Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Дисциплина «Программное обеспечение мобильных систем»

|  |  |
| --- | --- |
|  | «К ЗАЩИТЕ ДОПУСТИТЬ» |
|  | Руководитель курсового проекта  ассистент каф. ПИКС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.С. Крез |
|  | \_\_\_.\_\_\_\_.2023 |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

на тему:

**«ПРОГРАММНОЕ СРЕДСТВО МАГАЗИНА ОРГТЕХНИКИ»**

БГУИР КР 1-39 03 02 011 ПЗ

|  |  |
| --- | --- |
|  | Выполнил студент группы 113802  МИКУЛЬСКИЙ Иван Сергеевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  | Курсовой проект представлен на проверку \_\_\_.\_\_\_\_.2023  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |

Минск 2023

**РЕФЕРАТ**

БГУИР КР 1-39 03 02 011 ПЗ

**Микульский, И.С.** Программное средство магазина оргтехники: пояснительная записка к курсовому проекту / И.С. Микульский. – Минск: БГУИР, 2023. – 49 с.

Пояснительная записка 49 с., 25 рис., 25 источников, 3 приложения.

ПРОГРАММНОЕ СРЕДСТВО, УЧЕТ ТОВАРА НА СКЛАДЕ, УЧЕТ ВАЖНЫХ ДАННЫХ О СОТРУДНИКАХ, условия эксплуатации, проектная часть.

*Цель* *проектирования*: разработка программного средствамагазина оргтехники, предназначенного для упрощения и автоматизации процесса учета товаров на складе.

*Методология проведения работы*: в процессе решения поставленных задач использованы принципы системного подхода, был проведен анализ потребностей сотрудников магазина оргтехники, был реализован необходимый функционал.

*Результаты работы*: выполнен анализ потребностей и требований сотрудников магазина оргтехники, рассмотрено общетехническое обоснование разработки программного средства, осуществлена проверка исключительных ситуаций, уделено внимание вопросам корректной работы программы, разработана графическая часть проекта.

Программное средство обеспечивает учет данных о товарах и продажах.

*Область применения результатов*: могут быть использованы при проектировании аналогичных программных средств для смежных областей с повышенной надежностью работы.

# Содержание

Введение 6

1 Анализ исходных данных на курсовое проектирование 7

1.1 Анализ исходных данных к курсовой работе 7

1.2 Обоснование и описание выбора языка программирования,

средств раз работки, используемых технологий и сторонних

библиотек 7

2 Проектирование и разработка программного средства 12

2.1 Проектирование объектной модели и описание состояний

программного средства 12

2.2 Проектирование и разработка графического интерфейса 15

2.3 Описание и реализация используемых в программном

средстве алгоритмов 20

3 Эксплуатация программного средства 25

3.1 Ввод в эксплуатацию и обоснование минимальных

технических требований к оборудованию 25

3.2 Руководство по эксплуатации программного средства 27

Заключение 33

Список использованных источников 35

Приложение А (обязательное) Отчет о проверке на заимствование

в системе «Антиплагиат» 36

Приложение Б (обязательное) Листинги кода с комментариями 37

Приложение В (обязательное) Ведомость курсовой работы 48

**ВВЕДЕНИЕ**

Как всем известно, любой бизнес начинается именно с деловых бумаг. Множество различных договоров, юридических бумаг, служебных инструкций, бухгалтерских бланков, рекламных проспектов и афиш, технических заданий и технической документации, не говоря уже о визитках, этикетках и т.д. Море бумаг различного назначения. Этим обосновано развитие коммерческих организацией по продаже товаров оргтехники, в которых необходим компьютеризированный учет имеющихся товаров. В связи с этим и было разработано программное средство в рамках данной курсововй работы, направленное на помощь в автоматизации деятельности работников магазинов оргтехники.

Данное программное средство было разработано на языке программирования Java, что позволит беспрепятственно запускать разработанную программу на разных операционных системах: Windows, Linux, Mac OS X и другие.

Целью данной курсовой работы является создание программного средства, которое могло бы помочь автоматизировать деятельность работников магазина оргтехники и свести временные затраты на учет товаров к минимуму. Также были поставлены задачи обеспечить разносторонний функционал, при помощи которого начальство сможет взаимодействовать с данными не только о продажах и товарах, но и о самих сотрудниках.

При выполнении курсового проекта были применены государственные стандарты и нормативные документы, а также СТП БГУИР 01-2017. В пояснительной записке будут представлены три раздела: анализ исходных данных на курсовое проектирование, проектирование и разработка программного средства, эксплуатация программного средства.

Написанный материал пояснительной записки яввляется оригинальным на ХХ процентов. С отчетом о проверке на плагиат можно ознакомиться в приложении А на рисунке А.1.

Перечень графического материала данной курсовой работы будет включать в себя четыре диаграммы: схема алгоритма, UML диаграмма классов, диаграмма состояний и структура графического пользовательского интерфейса.

# Анализ исходных данных на курсовое проектирование

## 1.1 Анализ исходных данных к курсовой работе

В наше время уже тяжело представить жизнь без техники, которая, несомненно, облегчает наш быт. Но даже при таких условиях нигде не обойтись без бумажных документов, интсрукций, предписаний и иных разномастных документов. А это вызывает повышенный спрос на товары оргтехники. И это вполне оправданно, ведь никто не захочет переписывать один и тот же документ много раз от руки. Отсюда и вытекает большое число продаж различных принтеров, ксероксов и сканеров, что ставит вопрос об автоматизации рабочих процессов магазинов оргтехники [1]. Разработанное приложение направлено на решение данной проблемы.

Данная программа предоставляет возможность вести компьютизированный учет всех продаж, товаров и личных данных о сотрудниках. В приложении реализована возможность авторизоваться и регистрироваться. В случае необходимости восстановить логин или пароль пользователь сможет это сделать при помощи специального кода, высланного ему на почту. Также программа позволяет сортировать данные в необходимом порядке по алфавиту (по возрастанию) либо же против, вносить изменения в базу данных непосредственно через само оконное приложение, добавлять либо же удалять необходимые данные из выбранной пользователем таблицы.

Программа имеет интуитивно понятный интерфейс [2], что значительно облегчит освоение данного программного средства потенциальными пользователями.

Также после авторизации на этапе работы с программным средством пользователю будет доступна возможность перейти в настройки своего профиля и отредактировать свои персональные данные.

## 1.2 Обоснование и описание выбора языка программирования, средств разработки, используемых технологий и сторонних библиотек

Java – строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Разработка ведётся сообществом, организованным через Java Community Process; язык и основные реализующие его технологии распространяются по лицензии GPL. Права на торговую марку принадлежат корпорации Oracle [3].

Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, для которой существует реализация виртуальной Java-машины [4]. Дата официального выпуска – 23 мая 1995 года. Java занимает высокие места в рейтингах популярности языков программирования (2-е место в рейтингах IEEE Spectrum (2020) и TIOBE (2021)).

В 2006 году Sun Microsystems решила открыть исходный код Java под лицензией GNU General Public License (GPL). Это привело к созданию проекта OpenJDK, в рамках которого стала разрабатываться открытая реализация Java.

Java является языком программирования, построенным на принципах объектно-ориентированной парадигмы. Это означает, что разработка проекта на Java будет организована вокруг объектов, что упрощает разработку, обеспечивает повторное использование кода и делает его более подходящим для создания крупных и сложных проектов [5].

Java имеет встроенные механизмы безопасности, которые сводят к минимуму риск возникновения уязвимостей и защищают от нежелательных взаимодействий с внешними компонентами. Это особенно важно для проектов, связанных с обработкой и хранением конфиденциальной информации.

Выбор языка программирования Java для разработки оконного приложения можно обосновать множеством факторов. Java – это язык программирования, который изначально разрабатывался с учетом платформонезависи-мости. Это означает, что Java-приложения могут быть запущены на различных операционных системах без изменений в исходном коде. Эта особенность крайне важна при разработке оконных приложений, так как они могут быть запущены на разных версиях Windows, Linux или macOS.

Java обладает обширной библиотекой классов и фреймворков, которые облегчают создание графического пользовательского интерфейса. JavaFX - одна из популярных библиотек для разработки оконных приложений на Java, и она предоставляет богатые возможности для создания интерактивных и красочных пользовательских интерфейсов.

JavaFX – платформа на основе Java для создания приложений с насыщенным графическим интерфейсом. Может использоваться как для создания настольных приложений, запускаемых непосредственно из-под операционных систем, так и для интернет-приложений (RIA), работающих в браузерах, и для приложений на мобильных устройствах. JavaFX призвана заменить использовавшуюся ранее библиотеку Swing. Платформа JavaFX конкурирует с Microsoft Silverlight, Adobe Flash и аналогичными системами [6].

JavaFX предлагает богатый набор инструментов для создания привлекательных и многофункциональных пользовательских интерфейсов. Библиотека включает в себя множество компонентов, таких как кнопки, таблицы, графики и медиа. Она также поддерживает создание и анимацию 2D и 3D графики, что предоставляет разработчикам возможность создавать визуально привлекательные и эффектные элементы интерфейса.

JavaFX использует векторную графику, что позволяет создавать высококачественные и масштабируемые изображения и диаграммы [7]. Это особенно полезно при разработке приложений для разных устройств и экранов.

JavaFX предлагает широкий набор расширяемых и настраиваемых компонентов, что позволяет разработчикам создавать уникальные и индивидуальные пользовательские интерфейсы под требования проекта. Библиотека также легко интегрируется с другими технологиями Java, такими как базы данных, сетевые операции и многопоточность, что расширяет функциональные возможности проекта.

