ГБПОУ НСО «Новосибирский колледж электроники и

Вычислительной техники»

специальность 09.02.07 «Информационные системы и программирование»

Техническая документация

К клиент-лаунчеру для игры World of Warcraft Sirus

Исполнители

студенты группы 9ИС-325

Кутаков Сергей Викторович

Ильина Анастасия Александровна

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

г. Новосибирск, 2025 г.

Оглавление

[1. Общее описание 3](#_Toc192029601)

[1.1. Назначение лаунчера 3](#_Toc192029602)

[1.2. Архитектура системы 3](#_Toc192029603)

[2. Структура проекта 4](#_Toc192029604)

[3. Технологии и инструменты 4](#_Toc192029605)

[4. Установка и настройка 4](#_Toc192029606)

[4.1. Установка: 4](#_Toc192029607)

[4.2. Настройка: 4](#_Toc192029608)

[5. Программирование 5](#_Toc192029609)

[6. База данных 6](#_Toc192029610)

[6.1. Состав базы данных 6](#_Toc192029611)

[6.2. Основные операции 6](#_Toc192029612)

[7. Тестирование 7](#_Toc192029613)

[8. Обработка ошибок 7](#_Toc192029614)

# 1. Общее описание

## 1.1. Назначение лаунчера

Упростить процесс использования клиент-лаунчера и улучшить его визуальное оформление.

## 1.2. Архитектура системы

Клиент-лаунчер состоит из трех окон: главный экран, экран настроек, окно авторизации.

Главный экран состоит из следующих компонентов:

* Шапка с социальными сетями и технической поддержкой.
* Элемент отображения персонажа.
* Элемент отображения обновлений клиента.
* Информационный блок дополнительных окон (настройки и авторизации).
* Кнопка предзагрузки обновлений клиента.
* Кнопка запуска и входа в клиент.

Экран настроек состоит из следующих компонентов:

* Блок изменения расположения и установки клиента.
* Кнопка запуска лаунчера при загрузке системы.
* Кнопка отключения хеширования файлов.
* Блок просмотра пользовательских данных.
* Информационный блок о последнем установленном обновлении.
* Кнопка возвращения на главный экран

Окно авторизации состоит из следующих компонентов:

* Поле ввода логина
* Поле ввода пароля
* Кнопка входа в аккаунт
* Кнопка регистрации аккаунта

# 2. Структура проекта

Организация файлов и папок осуществляется начиная с основных файлов, которые занимают центральное место и являются ключевыми для функционирования системы. Последовательно переходя к вспомогательным файлам, которые, хотя и играют менее значимую роль, но все же способствуют поддержанию и оптимизации общей структуры проекта.

# 3. Технологии и инструменты

Используемые инструменты:

* **C# -** Используются для разработки лаунчера.
* **.NET Framework -** фреймворк для создания пользовательского интерфейса.
* **SQL Management Studio –** СУБД для создания базы данных.

# 4. Установка и настройка

## 4.1. Установка:

1. Скачайте актуальную версию лаунчера с официального сайта.
2. Запустите установочный файл и следуйте инструкциям по установке.
3. После установки на рабочем столе появится ярлык лаунчера.

## 4.2. Настройка:

1. Запустите лаунчер и выберите вариант загрузки клиента или укажите существующую папку с игрой.
2. Добавьте учетную запись для входа в игру.

# Программирование

Использованные в проекте встроенные методы из дополнительных библиотек:

* OpenFileDialog - класс, предоставляющий стандартное диалоговое окно для выбора одного или нескольких файлов. Он позволяет пользователю просматривать файловую систему и выбирать файлы для открытия.
* MessageBox - класс, который используется для отображения сообщений пользователю в виде всплывающего окна. Он может содержать текст сообщения, заголовок и кнопки для взаимодействия.
* Process.Start - метод, который позволяет запускать процессы на компьютере. Он может использоваться для открытия программ или документов в зависимости от переданных параметров.
* BitmapImage - класс, представляющий изображение в формате битовой карты. Он используется в WPF для загрузки и отображения изображений в приложениях. BitmapImage поддерживает различные форматы изображений и позволяет управлять такими свойствами, как размер и масштабирование изображения.
* FrameworkElement - базовый класс для всех элементов пользовательского интерфейса в WPF. Он предоставляет основные свойства и методы для работы с визуальными элементами, такие как размещение, размеры и привязка данных.
* DispatcherTimer - класс, который используется для выполнения кода через определённые промежутки времени в контексте пользовательского интерфейса. Он позволяет обновлять интерфейс или выполнять периодические задачи без блокировки пользовательского интерфейса.

# База данных

## Состав базы данных

База данных состоит из таблицы users, в которой присутствуют следующие данные:

* Login – уникальное имя пользователя, первичный ключ.
* Password – пароль пользователя, хранящийся в зашифрованном виде.
* CharImageName – название изображения персонажа.
* Level – уровень персонажа пользователя.

## Основные операции

* Добавление пользователя:

Когда новый пользователь регистрируется, в таблицу добавляется новая запись с уникальным логином, паролем, названием изображения и начальным уровнем.

Если логин уже существует, база данных не позволит добавить новую запись, так как логин является первичным ключом.

* Аутентификация пользователя:

При входе в систему проверяется логин и пароль пользователя.

Если логин и пароль совпадают с записью в базе данных, пользователь успешно проходит аутентификацию.

* Обновление данных пользователя:

Если пользователь меняет пароль, название изображения или достигает нового уровня, соответствующие поля в таблице обновляются.

Обновление выполняется только для существующих записей.

# Тестирование

# Обработка ошибок