Для разработки desktop-приложения необходимо выбрать подходящую для себя среду разработки, определиться с языком программирования и инструментом.

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.

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.

Недостаток Visual Studio является то, что эта среда не кроссплатформенная и доступна только на операционной системе Windows.

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).

Преимущества Visual Studio Code:

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

Но насколько мощным 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.

CLion - интегрированная среда разработки (ИСР) для языков программирования Си и C++, разрабатываемая компанией JetBrains. Подходит для операционных систем Windows, macOS и Linux.

Преимущества CLion:

* Поддержка многих языков программирования
* Кроссплатформенность

Недостатком CLion 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. Бóльший опыт работы с этой мощной средой разработки.

Определившись со средой разработки, осталось выбрать инструменты, с помощью которых будет создаваться приложение.

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.

Плюсы Electron:

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

Минусы Electron:

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

Qt – библиотека классов C++ и набор инструментального программного обеспечения для создания кросс-платформенных приложений с графическим интерфейсом (GUI).

Существуют вариации для других языков:

* PyQt для Python,
* QtRuby для Ruby,
* Qt Jambi для Java.

Плюсы Qt:

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

Минусы Qt:

* Большой вес приложения
* Высокая стоимость коммерческой лицензии

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

Плюсы WPF:

* 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.

Минусы WPF:

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

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.

И для хранения данных использовалась база данных SQLite. SQLite - компактная встраиваемая СУБД. Основной причиной выбора именно этой СУБД является то, что она не требует сервера баз данных и может быть интегрирована непосредственно в приложение.