Для начала для разработки desktop-приложения необходимо определиться с языком программирования, на котором будет писаться программное приложение. Выбор был сделан в пользу C#.

C# — объектно-ориентированный язык программирования общего назначения.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Выбран C# был из-за своего большого количества возможностей и функции. А также именно с этим языком был получен больший опыт работы, чем с другими языками программирования.

Теперь, когда язык программирования определён, следующим шагом будет выбор среды разработки, которая поддерживает выбранный язык программирования.

Первым кандидатом является Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки (IDE) программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать все виды приложений.

Visual Studio поддерживает множество языков программирования, а именно: Ajax, ASP.NET, DHTML, ASP.NET, JavaScript, Visual Basic, Visual C#, Visual C++, Visual F#, XAML и другие.

Из-за этого выделяются следующие преимущества Visual Studio:

* Возможность создать любое приложение.
* Поддержка большого количества языков программирования
* Имеет множество инструментов для разработчика
* Есть бесплатная версия (Community) для частного использования, студентов и создателей опенсорсовых проектов.

Но всё же недостатком Visual Studio является то, что эта среда не кроссплатформенная и доступна только на операционной системе Windows.

Второй на очереди это Visual Studio Code - лёгкий, но мощный редактор исходного кода, который работает на вашем настольном компьютере и доступен для Windows, macOS и Linux.

Он включает встроенную поддержку JavaScript, TypeScript и Node.js и имеет богатую экосистему расширений для других языков и сред выполнения (например, C++, C#, Java, Python, PHP, Go, .NET).

После вышесказанного можно определить следующие достоинства Visual Studio Code, а именно:

* Огромный набор плагинов, с помощью которого можно настроить редактор на свой вкус
* Поддержка большого количества языков программирования
* Кроссплатформенность
* Полностью бесплатен

Но насколько мощным редактором Visual Studio Code не казался, он больше подходит для разработки веб-приложений и сайтов. Если возникнет желание создать desktop-приложение, то могут возникнут трудности по настройке редактора под это дело.

И последний вариант на очереди Rider - кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#, VB.NET и F#. Работает на операционных системах Windows, macOS, Linux.

Узнав про эту среду разработки, можно выделить такие преимущества как:

* Поддержка многих языков программирования
* Кроссплатформенность

Недостатком Rider является то, что эта интегрированная среда разработки платная.

Из рассмотренных сред разработки была выбрана Visual Studio. Выбрана она по следующим причинам:

1. Немалый выбор инструментов для создания desktop-приложений.
2. Больший опыт работы с этой средой разработки по сравнению с другими.
3. Полная поддержка языка программирования C#.

Определившись со средой разработки, надо перейти к рассмотрению инструментов, с помощью которых будет создаваться приложение.

Первый популярный выбор разработчиков Electron - фреймворк, разработанный GitHub. Позволяет разрабатывать нативные графические приложения для операционных систем с помощью веб-технологий, комбинируя возможности Node.js для работы с back-end и браузера Chromium.

После обзора мнений других разработчиков, выделяются следующие достоинства фрейморка Electron:

* Использование наработок из Web (HTML, CSS, JS)
* Кроссплатформенность
* Качественная (и русифицированная) документация
* Поддержка сообщества и GitHub

Но без недостатков фрейморк Electron не обошёлся, выделяются следующие из них:

* Высокое потребление памяти (физическая и ОЗУ)
* Легко написать плохой код
* Плохая нативность

Вторым мощным инструментом является Qt – библиотека классов C++ и набор инструментального программного обеспечения для создания кроссплатформенных приложений с графическим интерфейсом (GUI).

Существуют вариации для других языков: PyQt для Python, QtRuby для Ruby, Qt Jambi для Java.

