

МИНИСТЕРСТВО ОБРАЗОВАНИЯ САРАТОВСКОЙ ОБЛАСТИ
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ САРАТОВСКОЙ ОБЛАСТИ
«ВОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»

Специальность: 09.02.07 «Информационные системы и программирование»

КУРСОВОЙ ПРОЕКТ
РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ МЕДИЦИНСКОГО ЦЕНТРА
ПМ.02 09.02.07.00.31.03

Руководитель:
преподаватель Мифтахов Н.И.

«___» _____ 2024 г.

Выполнил:
студент группы ИТ-31
_____ Бильтяев Д.А.

«___» _____ 2024 г.

Вольск 2024 г.

<i>ВВЕДЕНИЕ</i>	<i>3</i>
<i>1 ПРЕДВАРИТЕЛЬНОЕ ПЛАНИРОВАНИЕ И ПРОВЕРКА ОСУЩЕСТВИМОСТИ</i>	<i>5</i>
<i>2 ПОСТАНОВКА ЗАДАЧИ И ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ</i>	<i>8</i>
<i>3 РАЗРАБОТКА ПРИЛОЖЕНИЯ</i>	<i>11</i>
<i>3.1 Создание технического задания</i>	<i>11</i>
<i>3.2 Прототипирование</i>	<i>14</i>
<i>4 РЕАЛИЗАЦИЯ</i>	<i>17</i>
<i>4.1 Выбор и обоснование IDE</i>	<i>17</i>
<i>4.2 Разработка логики приложения</i>	<i>17</i>
<i>4.3 Написание кода</i>	<i>18</i>
<i>4.4 Создание UI</i>	<i>21</i>
<i>5. ТЕСТИРОВАНИЕ</i>	<i>24</i>
<i>5.1 Первичное тестирование</i>	<i>24</i>
<i>5.2 Повторное тестирование</i>	<i>25</i>
<i>5.3 Итоги тестирования</i>	<i>26</i>
<i>6 СОЗДАНИЕ СБОРКИ</i>	<i>28</i>
<i>Заключение</i>	<i>34</i>
<i>Список использованных источников</i>	

					ПМ.02 09.02.07.00.31.03			
Изм.	Лист	№ докум	Подпись	Дата	РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ МЕДИЦИНСКОГО ЦЕНТРА	Литера	Лист	Листов
Подготовил		Бильтяев Д.А.				у	2	13
Проверил		Мифтахов Н.И.				ГАПОУ СО «ВТК»		
Принял		Мифтахов Н.И.						

ВВЕДЕНИЕ

Актуальность разработки программного обеспечения для управления и аутентификации пользователей в современных условиях не вызывает сомнений. В эпоху цифровизации и глобального доступа к интернету защита персональных данных и обеспечение безопасности аккаунтов являются одними из ключевых аспектов в разработке приложений.

Обоснование выбора языка программирования.

Для выполнения данного задания выбран язык программирования C#. Основные причины выбора включают:

- Популярность и распространенность: C# широко используется в разработке корпоративных приложений, что делает его одним из самых востребованных языков программирования.
- Платформа .NET: C# является основным языком для разработки на платформе .NET, предоставляющей множество библиотек и инструментов для облегчения разработки.
- Поддержка сообществом и документация: Большая база знаний и активное сообщество разработчиков позволяет быстро находить решения для возникающих проблем.
- Безопасность: C# обладает встроенными механизмами для обеспечения безопасности, что особенно важно при обработке персональных данных.

Обоснование выбора IDE

Для разработки будет использована Visual Studio по следующим причинам:

- Интеграция с .NET: Visual Studio полностью интегрирована с платформой .NET, что облегчает работу с библиотеками, пакетами и инструментами.
- Мощные инструменты для разработки: Visual Studio предоставляет обширный набор функциональных возможностей, включая отладку, профилирование и управление версиями, что ускоряет процесс разработки.

– Поддержка различных языков и технологий: IDE поддерживает широкий спектр языков программирования и технологий, что делает её универсальным инструментом для разработчиков.

– Интуитивно понятный интерфейс: Пользовательский интерфейс Visual Studio разработан с учётом удобства и эффективности, что позволяет сократить время на разработку.

Описание предполагаемой логики работы приложения

Логика работы приложения включает следующие ключевые этапы:

1. Заполнение полей: Пользователь заполняет поля для ввода логина и пароля.

2. Проверка ввода: При нажатии на кнопку "Вход" приложение проверяет, заполнены ли все необходимые поля. Если какие-либо поля не заполнены, выводится сообщение "Не введен логин или пароль".

3. Проверка данных: Если все поля заполнены, приложение сверяет введенные логин и пароль с данными из базы данных.

4. Аутентификация:

– Если логин и пароль совпадают с данными из базы, выводится сообщение "Логин и пароль совпадают!" и пользователь получает доступ к системе.

– Если логин и пароль не совпадают, выводится сообщение "Неверный логин или пароль".

Известные аналоги

Существует множество аналогичных приложений, реализующих функции аутентификации пользователей:

– Google Accounts: Система аутентификации Google, предоставляющая возможность входа в различные сервисы Google через единый аккаунт.