Начиная с версии Java 11 больше не входит в Java SE и не разрабатывается компанией Oracle (как отдельный модуль поддерживается компанией Gluon) [8]. Oracle будет поддерживать JavaFX в составе Java JDK 8 до марта 2025.

Технология JavaFX была впервые продемонстрирована корпорацией [Sun Microsystems](https://ru.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) на конференции JavaOne в мае [2007 года](https://ru.wikipedia.org/wiki/2007_%D0%B3%D0%BE%D0%B4). [4 декабря](https://ru.wikipedia.org/wiki/4_%D0%B4%D0%B5%D0%BA%D0%B0%D0%B1%D1%80%D1%8F) [2008 года](https://ru.wikipedia.org/wiki/2008_%D0%B3%D0%BE%D0%B4) вышла версия 1.0 платформы, содержащая следующие компоненты:

– cредства разработки – [компилятор](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) и среда исполнения JavaFX, [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) JavaFX Script, а также графические, медийные и веб-библиотеки для создания RIA-приложений для настольных компьютеров, веб-сайтов и мобильных устройств.

– [интегрированная среда разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) [NetBeans IDE](https://ru.wikipedia.org/wiki/NetBeans" \o "NetBeans) (версии 6.\*) – средства для кодирования и отладки приложений, написанных на JavaFX Script. В редакторе JavaFX Script есть возможность быстрого добавления объектов JavaFX с уже готовыми геометрическими фигурами, компонентами интерфейса пользователя, средствами преобразования и анимацией.

Также JavaFX поддерживает стилизацию с использованием CSS [9], что делает процесс дизайна пользовательского интерфейса более гибким. Разработчики могут применять стили и темы для создания современного и уникального внешнего вида приложений.

JavaFX предоставляет богатые возможности для создания анимаций и трансформаций элементов интерфейса. Это делает приложения более интерактивными и привлекательными для пользователей.

Данная библиотека позволяет внедрять аудио и видео, обеспечивая возможность создавать мультимедийные приложения. Это особенно важно для разработки игр, мультимедийных приложений и образовательных программ.

JavaFX предоставляет средства для разработки сценариев и обработки многопоточности в интерфейсе, что делает возможным реализацию сложной логики и взаимодействия с пользователем.

Еще одним плюсом является наличие у JavaFX активного сообщества разработчиков и обширной документации, что облегчает получение поддержки и решение возникающих вопросов.

Обобщая все вышесказанное о JavaFX, данная библиотека предоставляет множество инструментов и функций для разработки современных и привлекательных приложений с удобными интерфейсами. Она остается популярным выбором для создания графических приложений на платформе Java. Именно по этой причине была выбрана именно эта библиотека.

Так же существует множество мощных интегрированных сред разработки (IDE) для Java, таких как Eclipse, IntelliJ IDEA и NetBeans. Они предоставляют разработчикам удобные инструменты для создания, отладки и тестирования приложений. При разработке программного средства для данной курсовой работы выбор пал на IntelliJ IDEA, поскольку эта IDE обладает мощными инструментами для интеграции с различными библиотеками и фреймворками Java. Это упрощает создание приложений с использованием сторонних библиотек и компонентов.

IntelliJ IDEA – это мощная и популярная интегрированная среда разработки (IDE) для языка программирования Java. Разработана компанией JetBrains и предоставляет широкий спектр функциональности для разработки Java-приложений [10].

Первая версия IntelliJ IDEA была выпущена в январе 2001 года и была одной из первых доступных Java IDE с интегрированными расширенными возможностями навигации по коду и [рефакторинга кода](https://en.wikipedia.org/wiki/Code_refactoring" \o "Рефакторинг кода).

В 2009 году JetBrains выпустила исходный код IntelliJ IDEA под [лицензией Apache](https://en.wikipedia.org/wiki/Apache_License)[с открытым исходным](https://en.wikipedia.org/wiki/Open-source_software) кодом 2.0. JetBrains также начала распространять ограниченную версию IntelliJ IDEA, состоящую из функций с открытым исходным кодом, под названием Community Edition. Коммерческая версия Ultimate Edition предоставляет дополнительные функции и остается доступной за определенную плату.

В отчете [InfoWorld](https://en.wikipedia.org/wiki/InfoWorld" \o "ИнфоМир) за 2010 год IntelliJ получил высшую оценку центра тестирования из четырех лучших инструментов программирования Java: [Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software)" \o "Затмение (программное обеспечение)) , IntelliJ IDEA, [NetBeans](https://en.wikipedia.org/wiki/NetBeans" \o "NetBeans) и [JDeveloper](https://en.wikipedia.org/wiki/JDeveloper" \o "JРазработчик).

IntelliJ IDEA предоставляет множество встроенных операций рефакторинга, таких как переименование переменных, извлечение методов, оптимизация импортов и многое другое [11]. Эти инструменты облегчают процесс изменения структуры и улучшения читаемости вашего кода. IDE предлагает контекстно-зависимые подсказки и автодополнение кода, что ускоряет написание кода и помогает избежать опечаток и ошибок. IntelliJ IDEA выполняет анализ кода на лету, предупреждая о потенциальных ошибках и предоставляя рекомендации по его улучшению.

IDEA предоставляет интуитивный мастер для создания новых проектов, а также позволяет импортировать существующие проекты. IntelliJ IDEA интегрируется с системами контроля версий, такими как Git, Subversion, и Mercurial, обеспечивая удобное отслеживание изменений в коде [12].

IntelliJ IDEA не только поддерживает язык программирования Java, но также имеет мощные функции для разработки других языков программирования, включая Kotlin, JavaScript, Python и многие другие. Это обеспечивает разработчикам универсальный инструмент для работы с различными языками программирования в рамках одной среды разработки.

Также IDEA включает инструменты для работы с базами данных и SQL-запросами [13]. Это дает возможность создавать и редактировать таблицы, выполнять запросы и многое другое, не покидая среду разработки.

IntelliJ IDEA интегрируется со многими сторонними инструментами и библиотеками. Это позволяет разработчикам использовать популярные инструменты и библиотеки для тестирования, инструменты управления версиями и т.д., чтобы сделать разработку проекта более удобной и эффективной.

Стоит отметить, что IntelliJ IDEA предлагает множество настраиваемых параметров, которые позволяют разработчикам адаптировать среду под свои потребности и предпочтения. Кроме того, с помощью плагинов и расширений разработчики могут расширять возможности IntelliJ IDEA и добавлять дополнительные функции и инструменты по мере необходимости [14].

Отдельным плюсом данной среды разработки является тот факт, что IDEA поддерживает различные серверы приложений, такие как Tomcat, WildFly и GlassFish, облегчая развертывание и отладку ваших приложений [15].

Из всего вышеперечисленного можно сделать вывод о том, что Java является очень сильным языком программирования для разработки программного средства с нуля. Наличие ранее описанных мощных и многофункциональных библиотек, а также крайне удобной среды разработки с множеством дополнительных функций, ускоряющих процесс разработки засчет своей производительности, предоставляет огромное количество возможностей для реализации массивного функционала программного средства. Также данное программное средство имеет огромный плюс в виде своей адаптивности под разные операционные системы, что позволит не испытывать проблем с его поддержкой в будущем.

# Проектирование и разработка

# программного средства

## 2.1 Проектирование объектной модели и описание состояний программного средства

В современном мире разработка программного обеспечения стала неотъемлемой частью технологического прогресса. Проектирование объектной модели и описание состояний программного средства представляют собой ключевые этапы в создании программ, обеспечивающих не только функциональность, но и легкость сопровождения, расширения и адаптации к изменяющимся требованиям [16].

В данном разделе будет представлена диаграмма состояний програмного средства для организации деятельности магазина оргтехники, которая отображает все возможные состояния системы и переходы между ними. Описание состояний программного средства представляет собой анализ того, как программа может находиться в различных состояниях в зависимости от внешних воздействий и внутренних процессов. Этот аспект позволяет разработчикам предусмотреть различные сценарии использования, обрабатывать ошибки и обеспечивать стабильную работу приложения.

В этом разделе также будет рассмотрен фрагмент диаграммы классов, а также её лаконичное описание.

Также в этом разделе будет представлена структура программного срества для лучшего понимания, как организована логика в разработанном приложении, по какому принципу организован алгоритм хранения файлов и какие именно файлы могут отвечать за тот или иной аспект программы.

Все вышеуказанные моменты будут рассмотрены подробно с помощью иллюстраций.

На рисунке 2.1 будет представлена диаграмма состояний программного средства.

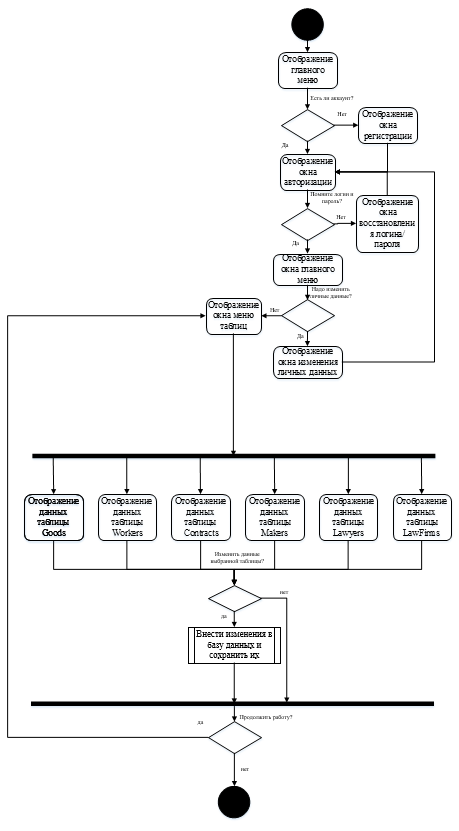


Рисунок 2.1 – Диаграмма состояний программного средства

Данная диаграмма показывает принцип работы разработанного программного средства. Программа при запуске открывает окно с выбором действия: авторизация или регистрация. При выборе варианта «авторизация» пользователю предоставится окно, где необходимо будет ввести логин и пароль от желаемой учетной записи. В случае выбора варианта «регистрация» от пользователя потребуется ввести адрес электронной почты, к которой он хочет привязать свой аккаунт, придумать логин и пароль, подтвердив последний повторным вводом. После входа в программу пользователю будет предоставлена возможность просматривать и редактировать данные в таблицах, а также в случае необходимости пользователь сможет изменить свои персональные данные.

