозрачность оборота документации в организациях.

3. Описание логической структуры

3.1. Анализ предметной области

3.1.1. Описание предметной области

Очевидно, системы электронного документооборота предназначены для работы с документами. В таком случае, прежде всего, определим, что такое традиционный бумажный документ.

Документ – письменный акт, установленной формы, составленный компетентными организациями, должностными лицами или гражданами для удостоверения или изложения фактов, имеющих юридическое значение, а также для подтверждения прав и обязанностей [1].

Соответственно, электронный документ можно определить как набор данных, записанных в электронном виде, при условии, что существует признанная участниками ЭДО или компетентными органами процедура преобразования этих данных в традиционный документ.

Таким образом, СЭД – автоматизированная система, сопровождающая опирающийся на человеко-читаемые документы процесс управления работой иерархической организации, при этом использоваться могут как электронные документы, так и образы документов (электронные формы бумажных документов, например, электронная скан-копия).

Итак, перейдем непосредственно к рассмотрению предметной области, для которой ведется разработка СЭД. В рамках курсовой работы была рассмотрена организация хозяйственно-эксплуатационной службы медицинского центра (далее МЦ). Ниже, на рисунке 1, приведена организационная схема данной структурной единицы МЦ с учетом информационных потоков.

Из схемы видно, что в организации есть информационные потоки, которые следует устранить путем ограничения круга адресатов в СЭД. Также следует устранить устную форму обращения между руководителями и их подчиненными. Так, если сотруднику требуется, например, заменить

оборудование, ему следует оформить заявку согласно установленной форме и направить ее руководителю вместо того, чтобы попросить об этом устно.

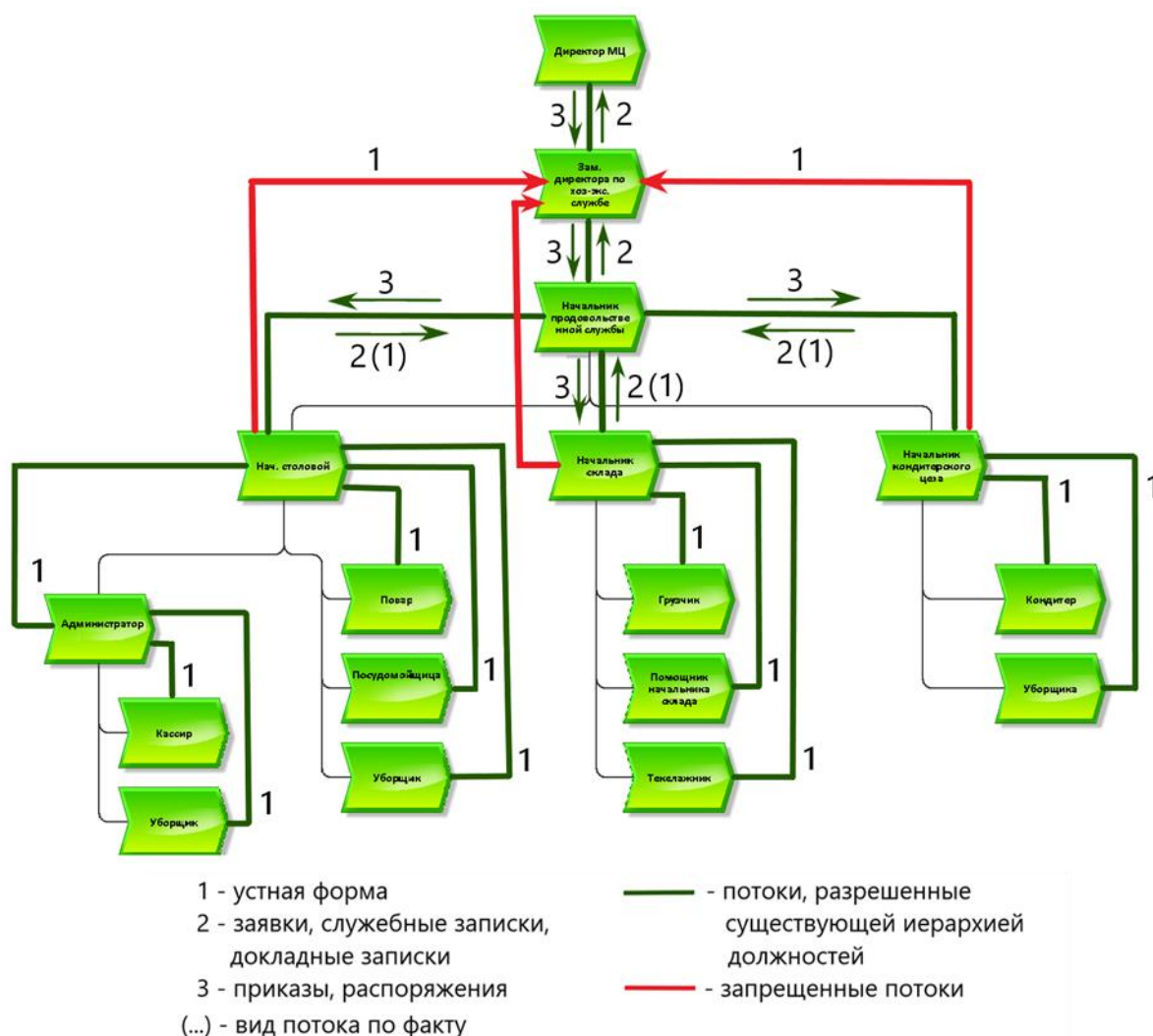


Рисунок 3.1 – Организационная схема хозяйственно-эксплуатационной службы МЦ с учетом информационных потоков

3.1.2. Анализ аналогов разрабатываемого приложения

На сегодняшний день, существует много различных систем электронного документооборота. Далее приведен анализ некоторых из них.

Рассмотрим СЭД «1С: Документооборот 8», «Tessa» и «А2Б». Некоторые из этих систем представлены в виде web-приложения, т.е. все взаимодействие организовано через браузер, и некоторые – как desktop-клиент.

Рассмотрим подробнее систему «1С: Документооборот 8». Система позволяет вести учет входящих и исходящих документов, которые можно

3. Описание логической структуры

3.1. Анализ предметной области

3.1.1. Описание предметной области

Очевидно, системы электронного документооборота предназначены для работы с документами. В таком случае, прежде всего, определим, что такое традиционный бумажный документ.

Документ – письменный акт, установленной формы, составленный компетентными организациями, должностными лицами или гражданами для удостоверения или изложения фактов, имеющих юридическое значение, а также для подтверждения прав и обязанностей [1].

Соответственно, электронный документ можно определить как набор данных, записанных в электронном виде, при условии, что существует признанная участниками ЭДО или компетентными органами процедура преобразования этих данных в традиционный документ.

Таким образом, СЭД – автоматизированная система, сопровождающая опирающийся на человеко-читаемые документы процесс управления работой иерархической организации, при этом использоваться могут как электронные документы, так и образы документов (электронные формы бумажных документов, например, электронная скан-копия).

Итак, перейдем непосредственно к рассмотрению предметной области, для которой ведется разработка СЭД. В рамках курсовой работы была рассмотрена организация хозяйственно-эксплуатационной службы медицинского центра (далее МЦ). Ниже, на рисунке 1, приведена организационная схема данной структурной единицы МЦ с учетом информационных потоков.

Из схемы видно, что в организации есть информационные потоки, которые следует устранить путем ограничения круга адресатов в СЭД. Также следует устранить устную форму обращения между руководителями и их подчиненными. Так, если сотруднику требуется, например, заменить

оборудование, ему следует оформить заявку согласно установленной форме и направить ее руководителю вместо того, чтобы попросить об этом устно.

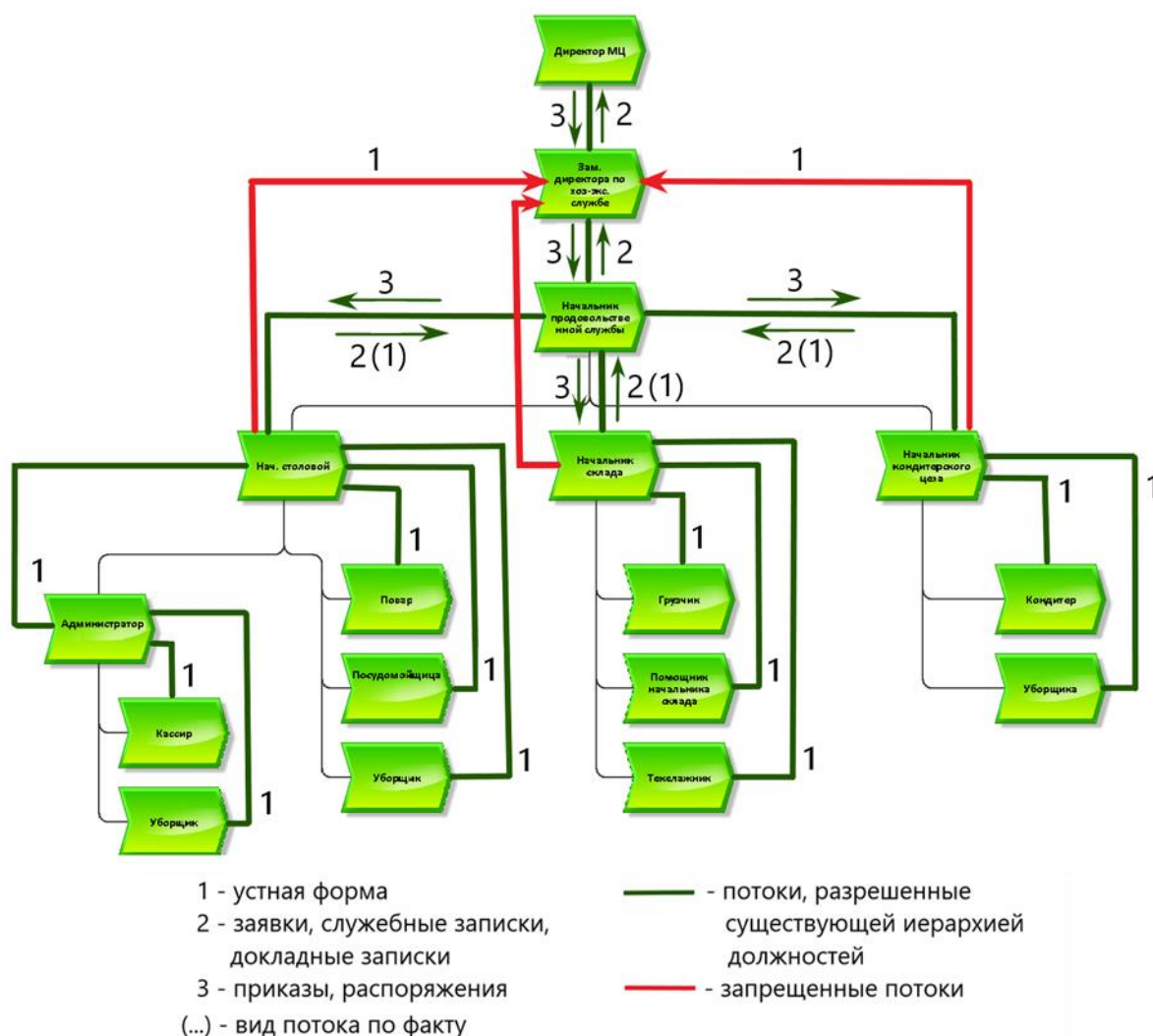


Рисунок 3.1 – Организационная схема хозяйственно-эксплуатационной службы МЦ с учетом информационных потоков

3.1.2. Анализ аналогов разрабатываемого приложения

На сегодняшний день, существует много различных систем электронного документооборота. Далее приведен анализ некоторых из них.

Рассмотрим СЭД «1С: Документооборот 8», «Tessa» и «А2Б». Некоторые из этих систем представлены в виде web-приложения, т.е. все взаимодействие организовано через браузер, и некоторые – как desktop-клиент.

Рассмотрим подробнее систему «1С: Документооборот 8». Система позволяет вести учет входящих и исходящих документов, которые можно

просматривать в виде списков, предусмотрен также контроль исполнения (учтены состояния документа и контроль возврата документов, находящихся у контрагентов). «1С: Документооборот 8» поддерживает три вида электронной подписи (простая, усиленная и квалифицированная), что позволяет удостовериться в целостности документа и авторстве подписавшего [2]. В системе так же реализовано архивное хранение и поиск по документам. Данная СЭД обладает широким функционалом, что является результатом разработки в течении нескольких лет большим штатом разработчиков.

Далее приведен краткий обзор СЭД Tessa. Данная система имеет как desktop-клиент (windows-приложение), так и полнофункциональный web-клиент. Также имеется почтовый клиент (мобильного согласование). Ниже, на рисунке 3.1, представлена схема архитектуры СЭД [3].