Microsoft Account: Универсальная система аутентификации пользователей для сервисов и продуктов Microsoft, таких как Windows, Office и Xbox.

Auth0: Коммерческая платформа для управления аутентификацией и авторизацией, предоставляющая комплексные решения для разработчиков.

1 ПРЕДВАРИТЕЛЬНОЕ ПЛАНИРОВАНИЕ И ПРОВЕРКА ОСУЩЕСТВИМОСТИ

Предварительное планирование

На данный момент аутентификация пользователей является одной из ключевых задач в системах управления доступом на предприятиях и в различных организациях. Процесс аутентификации направлен на обеспечение безопасности и защиту от несанкционированного доступа.

В существующих системах аутентификации используют:

- Ввод логина и пароля: Основной метод, включающий проверку введенных пользователем данных по заранее сохранённым данным в базе.
- Двухфакторная аутентификация (2FA): Дополнительный уровень защиты, требующий ввода одноразового кода, отправленного, например, на мобильный телефон или электронную почту.
- Биометрическая аутентификация: Использование отпечатков пальцев, распознавания лица, сетчатки глаза для подтверждения личности.
- Социальная аутентификация: Использование учётных записей социальных сетей (Google, Facebook, Microsoft) для входа в систему.

Аналоги приложения:

Google Accounts: Реализует проверку учётных данных и двухфакторную аутентификацию.

Microsoft Account: Аналогичная система с поддержкой дополнительных мер безопасности.

Auth0: Предоставляет комплексное решение для управления аутентификацией с возможностью интеграции различных методов проверки.

Новые приложения:

Passwordless Login: Проекты, разрабатывающие вход без паролей, используя одноразовые коды, биометрические данные или аппаратные ключи.

Децентрализованные системы аутентификации: Например, с использованием блокчейна для улучшения безопасности и конфиденциальности данных.

		Бильяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		5

Информация о существующих и новых приложениях будет использована для определения потенциальных связей между разрабатываемым приложением и другими системами, а также для внедрения лучших практик и современных технологий в процесс аутентификации.

Проверка осуществимости

Технологическая осуществимость

Оборудование: Для работы приложения потребуется ПК на ОС Windows 10 и Windows 11

Программное обеспечение: Для работы приложения не требуется наличие дополнительных приложений

Операционная осуществимость

Персонал: Для работы с разрабатываемым приложением потребуется персонал, обладающий навыками работы с системами аутентификации и базами данных, а также навыками программирования на языке C#. Эти навыки можно приобрести через обучение и практику, что делает операционную осуществимость реалистичной.

Поддержка пользователей: Потребуется наличие специалистов для оказания технической поддержки конечным пользователям, проведения инструктажей и обучения.

Экономическая целесообразность

Затраты на разработку: Затраты включают расходы на оплату труда разработчиков, тестировщиков, технической поддержки, приобретение лицензии на программное обеспечение и оборудование.

Затраты на внедрение: Возможные затраты на интеграцию системы в существующую инфраструктуру организации, обучение персонала и техническое сопровождение.

Ожидаемые выгоды: Внедрение системы аутентификации обеспечит безопасность и защиту данных пользователей, что может снизить вероятность несанкционированного доступа и потенциальные убытки от утечек данных. Также

		Бильяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
						6
Изм.	Лист	№ докум	Подпись	Дата		

это повысит доверие пользователей к системе и улучшит общий уровень безопасности на предприятии.

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
						7
Изм.	Лист	№ докум	Подпись	Дата		

2 ПОСТАНОВКА ЗАДАЧИ И ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Разработка приложения для аутентификации пользователей направлена на создание безопасного и удобного механизма управления доступом к информационным системам организации. Ниже приводится детальное описание всех аспектов постановки задачи, включая цели, информационные потребности, требования к оборудованию и программному обеспечению, а также поэтапный план создания продукта.

Цели приложения

- 1) *Обеспечение безопасности:* Предотвращение несанкционированного доступа к системе с помощью надёжной аутентификации пользователей.
- 2) *Удобство использования:* Создание интуитивно понятного интерфейса для удобства работы конечных пользователей.
- 3) *Снижение рисков:* Минимизация рисков утечек данных и улучшение общей безопасности на уровне каждого отдельного ПК пользователя.
- 4) *Автономность:* Обеспечение возможности работы приложения без подключения к серверу через локальное хранение базы данных.

Информационные потребности

Для разных отделов и служб организации приложение должно обеспечивать эффективное управление пользователями и доступ к их учётным записям:

Отдел IT:

- 1) *Управление локальными базами данных пользователей и учётных записей.*
- 2) *Мониторинг доступа и активностей пользователей на каждом ПК.*

Кадровый отдел:

- 1) *Управление доступом новых сотрудников на их выделенных рабочих местах.*
- 2) *Обеспечение безопасного доступа при увольнении сотрудников, локально на каждом их ПК.*

Пользователи:

- 1) *Возможность безопасного входа в систему со своего рабочего ПК.*
- 2) *Самостоятельное изменение пароля, хранящегося в локальной базе данных.*

		Бильяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
						8
Изм.	Лист	№ докум	Подпись	Дата		

Требования к оборудованию и программному обеспечению

Оборудование

Рабочие станции: ПК и ноутбуки, на которых будет производиться аутентификация и локально храниться база данных.