Проанализировав данную библиотеку, выделяются следующие её преимущества:

* Визуальный редактор интерфейса
* Огромное количество модулей в составе фреймворка
* Поддержка большого количества языков программирования
* Кроссфплатформенность

И у библиотеки Qt есть свои отрицательные моменты, а именно:

* Большой вес приложения
* Высокая стоимость коммерческой лицензии

Последний инструмент, который хочется рассмотреть это Windows Presentation Foundation (WPF). WPF — это платформа разработки графических интерфейсов от Microsoft, которая предлагает богатый набор инструментов для создания красивых и функциональных пользовательских интерфейсов.

Изучив возможности данной платформы, можно указать на следующие её достоинства:

* Широкая интеграция. WPF поддерживает ЗD-графику, видео, речевые технологии, работу с документами.
* Декларативное программирование. Конструктор доступен, но разработчики обычно предпочитают создавать проекты на основе кода с помощью декларативного XAML.
* Богатые возможности композиции и настройки.

Но без недостатков также не обошлось. Можно выделить следующее:

* Отсутствие кроссплатформенности. Созданные приложения будут работать только на ОС Windows.

В итоге в качестве инструмента разработки была выбрана платформа WPF по ряду причин:

1. Разделение пользовательского интерфейса и функционального программного кода в лице языка разметки XAML и языка программирования C#.
2. С помощью XAML можно сверстать любой пользовательский интерфейс любой сложности.
3. Встроенная поддержка паттерна разработки MVVM.
4. В отличие от других платформ, с WPF имеется больше опыта работы.

Осталось определиться, где будут храниться данные программного приложения. Лучшим решением в этом случае является подключение базы данных. А для работы с ней понадобится определённая СУБД.

Первым вариантом на рассмотрение является SQL Server Management Studio (SSMS). Это интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных SQL Azure. SSMS предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server и баз данных.

Преимущества SSMS:

* Обширный функционал для создания профессиональных и сложных программных продуктов;
* Интеграция с Visual Studio.

Недостатки SSMS:

* Нужны определённые знания для создания даже простого графического приложения с БД.

Вторым вариантом является Microsoft Office Access или просто Microsoft Access — реляционная система управления базами данных (СУБД) корпорации Microsoft. Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных.

Преимущества Microsoft Access:

* Имеет простой графический интерфейс, с помощью которого можно создавать различные объекты, такие как таблицы, запросы, формы, отчёты и т.д.;
* Благодаря встроенному языку VBA (макросам) можно писать приложения, работающие с базами данных;
* Вся созданная база данных и её объекты хранятся в одном файле;
* Для создания базы данных и графического интерфейса не обязательны знания языков программирования.

Недостатки Microsoft Access:

* Для создания более сложных вещей не хватает функционала;
* Нельзя чётко реализовать разграничение ролей администратора и обычного пользователя.

И третий вариант, который будет рассмотрен это DB Browser for SQLite. DB Browser for SQLite (DB4S) - это высококачественный визуальный инструмент с открытым исходным кодом для создания, проектирования и редактирования файлов баз данных, совместимый с SQLite. DB4S предназначен для пользователей и разработчиков, которые хотят создавать, искать и редактировать базы данных.

У этой СУБД можно выделить следующие преимущества:

* Имеет простой пользовательский интерфейс
* База данных является простым файлом, который можно встроить в приложение
* Её можно использовать во многих языках программирования

Недостатком SQLite для разработчиков может как раз оказаться неширокий список типов данных, используемых в таблицах.

После просмотра всех вариантов, была выбрана СУБД DB Browser for SQLite. DB Browser for SQLite.

Причинами данного выбора является то, что она не требует сервера для базы данных, проста в использовании и может быть интегрирована непосредственно в приложение.

В итоге для разработки программного приложения были выбраны следующие инструменты и программные средства:

* Среда разработки Visual Studio
* Платформа WPF (язык программирования C# и язык разметки XAML)
* СУБД DB Browser for SQLite.

Проанализировав предметную область, составив техническое задание, обозрев существующие программные решения и определившись с выбором программных средств, можно приступать к практической части.