

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1 Теоретические основы разработки приложения

1.1 Постановка задачи

Цель дипломного проекта состоит в разработке приложения «Телефонный справочник», которое будет служить средством для эффективного управления контактами, которые есть у пользователя. Основной задачей приложения «Телефонный справочник» является предоставление пользователям интуитивного и удобного инструмента для взаимодействия с его контактами. Оно должно обеспечивать возможность создания, редактирования, чтение и удаление контактов.

Основными функциями приложения будут:

- добавление новых контактов в справочник;
- чтение существующих контактов;
- редактирование и удаление контактов;
- создание, удаление и редактирование пользователей.

Среди ключевых характеристик приложения будет особое внимание уделено его простоте и удобству использования, что является важным аспектом для привлечения и удержания пользователей, желающих иметь под рукой эффективное и функциональное средство для организации своих контактов.

1.2 Способы решения задачи. Преимущества и недостатки

Разработка приложения «Телефонного справочника» представляет собой комплексное задание, требующее внимательного выбора технологического подхода. Ниже представлен подробный анализ различных способов решения задачи с учетом их преимуществ и ограничений.

Десктопное приложение с использованием Windows Forms или WPF.

Этот путь предполагает создание приложения для десктопных устройств под управлением операционной системы Windows. Технологии Windows Forms или WPF позволяют разработать интуитивно понятный интерфейс, полностью интегрированный с Windows.

Преимущества:

- интерфейс и функциональность. Возможность создания насыщенного интерфейса с использованием разнообразных элементов управления, обеспечивающих комфортное взаимодействие пользователя с приложением;

- производительность. Desktopные технологии обеспечивают высокую производительность, важную для работы с объемными данными, характерными для ежедневников;

- интеграция с Windows. Легкая интеграция с функционалом Windows, таким как календарь, напоминания и системные ресурсы.

Недостатки:

- ограниченность платформой. Приложение будет ограничено использованием только на устройствах под управлением Windows;

- переносимость. Затруднен перенос на другие платформы, что может стать проблемой в будущем.

Web приложение с использованием языка Java Script и библиотеки React.

Преимущества:

- высокая скорость разработки. JavaScript и React позволяют быстро создавать интерфейсы и функциональность приложений благодаря множеству готовых компонентов и библиотек;

- масштабируемость и гибкость. React обладает модульной архитектурой, что упрощает поддержку и масштабирование приложения, позволяет эффективно работать с большим объемом данных;

- производительность. Благодаря виртуальному DOM и эффективному рендерингу компонентов, React обеспечивает высокую скорость работы приложения.

Недостатки:

- сложность изучения. Для работы с React необходимо иметь определенные знания и опыт в программировании, что может потребовать времени на обучение;

- ограниченность функциональности. В некоторых случаях нативные приложения могут предоставлять больший набор функций и возможностей по сравнению с web версией;

- зависимость от браузера. Использование JavaScript и React web приложения подразумевает работу в рамках браузера, что может ограничить возможности взаимодействия с операционной системой и аппаратурой устройства.

Приложение с использованием Xamarin или Flutter

Разработка кроссплатформенного приложения с использованием Xamarin (на C#) или Flutter (на Dart), позволяющего работать на Android и iOS.

Преимущества:

- кроссплатформенность. Единый код для обеих операционных систем, что обеспечивает широкий охват аудитории;

- язык программирования в Xamarin. Преимущество для разработчиков, знакомых с языком C# и платформой .NET;

- масштабируемость. Возможность простого масштабирования приложения для удовлетворения растущих потребностей.

Недостатки:

- ограниченные возможности нативных приложений. Некоторые возможности, доступные в нативных приложениях, могут быть ограничены в кроссплатформенных решениях;

- производительность. Некоторые задачи могут иметь ограниченную производительность по сравнению с нативными приложениями.

Web-приложение с использованием ASP.NET Core Blazor.

Разработка web-приложения с использованием технологии ASP.NET Core Blazor, которая позволяет пользователям взаимодействовать с приложением через web-браузер.

Преимущества:

- универсальный доступ. Возможность использования приложения из любого устройства с доступом в интернет, что гарантирует широкий круг пользователей;

- легкая обновляемость. Изменения в приложении вступают в силу незамедлительно без необходимости обновления на устройстве пользователя;
- интеграция с облачными сервисами. Простая интеграция с облачными сервисами для хранения данных и синхронизации.

Недостатки:

- зависимость от интернета. Работа приложения требует постоянного подключения к интернету;
- ограниченность доступа к системным ресурсам. Некоторые системные ресурсы устройства могут быть недоступны через браузер.

Кроссплатформенные фреймворки.

Unity – это мощный кроссплатформенный фреймворк, изначально ориентированный на создание игр, но также прекрасно подходящий для приложений различного назначения, включая ежедневники. Unity поддерживает не только мобильные платформы, такие как iOS и Android, но и десктопные системы Windows и macOS, а также web-приложения.