Программное обеспечение

- 1) Язык программирования: C#.*
- 2) Среда разработки: Visual Studio.*
- 3) Система управления базами данных: Access*
- 4) Локальная инфраструктура: Установка и настройка баз данных на каждом отдельном ПК.*

План поэтапного создания продукта

1) Анализ требований:

- 1) Сбор и анализ требований к системе от различных отделов и служб.*
- 2) Определение функциональных и нефункциональных требований.*

2) Проектирование:

- 1) Разработка архитектуры системы.*
- 2) Проектирование структуры локальной базы данных.*
- 3) Создание прототипов пользовательского интерфейса.*

3) Разработка:

- 1) Разработка базовой функциональности: регистрация и аутентификация пользователей.*
- 2) Разработка интерфейса для работы с локальной базой данных*
- 3) Интеграция с локальной СУБД.*

4) Тестирование:

- 1) Юнит-тестирование отдельных модулей*
- 2) Интеграционное тестирование системы на каждом отдельном ПК*
- 3) Тестирование безопасности.*

5) Внедрение:

- 1) Развертывание системы на каждом рабочем ПК.*

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
						9
Изм.	Лист	№ докум	Подпись	Дата		

2) Настройка и инициализация локальных баз данных.

3) Пилотное внедрение и сбор отзывов.

6) Поддержка и обслуживание:

1) Мониторинг работы системы на каждом ПК.

2) Регулярные обновления и исправление ошибок.

3) Поддержка пользователей и обучение.

Информация, содержащаяся в документации

Приложение должно иметь подробную документацию, включающую следующее:

1) Руководство пользователя:

2) Как зарегистрироваться и войти в систему на своём ПК.

3) Как изменять личные данные и пароли, хранящиеся локально.

4) Процесс аутентификации и частые ошибки, с которыми может сталкиваться пользователь.

Руководство администратора:

1) Управление учетными записями пользователей на каждом отдельном ПК.

2) Настройка системы безопасности локально на каждом ПК.

3) Мониторинг и отчётность о доступах и активности пользователей.

Техническая документация:

1) Описание архитектуры системы.

2) Схемы локальной базы данных.

3) Руководство по установке и настройке базы данных на ПК.

Информация, которую пользователь будет получать

При удачной аутентификации:

Сообщение “Логин и пароль совпадают!”. Пользователь получает доступ к своим данным на своём рабочем ПК.

При ошибках в аутентификации:

Сообщение “Неверный логин или пароль”.

Сообщение “Не введен логин или пароль”, если хотя бы одно из полей не заполнено.

		Бильяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		10

3 РАЗРАБОТКА ПРИЛОЖЕНИЯ

3.1 Разработка технического задания:

Для разработки ТЗ будет использован ГОСТ 34.602.89, который включает в себя:

1) общие сведения;

1.1) Полное наименование системы и её условное обозначение;

1.2) Плановые сроки начала, и окончания работы по созданию системы;

2) Назначение и цели создания (развития) системы;

2.1) Назначение системы;

2.2) Цели создания системы;

3) Характеристика объектов автоматизации;

3.1) Краткие сведения об объекте автоматизации;

3.2) Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды;

4) Требования к системе;

4.1) Требования к функциям (задачам), выполняемой системой;

4.2) Требования к видам обеспечения.

5) Состав и содержание работ по созданию системы;

6) Порядок контроля и приемки системы;

7) Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;

8) Требования к документированию;

9) Источники разработки.

1) Общие сведения

1.1) Полное наименование системы и её условное обозначение

Полное наименование: Система аутентификации пользователей для локальной базы данных.

Условное обозначение: САУЛБД (Система Аутентификации Пользователей с Локальной Базой Данных).

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		11

1.2) Плановые сроки начала и окончания работы по созданию системы

Срок начала работ: 2 марта 2024 года.

Срок окончания работ: 10 марта 2024 года.

2) Назначение и цели создания (развития) системы

2.1) Назначение системы

Система предназначена для обеспечения безопасного и удобного входа пользователей в локальные информационные системы предприятия через аутентификацию по логину и паролю. Система будет реализована на клиентском ПК с локальной базой данных.

2.2) Цели создания системы

Обеспечение высокоуровневой безопасности доступа к локальным данным.

Повышение удобства использования для сотрудников.

Увеличение защиты от несанкционированного доступа.

Обеспечение автономной и стабильной работы приложения независимо от серверного окружения.

3) Характеристика объектов автоматизации

3.1) Краткие сведения об объекте автоматизации

Объект автоматизации — процесс аутентификации пользователей на рабочих станциях предприятия. Основная функциональность включает проверку введенных логина и пароля по данным, хранящимся в локальной базе данных SQLite.

