**Муниципальное автономное образовательное учреждение «Лицей №9»**

**Система методов анализа данных в социальных сетях**

*«ExtremeParse»*

* Выполнил ученик 10 «И» класса

***Янковский Фёдор***

* Руководитель класса

***Хархота Александр Дмитриевич***

г. Новосибирск

2023 г.

Оглавление

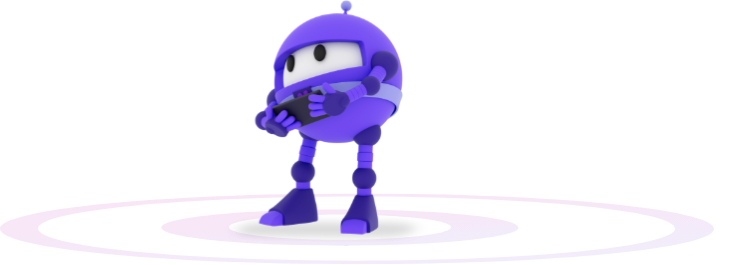
[Введение.Идея 3](#_Toc125733647)

[Overview.Инструменты 4](#_Toc125733648)

[Overview.Принцип работы 5](#_Toc125733649)

[Overview.Архитектура 7](#_Toc125733650)

[Заключение.О счастливом будущем 10](#_Toc125733651)

******

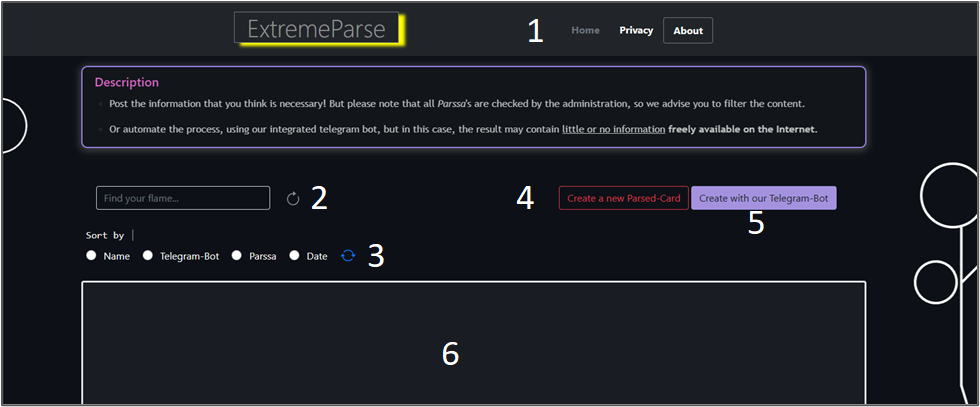
# Введение.Идея

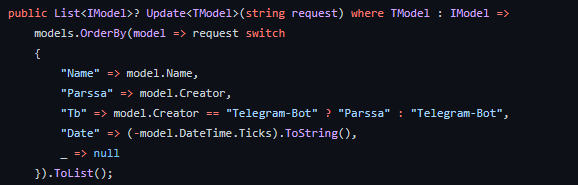
* Мотивация
  + Облегчение поиска основной, публичной информации о человеке, обладая лишь ее некоторым фрагментом
* Цель
  + Создать Open-Source проект-система, состоящая из:
    - Сервер-клиентской части; view-модель, визуализация информации – веб-сайт.
    - Клиентская часть; интерфейс отправки запроса по заданным аргументам – чат-бот.
* Задачи
  + Изучение технологии для создания веб-приложений - ASP.NET Core 7
  + Проектирование приложения на ранней стадии разработки
  + Декомпозиция задач
  + Деплой приложения на бесплатный хостинг
  + Целесообразное применение шаблонов проектирования, таких как: стратегия, медиатор, синглтон, команда, MVC, фабрика
  + Изучение основ сетевого программирования для создания клиент-сервер приложения

# Overview.Инструменты

* Платформы
  + .NET 7
  + ASP.NET Core 7
* Языки программирования\вёрстки
  + C# 11 + Razor Pages
  + Python 3.10
  + HTML5\CSS3
  + JavaScript
  + UML
* Среды разработки\редактор кода
  + Visual Studio 2022
  + Visual Studio Code
* Контейнеризация приложения\деплой
  + Docker
  + Railway (<https://railway.app>)
* Реляционная база данных
  + SQL Server (Windows\Unix-system)
  + LocalDB (Windows)
* Система управления версиями
  + Git
  + Github (<https://github.com>)
* Другое
  + ChatGPT (<https://openai.com>)

# Overview.Принцип работы

* Информация, опубликованная при помощи бота или пользователя, на сайте – Parssa (парсса, карточка)
* Веб-сервис для размещения сайтов в интернете – IIS
* Система является одним решением, внутри которого размещены 3 проекта (веб-сайт, клиент-бот, сервер-бот)
* Всё решение является Client\Server side
* Для запуска проекта на локальной машине можно воспользоваться Docker-контейнером (при условии, что Dockerfile существует), либо установить: .NET 7 SDK, Python 3.10
* Веб-сайт

1. <navbar> элемент, содержащий название бренда, 3 кнопки переадресации – Home, Privacy, About
2. Интерфейс поиска парссы[[1]](#footnote-1)
3. Интерфейс фильтрации карточек
4. <button> элемент для рендера блока класса ‘modal’ (см. <https://getbootstrap.com/docs/5.0/components/modal>).