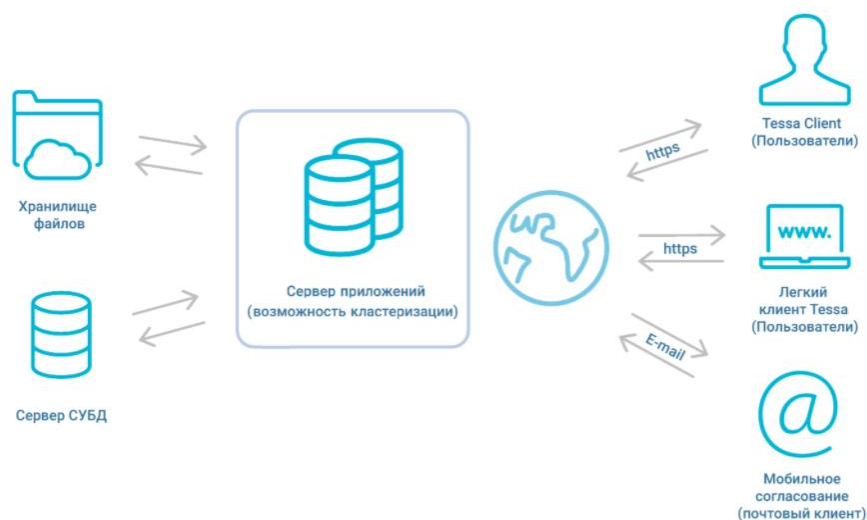


Рисунок 3.2 – Схема трехуровневой архитектуры СЭД Tessa

Итак, Tessa обладает следующим функционалом: регистрация входящей и исходящей корреспонденции, сканирование документов и доведение их до исполнителей в электронном виде, согласование договоров и платежей, обработка заявок/служебных записок, обработка жалоб и претензий, работа с протоколами совещаний/коллегияльных органов и др. Разработчики данной системы выделяют такие преимущества СЭД «Tessa», как современный интерактивный пользовательский интерфейс, мощная подсистема ролевого

доступа к объектам и функциям, передача данных по протоколу https, возможность работы через Интернет, в т.ч. поверх VPN и др.

Теперь перейдем к рассмотрению следующей системы электронного документооборота – «А2Б». Данная система реализует следующие функции: настройки доступа к папкам и документам, поиск по документам, по ключевым словам, уведомления о новых документах и любых изменениях, создание маршрутов работы с документами, настройка карточки документа и т.д. [4]. Система «А2Б» реализована как web-приложение, для работы с которым подойдет любой браузер. Также реализована интеграция с Google-календарем и «Диском».

Подводя итог, можно сказать, что каждая СЭД разграничивает доступ к документации и реализует механизм передачи документа от одного должностного лица к другому, что, в свою очередь, ведет к необходимости организации личных кабинетов для каждого сотрудника с учетом иерархических связей между должностями в организации. Кроме обеспечения этих базовых функций возможна разработка дополнительных опций.

3.2. Методология разработки сервиса

Курсовая работа направлена на разработку клиент-серверного приложения совместно со студентом группы ИКБО-01-18, Студенниковым М.Р. Согласно распределению обязанностей, в данной работе будут освещены front-end разработка (разработка ГИП) и создание API (скриптов, обрабатывающих запросы, поступающие от клиентского приложения).

3.2.1. Выбор языков разработки

Разработка пользовательских интерфейсов при грамотном подходе направлена на создание максимально привлекательного, удобного и интуитивно понятного интерфейса, поэтому для разработки программы был выбран язык Java при совместном использовании с платформой JavaFX. JavaFX

предназначена для создания насыщенных графических интерфейсов пользователя. За счет использования языков fxml и css разработка интерфейсов отличается быстротой и удобством, что, несомненно, является преимуществом платформы JavaFX. Таким образом, благодаря платформе JavaFX основной код, логическая структура страниц приложения и их оформление разделены на отдельные, самостоятельные блоки.

Для разработки серверного API использовался язык PHP. Данный язык сейчас активно применяется для web-разработки. Среди преимуществ данного языка можно выделить наличие строенных средств для автоматического извлечения POST- и GET-параметров запросов, взаимодействие с большим количеством различных СУБД, обработка файлов, загружаемых на сервер и др.

3.2.2. Выбор инструментов для разработки интерфейса приложения

Язык fxml позволяет просто и быстро размещать элементы управления в формах независимо от внутренней логики программы.

CSS – мощный и очень удобный инструмент настройки стилизации. CSS позволяет точно настроить позиционирование и внешний вид элементов (изменение фоновых изображений, шрифтов, создание элементарных анимаций таких как изменение стиля элемента при наведении на него мышью, например).

3.2.3. Выбор среды для разработки приложения и БД

Приложение разрабатывалось в IDE IntelliJ Idea на языке Java. Для создания и управления БД использовалась СУБД MySQL Ver. 10.3.25-MariaDB. Выбор сред разработки более подробно описан в пояснительной записке к курсовой работе студента группы ИКБО-01-18, Студенникова М.Р.

3.3. Реализация сервиса

3.3.1. Стадии и этапы разработки

Стадии и этапы разработки:

1. исследование концепции (01.09.2020 - 14.09.2020);
2. выработка требований к системе (14.09.2020 - 27.09.2020);
3. проектирование БД и файлового хранилища (27.09.2020 - 05.10.2020);
4. написание логики клиентского приложения и создание API (05.10.2020 - 16.11.2020);
5. разработка ГИП (16.11.2020 - 23.11.2020);
6. тестирование и отладка системы (23.11.2020 - 30.11.2020);
7. создание проектной документации (30.11.2020 - 14.12.2020);
8. сдача проекта и защита курсовой работы (14.12.2020 - 18.12.2020).

3.3.2. Структурная схема сервиса

Система состоит из 3 архитектурных частей: клиентское приложение DocTracer, web-сервер, сервер баз данных.

Клиентское приложение предназначено для обеспечения диалога между системой и пользователем. Получив от пользователя команду через ГИП, приложение отправляет запрос к web-серверу.

На данном этапе web-сервер только передает запросы от клиента серверу БД. Web-сервер не был исключен из архитектуры как лишнее звено, т.к. предполагается, что он будет задействован как сервер-ретранслятор при необходимости масштабирования системы.

Сервер баз данных выполняет две функции: реализация хранилища файлов пользователя и хранение персональных данных (таких как ФИО, логин, пароль и должность) в локальной БД. Также на сервере имеется программный интерфейс, отвечающий за обработку поступающих запросов с последующей отправкой ответа клиенту.

Далее, на рисунке 3.3, приведена схема архитектуры системы DocTracer.

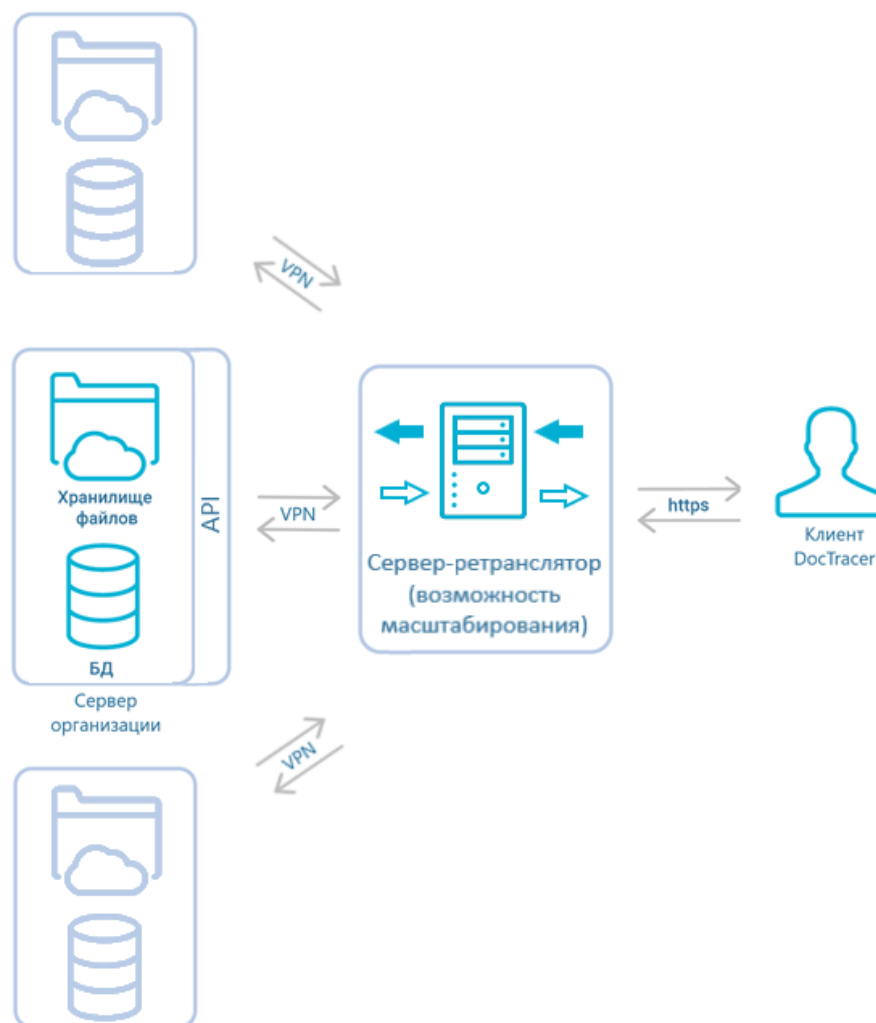


Рисунок 3.3 – Структурная схема сервиса

Такая архитектура была выбрана по причине того, что хранить данные различных организаций на одном сервере неправильно с точки зрения безопасности и соблюдения коммерческой тайны.

3.3.3. UML-диаграмма структуры БД

База данных используется для хранения данных пользователей (ФИО, должность, ключ пользователя, название пользовательской папки на сервере, логин и пароль). Также в БД имеется таблица, хранящая данные о поступлении документа сотруднику и о дате истечения срока рассмотрения.

На рисунке 3.4 представлена схема базы данных. Полное описание структуры, а также пояснения к SQL запросам находятся в работе студента группы ИКБО-01-18, Студенникова М.Р.

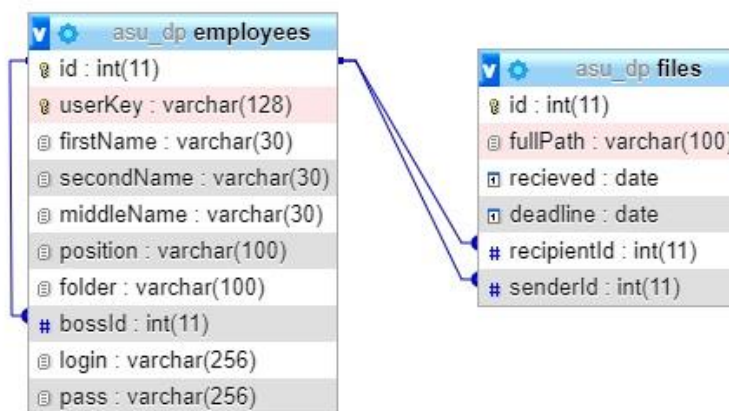


Рисунок 3.4 – Схема базы данных

3.3.4. Разработка серверной части

Как видно из рисунка 3.3, серверная часть состоит из 2 серверов: сервера-ретранслятора и сервера БД. Далее рассмотрим каждый из серверов подробнее.

3.3.4.1. Сервер-ретранслятор

На данный момент сервер-ретранслятор передает запросы от клиента серверу баз данных, и передает ответ клиенту от сервера, т.к. в системе существует пока только одна организация. Чтобы не исключать перспективу масштабирования системы (т.е. возможность подключения к системе новых серверов баз данных других компаний), данный сервер было решено оставить в архитектуре (сервер осуществлял бы распределение поступающих запросов по серверам разных организаций). Как было упомянуто ранее, такое решение было принято из соображений безопасности данных компаний.

3.3.4.2. Сервер баз данных

Сервер баз данных также можно условно разделить на три части: база данных, файловое хранилище и API.

База данных была уже рассмотрена в п. 3.3.3.

Структуру файлового хранилища также разрабатывал Студенников М.Р., студент группы ИКБО-01-18. Однако, для лучшего понимания работы сервиса далее будет дано кратное описание хранилища файлов.

Говоря о файловом хранилище, важно отметить, что каждый пользователь имеет в хранилище личную папку с названием «uploads» + токен. В папке пользователя все файлы разделены еще по двум папкам: отправленные (out) и полученные (in).

3.3.4.3. Скрипты обработки запросов (API)

Прежде чем перейти к рассмотрению принципов работы API осветим некоторые аспекты идентификации пользователей.

Каждый пользователь в системе имеет уникальный ключ, с помощью которого система определяет, кто из пользователей сделал запрос.

Однако при регистрации кроме ключа новому пользователю назначается еще и токен, который затем будет использоваться для повышения безопасности при авторизации, и как идентификатор папки пользователя с префиксом «uploads».

3.3.4.3.1. Скрипт bosslist.php

Скрипт используется для запроса списка сотрудников, из числа которых при регистрации нового сотрудника будет выбираться начальник.

Если в системе еще нет сотрудников, то в ответ клиент получит строку, содержащую «No», в противном случае скрипт отправит в ответ json массив со списком сотрудников.

3.3.4.3.2. Скрипт cryptouserkey.php

Скрипт используется для генерации ключа пользователя по токenu и паролю.

3.3.4.3.3. Скрипт decode.php

Данный скрипт генерирует токен из ключа пользователя. Данная функция используется при авторизации.