3.2) Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды

Условия эксплуатации: Рабочие станции предприятия.

Окружающая среда: Температура 10–35°C, влажность 20–80%, нормальные офисные условия. Система должна быть устойчивой к кратковременным отключениям электроэнергии и непредвиденным сбоям в работе ПК.

4) Требования к системе

4.1) Требования к функциям (задачам), выполняемой системой

Проверка наличия введенных логина и пароля.

		Бильяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		12

Сверка введенных данных с записями в локальной базе данных.

Вывод сообщения о результате аутентификации (успешная/неуспешная аутентификация, отсутствие данных).

Поддержка изменения пароля пользователем.

Обеспечение безопасности данных при хранении и передаче.

4.2 Требования к видам обеспечения

Программное обеспечение: Разработка на языке C# с использованием Visual Studio, база данных – SQLite.

Аппаратное обеспечение: Рабочие станции (ПК, ноутбуки) с ОС Windows 10 и выше.

5) Состав и содержание работ по созданию системы

Анализ требований.

Проектирование системы.

Разработка прототипов и интерфейсов.

Реализация функциональности.

Тестирование и отладка программы.

Развертывание системы на клиентских ПК.

Сбор отзывов и доработка.

Поддержка и обслуживание.

6) Порядок контроля и приемки системы

Контроль: Планируется проведение промежуточных проверок на всех этапах разработки.

Приемка: Финальное тестирование и приемка системы по завершении разработки с оценкой соответствия требованиям техзадания.

7) Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Установка и настройка SQLite на каждом ПК.

Установка клиента приложения.

Проведение обучения пользователей и администраторов.

8) Требования к документированию

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		13

Руководство пользователя.

Руководство администратора.

Техническая документация.

Документы по установке и настройке системы.

9) Источники разработки

Материалы по методологии разработки по ГОСТ 34.602.89.

Руководства и справочники по C# и .NET Framework.

Документация по SQLite.

Внутренние требования и стандарты предприятия.

3.2 Прототипирование

Основные шаги прототипирования

1) Анализ функционала программы

На этом этапе осуществляется проработка всех аспектов функционала модуля аутентификации:

Заполнение полей логина и пароля.

Проверка ввода данных.

Обработка результатов аутентификации.

Вывод сообщений в зависимости от исхода проверки.

2) Разработка схем экранов приложения

UI, интерфейс пользователя, он же пользовательский интерфейс (UI — англ. user interface) — интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы.

Для модуля аутентификации потребуется несколько ключевых экранов:

Экран логина: включает поля для ввода логина и пароля и кнопку "Вход".

1) Экран успешной аутентификации: отображает сообщение "Логин и пароль совпадают!".

2) Экран неудачной аутентификации: отображает сообщение "Неверный логин

		Бильяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		14

или пароль”.

3) Экран ошибки ввода: отображает сообщение “Не введен логин или пароль”, если поле логина или пароля не заполнено.

3) Структурирование экранов и переходов между ними

Разрабатывается схема переходов между экранами в зависимости от пользовательских действий и результатом аутентификации. Эта схема служит для понимания логики работы приложения и тестирования пользовательского пути:

1) Пользователь открывает приложение и видит экран логина.

2) Вводит логин и пароль, нажимает кнопку “Вход”.

3) Если оба поля не заполнены, приложение переходит на экран ошибки ввода.

4) Если введены логин и пароль, приложение проверяет данные.

5) Если данные корректны, выводит на экран сообщение об успешной аутентификации.

6) Если данные некорректны, выводит на экран сообщение о неудачной аутентификации.

Прототипирование UI (User Interface)

Для наглядного представления и оценки пользовательского интерфейса создаются визуальные макеты, представляющие ключевые экраны и элементы управляющих компонентов:

1) Экран логина

Поле ввода логина.

Поле ввода пароля.

Кнопка “Вход”.

Кнопка “Забыли пароль?” (при необходимости).

2) Экран успешной аутентификации

Сообщение “Логин и пароль совпадают!”.

Кнопка “ОК”.

3) Экран неудачной аутентификации

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		15

Сообщение "Неверный логин или пароль".

Кнопка "Попробовать снова".

4) Экран ошибки ввода

Сообщение "Не введен логин или пароль".

Кнопка "ОК".

Итоги прототипирования

На этом этапе происходит окончательная проработка всех интерфейсных элементов и их расположения, а также структурирование переходов между экранами. Прототипирование помогает визуализировать взаимодействие пользователя с приложением, выявить возможные проблемы и оптимизировать пользовательский путь. Это обеспечивает понимание того, как будет выглядеть и функционировать модуль аутентификации, прежде чем приступить к его непосредственной разработке.