Рассмотрим более детально состояние программы на этапе авторизации. Фрагмент диаграммы, отображающий этот этап, будет продемонстрирован на рисунке 2.2.

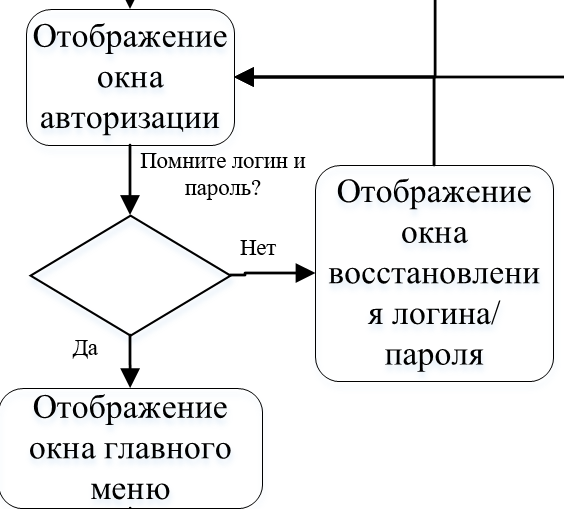


Рисунок 2.2 – Фрагмент диаграмы для состояния программы на этапе авторизации

При входе в аккаунт пользователю потребуется ввести свои логин и пароль. Так же этом месте предусмотрены функции восстановления логина и пароля на случай, если пользователь не может вспомнить свои данные для входа. Для каждого параметра создана своя сцена с восстановлением.

Также для данного программного средства была продумана своя определенная структура проекта, которая будет приведена на рисунке 2.3.

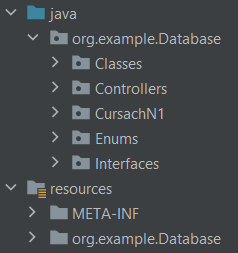


Рисунок 2.3 – Структура проекта

В папке «java» находятся все классы, отвечающие за логику проекта, а также за правильное выполнение всех функций и корректное подключение программного средства к базе данных. Также в проекте предусмотрена отдельная папка «resources» для хранения всех файлов с расширением .fxml, которые являются различными окнами разработанного программного средства.

Изучение и оптимизация процессов проектирования объектной модели и описания состояний программного средства существенно влияют на качество программ, обеспечивая их устойчивость, гибкость и эффективность.

## 2.2 Проектирование и разработка графического интерфейса

Для проектирования и разработки графического интерфейса программного средства для организации деятельности магазина оргтехники было принято решение воспользоваться библиотекой для графики JavaFХ языка программирования Java. Библиотека JavaFX предоставляет различные инструменты и функции для создания пользовательского интерфейса (UI) и оптимизации пользовательского опыта (UX) [18].

UI (User Interface) и UX (User Experience) - это два термина, связанных с проектированием пользовательского интерфейса приложений или веб-сайтов [19].

UI (User Interface) описывает то, как пользователь взаимодействует с интерфейсом приложения или веб-сайта. Это включает в себя дизайн элементов интерфейса, таких как кнопки, поля ввода, меню и т.д., а также расположение и организацию этих элементов. Цель UI-дизайна - сделать интерфейс интуитивно понятным, удобным в использовании и привлекательным для пользователя.

UX (User Experience) описывает общий опыт пользователя при использовании приложения или веб-сайта. Это включает в себя все аспекты взаимодействия пользователя с продуктом - от первичного впечатления до навигации, процесса взаимодействия и выполнения задач. Цель UX-дизайна - создать удовлетворительный и приятный пользовательский опыт, учитывая потребности и ожидания пользователей.

UI и UX тесно связаны и важны для создания успешных и эффективных пользовательских интерфейсов. Хороший дизайн UI и UX может повысить удобство использования, улучшить вовлеченность пользователей и повысить общую стоимость приложений и веб-сайтов.

Все окна данного программного средства будут оформлены в едином стиле для поддержания единообразия всей программы, чтобы у пользователей не возникало проблем со взаимодействием с ними и перемещением между окнами.

Как уже упоминалось ранее, разработанное программное средство будет начинаться с окна с выбором действия: авторизация и регистрация. Вид данного окна будет представлен на рисунке 2.4.

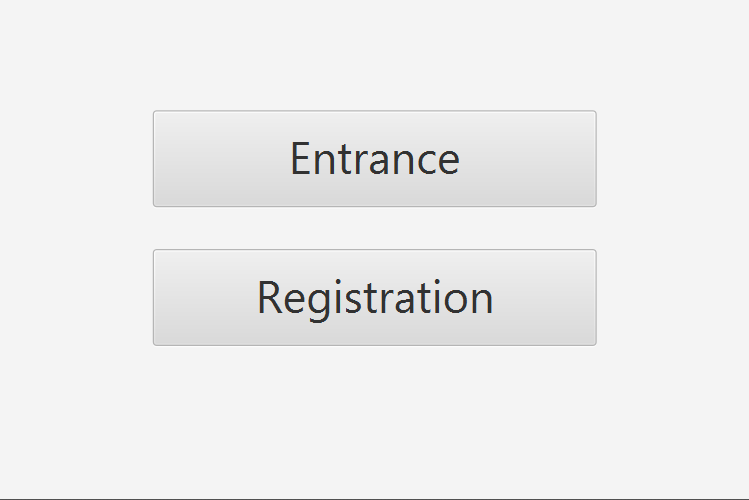


Рисунок 2.4 – Вид окна с выбором авторизация/регистрация

Ниже будет приведен фрагмент кода, отвечающий за окно с выбором авторизация/регистрация.

public class menuEntranceOrRegistrationControl implements Initializable {

@FXML

private Button entranceButton;

@FXML

private Button registrationButton;

@Override

public void initialize(URL url, ResourceBundle resourceBundle) {

(new DatabaseHandler()).truncateOnlineUser();

Tooltip t=new Tooltip("Нажмите");

entranceButton.setTooltip(t);

registrationButton.setTooltip(t);

entranceButton.setOnAction(actionEvent -> Scenes.ENTRANCE.setScene((Stage) entranceButton.getScene().getWindow()));

registrationButton.setOnAction(actionEvent -> Scenes.REGISTRATION.setScene((Stage) registrationButton.getScene().getWindow()));

dbHandler.truncateOnlineUser();

}

}

Рассмотрим детальнее окна авторизации и регистрации.

Все поля для заполнения должны быть подписаны, чтобы пользователь знал, какую информацию необходимо туда внести. Также каждое окно должно иметь название, которое будет отображаться вверху.

На рисунке 2.5 представлено окно авторизации пользователя.

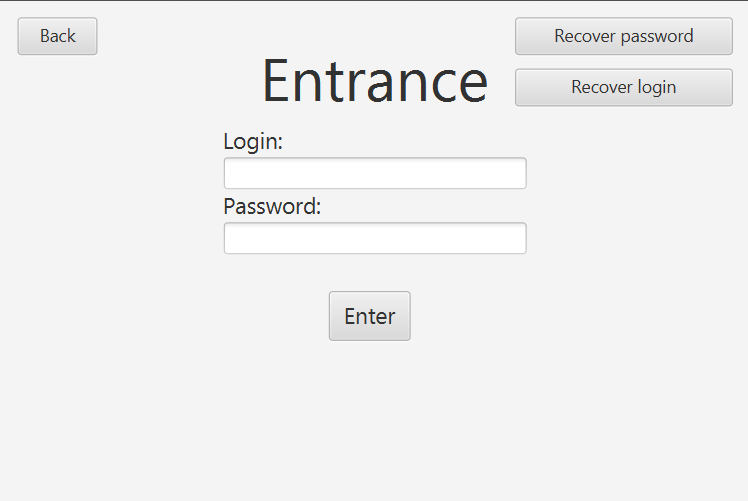


Рисунок 2.5 – Вид окна авторизации

На рисунке 2.6 представлено окно регистрации пользователя.

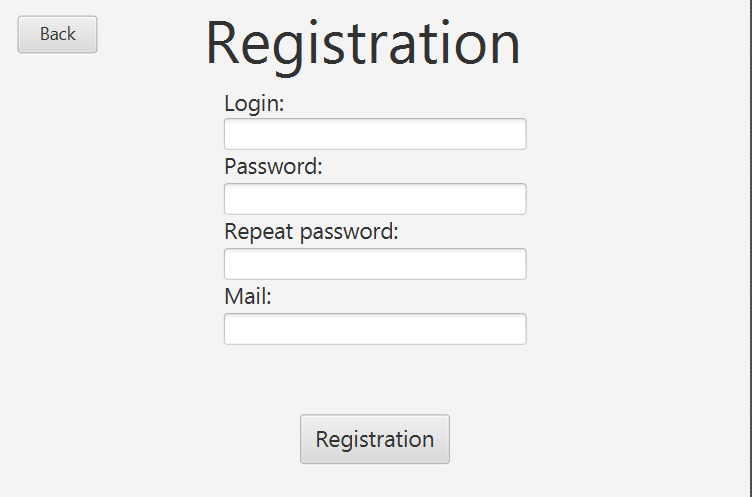


Рисунок 2.6 – Вид окна регистрации

После авторизации пользователя в программе запускается окно, содержащее главное меню, а также кнопку для перехода к настройкам профиля. Страница главного меню пользователя изображена на рисунке 2.7.