3.3.4.3.4. Скрипт delsubordinate.php

Данный скрипт используется для удаления сотрудника вместе с его «поддеревом». При удалении каждого сотрудника удаляется и его индивидуальная папка. Для удаления папки используется скрипт rmDir.php, описанный в п. 3.3.4.3.14.

В случае успешного выполнения запроса к БД, в ответ пользователь получает строку «Success», если же произошла ошибка, высылается текст самой ошибки.

3.3.4.3.5. Скрипт download.php

Данный скрипт позволяет пользователю скачать файл из хранилища. При поступлении запроса на скачивание файла, скрипт отвечает ссылкой, по которой можно загрузить требуемый файл, в случае ошибки скрипт отправит в ответ текст случившейся ошибки.

3.3.4.3.6. Скрипт drop.php

Этот скрипт удаляет файл из системы со стороны отправителя.

Сначала из ключа пользователя, переданного в запросе, генерируется токен, по которому производится поиск папки отправителя. Далее производится поиск ключа пользователя-получателя по таблице files. Затем с помощью скрипта decode.php из ключа генерируется токен, по которому, в свою очередь, производится поиск папки получателя.

Затем выполняется запрос на удаление строки из таблицы files, соответствующей удаляемому файлу. Если запрос успешно выполнен, то файл, идентификатор которого поступил в запросе, удаляется и из папки out отправителя, и из папки in получателя, а в ответ клиент получит строку «Success». Если же при удалении информации о файле из БД произошла ошибка, клиент получит ответ в виде текста ошибки.

3.3.4.3.7. Скрипт edituser.php

Данный скрипт производит замену персональных данных пользователя.

В запросе скрипт получает ключ пользователя, ФИО, новые логин и пароль. Затем генерируются новый ключ в соответствии с новыми логином и

паролем. Далее производится замена всех данных в БД по старому ключу на новые данные. Если попытка замены данных привела к ошибке в БД, система делает «вывод», что ключ пользователя не уникален, а следовательно, логин и пароль уже заняты. В таком случае клиент получит ответ от сервера в виде строки «Duplicate».

3.3.4.3.8. Скрипт filegetlist.php

Данный скрипт предоставляет список полученных файлов, т.е. тех файлов, которые данному сотруднику прислали его коллеги.

По ключу пользователя, полученному в запросе, производится поиск в таблице files тех записей, в которых id соответствует полученному ключу и записан в поле получателя. Также происходит поиск ФИО и должности отправителя файла по id из таблицы files, записанных в поле отправителя.

Если в ходе выполнения запроса к БД ошибок не возникло, скрипт отправляет в ответ клиенту json массив с данными о полученных файлах; в противном случае клиент получит строку с текстом ошибки.

3.3.4.3.9. Скрипт filelist.php

Данный скрипт работает аналогично скрипту filegetlist.php (п. 3.3.4.3.8) с разницей лишь в том, что поиск идентификатора пользователя в таблице files проводится по столбцу отправителей.

3.3.4.3.10. Скрипт login.php

Данный скрипт отвечает за авторизацию на сервисе. При попытке войти в личный кабинет пользователь вводит логин и пароль, которые после нажатия на кнопку «Войти» отправляются в форме запроса на сервер.

Сначала в скрипте генерируется токен на основе логина и пароля, затем генерируется ключ пользователя на основе токена и пароля. Затем выполняется запрос к БД, в котором по полученному ключу пользователя из таблиц выбираются данные о самом сотруднике и о его начальнике (эти данные понадобятся при дальнейшей работе с личным кабинетом).

Если запрос к БД выполнен успешно, клиент получает ответ в виде json массива, в противном случае придет ответ в виде строки «Nobody», сообщающей о том, что нет пользователя с полученными логином и паролем.

3.3.4.3.11. Скрипт notifylist.php

Скрипт используется для получения списка документов, ответ на которые подчиненные данного сотрудника просрочил. Скрипт получает из запроса ключ пользователя, сделавшего запрос. Затем выполняется запрос к БД, результатом которого являются строки таблицы, содержащие данные об отправителе, получателе и самом документе. Если в ходе выполнения запроса не возникло ошибок и результат является непустой таблицей, результат выполнения запроса помещается в json массив, в противном случае клиент получает в ответ на запрос строку «no notify».

3.3.4.3.12. Скрипт registration.php

При регистрации скрипт получает из запроса логин, пароль, ФИО и должность нового сотрудника. Сначала генерируются токен и ключ. Затем, если в поле boss_userKey запроса содержится символ @, скрипт выполняет запрос к БД для регистрации самого первого сотрудника (директора, например) с указанием в поле начальника собственного id. Если же в указанном поле не символ @, а id начальника, выбранного из списка при регистрации, то выполняется другой запрос, результатом которого является новая строка в таблице employees. Если добавить строку не удалось, то считается, что сгенерированный ключ не уникален, что говорит о том, что введенные пользователем логин и пароль уже заняты. В таком случае клиент получит в ответ строку «Duplicate».

3.3.4.3.13. Скрипт rmDir.php

Данный скрипт содержит рекурсивную функцию, которая удаляет директорию по пути, передаваемому в качестве параметра. Рассматривая файловую систему как структуру данных «Дерево», можно сказать, что сначала рекурсивно удаляются файлы-листы, затем вышележащие уровни.

3.3.4.3.14. Скрипт subordnatelist.php

Скрипт возвращает клиенту список подчиненных сотрудников. После получения из запроса ключа пользователя, выполняется запрос к БД, где выбираются строки из таблицы employees, в которых в поле начальника содержится id, соответствующий полученному ключу пользователя. В случае успешного выполнения запроса, в ответ клиент получает json массив со списком подчиненных сотрудников, иначе строку «No» (нет подчиненных).

3.3.4.3.15. Скрипт upload.php

Скрипт реализует отправку файла от одного сотрудника к другому. Суть состоит в том, что новый файл загружается в папку out (отправленные) персональной папки отправителя, и также загружается в папку in (полученные) персональной папки получателя.

Скрипт получает на вход ключ отправителя и ключ получателя. Затем для ключей генерируются токены, которые будут использованы для определения путей, по которым будут сохраняться файл.

Затем выполняется проверка, заведена ли персональная папка на каждого из двух пользователей. Соответственно, если папка не заведена, то выполняется команда создания новой директории.

Далее, необходимо внести данные о файле в БД в таблицу files. Если запрос не был успешно выполнен, клиент получит в ответ текст ошибки. Далее, происходит попытка перемещения загруженного файла в папку отправителя. Если перемещение удалось, то затем файл копируется в папку получателя и в ответ отправляется строка «success», иначе – строка «nosuccess».

3.3.5. UML-диаграммы классов экранов

В приложении предусмотрено несколько окон:

- окно авторизации;
- окно регистрации;
- окно личного кабинета;
- окно изменения персональных данных;
- окно отправки файла;
- окно просмотра полученных документов;
- окно просмотра отправленных документов;
- окно просмотра документов, рассмотрение которых подчиненные просрочили;

Работа окон клиентского приложения контролируется специальными классами – контроллерами. Рассмотрим далее каждый класс-контроллер.

Отметим, что программа также содержит и вспомогательные классы, которые используются только во внутреннем устройстве программы и никак не связаны с интерфейсом приложения. Реализация этих классов (Main, DocFile, User, Log_Pass, HTTPRequest) подробно описана в пояснительной записке к курсовой работе студента группы ИКБО-01-18, Студенникова М.Р.

3.3.5.1. Класс Controller

Класс Controller отвечает за работу страницы авторизации в приложении (стартовая страница). Далее, на рисунке 3.5 приведена его UML-диаграмма.

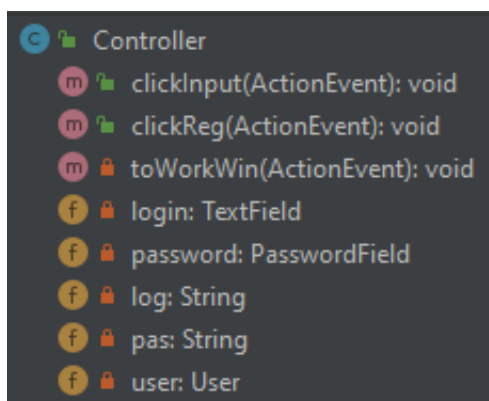


Рисунок 3.5 – UML-диаграмма класса Controller

Методы `clickInput` и `clickReg` являются обработчиками нажатия на кнопки входа и регистрации соответственно.

Метод `toWorkWin` отвечает за переход к окну личного кабинета.

3.3.5.2. Класс `RegistrationPageController`

Класс `RegistrationPageController` отвечает за работу окна регистрации, его UML-диаграмма представлена на рисунке 3.6.

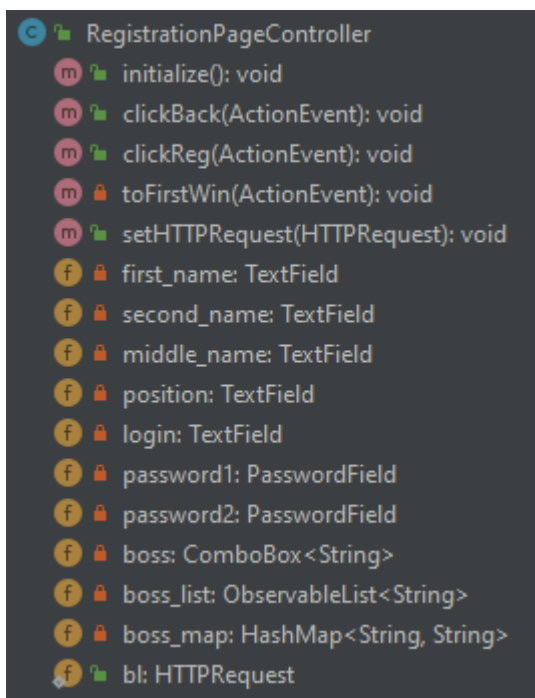


Рисунок 3.6 – UML-диаграмма класса `RegistrationPageController`

Метод `initialize` реализует загрузку окна регистрации, а также выполняется запрос на получение списка сотрудников, которые могут быть начальником регистрируемого сотрудника.

Метод `toFirstWin` отвечает за переход к стартовому окну (окну авторизации).

Метод `clickBack` является обработчиком нажатия кнопки «Назад» и вызывает метод `toFirstWin`.

Метод `clickReg` – обработчик нажатия на кнопку регистрации. В данном методе происходит проверка корректности введенных на форму данных, и отправка запроса на регистрацию.

3.3.5.3. Класс WorkPageController

Данный класс является контроллером для окна личного кабинета пользователя. Структура класса представлена на рисунке 3.7.

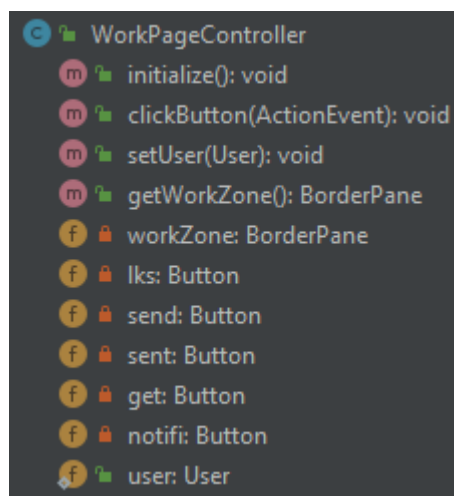


Рисунок 3.7 – UML-диаграмма класса WorkPageController

Окно разделено на две области: левую часть окна занимает панель с кнопками перехода на другие вкладки, а правую часть занимает сама рабочая область, куда по нажатию на кнопки загружаются другие страницы.

Метод initialize выполняет загрузку окна и по умолчанию загружает в основную часть окна страницу со сведениями об авторизованном сотруднике.

Метод clickButton является обработчиком нажатий на кнопки. В зависимости от того, какая кнопка нажата, загружается тот или иной фрагмент окна (страница).

Самая последняя кнопка – кнопка выхода из личного кабинета, выполняет переход к окну авторизации.

3.3.5.4. Класс LksController

Класс LksController отвечает за страницу, на которой размещена информация о текущем авторизованном сотруднике. Данная страница загружается в основную часть окна личного кабинета. Диаграмма класса представлена на рисунке 3.8.

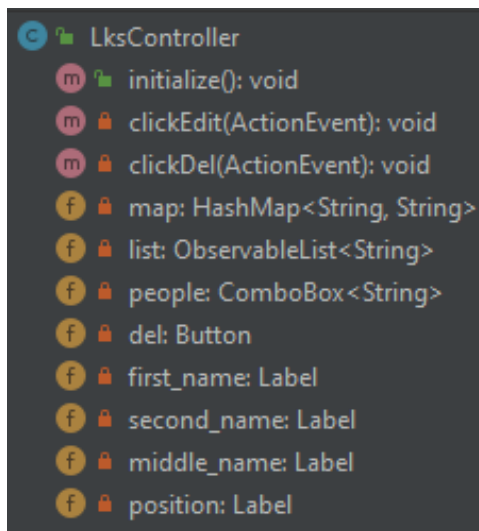


Рисунок 3.8 – UML-диаграмма класса LksController

Метод `initialize` осуществляет загрузку страницы, а также отправляет запрос на получение списка подчиненных сотрудников, который затем размещается в списке в разделе «Мои сотрудники».