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		16

4 РЕАЛИЗАЦИЯ

4.1 Выбор и обоснование IDE

Для разработки приложения была выбрана среда разработки Microsoft Visual Studio. Обоснование этого выбора включает следующие аспекты:

- 1) Интеграция с .NET Framework: Visual Studio обеспечивает тесную интеграцию с .NET Framework, что облегчает разработку приложений на языке C# и использование встроенных библиотек и классов.
- 2) Мощные инструменты для отладки и тестирования: Visual Studio предоставляет удобные и мощные инструменты для отладки, профилирования и юнит-тестирования, позволяя быстро находить и исправлять ошибки.
- 3) Поддержка различных языков и технологий: Помимо C#, Visual Studio поддерживает множество других языков и технологий, что делает её универсальным инструментом для различных видов разработок.
- 4) Интуитивный интерфейс: Удобный и интуитивно понятный интерфейс снижает время на освоение и позволяет быстро переходить к продуктивной работе.

4.2 Разработка логики приложения

Основной логикой приложения является процесс аутентификации пользователя. Эта логика включает следующие основные этапы:

- 1) Проверка ввода данных:
 - Проверка на наличие введенных логина и пароля.
 - Подача соответствующих сообщений при отсутствии данных.
- 2) Сверка данных с базой данных:
 - Взаимодействие с локальной базой данных Access.
 - Проверка совпадения введенных логина и пароля с сохраненными в базе данных учетными записями.
- 3) Обработка результатов аутентификации:
 - Если введенные данные совпадают с данными в базе, вывод сообщения об успешной аутентификации.
 - Если данные не совпадают, вывод сообщения об ошибке.

		Бильяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		17

- Если данные отсутствуют, вывод сообщения о необходимости ввода данных.

4.3 Написание кода

Разработка кода приложения включает следующие шаги:

1. Создание базы данных:

Подготовка таблицы Аккаунты в локальной базе данных Access с полями для логина и пароля.

2. Взаимодействие с базой данных:

- Написание кода для подключения к базе данных Access.
- Разработка функций для выполнения SQL-запросов по проверке логина и пароля.

3. Основная логика аутентификации:

- Написание функций для проверки введенных данных.
- Реализация обработки результатов аутентификации и генерация соответствующих сообщений.

Пример кода для проверки логина и пароля:

```
using System;
```

```
using System.Data.OleDb;
```

```
using System.Windows;
```

```
using System.Windows.Controls;
```

```
namespace MEDICINSKI_CENTRE
```

```
{
```

```
    /// <summary>
```

```
    /// Логика взаимодействия для Login.xaml
```

```
    /// </summary>
```

```
    public partial class Login : Window
```

```
    {
```

```
        public static string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=MedCenter.accdb";
```

```
        private OleDbConnection Connection;
```

		Бильяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		18

```

public Login()
{
    InitializeComponent();
    Connection = new OleDbConnection(connectString);
}

private void LoginEnter_TextChanged(object sender, TextChangedEventArgs e)
{
}

private void LoginEnter_Click(object sender, RoutedEventArgs e)
{
    string login = LoginTxt.Text;
    string password = PassTxt.Password;

    if (string.IsNullOrEmpty(login) || string.IsNullOrEmpty(password))
    {
        MessageBox.Show("Не введен логин или пароль");
        return;
    }

    OleDbCommand command = new OleDbCommand("SELECT * FROM Аккаунты
WHERE (Логин = @Login) AND Пароль = @Password", Connection);
    command.Parameters.AddWithValue("@Login", login);
    command.Parameters.AddWithValue("@Password", password);

    try
    {
        if (Connection.State != System.Data.ConnectionState.Open)

```

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		19

```

    {
        Connection.Open();
    }

    OleDbDataReader reader = command.ExecuteReader();

    if (reader.HasRows)
    {
        MessageBox.Show("Логин и пароль совпадают!");
    }
    else
    {
        MessageBox.Show("Не верный логин или пароль");
        return;
    }

    reader.Close();
}
catch (Exception ex)
{
    MessageBox.Show($"Произошла ошибка: {ex.Message}");
}
finally
{
    if (Connection.State == System.Data.ConnectionState.Open)
    {
        Connection.Close();
    }
}

```

}

}

}

}

4.4 Создание UI

Создание пользовательского интерфейса включает разработку всех необходимых экранов и элементов управления для удобного взаимодействия пользователей с приложением.

1. Экран логина:

- Поля для ввода логина и пароля.
- Кнопка "Вход".
- Сообщение об ошибке или успехе под формой ввода.

2. Экран сообщений:

- Модальные окна для вывода сообщений пользователю (например, "Логин и пароль совпадают!" или "Не введен логин или пароль").

3. Кнопка показать или спрятать пароль нужна с целью обеспечения конфиденциальности пользователя.