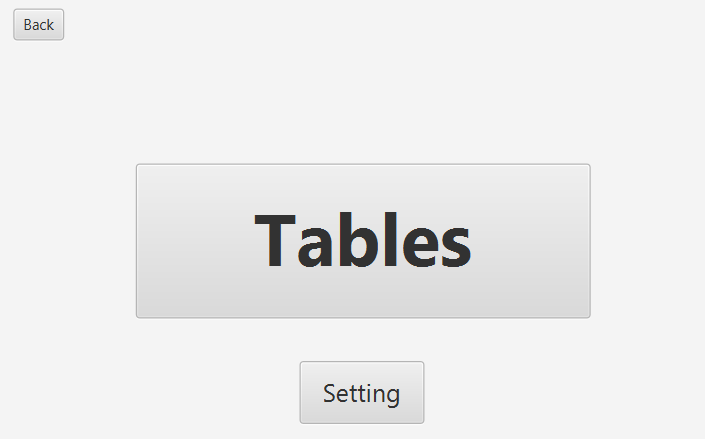


Рисунок 2.7 – Вид окна с главным меню

На рисунке 2.8 изображено окно профиля пользователя, где пользователь может редактировать его личную информацию.

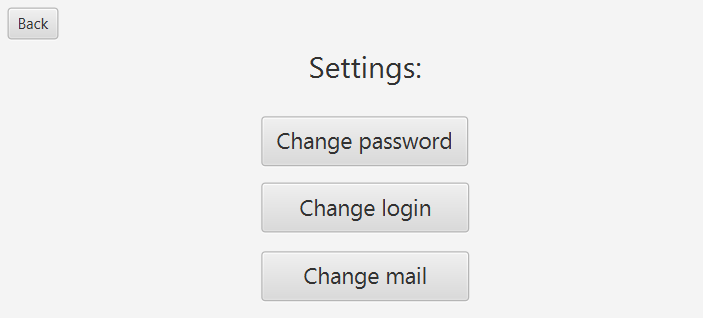


Рисунок 2.8 *–* Окно профиля пользователя

В каждом окне должна быть навигация, то есть с любого окна пользователь может вернуться в главное меню. На рисунке 2.9 изображено окно, содержащее все таблицы, имеющиеся в системе. При нажатии на любую из них отобразится окно с информацией по данной таблице. При нажатии на кнопку «Back» пользователь перейдёт на страницу с главным меню.

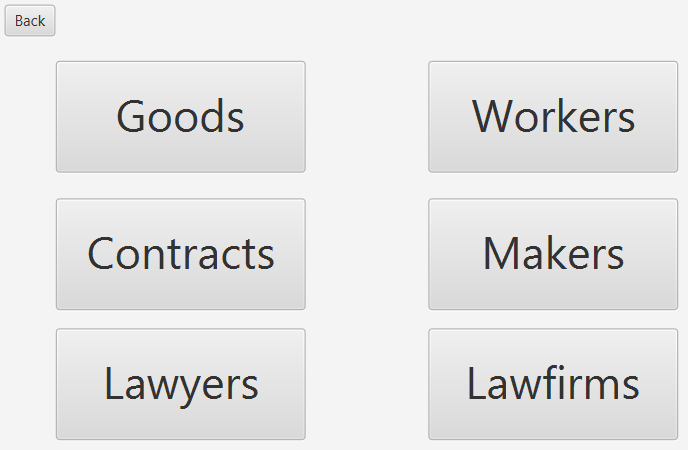


Рисунок 2.9 *–* Окно списка таблиц

На рисунке 2.10 изображено окно информации о конкретной таблице. На ней пользователь может сортировать данные по любому столбцу, удалять данные, а также производить поиск необходимой информации.



Рисунок 2.10 *–* Окно информации о таблице

Таким образом, были спроектированы прототипы пользовательского интерфейса разрабатываемого программного средства и внедрены в проект.

## 2.3 Описание и реализация используемых в программном средстве алгоритмов

В мире программирования алгоритмы играют решающую роль, определяя эффективность и функциональность создаваемых программ. Алгоритм – это набор инструкций, предписывающих последовательность шагов для достижения конкретной цели. Он подобен рецепту, по которому готовится блюдо, где каждый шаг определен и направлен на получение определенного результата [20].

Алгоритмы широко используются в программировании для решения разнообразных задач, начиная от сортировки данных и поиска оптимальных путей до шифрования информации. Важной характеристикой алгоритма является его инструктивность – способность быть понятым и легко интерпретированным человеком.

Взаимодействие алгоритмов с различными структурами данных, такими как массивы, списки и деревья, играет ключевую роль в эффективной обработке и хранении информации. Выбор подходящей структуры данных в сочетании с оптимальным алгоритмом обеспечивает оптимальную производительность программы.

Известные алгоритмы, такие как сортировка QuickSort, поиск в ширину (BFS), шифрование RSA, являются краеугольными камнями мира программирования. Их анализ и понимание обеспечивают необходимый фундамент для разработчиков.

Существуют различные типы алгоритмов [21]:

‒ Линейный алгоритм − последовательность команд (инструкций), выполняемых последовательно во времени.

‒ Разветвляющийся алгоритм − алгоритм, содержащий в качестве результата управления хотя бы одно условие, которое можно разделить на несколько альтернативных ветвей алгоритма.

‒ Циклический алгоритм − алгоритм, в котором одно и то же действие (один и тот же процесс) повторяется несколько раз. Большинство вычислительных методов сводятся к круговым алгоритмам. Цикл программы − последовательность инструкций (серия, тело цикла), которая может выполняться многократно.

Основным способом визуализации последовательности алгоритмов является создание схем алгоритмов. Схема алгоритма представляет собой визуальное представление шагов, которые необходимо выполнить для решения определенной задачи или достижения конкретной цели. С помощью схемы алгоритма можно проиллюстрировать последовательность действий, операторы и управляющие конструкции, используемые в процессе выполнения алгоритма.

Схема алгоритма позволяет легко визуализировать и понять последовательность выполнения операций, структуру алгоритма и поток управления программы. Она может быть использована для разработки, исправления и отладки алгоритмов перед переходом к написанию программного кода. Благодаря графическому представлению, схема алгоритма удобна для коммуникации и обмена информацией между разработчиками или командами, работающими над одной задачей.

Далее будут представлены примеры используемых в проекте алгоритмов каждого типа с их схемами для лучшего понимания работы и наглядности.

Первым следует представить пример линейного алгоритма в связи с его простотой. Таким алгоритмом является алгоритм очистки полей поиска. Его код представлен ниже:

private void clearFields() {

firmField.setText("");

yearsField.setText("");

reputationField.setText("");

}

Схема данного алгоритма будет приведена на рисунке 2.11.

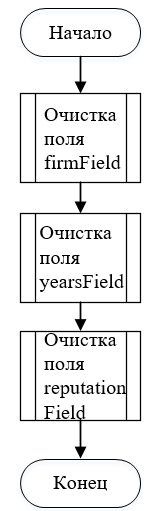


Рисунок 2.11 *–* Схема линейного алгоритма

Далее представлен пример разветвляющегося алгоритма. Данным алгоритмом является алгоритм подтверждения. Его код представлен ниже:

public void verification() {

if (password.equals(hashClass.getSecurePasswordWithSalt(passwordField.getText(), salt)) && code.equals(codeField.getText())) {

Scenes.NEW\_LOGIN.setScene((Stage) verificationButton.getScene().getWindow());

} else errorLabel.setText(Const.INCORRECT);

}

Схема данного алгоритма будет приведена на рисунке 2.12.

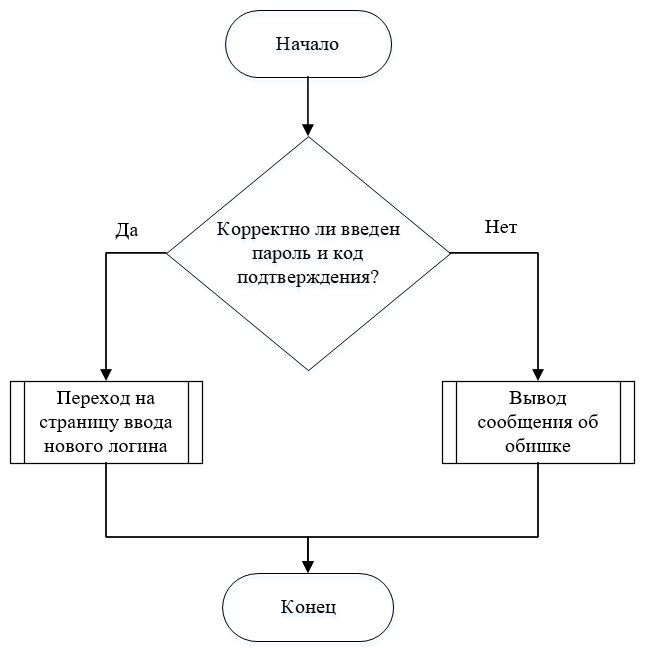


Рисунок 2.12 *–* Схема разветвляющегося алгоритма

Последним представлен циклический алгоритм. Чаще всего алгоритмы данного типа применяются при необходимости работы с большим массивом связанных данных. Одним из таких алгоритмов в проекте является алгоритм добавления флага. Его код представлен ниже:

private void addFlags() {

for (int i = 0; i < 3; ++i) {

searchFlags.add(true);

changeFlags.add(false);

}

}

Схема данного алгоритма будет приведена на рисунке 2.13.