Создание произвольной карточки

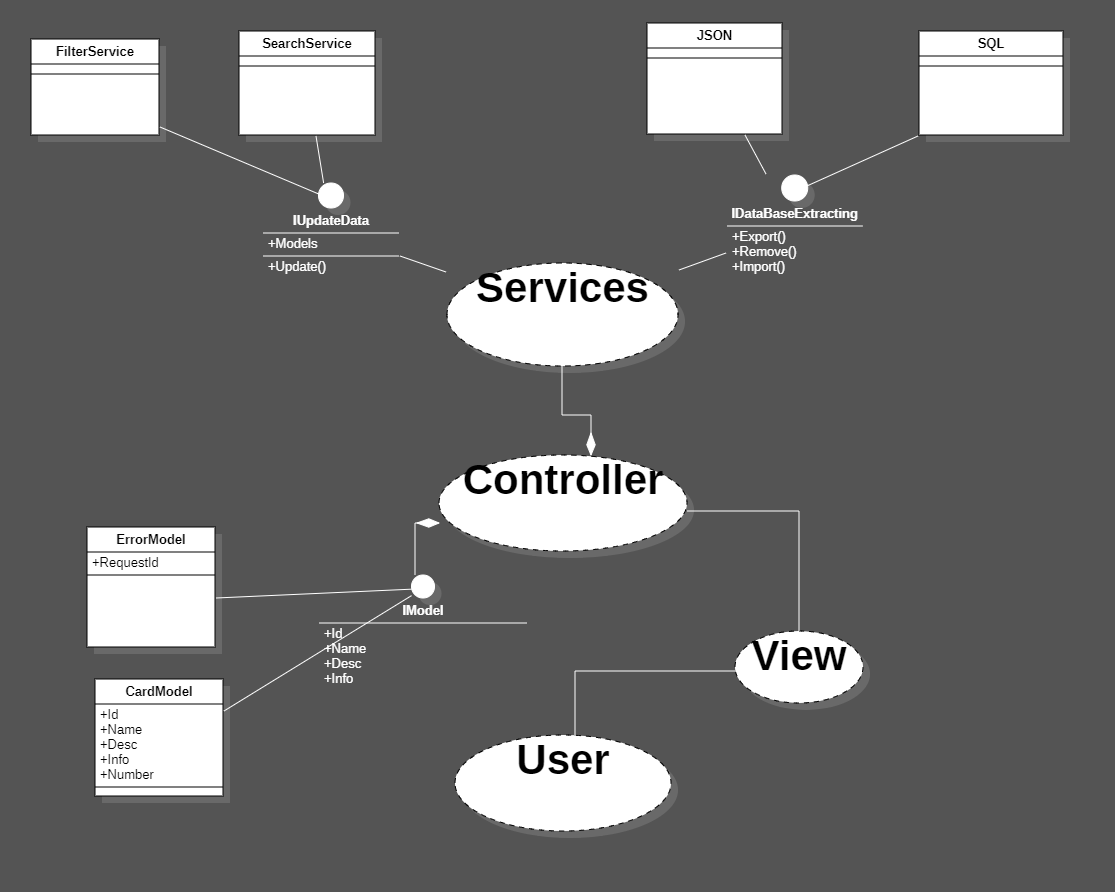
1. <button> элемент для переадресации на бота
2. Контейнер карточек