Преимущества:

- кроссплатформенность. Позволяет создавать приложения, которые могут работать на разных устройствах и операционных системах, что обеспечивает максимальное покрытие аудитории;
- мощный движок. Обладает широким спектром инструментов для создания красочных и интерактивных пользовательских интерфейсов.

Недостатки:

- нетрадиционное использование. Unity чаще ассоциируется с игровой разработкой, и его применение для создания ежедневника может показаться избыточным и усложнять процесс разработки.

1.3 Выбор алгоритма решения задачи и определение его сложности

При выборе алгоритма решения задачи необходимо учитывать следующие факторы:

- сложность задачи;
- ресурсы, необходимые для выполнения алгоритма;

- скорость выполнения алгоритма.

Сложность задачи разработки настольного приложения «Телефонный справочник» на языке С# с использованием фреймворка WPF заключается в обеспечении функциональности, позволяющей пользователям вести свой справочник, эффективно взаимодействовать с интерфейсом приложения и хранить, редактировать информацию о своих контактах.

Для успешного решения этой задачи требуется:

- определить необходимый функционал и структуру приложения "Телефонный справочник". Например, приложение может содержать возможность создания, редактирования и удаления контактов, отображение списка контактов на определенную дату и т. д.;

- создайте компоненты WPF для каждой функциональности приложения. Например, компоненты для формы создания и редактирования контактов, компонент для отображения списка контактов и т. д.;

- реализовать логику взаимодействия между компонентами. Например, при создании нового контакта, данные должны сохраняться в состоянии приложения и отображаться на странице;

- оптимизировать код приложения и убедиться, что все компоненты работают корректно;

- определить сложность приложения. Сложность wpf-приложения "Телефонный справочник" на С# с использованием фреймворка WPF зависит от количества и сложности компонентов, используемых при создании приложения;

- протестировать приложение, чтобы убедиться, что оно работает корректно и соответствует требованиям;

- скомпилируйте приложение и используйте в операционной системе Windows.

Эффективность выполнения алгоритма играет ключевую роль в разработке приложения «Телефонный справочник». Более оперативный алгоритм обеспечивает быструю обработку пользовательских действий, что существенно влияет на отзывчивость и общую производительность

приложения. Приложение может иметь множество пользователей, каждый из которых будет вносить изменения в свой справочник. Поэтому оптимальное время выполнения алгоритмов для обработки данных, хранения и отображения контента, а также реакции на пользовательские действия крайне важно.

Использование фреймворка WPF позволяет быстро создавать настольные приложения, что повышает производительность труда программиста. Кроме того программисту не нужно задумываться об низкоуровневых операциях которые должно выполнять приложение, за него это делает WPF.

Таким образом, эффективность выполнения алгоритмов в wpf-приложении "Телефонный справочник" на C# с использованием фреймворк WPF имеет принципиальное значение для обеспечения качественного пользовательского опыта и успешного функционирование приложения.

Важные аспекты в контексте легкости написание приложение выполнения включают:

- Сборщик мусора C#. Позволяет не задумываться об выделении и удалении динамической памяти;
- Не нужно учить низкоуровневые операции. Фреймворк WPF обеспечивает удобный интерфейс для использование функций операционной системы windows, таких как использование win api;
- Использование готовый элементов. Готовые элементы позволяют не писать повторно код;
- оптимизация базы данных. Проводите регулярное обслуживание базы данных, удаляя неиспользуемые данные, оптимизируя запросы и индексы, чтобы улучшить скорость выполнения запросов;
- доступность серверов. Обеспечьте хорошее качество хостинга и масштабируемость серверов, чтобы гарантировать быструю загрузку и обработку запросов;
- использование кэширования на стороне клиента. Используйте локальное хранилище браузера для кэширования данных и ускорения выполнения операций;

- минимизация зависимостей. Избегайте излишнего использования внешних библиотек и плагинов, которые могут замедлять работу приложения. Предпочтительно использовать только необходимые компоненты;

- асинхронная загрузка данных. Загружайте данные асинхронно, используя асинхронные запросы, чтобы ускорить отображение контента на странице.

Учитывая вышеперечисленные аспекты, можно значительно улучшить скорость выполнения работы wpf приложения "Телефонный справочник" и обеспечить удовлетворительный пользовательский опыт.

Оценка эффективности алгоритмов проводится с использованием следующих методов:

- тестирование. Проверка работы алгоритмов на различных объемах данных для оценки времени выполнения и использования памяти в реальных условиях;

- профилирование. Анализ времени выполнения каждой операции в алгоритме для выявления узких мест и возможных улучшений.

Оценка эффективности алгоритмов позволяет выбрать оптимальное решение для работы с данными в ежедневнике, обеспечивая высокую производительность и отзывчивость приложения.

1.4 Определение модели проектирования и технологии программирования

Модель проектирования – это архитектурный подход к разработке программного обеспечения, который определяет структуру и взаимодействие компонентов системы. Для разработки приложения для планирования и организации была выбрана модель проектирования «Waterfall Model» (каскадная модель или «водопад»).