Пример кода для создания простого UI с использованием WPF:

```
<Window x:Class="MEDICINSKI_CENTRE.Login"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:MEDICINSKI_CENTRE"
    mc:Ignorable="d"
    Title="Login" Height="350" Width="400">
    <Grid>
```

<TextBlock VerticalAlignment="Top" HorizontalAlignment="Center" FontSize="24"
Margin="0,43,0,0">

<Run Language="ru-ru" Text="Boïmu"/>

</TextBlock>

<TextBox x:Name="LoginTxt" HorizontalAlignment="Center" TextWrapping="Wrap"
VerticalAlignment="Top" Width="120" TextChanged="LoginEnter_TextChanged"
Margin="0,124,0,0"/>

<PasswordBox x:Name="PassTxt"
PasswordChanged="PasswordInput_PasswordChanged" HorizontalAlignment="Center"
Margin="0,167,0,0" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="PasswordText" Height="18" VerticalAlignment="Top"
Visibility="Collapsed" Width="120" HorizontalAlignment="Center" Margin="0,167,0,0"/>

<CheckBox x:Name="PasswordShow" Content="Показать пароль"
Margin="0,214,0,0" HorizontalAlignment="Center" VerticalAlignment="Top"
Checked="ShowPassword_Checked" Unchecked="ShowPassword_Unchecked"/>

<TextBlock HorizontalAlignment="Center" Margin="0,103,0,0"
TextWrapping="Wrap" VerticalAlignment="Top">

<Run Language="ru-ru" Text="Логин"/>

</TextBlock>

<TextBlock HorizontalAlignment="Center" Margin="0,147,0,0"
TextWrapping="Wrap" VerticalAlignment="Top" Height="16">

<Run Language="ru-ru" Text="Пароль"/><LineBreak/><Run Language="ru-
ru"/>

</TextBlock>

<Button x:Name="LoginEnter" Content="Boïmu" HorizontalAlignment="Center"
Margin="0,245,0,0" VerticalAlignment="Top" Click="LoginEnter_Click" Height="48"
Width="120"/>

<TextBlock HorizontalAlignment="Center" Margin="0,258,0,0"
TextWrapping="Wrap" VerticalAlignment="Top">

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		22

</TextBlock>

</Grid>

</Window>

5 ТЕСТИРОВАНИЕ

5.1 Первичное тестирование.

Модуль для аутентификации проходит тщательное тестирование для выявления всех возможных ошибок и недочетов. Для этого создается таблица проверок, в которой указываются и подробно описываются все обнаруженные ошибки. В процессе проектирования невозможно предусмотреть все погрешности реальной эксплуатации, поэтому на этапе тестирования выявляются:

1) Функциональные ошибки:

- Ошибки в логике аутентификации.
- Некорректная обработка ввода данных.
- Неправильное взаимодействие с базой данных.

2) Интерфейсные ошибки:

- Несоответствия макетов экранов дизайну.
- Неправильное расположение или неработоспособность кнопок и элементов управления.
- Неудобства в использовании интерфейса.

№	Описание ошибки	Категория	Детали	Сроки устранения	Итог
1	Не выводится сообщение об ошибке	Интерфейсная	Не отображается сообщение “Неверный логин или пароль”	1 день	Исправлено
2	Ввод пустых данных	Функциональная	В случае ввода пустого логина или пароля не	1 день	Исправлено

			<i>генерируется сообщение</i>		
3	<i>Неправильная сверка данных</i>	<i>Функциональная</i>	<i>Неправильная сверка логина и пароля с базой данных</i>	<i>2 дня</i>	<i>Исправлено</i>
4	<i>Проблемы с подключением к базе</i>	<i>Функциональная</i>	<i>Ошибки при подключении к Access базе данных</i>	<i>1 день</i>	<i>Исправлено</i>

5.2 Повторное тестирование

После исправления всех выявленных ошибок приложение устанавливается на тестовые устройства или тестируется при помощи эмуляторов. Повторное тестирование проводится для проверки исправлений и функциональности приложения в реальных условиях эксплуатации.

1) Повторная проверка логики приложения:

- Аутентификация пользователя с правильными и неправильными данными.*
- Поведение приложения при пустом вводе данных.*
- Сравнение результатов тестирования с ожидаемыми результатами.*

2) Проверка работы с базой данных:

- Успешное подключение к локальной базе данных.*
- Корректная работа функций чтения и записи.*

3) Тестирование в разных условиях:

- Работа приложения при различных версиях операционной системы.*
- Проверка поведения приложения при отключении от интернета (если применимо).*
- Тестирование на устройствах с различными конфигурациями аппаратного обеспечения.*

№	Описание проверки	Условия тестирования	Результат проверки	Изменения
1	Логин и пароль совпадают	Ввод корректных данных	Успех, вход выполнен	Нет
2	Неверный логин или пароль	Ввод некорректных данных	Ошибка, сообщение отображается	Нет
3	Пустой логин и пароль	Ввод пустых полей	Ошибка, сообщение отображается	Нет
4	Проверка подключения к базе данных	Имитация отключения базы	Ошибка, сообщение отображается	Нет
5	Разработка на различных ОС	Windows 10, Windows 11, разные версии	Успех, корректная работа	Нет
6	Проверка многопользовательского режима	Несколько пользователей одновременно	Успех, корректная работа	Нет

5.3 Итоги тестирования

После проведения как первичного, так и повторного тестирования, был создан перечень обнаруженных проблем и недочетов с указанием сроков на их устранение. После устранения всех проблем приложение пересобирается и выпускается с исправленными ошибками и при необходимости с измененным функционалом. Это документируется в обновлённой таблице проверок, которая включает информацию о всех исправлениях и итогах тестирования.