* Бот

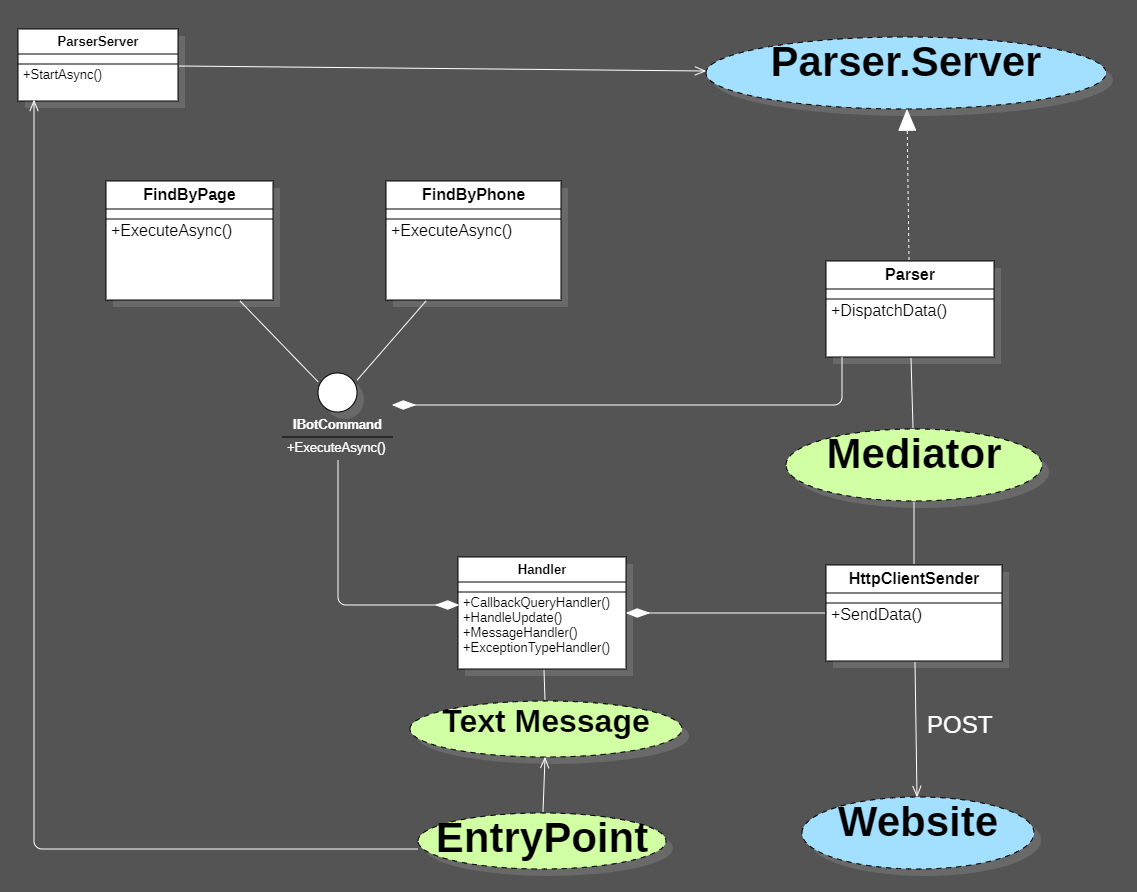
1. На данный момент существует только две команды:
   1. /findbyphone – поиск по номеру телефона (только +7)
   2. /findbypage – анализ страницы в ВКонтакте (также проверяется регулярными выражениями)
2. На данный момент (27.01.23) серверная часть бота не реализована
3. Принцип действия:
   1. Запуск бота, конфигурирование TCP-соединения между клиентом и сервером
   2. Ожидание запроса
   3. Отправка запроса с аргументами на сервер
   4. Поиск информации, возврат
   5. Отправка ответа пользователю, ожидание запроса на публикацию информации на веб-сервере
   6. В случае положительного запроса со стороны пользователя, создание POST-запроса на сайт.
   7. Возврат ответа веб-сервера пользователю

# Overview.Архитектура

* Веб-сайт
  + Является SPA приложением, динамичное обновление страниц происходит с помощью AJAX
  + Построен на платформe ASP.NET Core 7
  + Архитектурный стиль взаимодействия компонентов – REST API
  + Интерфейс базы данных CRUD (Create, Read, Update, Delete)
  + Серверная часть – C#11
  + Клиентская часть – Razor Pages, CSS3, JavaScript, Bootstrap5
  + Паттерны проектирования
    - Strategy
    - MVC (Model-View-Controller)
  + Дополнительные NuGet пакеты:
    - EFCore.SqlServer
    - EFCore.Tools
    - Azure.Containers
    - Extensions.Logging
    - Extensions.DependencyInjection
  + UML представление приложения:



* Бот
  + Клиентская часть – C#11
  + Серверная часть – Python 3.10
  + Дополнительные NuGet пакеты
    - Telegram.Bot
    - Newtonsoft.Json
  + Паттерны проектирования
    - Mediator
    - Singleton
    - Strategy
    - Command
  + UML представление приложения:



# Заключение.О счастливом будущем

* Планы
  + Реализовать серверную часть бота
  + Перейти на другую технологию: Razor Pages -> React/Angular, так как
    - Razor уже устаревшая технология
    - Загрузка страниц с фреймворком Razor осуществляется путем трансляции C# фрагментов кода в HTML, тем самым, сильно проигрывая в скорости работы
  + Доработать клиентскую часть сайта
    - Устранить недочеты в верстке
    - Покрыть UI тестами
  + Наконец, опубликовать проект
* Что потом?
  + Развиваться в области веб-технологий.
  + Использовать данный проект, как шаблонный репозиторий

1. Алгоритмом поиска карточки является конкатенация строк – Description, Name, Phone, Info; с последующим приведением в lowercase вид. Поиск линеен, т.к отсутствует сортировка карточек по их значению [↑](#footnote-ref-1)