«Waterfall Model» (каскадная модель или «водопад») – одна из самых старых, подразумевает последовательное прохождение стадий, каждая из которых должна завершиться полностью до начала следующей. В модели Waterfall легко управлять проектом. Благодаря её жесткости, разработка проходит быстро, стоимость и срок заранее определены. Но это палка о двух

концах. Каскадная модель дает отличный результат только в проектах с четко и заранее определенными требованиями и способами их реализации. Нет возможности сделать шаг назад, тестирование начинается только после того, как разработка завершена или почти завершена.

Продукты, разработанные по данной модели без обоснованного ее выбора, могут иметь недочеты (список требований нельзя скорректировать в любой момент), о которых становится известно лишь в конце из-за строгой последовательности действий. Стоимость внесения изменений высока, так как для ее инициализации приходится ждать завершения всего проекта. Тем не менее, фиксированная стоимость часто перевешивает минусы подхода. Исправление осознанных в процессе создания недостатков возможно, и, по нашему опыту, требует от одного до трех дополнительных соглашений к контракту с небольшим ТЗ.

С помощью каскадной модели можно создавать множество проектов «с нуля», включая разработку только ТЗ.

WPF приложение "Телефонный справочник", должно иметь простой и интуитивно понятный интерфейс, который позволит пользователям легко создавать и редактировать контакты. Необходимо предусмотреть меры безопасности для защиты от атак, таких как SQL-инъекции и не санкционированного доступа.

Выбор технологий программирования определяет, каким образом будет реализован функционал приложения. Для разработки приложения «Телефонный справочник» были выбраны следующие технологии:

- язык программирования: C#;
- среда разработки: Visual Studio предоставляет обширные возможности для работы с различными языками программирования, обеспечивая удобство отладки и разработки.

Преимущества и недостатки технологий зависят от конкретных требований проекта.

Преимущества C#:

- использование ООП подхода, что позволяет эффективно строить приложение;

- является быстрым языком, что позволяет создавать эффективные функции в приложении.

Недостатки C#:

- может быть сложным из-за большого количество инструментов языка;

- долгая компиляция проекта, что усложняет тестирование;

- может требовать больше ресурсов, чем более легковесные технологии.

Преимущества WPF:

- 1) набор готовых шаблонов для использования в приложении;

- 2) предоставляет удобный и эффективный способ организации кода с использованием компонентов, что облегчает разработку, тестирование и поддержку приложения.

- 3) большое сообщество разработчиков и хорошую документацию, что облегчает изучение и использование фреймворка;

Недостатки WPF:

- 1) нет кроссплатформенности. WPF приложение можно запустить только из под операционной системы windows, что ограничивает круг пользователей;

- 2) использует много памяти для простых приложений;

Для создания wpf-приложения "Телефонный справочник" на C# с использованием фреймворка WPF, мы можем использовать следующие технические шаги:

- 1) настройка окружения разработки:

- установка Dot net для работы с C# и WPF;

- создание нового проекта WPF с помощью готовой шаблона в Visual Studio.

- 2) создание компонентов:

- разработка компонентов для отображения списка записей справочника, добавления новой записи и редактирование существующей;

— создание компонентов для регистрации и авторизации пользователей.

— создание пользовательского интерфейса. Использование языка разметки xaml;

3) роутинг. Настройка маршрутизации приложения с помощью Frame для переключения между различными страницами.

4) взаимодействие с сервером базы данных. Работа с сервером для отправки и получения данных, используя ADO NET.

5) тестирование. Написание тестов для проверки корректности работы приложение.

6) сборка и развертывание. Сборка приложения с помощью платформы dot net.

Эти шаги помогут нам разработать функциональное настольные приложение "Телефонный справочник" на C# с использованием фреймворка WPF.

1.5 Выбор инструментальных средств

Программный продукт к данному проекту разработан на языке C#.

C# - это язык программирования, который применяется для создания web-приложений и настольных приложений. С его помощью можно создавать быстрые приложение и не задумываться о динамической памяти.

Основные понятия C#:

1) переменные. В C# переменная используется для хранения данных. Она типизированная, то есть может содержать только один тип данных. Например: числа, строки, булевы значения и другие типы данных;

2) условные операторы if, else и else if используются для выполнения различных действий в зависимости от условий;

3) циклы позволяют выполнять один и тот же блок кода несколько раз. В C# есть циклы foreach, for, while и do...while;

4) функции. Используются для организации повторяющихся действий в коде. Они позволяют уменьшить дублирование кода и сделать его более читаемым.

Инструменты C#:

1) консоль. Используется для вывода информации о переменных, отладки кода и выполнения различных команд.

2) Сборщик мусора. Позволяет не задумываться о выделении и освобождении динамической памяти. Самостоятельная работа с память приводит к утечкам памяти и ошибками доступа.