Метод `clickEdit` – обработчик нажатия на кнопку изменения данных о пользователе. Метод загружает странице изменения персональных данных.

Метод `clickDel` – обработчик нажатия на кнопку удаления сотрудника. В данном методе происходит отправка запроса удаление сотрудника и всего его поддерева подчиненных.

3.3.5.5. Класс EditController

Класс `EditController` – контроллер окна изменения персональных данных сотрудника (ФИО, логин и пароль). UML-диаграмма класса представлена на рисунке 3.9.

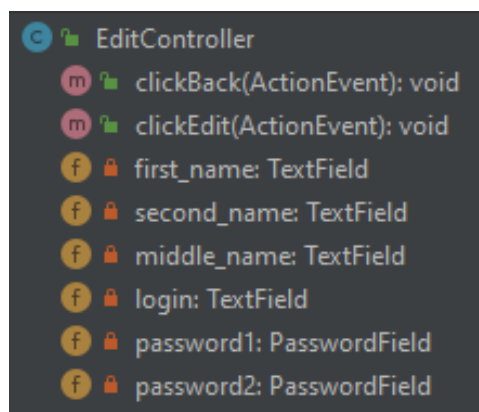


Рисунок 3.9 – UML-диаграмма класса EditController

Метод `clickBack` выполняет переход к окну личного кабинета, с загрузкой страницы с информацией о пользователе.

Метод `clickEdit` отправляет запрос на изменение данных о сотруднике, а затем выполняет переход к окну авторизации (после смены пароля необходимо заново авторизоваться).

3.3.5.6. Класс `SendController`

Данный класс реализует работу страницы отправки документа другому сотруднику. Структура класса представлена на рисунке 3.10.

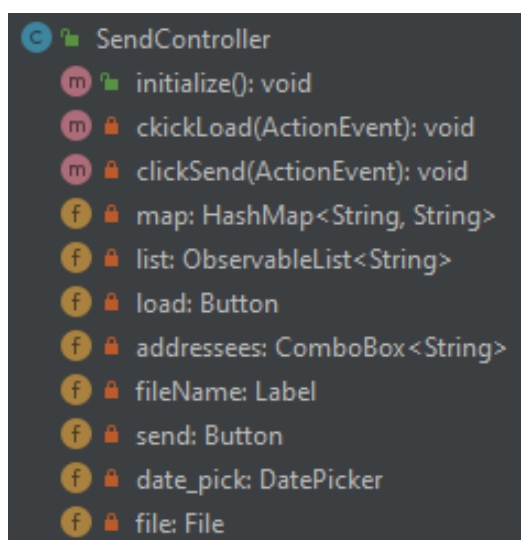


Рисунок 3.10 – UML-диаграмма класса `SendController`

Метод `initialize` выполняет загрузку страницы и формирует список сотрудников, которым можно отправить документ (непосредственные начальник и подчиненные).

Метод `clickLoad` это обработчик нажатия на кнопку выбора файла. Этот метод открывает диалоговое окно проводника для выбора документа.

Метод `clickSend` обрабатывает нажатия на кнопку отправки файла. Метод отправляет запрос на загрузку нового файла. После отправления документ появится в таблице отправленных документов у отправителя, и в таблице полученных документов у получателя.

3.3.5.7. Класс SentController

Этот класса отвечает за работу окна для просмотра полученных документов. Структура класса представлена на рисунке 3.11.

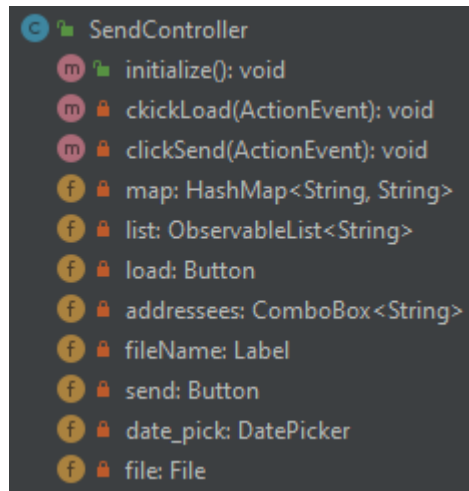


Рисунок 3.11 – UML-диаграмма класса SentController

Initialize – метод выполняющий загрузку списка полученных файлов с сервера и размещающий его в списке на экране.

Метод clickDel – обработчик нажатия на кнопку удаления файла и у отправителя, и у получателя. По нажатию на эту кнопку выполняется отправка запрос на удаление файла по ключу пользователя и id файла.

3.3.5.8. Класс GetController

Данный класс отвечает за работу страницы полученных файлов. Диаграмма данного класс представлена на рисунке 3.12.

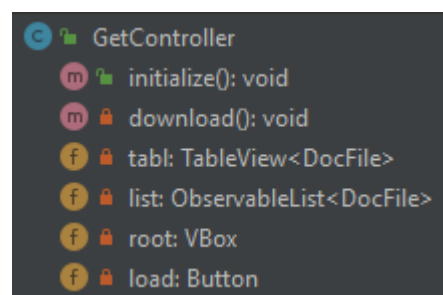


Рисунок 3.12 – UML-диаграмма класса GetController

Метод initialize реализует логику аналогичную логике одноименного метода класса SentController, с разницей лишь в том, что запрашиваются

полученные файлы. Также вместо кнопки удаления в этом окне размещена кнопка скачивания файла.

Метод `download` является обработчиком нажатия на кнопку загрузки. В этом методе отправляется запрос на скачивание файла с последующим сохранением его в папку «Загрузки».

3.3.5.9. Класс `NotifiController`

Этот класс отвечает за работу страницы со списком сотрудников, не рассмотревших полученные документы в срок. Диаграмма класса представлена на рисунке 3.13.

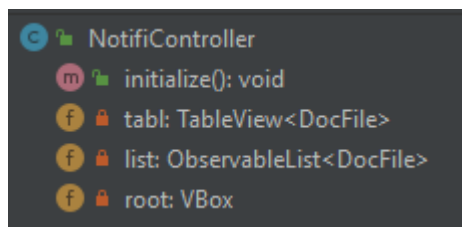


Рисунок 3.13 – UML-диаграмма класса `NotifiController`

Метод `initialize` выполняет загрузку окна (создание таблицы). Затем отправляется запрос на получение списка подчиненных, просрочивших ответ на документ.

3.3.6. Разработка графического интерфейса

ГИП разрабатывался с использованием языка `fxml`, очень похожего на язык разметки `xml`. В сущности, `fxml` также представляет собой язык разметки, разработанный специально для платформы `JavaFX`.

Кроме того, при создании интерфейса приложения для настройки стилизации элементов окон использовался `CSS` – формальный язык описания внешнего вида.

Приложение имеет четыре окна: окна авторизации, регистрации, личного кабинета и изменения персональных данных.

Далее, на рисунке 3.14, представлено окно авторизации.

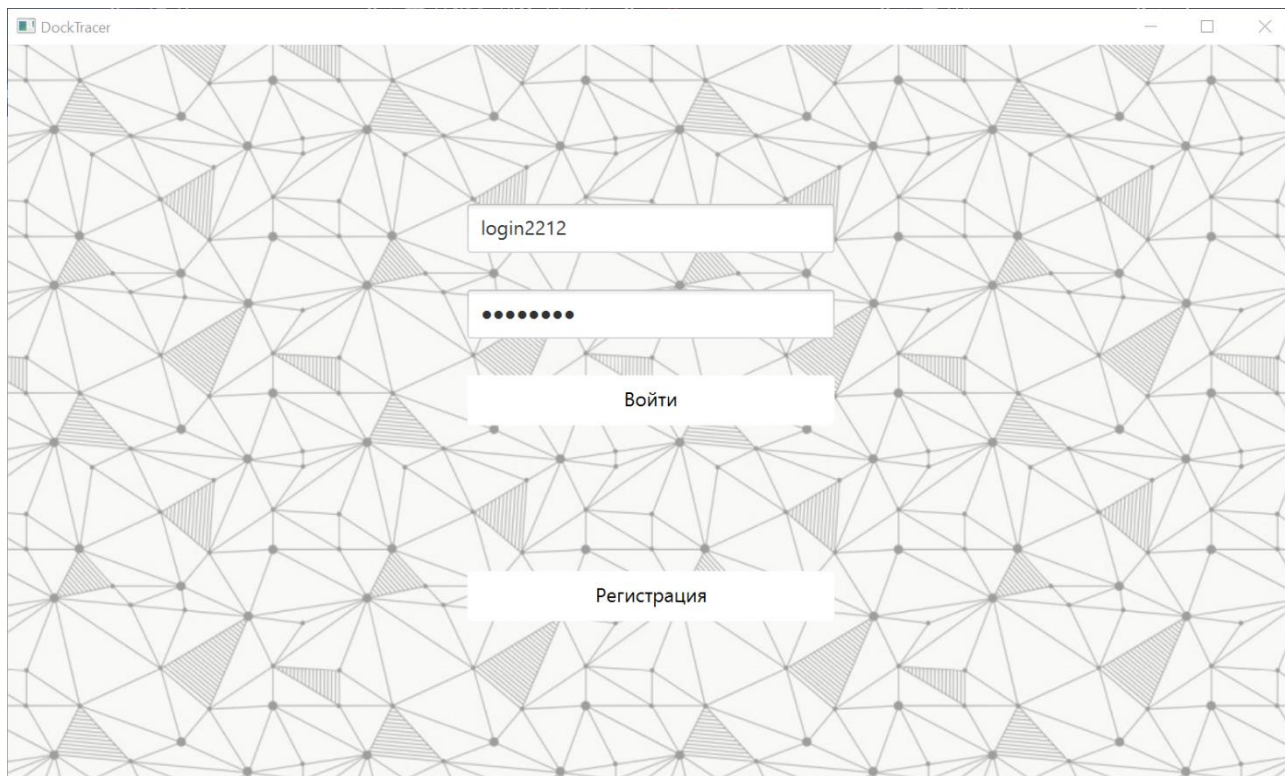


Рисунок 3.14 – Графический интерфейс окна авторизации

Как видно из рисунка, окно авторизации имеет два поля ввода для логина и пароля, а также кнопку перехода к регистрации.

Ниже на рисунке 3.15 изображено окно регистрации нового сотрудника.

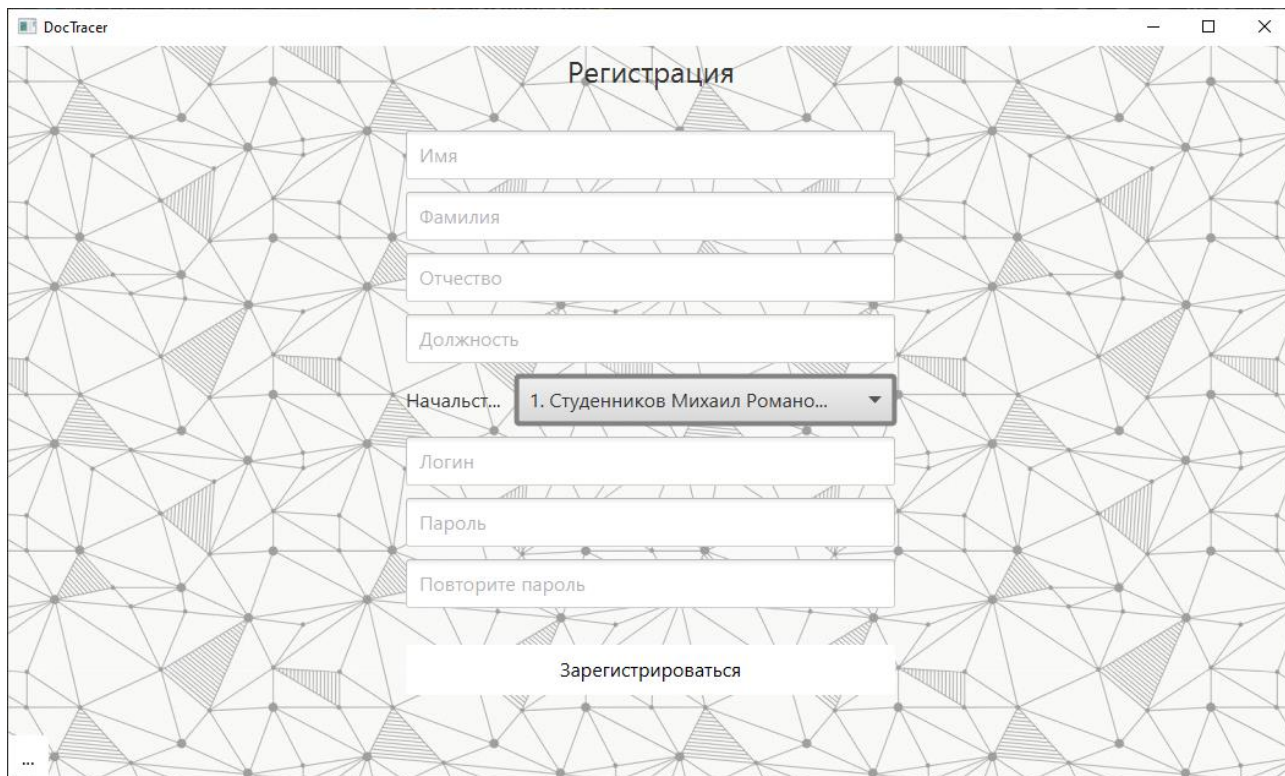


Рисунок 3.15 – Окно регистрации

Окно регистрации имеет поля ввода имени, фамилии, отчества и должности, список, из которого можно выбрать начальника регистрируемого сотрудника, поля для логина и пароля. Также предусмотрено поле для повторного пароля, чтобы избежать опечаток во время набора.