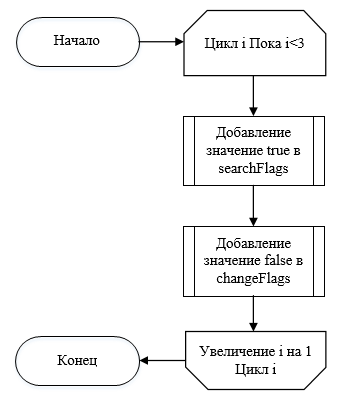


Рисунок 2.13 *–* Схема циклического алгоритма

Также в перечне графического материала будет представлена схема алгоритма всей программы. Эта схема позволит лучше отобразить все внутренние процессы разработанного программного средства.

# Эксплуатация программного средства

## 3.1 Ввод в эксплуатацию и обоснование минимальных технических требований к оборудованию

Ввод в эксплуатацию программного средства на платформе JavaFX представляет собой критически важный этап в жизненном цикле разработки программного обеспечения [22]. Этот этап включает в себя развертывание приложения, подготовку к работе конечных пользователей и обеспечение необходимых технических ресурсов для эффективной работы приложения. В данном разделе будет рассмотрен процесс ввода в эксплуатацию программного средства на платформе JavaFX и обоснование минимальных технических требований к оборудованию.

Ввод в эксплуатацию предполагает выполнение ряда шагов, направленных на успешное развертывание программного продукта и готовность его к использованию конечными пользователями. Основные этапы ввода в эксплуатацию на JavaFX включают в себя:

– Создание исполняемого файла: сборка и создание исполняемого файла (JAR или EXE) из исходного кода JavaFX приложения. Этот файл будет предоставлен конечным пользователям для установки и запуска.

– Тестирование на совместимость: проведение тестирования на различных конфигурациях оборудования и операционных систем, чтобы обеспечить совместимость приложения с различными устройствами.

– Развертывание на сервере или хосте: если приложение требует централизованного доступа, необходимо убедиться, что оно развернуто на сервере или хосте, готовом обрабатывать запросы от пользователей.

– Установка и конфигурирование: также необходимо разработать инструкции по установке и настройке приложения на компьютерах конечных пользователей. Необходимо убедиться, что процесс установки интуитивно понятен.

– Обеспечение безопасности: разработчику следует разработать и реализовать меры безопасности для защиты приложения от несанкционированного доступа и вредоносных атак.

– Обучение пользователей: предоставление руководства пользователя и обучение для конечных пользователей, чтобы обеспечить эффективное использование приложения.

Обоснование минимальных технических требований к оборудованию включает в себя определение минимальных характеристик, которые должны быть у конечных пользователей для нормального функционирования программного средства. Эти требования могут включать в себя:

– Типы устройств: ПК, ноутбуки или планшеты с операционной системой, поддерживающей JavaFX (например, Windows, Linux, MacOS).

– Процессор и память: минимальные требования к процессору и оперативной памяти для стабильного выполнения приложения.

Разрешение экрана: рекомендуемое разрешение экрана для оптимального визуального восприятия приложения.

Версия Java: указание минимальной версии Java, которую должен поддерживать компьютер пользователя.

Другие зависимости: любые другие технические зависимости, такие как наличие определенных библиотек или поддержка определенных технологий (например, OpenGL).

В рамках данной курсовой работы проект был реализован при помощи JavaFX, что накладывает ограничение на версию языка Java, то есть Java должна быть 11 версии и старше [23].

Рассмотрим вариант для ввода в эксплуатацию и дальнейшего распространения с созданием исполняемого файла .ехе и последующей загрузкой на GutHub, откуда его сможет скачать любой желающий. Чтобы создать такой файл сперва необходимо собрать jar для своего проекта. Это можно сделать в самой среде разработки IntelliJ IDEA, перейдя в окно File -> Project Structure -> Project Settings -> Artifacts -> Add -> JavaFx application -> From Module «имя модуля». После этого нужно добавить к созданному артефакту все библиотеки, использованные в проекте, после заполнить сведения об основном классе, который запускает приложение (поле Application class). В той же вкладке надо указать значение «all» в поле «Native bundle» и применить настройки. После этого необходимо войти в режим Build -> Build Artifacts и выбрать только что созданный артефакт.

Также ехе файл можно сделать и с помощью сторонних приложений, таких как Launch4j или Inno Setup [24].

Существуют и иные способы, но рекомендации при создании ехе файла, как правило, одни: во избежание ошибок при создании jar или exe файлов в наименовании пути к директорию не должно быть ни кириллицы, ни специальных символов.

После создания исполняемого файла остается загрузить его на GutHub [25]. Также на GutHub следует загрузить и инструкцию для будующих пользователей разработанного программного средства. GutHub позволяет создавать в своих репозиториях характерные для описания разработанных программ README.md файлы. В таких файлах обычно прописывают всю инструкцию для корректного скачивания и установки.

Также для обеспечения работоспособности программы в условиях корпоративного использования необходимо убедиться, что на сервере установлена подходящая версия Java, необходимая для выполнения разработанного приложения. Затем необходимо перенесьти собранный исполняемый файл и все необходимые ресурсы на сервер, в том числе и разработанная база данных. Это можно сделать с использованием scp (Secure Copy Protocol) или других инструментов передачи файлов.

## Руководство по эксплуатации программного средства

Представим руководство пользователя сотрудника магазина оргтехники, который пользуется программным средством для работы с информацией о покупках и поставщиках.

При запуске программы открывается главная страница, на которой есть кнопки «Entrance» и «Registration» (см. рисунок 3.1).

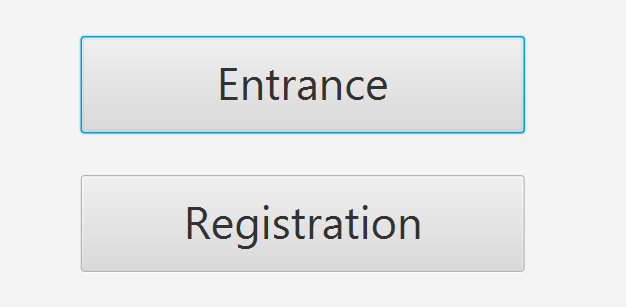


Рисунок 3.1 *–* Начальная страница

В случае прихода в компанию нового сотрудника необходимо нажать на кнопку «Registration», тогда пользователь попадает на страницу регистрации в программном средстве (см. рисунок 3.2).

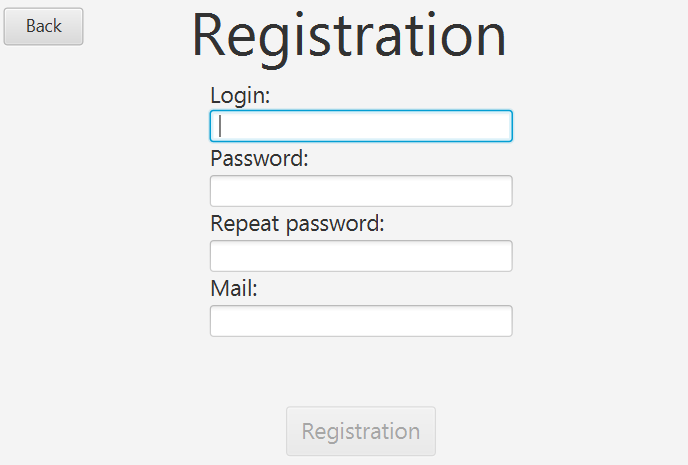


Рисунок 3.2 *–* Страница регистрации

После прохождения регистрации пользователь будет перенаправлен на начальную страницу.

Если пользователь уже зарегистрирован в программном средстве, необходимо нажать на кнопку «Entrance». Откроется страница авторизации пользователя, где необходимо ввести логин и пароль. Далее нужно нажать на кнопку «Enter» и пользователь будет успешно авторизован (см. рисунок 3.3).

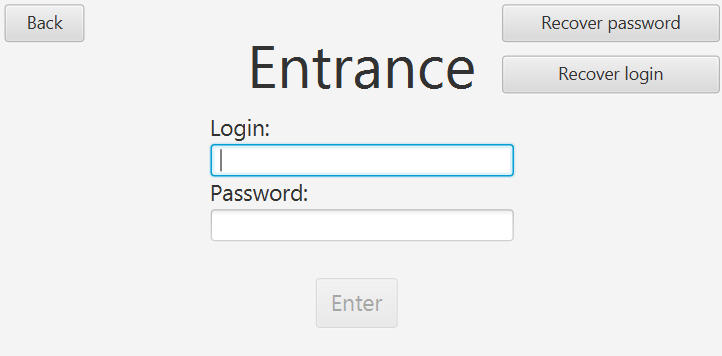


Рисунок 3.3 *–* Страница авторизации

В случае, если пользователь забыл свои учетные данные, в программе предусмотрена возможность восстановления логина и пароля. Для восстановления логина пользователю необходимо нажать на кнопку «Recover login», после чего программа перейдет к окну восстановления логина (см. рисунок 3.4).

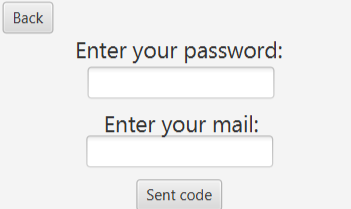


Рисунок 3.4 *–* Страница восстановления логина

Если же возникла необходимость восстановить пароль, пользователю требуется нажать на кнопку «Recover password», после чего программа перейдет к окну восстановления пароля (см. рисунок 3.5).

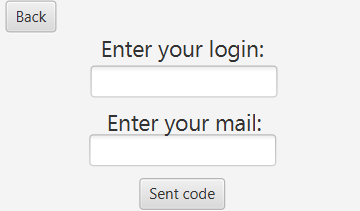


Рисунок 3.5 *–* Страница восстановления пароля

Выполнив вход в аккаунт, пользователь попадает на страницу главного меню, где ему будет предоставлена возможность редактировать свои персональные данные, а также возможность взаимодействовать с таблицами из базы данных (см. рисунок 3.6).