3) события. Позволяют отслеживать действия пользователя на странице, такие как клики мышью, нажатия клавиш и прокрутка страницы. События могут быть привязаны к элементам `html` с помощью обработчиков событий.

C# - мощный инструмент для создания быстрых и безопасных настольных приложений. Понимание основных понятий и инструментов C# поможет вам эффективно разрабатывать настольные приложения и улучшать пользовательский опыт в приложениях.

Есть несколько способов создания настольных приложений на C#:

1) использование фреймворков и библиотек, таких как WPF, Avalonia ui. Эти инструменты позволяют создавать сложные настольные приложения с помощью C#;

2) разработка настольных приложений с нуля с использованием XAML, C#. Вы можете создать простые настольные приложения, используя только базовые знания XAML и C#;

Dot net - это среда выполнения C#. Она позволяет запускать C# под операционной системой Windows, что делает Dot Net идеальным выбором для создания высокопроизводительных настольных приложений под Windows. Dot Net имеет множество возможностей:

1) высокая производительность. Поддержки много процессорной выполнении функций, приложение может эффективно использовать процессор ПК;

2) ООП Оно упрощает, ускоряет и делает разработку более согласованной;

3) Языковая поддержка. Dot Net поддерживает C#, F#, Visual Basic;

4) Универсальные стандарты Dot Net. Они упрощают разработки библиотек и компонентов для Dot Net;

5) кросс-платформенность. Dot Net поддерживается на большинстве операционных систем, что позволяет разрабатывать приложения на различных платформах (кроме WPF приложений, которые работают на старом Dot Net);

В целом, Dot Net предоставляет множество возможностей для создания эффективных и мощных настольных приложений.

WPF – это C# фреймворк для создания надежных и быстрых настольных приложений. Она позволяет разработчикам строить приложения, которые реагируют на изменения данных и отображают их на странице.

WPF основан на компонентной архитектуре, где интерфейс и реализация функционала разбиты на зависимые друг от друга компоненты.

WPF также поддерживает XAML – специальный язык разметки, который создавать красивые и понятные интерфейсы приложений. Благодаря своей простоте и эффективности, WPF стал популярным инструментом для создания настольных приложений.

При выборе инструментальных средств разработки настольных приложения на C# важно учитывать ряд факторов, таких как производительность, удобство использования, масштабируемость, сообщество разработчиков и интеграция с другими технологиями. Перед принятием решения нужно провести анализ требований к проекту, объема работы, опыта команды разработчиков.

Необходимо также учитывать экосистему инструментов, которые поддерживают выбранный фреймворк или библиотеку.

Важно также следить за актуальностью выбранных инструментов и их поддержкой сообществом разработчиков, чтобы обеспечить стабильность и безопасность разрабатываемого настольного приложения на C#.

2 Практическая реализация разработки приложения «Телефонный справочник»

2.1 Построение математической модели решения задачи

В этой главе будет рассмотрено построение математической модели, которая будет описывать функциональные требования и основные компоненты приложения «Телефонный справочник».

На сегодняшний день существует множество готовых решений и шаблонов для создания настольных приложений с использованием C#. Однако большинство из них ограничиваются простыми функциональными возможностями. Для создания более сложных приложений требуется построение уникальных математических моделей, которые будут определять структуру и функционал приложения.

Для успешного создания настольного приложения необходимо определить его структуру, функционал и взаимосвязи между компонентами. Для этого строится математическая модель, которая описывает все необходимые аспекты приложения. В данной работе будет использоваться теория графов, матриц и алгоритмы для построения математической модели веб-приложения на C# с использованием WPF.

После построения математической модели необходимо приступить к реализации ее на практике. Для этого используются средства программирования на C# и фреймворка WPF. В данной работе будет предложен алгоритм разработки настольного приложения на основе построенной математической модели, который включает в себя этапы создания компонентов, их взаимодействия, обработки данных и пользовательского интерфейса.

При построении математической модели решения задачи разработки веб-приложения на C# с использованием WPF необходимо учитывать несколько ключевых аспектов, которые помогут создать эффективное и функциональное приложение.

Сначала необходимо определить цель разработки приложения и его основные функциональные возможности. Это поможет понять, какие данные и

какие операции с ними необходимо реализовать. Далее следует провести анализ всех возможных вариантов решения задачи и выбрать наиболее оптимальный и эффективный.

Для успешной разработки настольного приложения на C# с использованием WPF необходимо создать структуру приложения, определить компоненты и их взаимодействие друг с другом. WPF и C# позволяет строить приложение из небольших, независимых компонентов, что облегчает разработку и поддержку кода.

Математическая модель разработки настольного приложения на C# может включать в себя определение основных алгоритмов обработки данных, формул расчета и методов взаимодействия с пользователем. Например, для создания интерактивных элементов на странице можно использовать различные математические вычисления и алгоритмы.