Окно личного кабинета разделено на две части. Слева находится панель с кнопками навигации, в правую часть загружаются рабочие страницы. Личный кабинет состоит из пяти страниц: страница с данными пользователя, отправка документа, список полученных документов, список отправленных документов и список сотрудников, просрочивших ответ на документ. Все страницы личного кабинета представлены на рисунках 3.16 – 3.20.

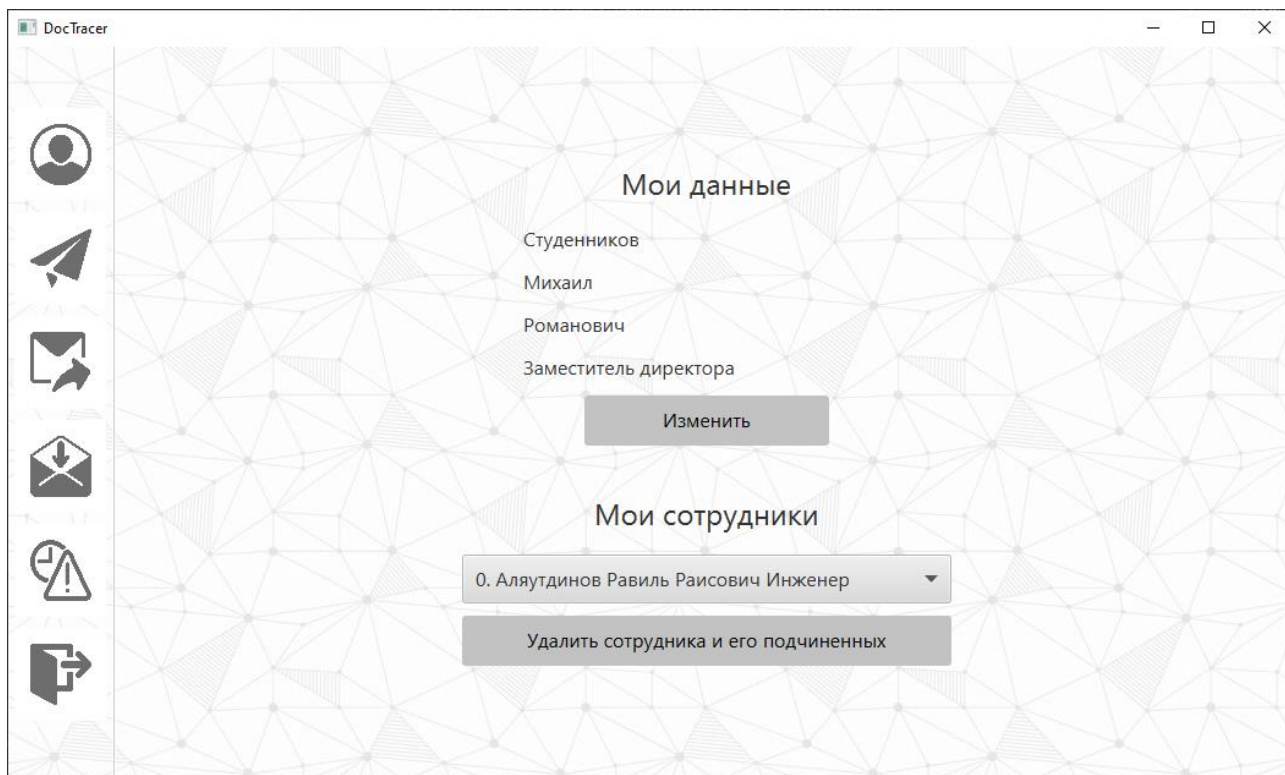


Рисунок 3.16 – Графический интерфейс страницы «Мои данные» личного кабинета

Страница данных пользователя имеет две кнопки: кнопка изменения персональных данных и кнопка удаления сотрудника вместе с его подчиненными (может использоваться, например, в случае реструктуризации компании).

Кнопка «Изменить» переводит пользователя на окно изменения персональных данных (рис. 3.21).

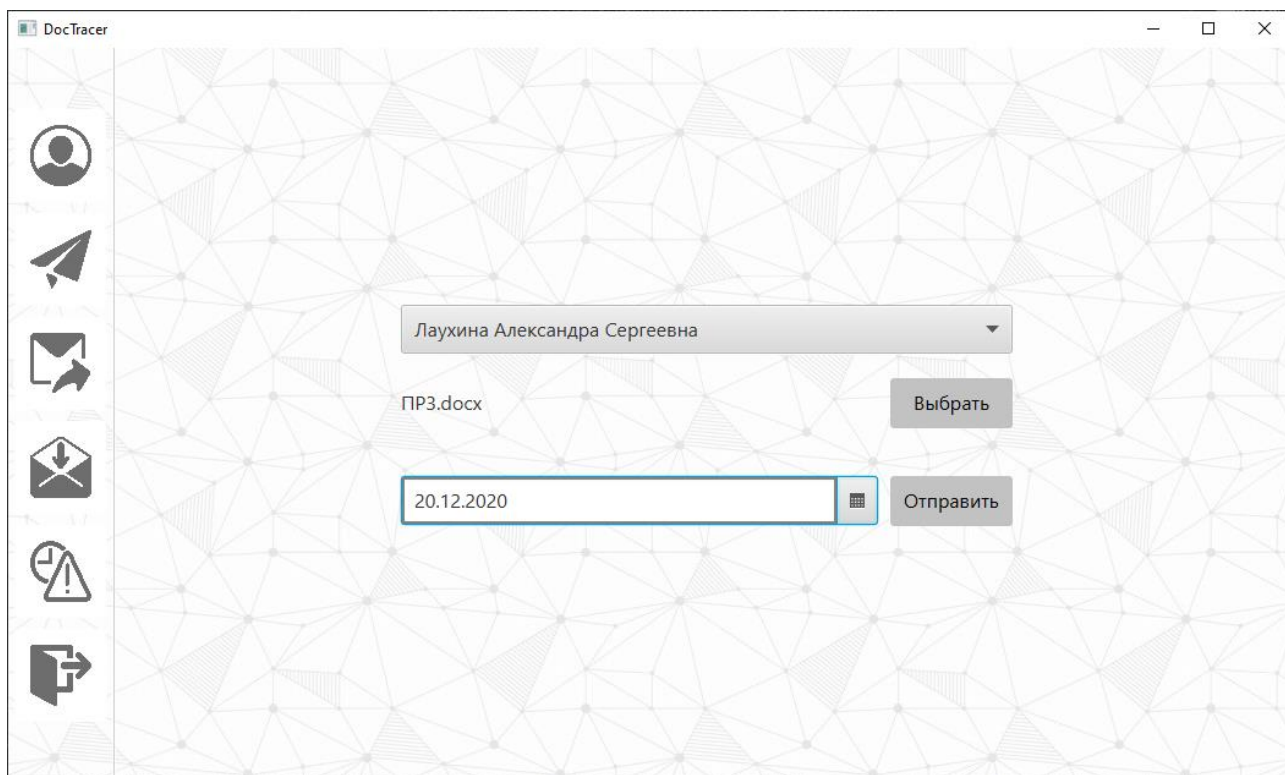


Рисунок 3.17 – Страница отправки файла

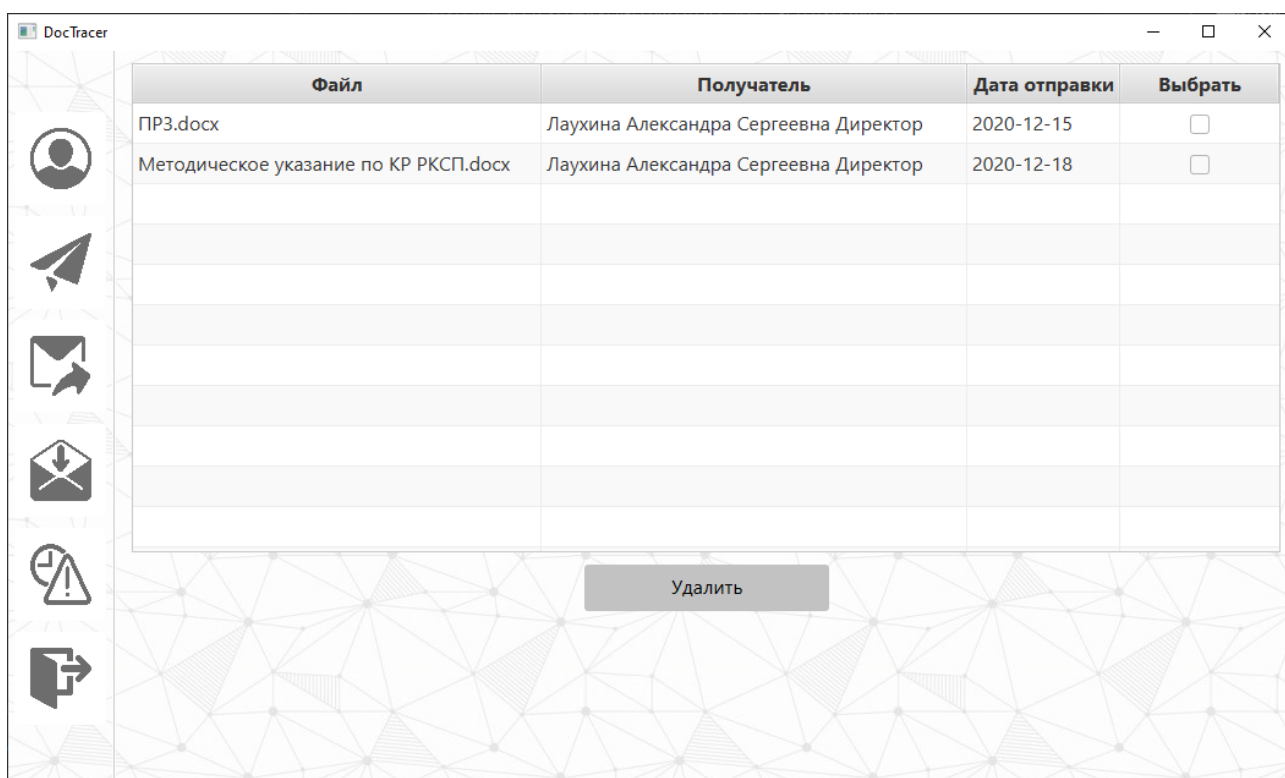


Рисунок 3.18 – Страница просмотра списка отправленных файлов

Страница просмотра списка отправленных файлов также содержит кнопку удаления отправленного файла. При удалении файл удаляется как у отправителя, так и у получателя.

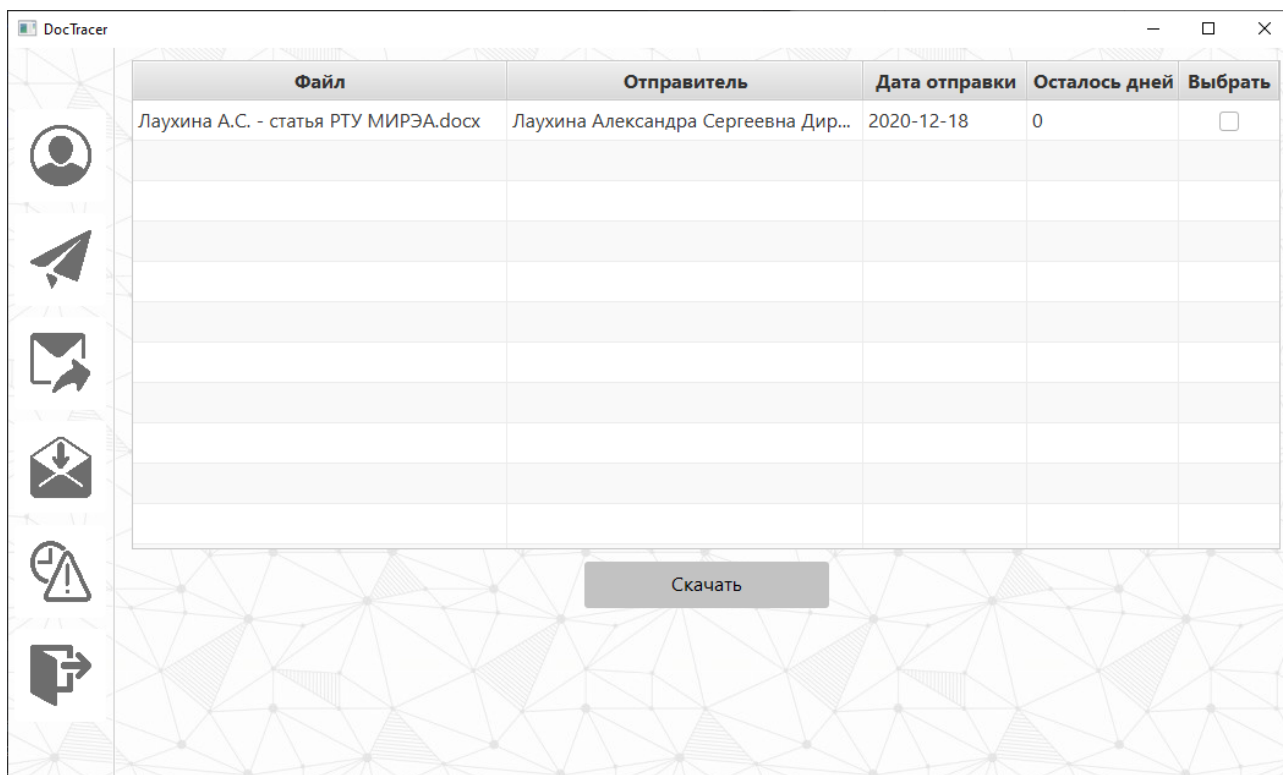


Рисунок 3.19 – Страница просмотра списка полученных файлов

На странице просмотра списка полученных документов есть кнопка, позволяющая скачать файл. Файл по умолчанию загружается в папку «Загрузки», о чем пользователю сообщит небольшое окошко с уведомлением (рис.3.29).