Такая структура тестирования обеспечивает детальный анализ работы приложения и позволяет удостовериться в его надежности и соответствии заданным требованиям перед выпуском.

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
						27
Изм.	Лист	№ докум	Подпись	Дата		

6 СОЗДАНИЕ СБОРКИ

После завершения разработки и тестирования приложения, следующим этапом является создание сборки программного продукта для установки на конечные устройства пользователей. Этот процесс включает в себя оценку, внесение необходимых исправлений, упаковку приложения и размещение проекта на репозитории для последующего использования.

Программный продукт

1) Оценка выполнения требований

После завершения реализации приложение проходит окончательную оценку, чтобы удостовериться, что все предъявленные требования выполнены:

- **Функциональность:** Все функции аутентификации реализованы в соответствии с требованиями.
- **Интерфейс:** Пользовательский интерфейс соответствует разработанным макетам и является интуитивно понятным.
- **Безопасность:** Приложение корректно обрабатывает все возможные сценарии ввода данных и обеспечивает безопасность используемых данных.

2) Внесение изменений

При обнаружении отклонений от требований или выявлении дополнительных улучшений, в код вносятся необходимые изменения.

3) Создание сборки

- **Упаковка приложения:** Приложение пакуется в исполнимый файл (.exe), который можно запускать на устройствах конечных пользователей.
- **Тестирование сборки:** Перед финальным выпуском сборка проходит дополнительное тестирование на реальных устройствах для проверки корректности установки и работы.

4. Размещение на веб-сервисе GitHub

Проект размещается на GitHub или аналогичном веб-сервисе репозитория для обеспечения доступа к исходному коду, а также удобного отслеживания версий и изменений.

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		28

Пример кода ключевых элементов

Для наглядности в приложении пояснительной записки приводятся коды ключевых элементов с комментариями, поясняющими работу приложения. Код для проверки логина и пароля:

```
using System;
```

```
using System.Data.OleDb;
```

```
using System.Windows;
```

```
using System.Windows.Controls;
```

```
namespace MEDICINSKI_CENTRE
```

```
{
```

```
    /// <summary>
```

```
    /// Логика взаимодействия для Login.xaml
```

```
    /// </summary>
```

```
    public partial class Login : Window
```

```
    {
```

```
        public static string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=MedCenter.accdb";
```

```
        private OleDbConnection Connection;
```

```
        //подключение к БД с ссылкой на файл БД
```

```
        public Login()
```

```
        {
```

```
            InitializeComponent();
```

```
            Connection = new OleDbConnection(connectionString);
```

```
        }
```

```
        private void ShowPassword_Checked(object sender, RoutedEventArgs e)
```

```
        {
```

```
            PasswordText.Text = PassTxt.Password;
```

```
            PassTxt.Visibility = Visibility.Collapsed;
```

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
Изм.	Лист	№ докум	Подпись	Дата		29

```

        PasswordText.Visibility = Visibility.Visible;
    }

    //маска для ввода пароля 1 если нажата галочка на видимость пароля
    маска убирается
    private void ShowPassword_Unchecked(object sender, RoutedEventArgs e)
    {
        PassTxt.Password = PasswordText.Text;
        PassTxt.Visibility = Visibility.Visible;
        PasswordText.Visibility = Visibility.Collapsed;
    }

    //маска для ввода пароля 2 галочка не нажата, пароль скрыт маской
    private void PasswordInput_PasswordChanged(object sender, RoutedEventArgs
e)
    {
        if (PassTxt.Visibility == Visibility.Visible)
        {
            PasswordText.Text = PassTxt.Password;
        }
    }

    //если убрана маска пароля то его становится видно
    private void LoginEnter_TextChanged(object sender, TextChangedEventArgs e)
    {
    }

    private void LoginEnter_Click(object sender, RoutedEventArgs e)
    {
        string login = LoginTxt.Text;
        string password = PassTxt.Password;

        if (string.IsNullOrEmpty(login) || string.IsNullOrEmpty(password))

```

```

    {
        MessageBox.Show("Не введен логин или пароль");
        return;
    }

    //проверка на введен логин и пароль или нет

    OleDbCommand command = new OleDbCommand("SELECT * FROM Аккаунты
WHERE (Логин = @Login) AND Пароль = @Password", Connection);
    command.Parameters.AddWithValue("@Login", login);
    command.Parameters.AddWithValue("@Password", password);
    //запрос к базе данных на проверку данных логина и пароля, сделана
через параметры с целью защиты от SQL инъекций
    try
    {
        if (Connection.State != System.Data.ConnectionState.Open)
        {
            Connection.Open();
        }

        //проверка на открытое подключение, если не открыто то
открывается

        OleDbDataReader reader = command.ExecuteReader();

        //команда ридера который читает наши данные и сверяет их с
данными в базе

        if (reader.HasRows)
        {
            MessageBox.Show("Логин и пароль совпадают!");
        }
    }