При разработке настольного приложения на WPF важно учитывать аспекты производительности и оптимизации кода. Модель решения задачи должна быть масштабируемой и гибкой, чтобы при необходимости можно было легко внести изменения и улучшения.

Также важно проводить тестирование приложения на всех этапах разработки, чтобы обнаружить и устранить возможные ошибки и недочеты. Тестирование позволяет убедиться в корректности работы приложения и его соответствии заявленным требованиям.

При разработке телефонного справочника важно учесть следующие основные требования:

- 1) возможность создания и управления пользователями. Приложение должно позволять создавать новых пользователей, редактировать и удалять их;
- 2) возможность создания и управления контактами. Приложение должно позволять пользователям создавать контакты, редактировать и удалять их;
- 3) доступность. Приложение должно быть доступно в любое время.

Построение математической модели решения задачи разработки настольного приложения на C# с использованием WPF является ключевым

этапом создания качественного и профессионального приложения. Систематический подход к разработке, учет всех особенностей технологии и правильный выбор методов и подходов помогут создать успешное приложение, которое будет удовлетворять потребности пользователей.

Последним шагом разработки математической модели является описание алгоритма в виде блок-схемы. В соответствии с рисунком 2.1 представлена блок-схема алгоритма работы приложения.

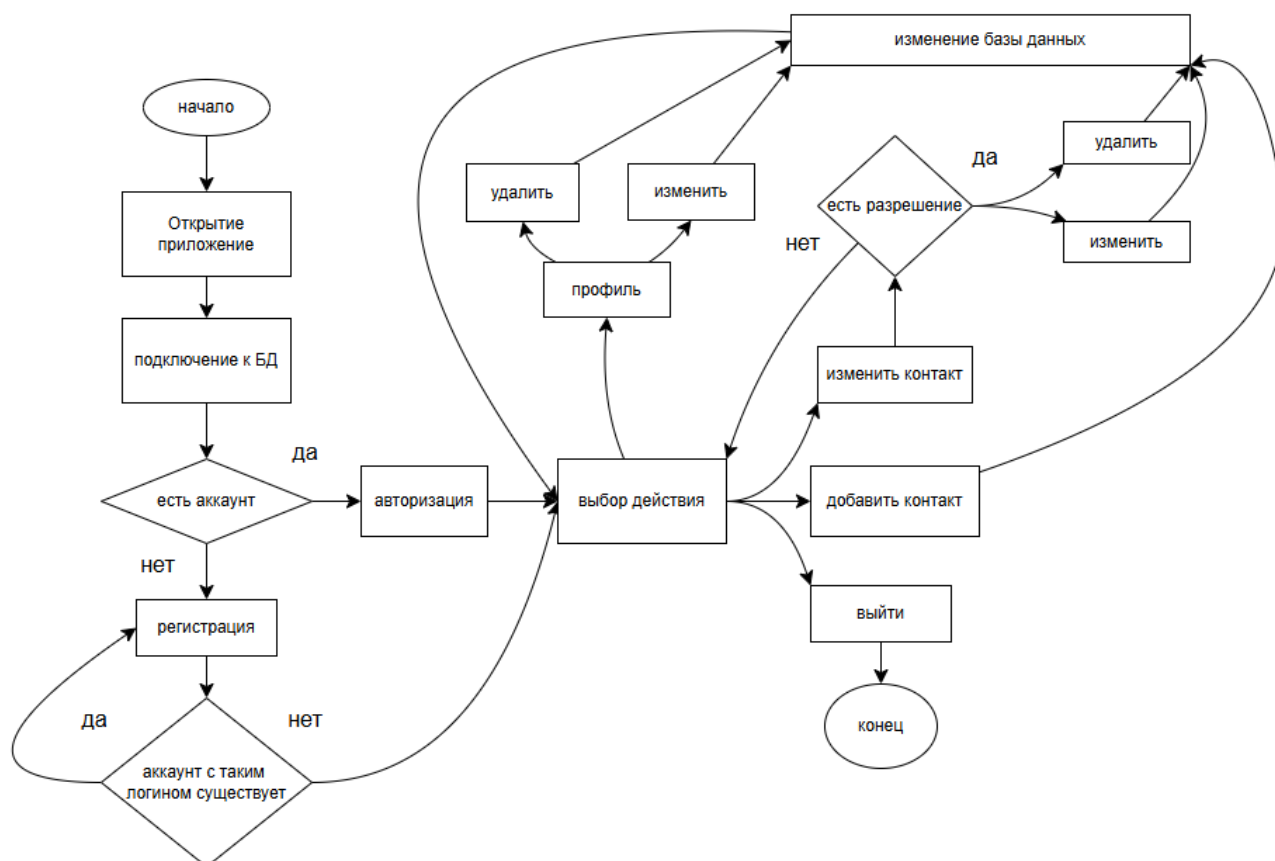


Рисунок 2.1 – Блок-схема программы

На рисунке 1 представлена блок-схема последовательности действий, выполняемых приложением, а также принципы обработки входных данных и формирования выходных данных. Настольное приложение «Телефонный справочник» предназначено для ведения контактов пользователей. Пользователь может создавать новые контакты, редактировать существующие и удалять.