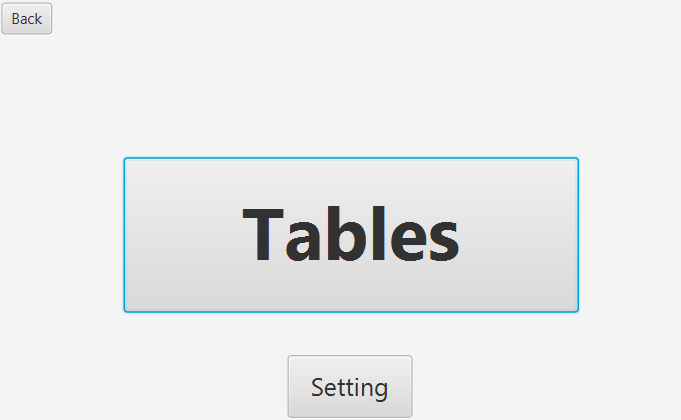


Рисунок 3.6 *–* Страница главного меню

Представим, что пользователю нужно отредактировать свои персональные данные. Для этого необходимо нажать на кнопку «Settings», после чего программа перенаправит пользователя на страницу редактирования своего профиля (см. рисунок 3.7).

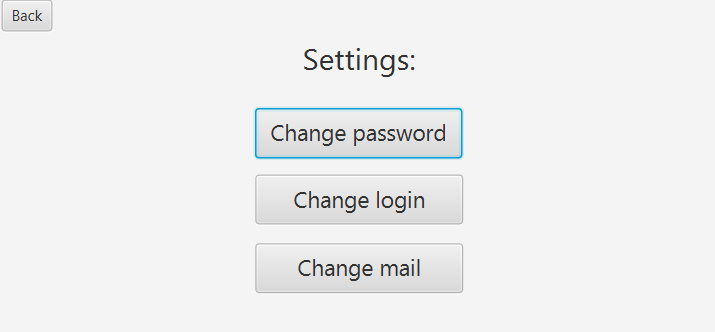


Рисунок 3.7 *–* Страница редактирования своего профиля

На данной странице пользователю будет предоставлена возможность редактировать следущее данные своего профиля: пароль, логин и адрес электронной почты. Для этого в открывшемся окне присутствуют три кнопки, отвечающие за указанные выше возможности соответственно.После выбора необходимого действия запустится страница с формой для заполнения необходимых данных, после ввода которых пользователю будет предложено указать новые корректные данные. В качестве примера будет рассмотрен пример смены логина пользователя (см. рисунок 3.8).

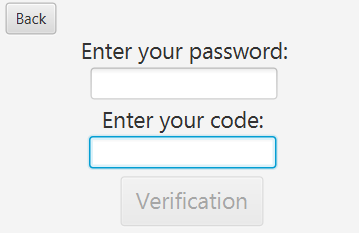


Рисунок 3.8 *–* Страница смены логина

Как было написано выше, после указания необходимых данных пользователю будет предложена форма для ввода новых данных. В контексте разбираемого примера программа предоставила форму для ввода нового логина. После ввода новых данных программа перенаправит пользователя на начальную страницу, где пользователь при необходимости сможет войти в систему, исользуя новые корректные данные (см. рисунок 3.9).

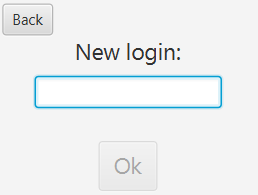


Рисунок 3.9 — Страница ввода нового логина

Если же пользователю необходимо приступить к работе с базой данных, то на странице главного меню требуется нажать кнопку «Tables», после чего программа откроет страницу с выбором таблиц (см. рисунок 3.10).

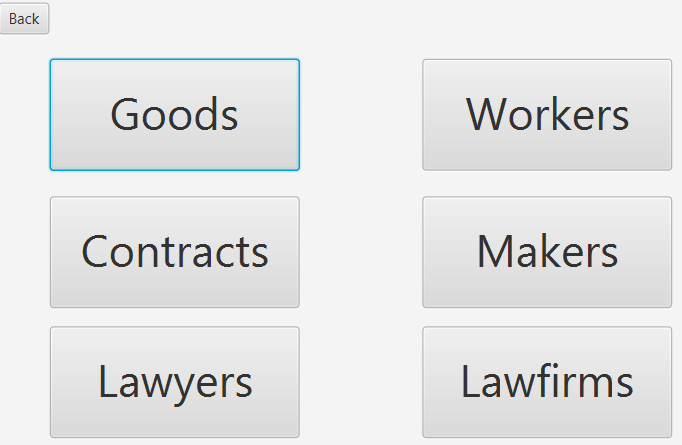


Рисунок 3.10 *–* Страница с выбором таблиц

Сделав выбор необходимой таблицы, пользователь перейдет к странице, которая содержит все интересующие его данные. На этой странице пользователь сможет просматривать данные, выполнять поиск и сортировку данных, а также удалять и добавлять данные в таблицу (см. рисунок 3.11).

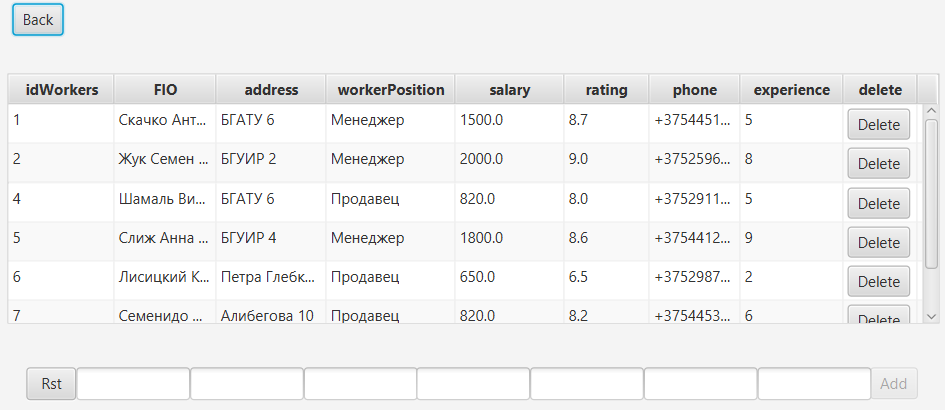


Рисунок 3.11 *–* Страница с данными выбранной таблицы

Для выхода из своего аккаунта пользователю необходимо перейти на страницу главного меню и нажать на кнопку «Back».

**ЗАКЛЮЧЕНИЕ**

В результате выполнения работы была достигнута цель – разработано программное средство для магазина оргтехники. В ходе разработки были решены следующие задачи:

Был проведен анализ информационных потребностей пользователей и сформулирована задача, реализация которой позволила удовлетворить их потребности. Также был проведен анализ исходных данных, была спроектирована объектная модель программного средства, составлено описание его состояний, был спроектирован и разработан графический интерфейс программы. Вместе с этим было составлено описание используемых в программном средстве алгоритмов, руководство по эксплуатации, обоснование минимальных технических требований к оборудованию, а также описана среда и язык реализации с подробной аргументацией выбора используемых технологий.

Были разработаны инфологическая и физическая модели БД, определены типы данных и ограничения. Были освоены некоторые приемы написания запросов и представлений на языке SQL.

В данной курсовой работе была спроектирована (до 3НФ) и реализована реляционная база данных (СУБД MySQL SERVER). В базе данных имеются ограничения: на всех полях установлено ограничение not null, где это возможно без нарушения корректной работы программного средства, во всех таблицах выбрано ключевое поле, на него наложено ограничение auto\_increment.

В разработанном программном средстве также созданы ограничения на ввод некорректных символов, проверка на правильность введенных логинов и паролей, а также были предусмотрены ограничения для ненадежных паролей.

В дальнейшем разработанную систему можно усовершенствовать, добавив дополнительный функционал программного средства.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Оргтехника – что к ней относится? [Электронный ресурс]. – Режим доступа: https://flex-system.ru/novosti-i-stati/articles/3576000/ – Дата доступа: 19.09.2023.

[2] Интуитивно понятный интерфейс: 4 простых принципа [Электронный ресурс]. – Режим доступа: https://ru.experrto.io/blog/2019/06/28/intuitivno-ponyatnyj-interfejs-4-prostyh-principa/ – Дата доступа: 21.09.2023.

[3] История создания Java [Электронный ресурс]. – Режим доступа: https://vertex-academy.com/tutorials/ru/istoriya-sozdaniya-java-2/ – Дата доступа: 21.09.2023.

[4] Язык программирования Java: что нужно знать новичку [Электронный ресурс]. – Режим доступа: https://ru.hexlet.io/blog/posts/yazyk-programmirovaniya-java-osobennosti-populyarnost-situatsiya-na-rynke-truda – Дата доступа: 20.09.2023.

[5] Что такое Java? [Электронный ресурс]. – Режим доступа: https://aws.amazon.com/ru/what-is/java/ – Дата доступа: 19.09.2023.

[6] Введение в Java FX [Электронный ресурс]. – Режим доступа: https://javarush.com/groups/posts/2560-vvedenie-v-java-fx – Дата доступа: 23.09.2023.

[7] JavaFX: описание и особенности [Электронный ресурс]. – Режим доступа: https://otus.ru/journal/javafx-pisanie-i-osobennosti/ – Дата доступа: 23.09.2023.

[8] Вышла Java 11: новые фичи и возможности [Электронный ресурс]. – Режим доступа: https://javarush.com/groups/posts/1961-vihshla-java-11-novihe-fichi-i-vozmozhnosti- – Дата доступа: 23.09.2023.