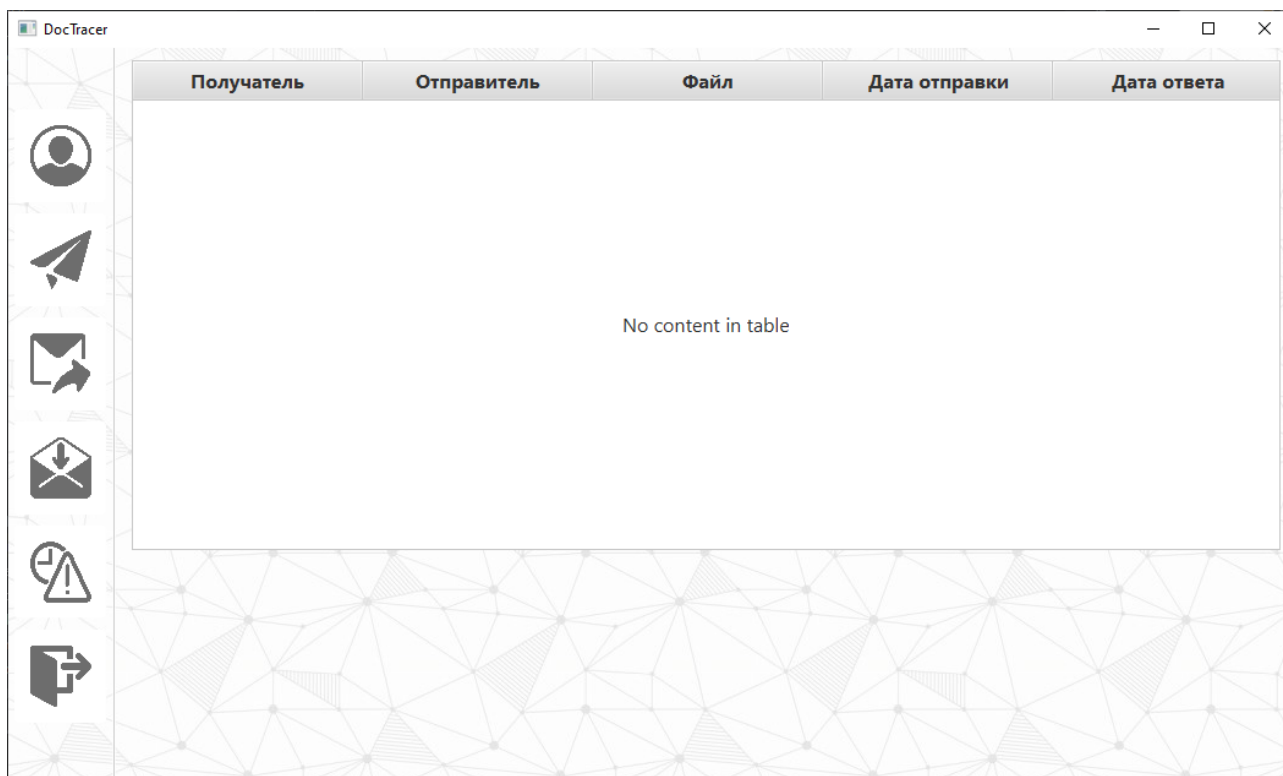


Рисунок 3.20 – Страница просмотра списка файлов, ответ на которые подчиненные данного сотрудника просрочили

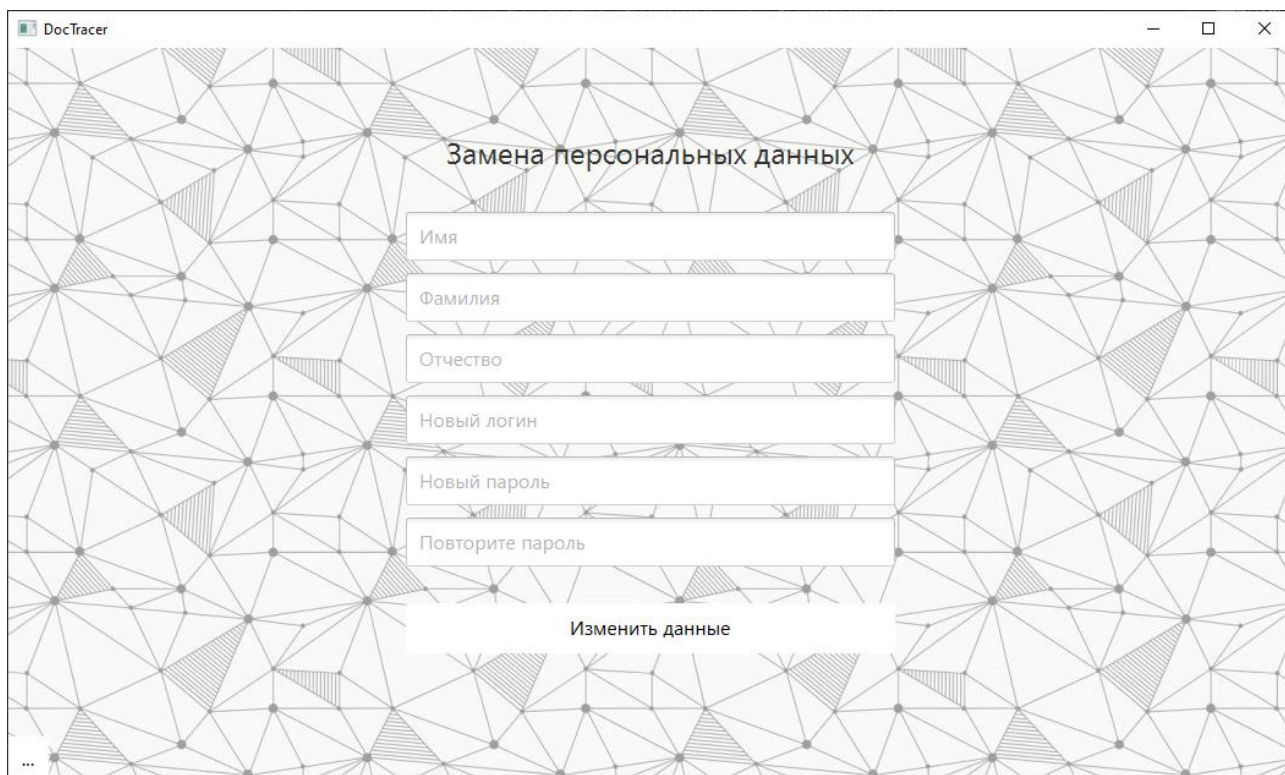


Рисунок 3.21 – Страница изменения данных о сотруднике

На странице изменения персональных данных сотрудника размещены поля ввода для новых имени, фамилии, отчества и данных для авторизации. Изменение должности не предусмотрено, т.к. это может нарушить логическую связь в иерархии должностей организации.

В системе предусмотрены уведомления для информирования пользователя, например, об отсутствии соединения с сервером. Далее, на рисунках 3.22 – 3.30, представлены все окна уведомлений, предусмотренные при разработке системы.

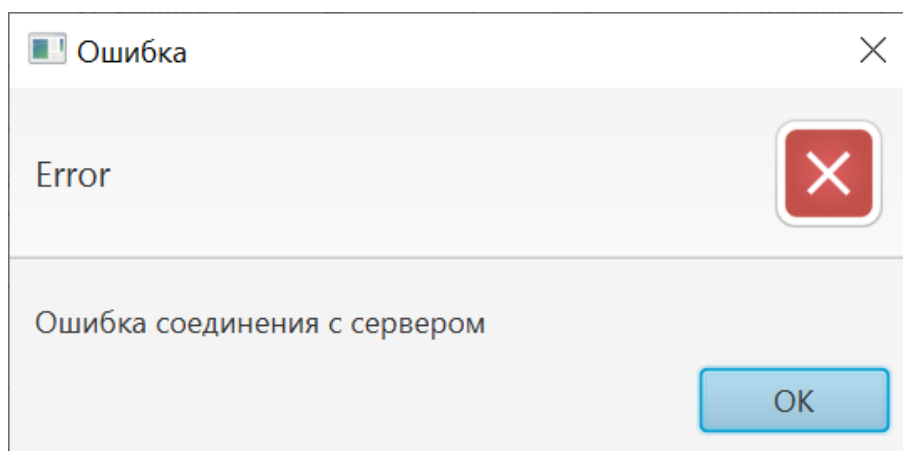


Рисунок 3.22 – Уведомление об отсутствии соединения с сервером

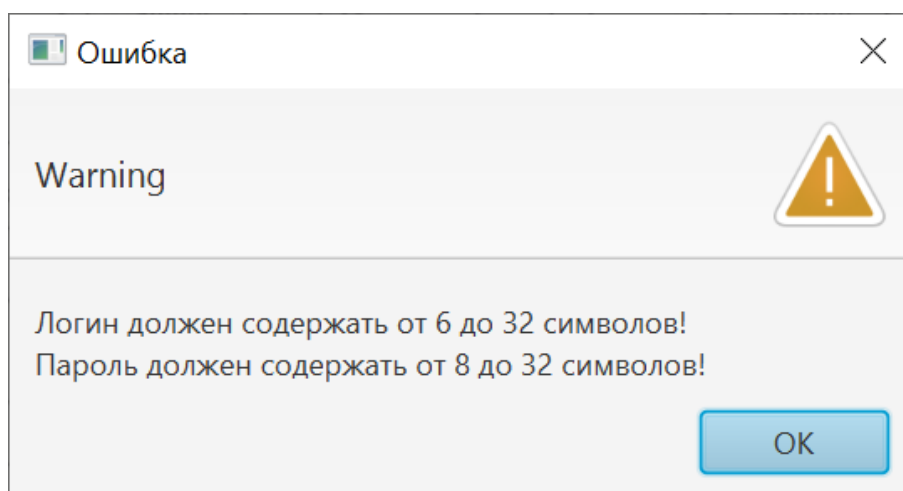


Рисунок 3.23 – Уведомление о несоответствии пароля и логина требованиям системы

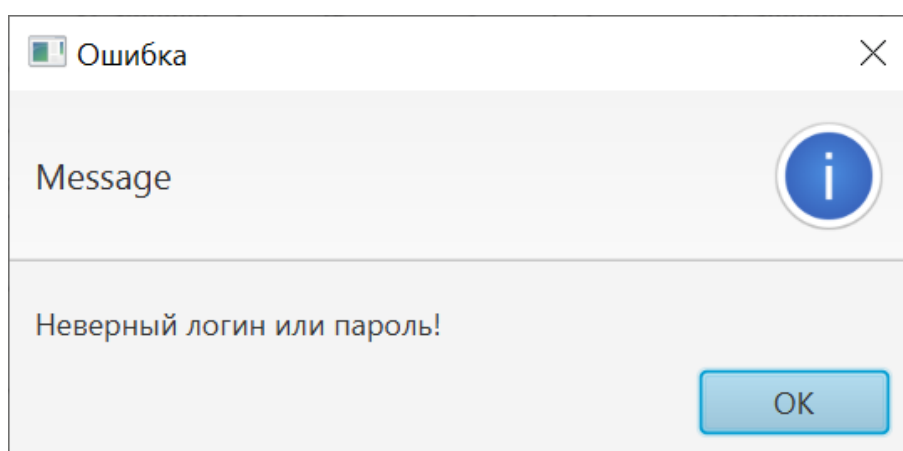


Рисунок 3.24 – Уведомление о том, что введены неверные логин и пароль

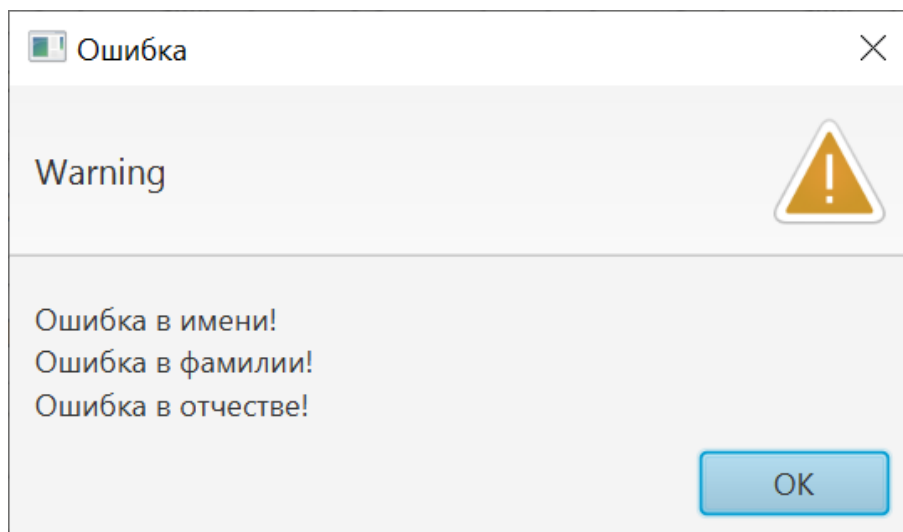


Рисунок 3.25 – Уведомление об ошибках в указанных полях (обычно возникает, если указанное поле ввода не заполнено)