Работа приложения начинается с «Открытия приложения». Затем происходит «Подключение к базе данных». Для работы с приложением мы

можем: пройти авторизацию или зарегистрироваться. Если мы решили авторизоваться, то нам следует сделать «Выбор действия»:

Для того, чтобы наши действия вступили в силу, происходит «Изменение базы данных» в соответствующих пунктах.

Далее мы можем продолжить работу и выбрать «Выбор действия» для продолжения.

Последовательность действий настольного приложения «Телефонный справочник» включает в себя следующие возможные шаги:

1) вход пользователя. Пользователь вводит свои учетные данные или создает новый аккаунт;

2) создание новых контактов. Пользователь нажимает на кнопку «Добавить контакт» и вводит имя, фамилию и номер нового контакта;

3) редактирование записи. Пользователь может отредактировать текст записи, изменить дату и время события, добавить или удалить теги, изменить прикрепленные файлы;

4) управление контактами. Пользователь может редактировать созданные контакты и удалять;

5) выход из учетной записи. Пользователь может выйти из своего аккаунта, чтобы обеспечить безопасность своих данных.

Таким образом, настольное приложение «Телефонный справочник» поможет пользователям эффективно организовать и хранить свои контакты.

2.2 Организация входных и выходных данных

Организация входных и выходных данных является важной частью разработки приложения «Телефонный справочник». В этом пункте курсовой работы необходимо описать, каким образом будет осуществляться ввод и вывод данных в приложении.

Ввод данных:

- пользовательский интерфейс. Разработанный справочник должен иметь удобный и интуитивно понятный пользовательский интерфейс, позволяющий пользователям вводить информацию о своих задачах, делах и событиях. Интерфейс может быть реализован в виде настольного приложения;
- формат ввода. Пользователям должны быть предоставлены формы и поля для ввода различных данных, таких как имя пользователя, номер телефона и прочее. Правильные форматы ввода должны быть определены и проверены на корректность;
- валидация данных. Важно обеспечить проверку введенных пользователем данных на корректность, чтобы предотвратить возможные ошибки. Некорректные данные, такие как пустые поля, должны быть обнаружены и сообщены пользователю, чтобы он мог внести необходимые корректировки.

Вывод данных:

- отображение элементов. Разработанный справочник должен отображать введенные пользователем данные, такие как имя контакта, его номер и прочее. Для этого могут использоваться списки, поля и другие удобные для визуализации средства;
- формат вывода. Отображаемая информация должна быть представлена в понятном и удобном формате для пользователя;

2.3 Описание базы данных

База данных используется в настольном приложении «Телефонный справочник» для хранения данных пользователя, такие как: имя и номер контактов и прочие.

Для приложения «Телефонный справочник» была выбрана СУБД MS SQL. СУБД (система управления базами данных) – это совокупность программ и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных. СУБД - комплекс программ позволяющих создать базу данных и манипулировать данными. MS SQL был выбран по ряду причин:

- Интеграция и совместимость. MS SQL и WPF имеют множество совместных инструментов, которые упрощают работу программиста;
- Производительность и масштабируемость. SQL Server оптимизирован для высоких нагрузок и может легко масштабироваться, что идеально подходит для приложений с увеличивающимся объемом данных и пользователей;
- Безопасность. MS SQL Server предлагает множество встроенных функций безопасности, таких как шифрование данных, аутентификация и авторизация пользователей, что позволяет обеспечивать защиту данных;
- Управление транзакциями. Поддержка надежных механизмов управления транзакциями помогает избежать проблем с целостностью данных, особенно в многопользовательских приложениях;
- Поддержка хранимых процедур и триггеров. Хранимые процедуры могут улучшать производительность приложения за счет уменьшения объема переданных данных и централизованного бизнес-логики на серверной стороне;
- Разнообразные инструменты для разработки и администрирования. Существует множество инструментов, таких как SQL Server Management Studio (SSMS), которые упрощают управление базами данных и разработку;

- Обширная документация и поддержка сообщества. Microsoft предоставляет обширную документацию и ресурсы, а также большую поддержку сообщества, что упрощает поиск решений и помощь при разработке.