[9] JavaFX, CSS-стили и пользовательские свойства компонентов [Электронный ресурс]. – Режим доступа: https://ru.intechcore.com/javafx-and-css-styles-2/ – Дата доступа: 24.09.2023.

[10] IntelliJ IDEA [Электронный ресурс]. – Режим доступа: https://blog.skillfactory.ru/glossary/intellij-idea/ – Дата доступа: 27.09.2023.

[11] Функциональность – IntelliJ IDEA [Электронный ресурс]. – Режим доступа: https://www.jetbrains.com/ru-ru/idea/features/ – Дата доступа: 27.09.2023.

[12] Что такое IntelliJ IDEA [Электронный ресурс]. – Режим доступа: https://apptractor.ru/develop/ide/chto-takoe-intellij-idea.html – Дата доступа: 27.09.2023.

[13] Начало работы с IntelliJ IDEA и распределенным SQL [Электронный ресурс]. – Режим доступа: https://proglib.io/p/nachalo-raboty-s-intellij-idea-i-raspredelennym-sql-2020-11-13 – Дата доступа: 27.09.2023.

[14] Intellij IDEA плюсы и минусы в 2023 [Электронный ресурс]. – Режим доступа: https://blog.uniwex.io/intellij-idea-plyusy-i-minusy-v-2023/ – Дата доступа: 27.09.2023.

[15] Как запустить Tomcat в IntelliJ IDEA [Электронный ресурс]. – Режим доступа: https://uchet-jkh.ru/i/kak-zapustit-tomcat-v-intellij-idea/ – Дата доступа: 27.09.2023.

[16] Объектная модель системы [Электронный ресурс]. – Режим доступа: https://old.mista.ru/oop\_book/glava2\_1.htm – Дата доступа: 28.09.2023.

[17] UML для самых маленьких: диаграмма классов [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/511798/ – Дата доступа: 28.09.2023.

[18] Создание Java программы с дизайном! Изучение библиотеки JavaFx [Электронный ресурс]. – Режим доступа: https://itproger.com/news/sozdanie-java-programmi-s-dizaynom-izuchenie-biblioteki-javafx – Дата доступа: 29.09.2023.

[19] Информация про UI/UX [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/321312/ – Дата доступа: 29.09.2023.

[20] Алгоритм [Электронный ресурс]. – Режим доступа: https://blog.skillfactory.ru/glossary/algoritm/ – Дата доступа: 01.10.2023.

[21] Виды алгоритмов и типы их схем [Электронный ресурс]. – Режим доступа: https://otus.ru/nest/post/1784/ – Дата доступа: 01.10.2023.

[22] Внедрение программного продукта. Особенности работы бизнес- консультанта [Электронный ресурс]. – Режим доступа: https://trinion.org/blog/vnedrenie-programmnogo-produkta-osobennosti-raboty-biznes-konsultanta-chast-i – Дата доступа: 20.10.2023.

[23] Как в IntelliJ IDEA написать и собрать в исполняемый .exe файл приложение на JavaFX [Электронный ресурс]. – Режим доступа: <https://habr.com>

/ru/articles/723798/ – Дата доступа: 21.10.2023.

[24] Создание exe-файла из jar [Электронный ресурс]. – Режим доступа: https://java-online.ru/maven-launch4j.xhtml – Дата доступа: 23.10.2023.

[25] Добавление проект в репозиторий [Электронный ресурс]. – Режим доступа: https://docs.github.com/ru/enterprise-cloud@latest/issues/planning-and-tracking-with-projects/managing-your-project/adding-your-project-to-a-repository – Дата доступа: 23.10.2023.

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Отчет о проверке на заимствование**

**в системе «Антиплагиат»**

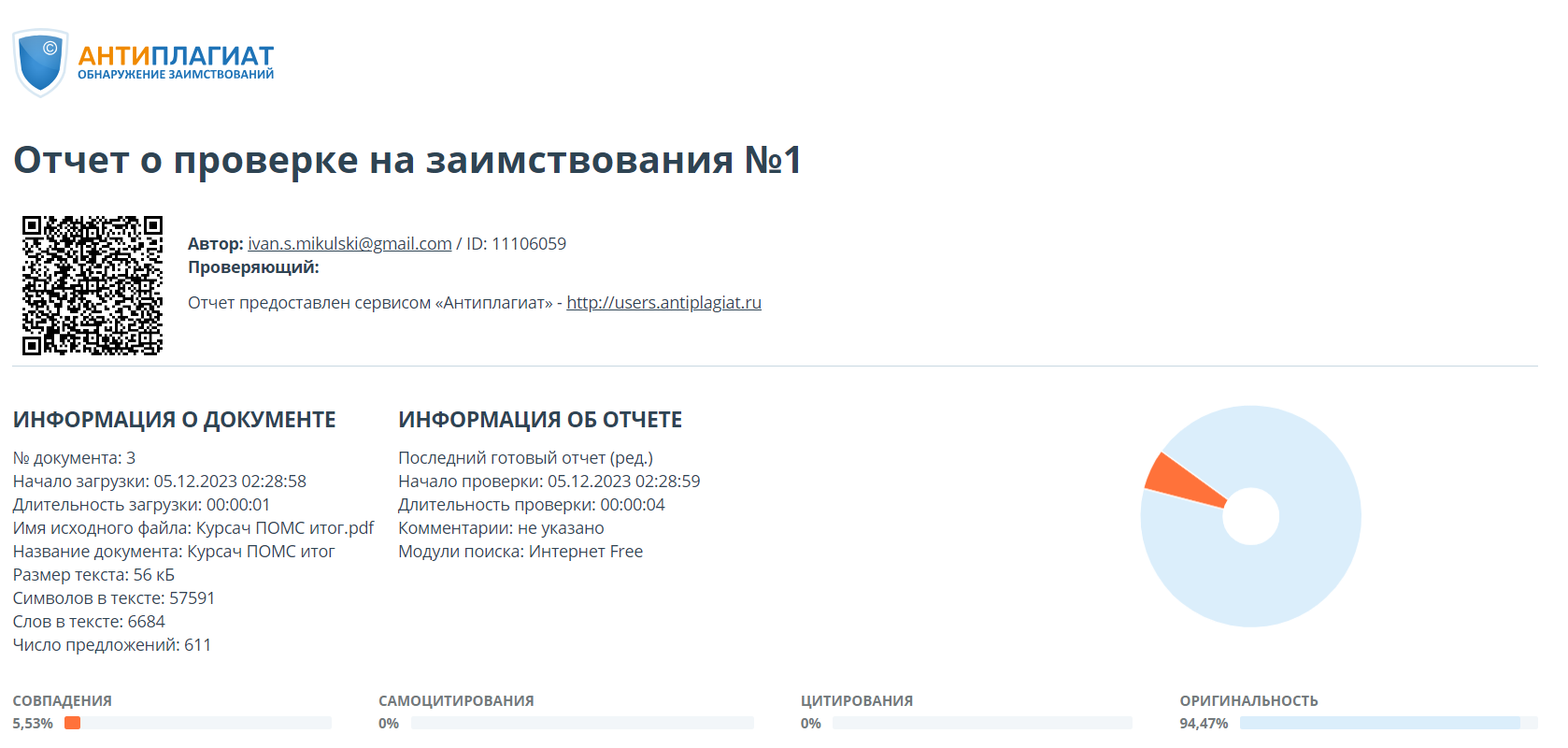
****

Рисунок А.1 *–* Отчет о проверке на заимствование

в системе «Антиплагиат»

**ПРИЛОЖЕНИЕ Б**

**(обязательное)**

**Листинг кода с комментариями**

**Файл Main.java**

import javafx.application.Application;

import javafx.stage.Stage;

import org.example.Database.Classes.HandlerClasses.DatabaseHandler;

import org.example.Database.Enums.EnumsForFX.Scenes;

public class Main extends Application {

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage stage) {

Scenes.MENU.setScene(stage);

stage.show();

}

@Override

public void stop() throws Exception {

(new DatabaseHandler()).truncateOnlineUser();

}

}

**Файл verificationForChangeLoginControl.java**

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.stage.Stage;

import org.example.Database.Classes.HandlerClasses.DatabaseHandler;

import org.example.Database.Enums.EnumsForFX.Scenes;

import java.net.URL;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class verificationForChangeLoginControl implements Initializable {

@FXML

private Label errorLabel;

@FXML

private Button verificationButton;

@FXML

private Button backButton;

@FXML

private TextField codeField;

@FXML

private TextField passwordField;

private final DatabaseHandler dbHandler = new DatabaseHandler();

final methodsWithConnectionToInternet connection=new methodsWithConnectionToInternet();

final String code = connection.getVerificationCode();

String password;

String salt;

{

try {

password = dbHandler.getOnlineUser().getString(3);

salt=dbHandler.getOnlineUser().getString(4);

} catch (SQLException e) {

e.printStackTrace();

}

}

public void back() {

Scenes.MAIN\_MENU.setScene((Stage) backButton.getScene().getWindow());

}

@Override

public void initialize(URL url, ResourceBundle resourceBundle) {

verificationButton.setDisable(true);

sentCode(dbHandler, connection, code, codeField, passwordField);

codeField.textProperty().addListener((observable, oldValue, newValue) -> {

if (newValue != null && newValue.length() != 0) {

if (checkForCorrectSymbols.isCorrectSymbolsForCode(newValue)) {

codeField.setText(oldValue);

}

verificationButton.setDisable(codeField.getText().isEmpty() || passwordField.getText().isEmpty());

} else verificationButton.setDisable(true);