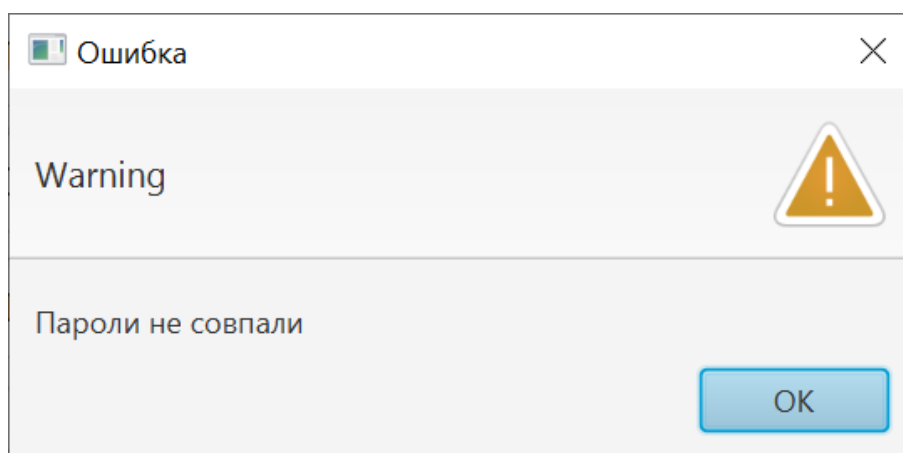


Рисунок 3.26 – Уведомление о несовпадении паролей при регистрации или изменении данных пользователя

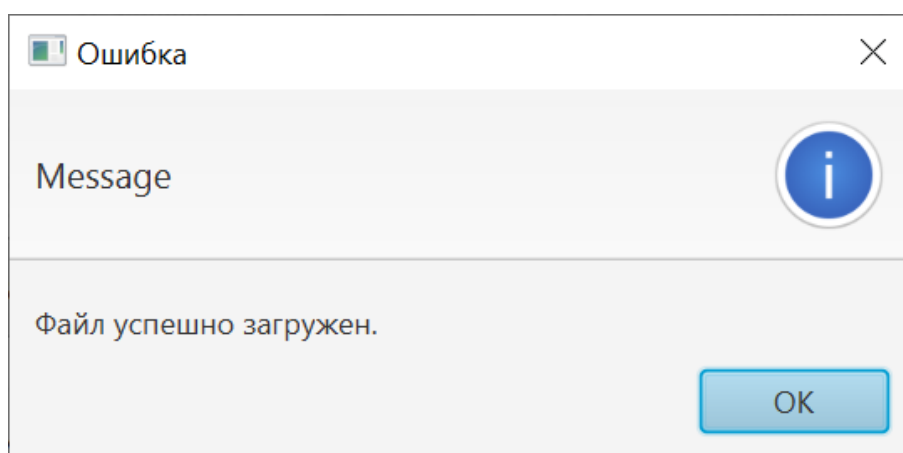


Рисунок 3.27 – Информирование об успешной загрузке файла в систему

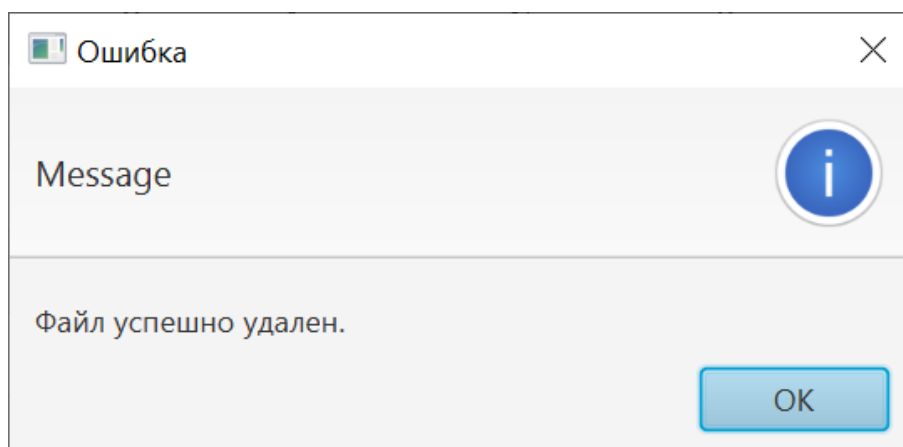


Рисунок 3.28 – Информирование об успешном удалении файла из системы

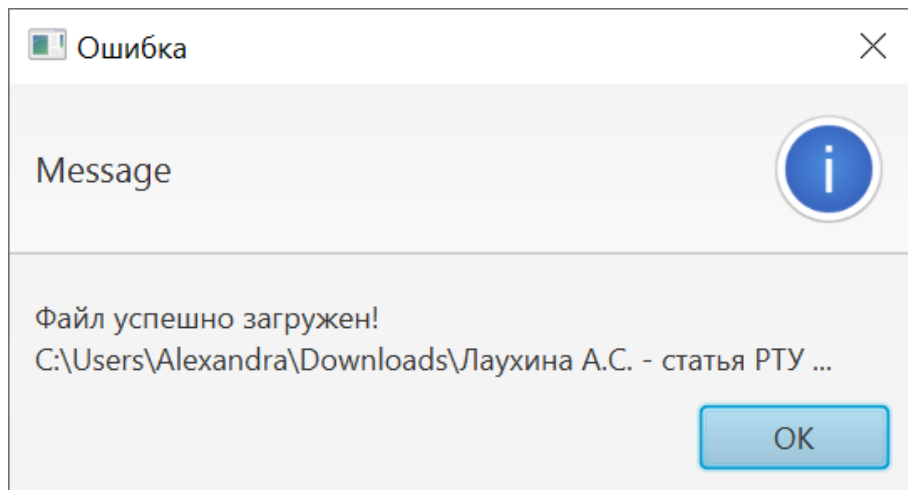


Рисунок 3.29 – Уведомление об успешной загрузке файла на рабочий компьютер сотрудника

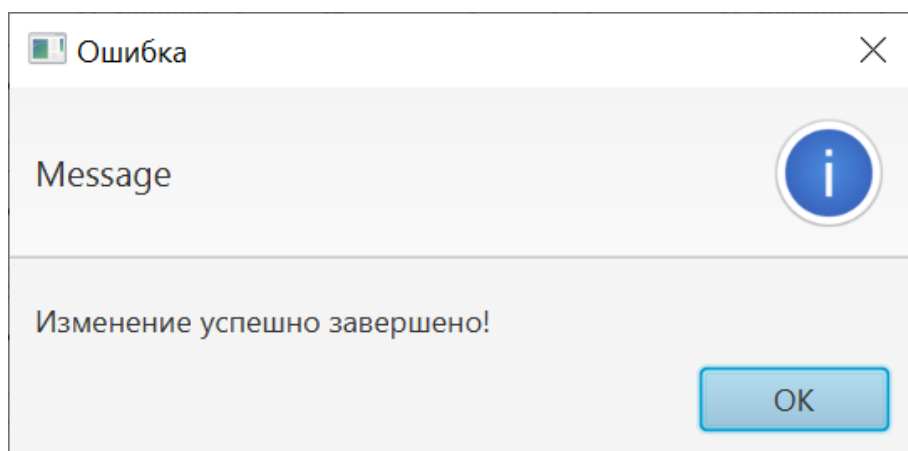


Рисунок 3.30 – Уведомление об успешном изменении данных пользователя

3.3.7. Диаграммы деятельности

В этом разделе будут рассмотрены диаграммы деятельности в нотации BPMN 2.0 для каждого из окон клиентского приложения.

Далее, на рисунках 3.31 – 3.33 приведены диаграммы деятельности основных окон приложения (авторизация, регистрация, личный кабинет).