Для настольного приложение «Телефонный справочник» используется схема базы данных в соответствии и рисунком 2.3 – Блок-схема базы данных

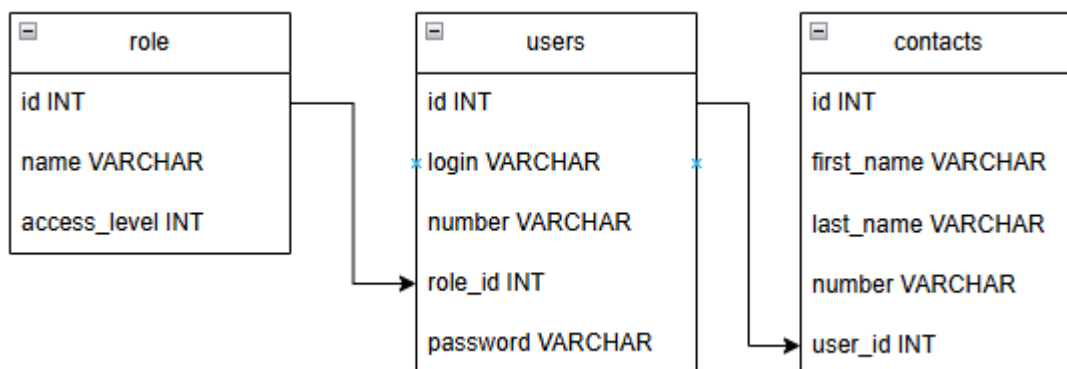


Рисунок 2.3 – Блок-схема базы данных

В этой базе есть следующие таблицы:

- Таблица ролей. В ней будут храниться роли, которые будут указывать какой уровень доступа есть у пользователя. Уровень доступа определяется по значению поля `access_level`, если оно равно 1, то доступ на редактирование контактов доступ иначе нет.
- Таблица пользователей. В ней будет храниться данные о пользователях такие как: логин, номер телефона, роль в системе, пароль.
- Таблица контактов. В ней будет храниться данные, о созданных контактах привязанных к пользователю такие как: имя, фамилия, номер телефона и `id` пользователя.

2.4 Описание основных модулей программного продукта

Приложения состоит из нескольких модулей. Ниже описание основных модулей:

1) модуль авторизации и регистрации. Этот модуль позволяет пользователям создавать учетные записи в системе, а затем входить в свои учетные записи с помощью логина и пароля;

При авторизации окно будет выглядеть в соответствии с рисунком 2.4.1 – окно авторизации.

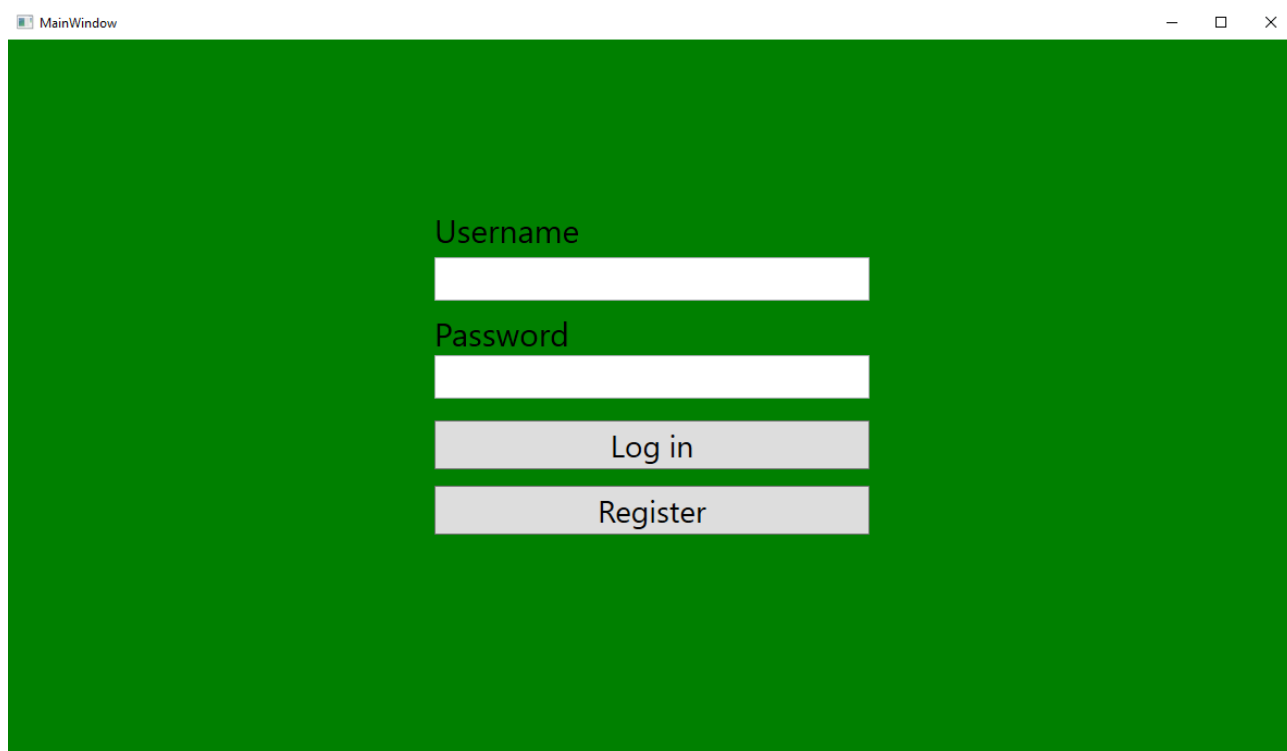


Рисунок 2.4.1 – окно авторизации

При регистрации нового пользователя окно будет выглядеть в соответствии с рисунком 2.4.2 – окно регистрации.