```

```

    }
    //проверка данных завершена и данные совпали
    else
    {
        MessageBox.Show("Не верный логин или пароль");
        return;
    }
    // проверка завершена и данные неверны
    reader.Close();
    //закрытие ридера
}
catch (Exception ex)
{
    MessageBox.Show($"Произошла ошибка: {ex.Message}");
}
//вывод сообщения об ошибках :P
finally
{
    if (Connection.State == System.Data.ConnectionState.Open)
    {
        Connection.Close();
    }
}
//проверка, если подключение открыто то закрывает и тем самым
завершает процесс входа
}
}
}

```


Ссылка на проект GitHub

Проект размещен на веб-сервисе GitHub для удобного доступа и отслеживания изменений. Ссылка на проект:

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
						33
Изм.	Лист	№ докум	Подпись	Дата		

Заключение

В ходе выполнения курсовой работы по разработке модуля аутентификации пользователей для локальной базы данных были достигнуты следующие результаты и выполнены поставленные задачи:

- 1. Актуальность работы: Подтверждена значимость разработки системы аутентификации для обеспечения безопасности и защиты данных пользователей.*
- 2. Выбор технологий: Язык программирования C# и среда разработки Microsoft Visual Studio выбраны на основании их широкого использования, мощных инструментов и тесной интеграции с .NET платформой.*
- 3. Проектирование и реализация: Разработана и реализована логика аутентификации пользователей с использованием локальной базы данных Access. Были созданы и протестированы основные экраны интерфейса и бизнес-логика приложения.*
- 4. Тестирование и улучшения: Приложение прошло этапы первичного и повторного тестирования, в ходе которых выявлены и устранены ошибки. Проведено финальное тестирование для подтверждения соответствия требованиям.*
- 5. Создание сборки: Приложение упаковано в исполнимый файл и создан инсталлятор для удобной установки пользователями. Проект размещен на GitHub для отслеживания изменений и предоставления доступа сообществу разработчиков.*

Общий вывод по теме работы

Разработка модуля аутентификации пользователей для локальных баз данных является важным шагом на пути обеспечения безопасности доступа к информационным системам различных организаций. Созданное приложение является надежным и удобным инструментом для пользователей, предоставляя необходимые функции для безопасной аутентификации.

Возможности внедрения программного продукта

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
						34
Изм.	Лист	№ докум	Подпись	Дата		

Программный продукт может быть успешно внедрен в различных организациях, направленных на повышение безопасности и оптимизацию управления доступом к системам. Возможности использования включают:

- Корпоративные системы: Обеспечение безопасного доступа сотрудников к конфиденциальным данным и внутренним ресурсам компании.
 - Образовательные учреждения: Управление доступом студентов и преподавателей к учебным ресурсам и системам, требующим аутентификации.
 - Медицинские учреждения: Защита медицинских данных и учетных записей пациентов для соблюдения конфиденциальности и безопасности информации.
- Направления дальнейшего совершенствования

1. Расширение функциональности: Внедрение дополнительных методов аутентификации, таких как двухфакторная аутентификация (2FA), биометрическая аутентификация, или использование аппаратных токенов для повышения уровня безопасности.
2. Интеграция с облачными сервисами: Разработка возможностей для работы с удаленными базами данных и интеграция с облачными платформами, что позволит использовать модуль в крупных распределенных системах.
3. Поддержка кросс-платформенности: Адаптация приложения для работы на различных операционных системах и устройствах, включая мобильные платформы.
4. Оптимизация пользовательского интерфейса: Проведение дополнительных исследований и улучшение UI/UX для повышения удобства и эффективности взаимодействия пользователей с приложением.
5. Мониторинг и аналитика: Внедрение инструментов для мониторинга использования приложения и сбора аналитических данных для дальнейшего улучшения безопасности и производительности.

Таким образом, созданный модуль аутентификации является мощным и гибким решением для обеспечения безопасности доступа к системам, и его дальнейшее

развитие и совершенствование откроет новые возможности для применения и улучшения безопасности в различных сферах деятельности.

		Бильтяев Д.А.			ПМ.02 09.02.07.00.31.03	Лист
						36
Изм.	Лист	№ докум	Подпись	Дата		

Список использованных источников

1. Документация по C# и .NET Framework:
 - Microsoft Docs. C# Guide. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/>
2. Документация Visual Studio:
 - Microsoft Docs. Visual Studio Documentation. URL: <https://docs.microsoft.com/en-us/visualstudio/>
3. Методические рекомендации по разработке программных систем, соответствующих ГОСТ 34.602.89:
 - Государственный стандарт Союза ССР. Единая система программной документации (ЕСПД).
4. GitHub:
 - Официальный репозиторий. URL: <https://github.com/>
5. Статьи и руководства по разработке пользовательских интерфейсов:
 - UX Design. Basic Principles of UX/UI Design. URL: <https://uxdesign.cc/>
6. Учебные ресурсы по программированию на C#:
 - W3Schools. C# Tutorial. URL: <https://www.w3schools.com/cs/>
7. Руководства по безопасности данных и аутентификации:
 - OWASP (Open Web Application Security Project). URL: <https://owasp.org/>