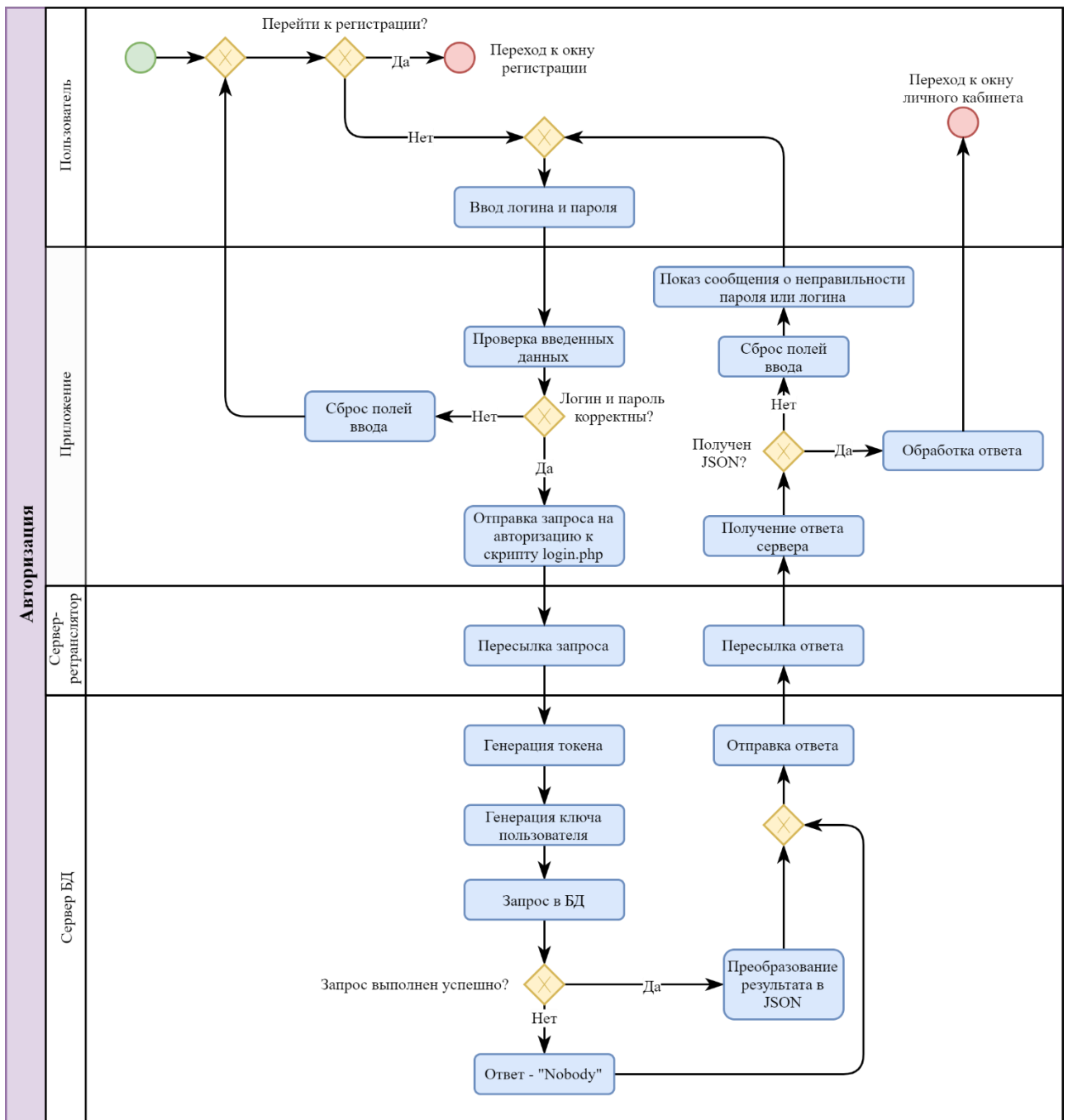


Рисунок 3.31 – Диаграмма деятельности процесса авторизации

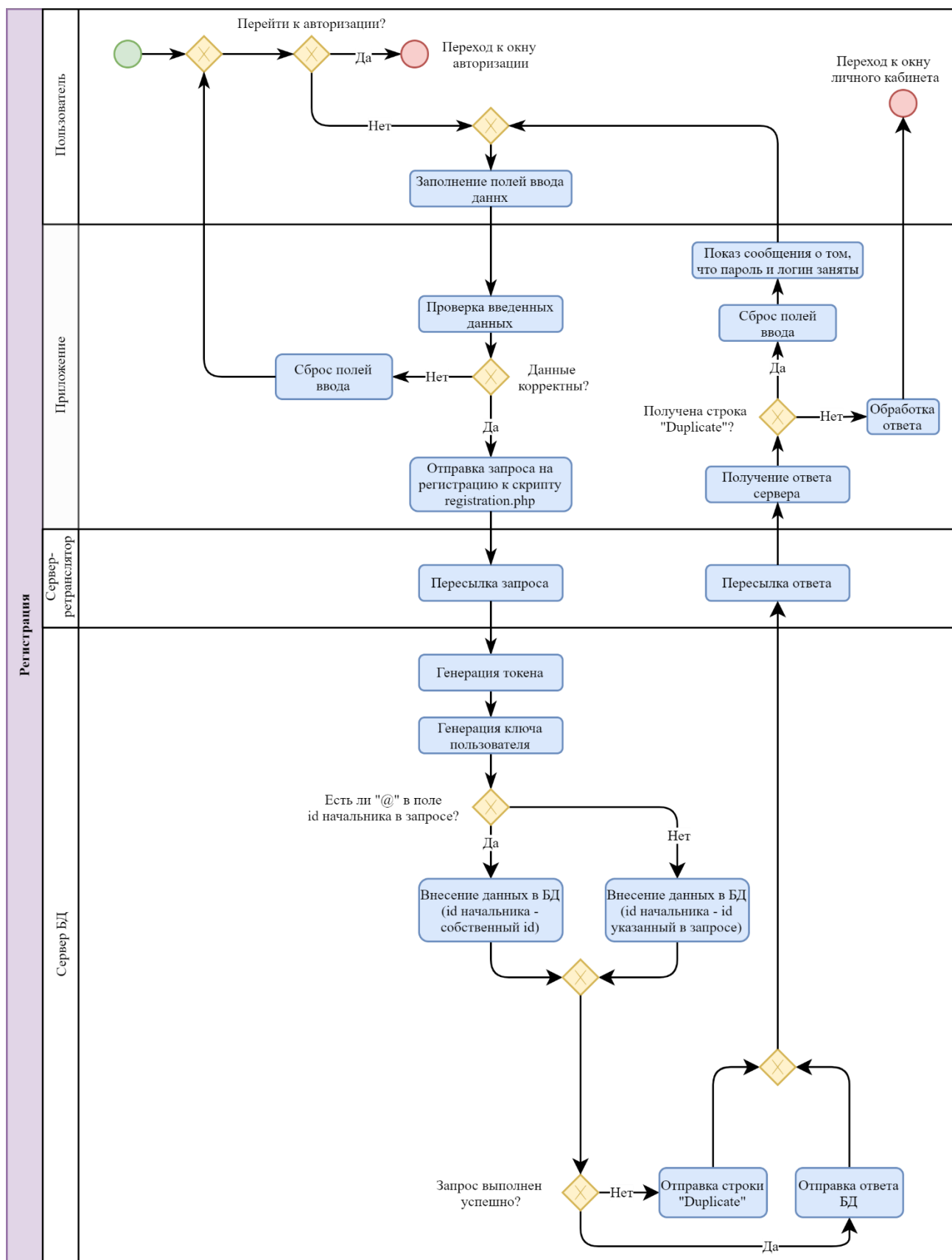


Рисунок 3.32 – Диаграмма деятельности процесса регистрации

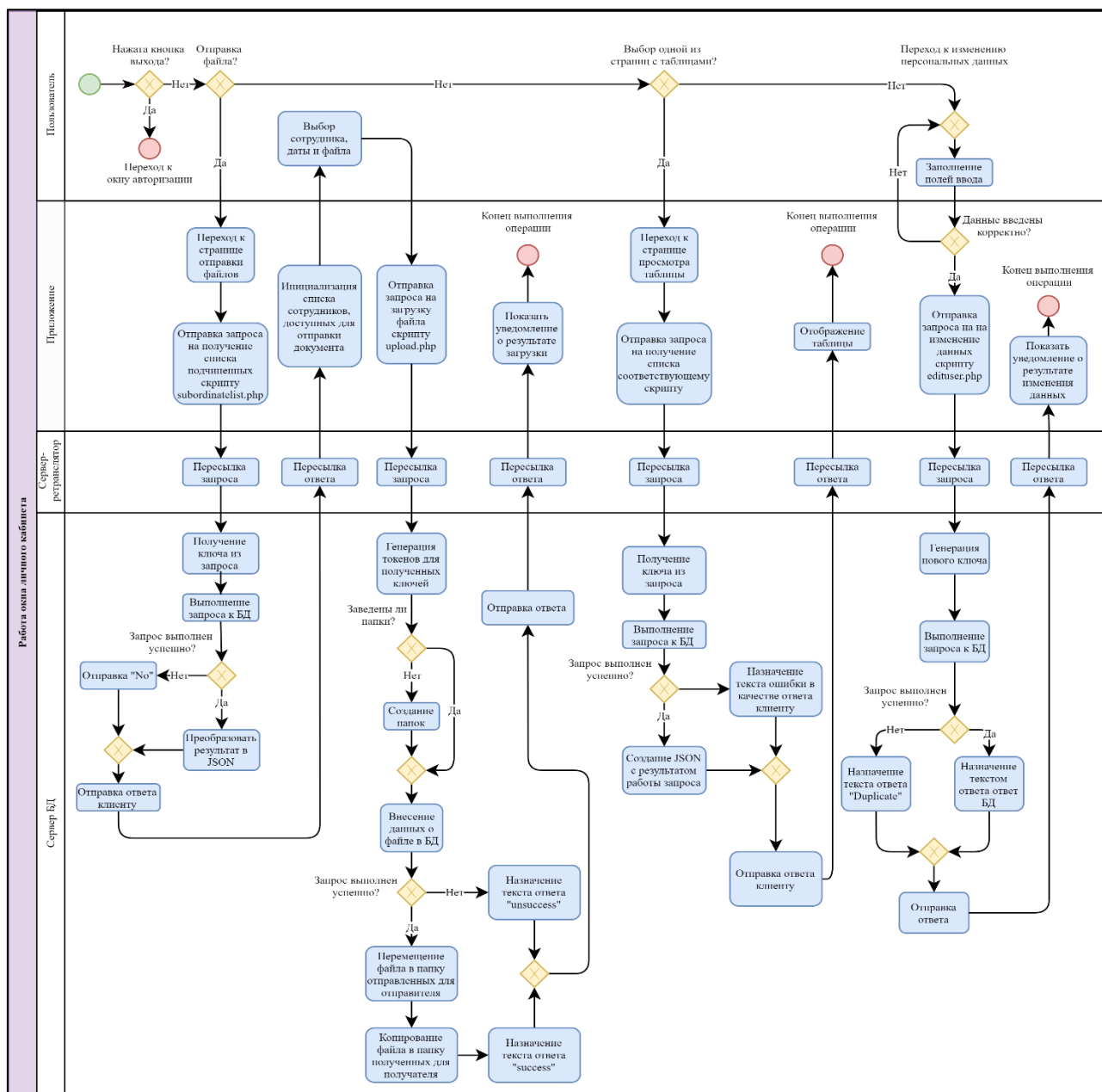


Рисунок 3.33 – Диаграмма деятельности работы окна личного кабинета

3.3.8. Выводы к разделу

В данном разделе был проведен предметный анализ области, в которой велась разработка. Также был представлен краткий обзор аналогов создаваемой системы с выводами о базовом, необходимом функционале СЭД. Затем были описаны такие аспекты разработки, как стадии и этапы разработки, разработка архитектуры сервиса, создание базы данных и файлового хранилища, разработка серверной части, и разработка интерфейса приложения. Также было дано наглядное пояснение к работе каждой из функций в виде диаграмм деятельности.

4. Руководство по использованию сервиса

Рассмотрим особенности использования СЭД «DocTracer».

Внутренне устройство системы реализовано таким образом, что при интеграции системы в предприятие сотрудников нужно регистрировать начиная с главной управляющей должности и заканчивая низшими.

При запуске приложения открывается окно авторизации (рисунок 3.14). Если пользователь еще не зарегистрирован в системе, он может перейти в окно регистрации (рисунок 3.15), нажав на кнопку «Регистрация». В окне регистрации пользователь вводит ФИО, должность, выбирает из выпадающего списка сотрудника, который является его непосредственным начальником и устанавливает логин и пароль. Логин не должен быть короче 6 символов или длиннее 32 символов, а пароль – короче 8 или длиннее 32. Также логин и пароль не должны содержать символы: пробел, «\», «/», кавычки. Имя, фамилия или отчество пользователя не могут содержать те же символы и могут содержать только кириллические символы. Для регистрации в системе пользователь должен нажать кнопку «Зарегистрироваться».

Если пользователь уже зарегистрирован в системе, ему нужно ввести свои логин и пароль в соответствующие поля и нажать кнопку «Войти». Если пользователь правильно ввел свой логин и пароль и нажал кнопку «Войти» на стартовой странице, то он попадает в личный кабинет сотрудника (рис. 3.16), в котором указаны ФИО, должность и также приведен список подчиненных. В выпадающем списке подчиненных можно выбрать сотрудника, которого нужно удалить вместе со всеми его подчиненными. Для этого необходимо нажать соответствующую кнопку. Также в окне личного кабинета находится кнопка «Изменить», при нажатии на которую открывается окно изменения личных данных (рисунок 3.21), в котором пользователь может изменить ФИО, логин и пароль. Чтобы внесенные изменения вступили в силу, пользователю необходимо нажать кнопку «Изменить данные».

На панели инструментов, которая располагается в левой части окна личного кабинета, пользователь может нажать кнопку отправки сообщения, которая откроет окно отправки документа (рисунок 3.17). В данном окне пользователь в выпадающем списке выбирает сотрудника, которому нужно отправить документ, далее по нажатию на кнопку «Выбрать» в открывшемся диалоговом окне выбирает файл, который хочет отправить, затем выставляет дату, до наступления которой получатель может обрабатывать отправляемый документ, и нажимает кнопку «Отправить».

Также с помощью панели инструментов пользователь может перейти на странице просмотра отправленных файлов (рисунок 3.18), в котором по нажатию на кнопку «Удалить» из системы будут удалены выбранные пользователем файлы.

Пользователь также может перейти на страницу просмотра полученных документов (рисунок 3.19), в котором при нажатии на кнопку «Скачать» на компьютер пользователя в папку «Загрузки» будут скачены, выбранные им файлы.

Предпоследняя кнопка на панели инструментов открывает окно просмотра недочетов сотрудников (рисунок 3.20). В этом окне пользователь может увидеть, кто из его подчиненных вовремя не выполнили свои обязанности по ответу на полученные документы.

Последняя кнопка на панели инструментов осуществляет переход на стартовое окно приложения (выход из личного кабинета).

Далее будут описаны некоторые особенности настроенной сети.

Клиентское приложение подключается к промежуточному серверу, который выполняет функции ретранслятора, по протоколу <https>. Промежуточный сервер соединяется с сервером БД, который находится в локальной сети компании, по VPN тоннелю (см. рисунок 3.3). Подключение через VPN сделано из соображений безопасности и по причине того, что поставщик связи не позволяет пробрасывать порты в локальную сеть. Подключение происходит по стандартным сертификатам.

5. Вызов и загрузка

Для эксплуатации ПО его требуется загрузить на устройство в виде .jar архива, предварительно установив на ПК пользователя JDK 11. Затем необходимо настроить сервер БД для подключения к серверу-ретранслятору (установить и настроить web-сервер nginx, а также VPN), создать структуру БД и файлового хранилища, используя СУБД MySQL, настроить API сервера БД. Для обеспечения функционирования API необходимо, чтобы на сервере БД был установлен интерпретатор php.

Заключение

В ходе выполнения данной курсовой работы было создано клиент-серверное приложение для контроля документооборота в организации – «DocTracer». Спроектированное приложение предоставляет пользователю возможность пересылать файлы другим пользователям, скачивать полученные файлы, следить за работой своих подчиненных. Приложение готово и может быть интегрировано, но остается большое поле деятельности для модифицирования приложения и расширения его функционала.

Создание такого приложения позволило закрепить знания и навыки написания клиент-серверных приложений, а также научиться разрабатывать, проектировать и тестировать программы, актуальные в современном мире.

В процессе разработки курсовой работы были приобретены следующие компетенции: ПК-13 (готовностью к использованию методов и инструментальных средств исследования объектов профессиональной деятельности).

Список использованных источников

1. Курченков К.Б. Электронный документооборот. Критерии разработки систем электронного документооборота. – Воронеж: Воронежский институт высоких технологий, 2014. – 5 с.
2. 1С: Документооборот: о продукте (Официальный сайт). [Электронный ресурс] – URL: <https://v8.1c.ru/doc8/>
3. О системе документооборота TESSA (Официальный сайт). [Электронный ресурс] – URL: <https://mytessa.ru/system/about/>
4. Ключевые возможности системы А2Б (Официальный сайт). [Электронный ресурс] – URL: <https://a2b.su/modules/>