A screenshot of a registration window titled 'MainWindow'. The window has a green background. It contains four text input fields labeled 'Username', 'Number', 'Role', and 'Password'. Below these fields are two buttons: 'Register' and 'Log In'.

Рисунок 2.4.2 – окно регистрации

2) модуль контактов. Этот модуль позволяет пользователям просматривать свои контакты. Они могут добавлять, редактировать и удалять свои контакты;

При просторе существующих контактов окно будет выглядеть в соответствии с рисунком 2.4.3 – список контактов.

A screenshot of a contacts list window titled 'MainWindow'. The window has a blue background. At the top, there is a header bar with three buttons: 'Me', 'Контакты', and 'Log out'. Below the header is a table with four columns: 'id', 'first name', 'last name', and 'number'. The table contains three rows of data. Below the table is a large empty rectangular area. At the bottom of the window is a button labeled 'add new contact'.

id	first name	last name	number
6	asd	asd	asd
7	aa	aa	aa
8	aa	aa	aa

Рисунок 2.4.3 – список контактов

При редактировании пользователя окно будет выглядеть в соответствии с рисунком 2.4.4 – окно редактирование контакта.

The screenshot shows a window titled 'MainWindow' with a solid orange background. In the top right corner, there is a 'back' button. The form contains three text input fields, each with the label 'First name', 'Last name', and 'Number' respectively, and each containing the text 'aa'. Below the input fields are two buttons: 'Update' and 'Delete'.

Рисунок 2.4.4 – окно редактирование контакта

При создании нового контакта окно будет выглядеть в соответствии с рисунком 2.4.5 – окно создание нового контакта

The screenshot shows a window titled 'MainWindow' with a solid purple background. In the top right corner, there is an 'Exit' button. The form contains three text input fields, each with the label 'first name', 'last name', and 'Number' respectively. Below the input fields is a single button labeled 'Create new contact'.

Рисунок 2.4.5 – окно создание нового контакта

3) модуль изменение и удаление данных о пользователе. Пользователь может редактировать свои данные такие как логин или пароль ил удалить аккаунт;

При просмотре, изменении и удалении данных о пользователе окно будет выглядеть в соответствии с рисунком 2.4.6 – окно профиля.

The screenshot shows a window titled "MainWindow" with a standard Windows title bar (minimize, maximize, close buttons). The window has an orange background. In the top-left corner, there is a dashed rectangular box. In the top-right corner, there is a button labeled "back". The main content area contains the following elements:

- Username:** A text input field containing the letter "a".
- Number:** A text input field containing the letter "a".
- Role:** A dropdown menu with "admin" selected.
- Password:** A text input field containing the letter "a".
- Update:** A button located below the password field.
- Delete:** A button located below the update button.

Рисунок 2.4.6 – окно профиля

ЗАКЛЮЧЕНИЕ

Разработка приложения для управления телефонным справочником представляет собой актуальную задачу, которая может существенно улучшить организацию и хранение контактов телефонов пользователей. Все больше людей, независимо от их профессиональной деятельности и стиля жизни, ищут современные инструменты, способные облегчить процесс хранения и управления контактами.

В процессе работы над проектом были рассмотрены различные интерактивные возможности настольных приложений, которые были использованы для реализации функционала приложения. Было разработано несколько команд, которые позволяют пользователям получать необходимую информацию.

Кроме того, были использованы инструменты, такие как Microsoft Visual Studio, благодаря которым была обеспечена более эффективная и функциональная работа приложения.

В процессе разработки была использован фреймворк WPF – это .Net платформа для разработки настольных приложений, созданная компанией Microsoft. Основным принципом WPF заключается в разделении представления и логики приложения, что обеспечивает гибкость, расширяемость и упрощает поддержку пользовательского интерфейса.

В ходе выполнения поставленных задач был проведен анализ потребностей пользователей, определены ключевые требования к приложению. Разработан интуитивно понятный дизайн и пользовательский интерфейс, обеспечивающие удобство использования и простую навигацию. Функциональность приложения включает в себя широкий спектр возможностей, таких как добавление контактов, хранение, редактирование созданных контактов и их удаление. Авторизация с уже существующим пользователем или создание нового. В результате выполнения дипломного проекта – по выбранному алгоритму было разработано приложение, а также выполнен полный процесс разработки программного продукта в соответствии с выбранной моделью проектирования.

Процесс разработки включал в себя тестирование приложения среди пользователей, что позволило собрать важную обратную связь и внести необходимые коррективы для повышения качества приложения. В завершение проекта приложение было подготовлено к выпуску на рынок.

Преимуществами данного программного продукта являются:

- 1) простое и удобное управление;
- 2) понятный интерфейс.

Цели и задачи дипломного проектирования были выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Приложение А