});

passwordField.textProperty().addListener((observable, oldValue, newValue) -> {

if (newValue != null && newValue.length() != 0) {

if (checkForCorrectSymbols.isCorrectSymbolsForPasswordAndLogin(newValue)) {

passwordField.setText(oldValue);

}

verificationButton.setDisable(codeField.getText().isEmpty() || passwordField.getText().isEmpty() || newValue.length()<6);

} else verificationButton.setDisable(true);

});

}

public void verification() {

if (password.equals(hashClass.getSecurePasswordWithSalt(passwordField.getText(), salt)) && code.equals(codeField.getText())) {

Scenes.NEW\_LOGIN.setScene((Stage) verificationButton.getScene().getWindow());

} else errorLabel.setText(Const.INCORRECT);

}

static void sentCode(DatabaseHandler dbHandler, methodsWithConnectionToInternet connection, String verificationCode, TextField codeField, TextField parameterField) {

String mail = "";

try {

mail = dbHandler.getOnlineUser().getString(5);

} catch (SQLException e) {

e.printStackTrace();

}

String finalMail = mail;

if (!connection.isSentMessage(verificationCode, finalMail, Const.VERIFICATION\_CODE)) {

codeField.setDisable(true);

parameterField.setDisable(true);

}

}

}

**Файл checkForCorrectSymbols.java**

public class checkForCorrectSymbols {

//проверяет последний символ пароля и логина на корректность-----------------------------------------------------------------------------------------

public static Boolean isCorrectSymbolsForPasswordAndLogin(String newValue) {

return (newValue.charAt(newValue.length() - 1)<'0'||

(newValue.charAt(newValue.length() - 1)>'9'&& newValue.charAt(newValue.length() - 1) < 'A') ||

(newValue.charAt(newValue.length() - 1) > 'Z' && newValue.charAt(newValue.length() - 1) < 'a') ||

newValue.charAt(newValue.length() - 1) > 'z');

}

//проверяет последний символ кода на корректность-----------------------------------------------------------------------------------------

public static Boolean isCorrectSymbolsForCode(String newValue) {

return (newValue.charAt(newValue.length() - 1) < '0' ||

newValue.charAt(newValue.length() - 1) > '9' ||

newValue.charAt(0) == '0');

}

//проверяет последний символ почты на корректность-----------------------------------------------------------------------------------------

public static Boolean isCorrectSymbolsForMail(String newValue) {

return (newValue.charAt(newValue.length() - 1) < '-' ||

(newValue.charAt(newValue.length() - 1) > '.' && newValue.charAt(newValue.length() - 1) < '0') ||

(newValue.charAt(newValue.length() - 1) > '9' && newValue.charAt(newValue.length() - 1) < '@') ||

(newValue.charAt(newValue.length() - 1) > 'Z' && newValue.charAt(newValue.length() - 1)<'\_') ||

(newValue.charAt(newValue.length() - 1) > '\_' && newValue.charAt(newValue.length() - 1)<'a') ||

newValue.charAt(newValue.length() - 1)>'z');

}

}

**Файл checkForCorrectSymbols.java**

public class checkForCorrectSymbols {

//проверяет последний символ пароля и логина на корректность-----------------------------------------------------------------------------------------

public static Boolean isCorrectSymbolsForPasswordAndLogin(String newValue) {

return (newValue.charAt(newValue.length() - 1)<'0'||

(newValue.charAt(newValue.length() - 1)>'9'&& newValue.charAt(newValue.length() - 1) < 'A') ||

(newValue.charAt(newValue.length() - 1) > 'Z' && newValue.charAt(newValue.length() - 1) < 'a') ||

newValue.charAt(newValue.length() - 1) > 'z');

}

//проверяет последний символ кода на корректность-----------------------------------------------------------------------------------------

public static Boolean isCorrectSymbolsForCode(String newValue) {

return (newValue.charAt(newValue.length() - 1) < '0' ||

newValue.charAt(newValue.length() - 1) > '9' ||

newValue.charAt(0) == '0');

}

//проверяет последний символ почты на корректность-----------------------------------------------------------------------------------------

public static Boolean isCorrectSymbolsForMail(String newValue) {

return (newValue.charAt(newValue.length() - 1) < '-' ||

(newValue.charAt(newValue.length() - 1) > '.' && newValue.charAt(newValue.length() - 1) < '0') ||

(newValue.charAt(newValue.length() - 1) > '9' && newValue.charAt(newValue.length() - 1) < '@') ||

(newValue.charAt(newValue.length() - 1) > 'Z' && newValue.charAt(newValue.length() - 1)<'\_') ||

(newValue.charAt(newValue.length() - 1) > '\_' && newValue.charAt(newValue.length() - 1)<'a') ||

newValue.charAt(newValue.length() - 1)>'z');

}

}

**Файл entranceControl.java**

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.PasswordField;

import javafx.scene.control.TextField;

import javafx.stage.Stage;

import org.example.Database.Classes.HandlerClasses.DatabaseHandler;

import org.example.Database.Enums.EnumsForFX.Scenes;

import java.net.URL;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class entranceControl implements Initializable {

@FXML

private Button recoverLoginButton;

@FXML

private Button recoverPasswordButton;

@FXML

private Button backButton;

@FXML

private Button entranceButton;

@FXML

private TextField loginField;

@FXML

private PasswordField passField;

@FXML

private Label incorrectParametersLabel;

final DatabaseHandler dbHandler = new DatabaseHandler();

public void authorisation() {

String login = loginField.getText();

String password = passField.getText();

User user = new User();

user.setLogin(login);

ResultSet foundUser = dbHandler.findUser(user);

try {

if (!foundUser.next()) {

incorrectParametersLabel.setText(Const.NO\_SUCH\_USER);

} else {

String salt = foundUser.getString(4);

if (foundUser.getString(3).equals(hashClass.getSecurePasswordWithSalt(password, salt))) {

dbHandler.insertOnlineUserAfterAuth(user);

Scenes.MAIN\_MENU.setScene((Stage) entranceButton.getScene().getWindow());

} else incorrectParametersLabel.setText(Const.NO\_SUCH\_USER);

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

@Override

public void initialize(URL url, ResourceBundle resourceBundle) {

entranceButton.setDisable(true);

backButton.setOnAction(actionEvent -> {

Scenes.MENU.setScene((Stage) backButton.getScene().getWindow());

});

recoverLoginButton.setOnAction(actionEvent -> {

Scenes.RECOVER\_LOGIN.setScene((Stage) recoverLoginButton.getScene().getWindow());

});

recoverPasswordButton.setOnAction(actionEvent -> {

Scenes.RECOVER\_PASSWORD.setScene((Stage) recoverPasswordButton.getScene().getWindow());

});

checkFieldsForFill(loginField);

checkFieldsForFill(passField);

}

private void checkFieldsForFill(TextField field) {

field.textProperty().addListener((observable, oldValue, newValue) -> {

if (newValue != null && newValue.length() != 0) {

if (checkForCorrectSymbols.isCorrectSymbolsForPasswordAndLogin(newValue)) {

field.setText(oldValue);

}

entranceButton.setDisable(loginField.getText().isEmpty() || passField.getText().isEmpty() || newValue.length()<6);

} else entranceButton.setDisable(true);

});

}

}

**Файл newMailControl.java**

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.stage.Stage;

import org.example.Database.Classes.HandlerClasses.DatabaseHandler;

import org.example.Database.Enums.EnumsForFX.Scenes;

import java.net.URL;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class newMailControl implements Initializable {

@FXML

private Button backButton;

@FXML

private Button exceptButton;

@FXML

private Label errorLabel;

@FXML

private TextField newMailField;

final DatabaseHandler dbHandler = new DatabaseHandler();

final ResultSet onlineUser = dbHandler.getOnlineUser();

final String oldMail;

{

try {

oldMail = onlineUser.getString(5);

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

final methodsWithConnectionToInternet connection = new methodsWithConnectionToInternet();

public void back() {

Scenes.MENU.setScene((Stage) backButton.getScene().getWindow());

}

@Override

public void initialize(URL url, ResourceBundle resourceBundle) {

exceptButton.setDisable(true);

newMailField.textProperty().addListener((observable, oldValue, newValue) -> {

if (newValue != null && newValue.length() != 0) {

if (checkForCorrectSymbols.isCorrectSymbolsForMail(newValue)) {

newMailField.setText(oldValue);

}

exceptButton.setDisable(newValue.equals(oldMail));

} else exceptButton.setDisable(true);

});

exceptButton.setOnAction(actionEvent -> except());

}

public void except() {

String newMail = newMailField.getText();

boolean wasMessageSent = false;

if (connection.isSentMessage(Const.RE\_LINKED\_TO\_NEW\_MAIL\_MESSAGE\_FIRST\_PART + oldMail + Const.RE\_LINKED\_TO\_NEW\_MAIL\_MESSAGE\_SECOND\_PART, newMail, Const.RE\_LINKED\_THEME)) {

errorLabel.setText(Const.NO\_CONNECTION\_OR\_INCORRECT\_MAIL);

wasMessageSent = true;

}

final boolean finalWasMessageSent = wasMessageSent;

if (finalWasMessageSent) {

connection.isSentMessage(Const.RE\_LINKED\_TO\_OLD\_MAIL\_MESSAGE + newMail, oldMail, Const.RE\_LINKED\_THEME);

}

if (finalWasMessageSent) {

User user = new User();

try {

user.setID(onlineUser.getString(1));

} catch (SQLException e) {

throw new RuntimeException(e);

}

user.setMail(newMail);

dbHandler.updateUserMail(user);

}

if (wasMessageSent) {

Scenes.MENU.setScene((Stage) backButton.getScene().getWindow());

}

}

}

**ПРИЛОЖЕНИЕ В**

**(обязательное)**

**Ведомость курсовой работы**