

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота
на здобуття освітнього рівня бакалавра
за спеціальністю 121 Інженерія програмного забезпечення
на тему:

**РОЗРОБКА ВЕБ-СЕРВІСУ ДЛЯ МОНІТОРИНГУ КІБЕРСПОРТИВНИХ
МАТЧІВ**

Виконав студент 4-го курсу
Михайло БУХАЛО

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Лариса КАТЕРИНИЧ

(підпис)

Засвідчую, що в цій роботі немає запозичень з
праць інших авторів без відповідних посилань.

Студент

(підпис)

Роботу розглянуто й допущено до захисту на
засіданні кафедри інтелектуальних програмних
систем

«_____» _____ 2021р.,

протокол № _____

Завідувач кафедри

Олександр ПРОВОТАР

(підпис)

РЕФЕРАТ

Обсяг роботи 52 сторінки, 15 ілюстрацій, 25 джерел посилань

COUNTER STRIKE, DEPENDENCY INJECTION, DOTA 2, LEAGUE OF LEGENDS, WEB-API, WEB-SERVICE, .NET, КІБЕРСПОРТ, АВТОРИЗАЦІЯ, АЛГОРИТМИ СИНТИКАСИЧНОГО АНАЛІЗУ, ДОКУМЕНТАЦІЯ, ТЕСТУВАННЯ, ШАБЛОН ПРОЕКТУВАННЯ.

Об'єктом розробки є веб-сервіс для моніторингу даних кіберспортивних матчів у режимі реального часу з мінімально можливою затримкою, реалізований у вигляді web-арі під платформу .NET.

Метою роботи є створення веб-сервісу, який надав би можливість користувачу швидко та зручно отримувати дані кіберспортивних матчів по своїй улюбленій дисципліні. Веб сервіс має підтримуватися будь якою мовою програмування та працювати на найпопулярніших операційних системах. Сервіс повинен бути достатньо оптимізований, щоб мінімізувати затримку між відправкою запиту і отриманням необхідних результатів. Також він мусить бути захищений, щоб запобігти його перенавантаженню і при цьому покритий тестами для забезпечення надійності роботи всіх методів.

Методи розробки: використання теорії та технології синтаксичного аналізу веб сторінок, розроблення зручного програмного продукту для забезпечення необхідних потреб користувачів. Інструменти розробки: безкоштовне, вільно поширюване інтегроване середовище розробки Visual Studio 2019, мова програмування C#, відкрита бібліотека для вирішення задач автоматизації роботи веб-браузерів Selenium 4.0.0.

Результат роботи: був розроблений веб-сервіс, який вирішує проблему отримання даних кіберспортивних матчів з мінімально можливою затримкою. Було

проведено аналіз функціоналу вже існуючих на сьогоднішньому ринку ресурсів та сайтів які займаються наданням статистики та новин з різного роду кіберспортивних дисциплін. Створено веб-арі для отримання актуальних даних з турнірних матчів по комп'ютерним іграм з використанням Open Source бібліотеки Selenium.

Сфера застосування: Оскільки веб сервіс поставляється у вигляді web-арі, то дане програмне забезпечення може бути використане для інтеграції з будь якими додатками які пов'язані з кіберспортом.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	5
ВСТУП	6
РОЗДІЛ 1 КІБЕРСПОРТ	8
1.1 Перші кіберспортивні ігри та турніри.....	8
1.2 Популяризація комп'ютерного спорту	10
1.3 Сучасний кіберспорт.....	12
1.4 Трансляції кіберспортивних матчів.....	14
РОЗДІЛ 2 ОПИС ПРОЄКТУ	17
2.1 Призначення проєкту	17
2.2 Порівняння існуючих альтернатив	18
2.3 Технічні вимоги до проєкту.....	19
2.4 Перспективи розвитку	19
РОЗДІЛ 3 РОЗРОБКА WEB-СЕРВІСУ	21
3.1 Аналіз існуючих можливостей отримання даних	21
3.2 Вибір інструментів розробки.....	27
3.3 Розробка програмного коду	30
3.3.1 Процес синтаксичного аналізу сайтів.....	30
3.3.2 Архітектура проєкту.....	40
3.3.3 Авторизація	44
3.3.4 Тестування веб-сервісу	45
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	50

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

FFA – Free For All, кожен сам за себе

PGL - Professional Gamers League

MOBA - Multiplayer Online Battle Arena

TI – The International

SDK - Software Development Kit

CLR - Common Language Runtime

XPath - XML Path Language

DOM - Document Object Model

DI – Dependency Injection

IoC - Inversion of Control

CQRS - Command Query Responsibility Segregation

JWT - JSON Web Token

ВСТУП

Оцінка сучасного стану об'єкта розробки. Далеко не факт, що комп'ютер та інтернет стали частиною сучасного життя і небагато людей в світі можуть уявити звичайне життя без них. Молодше покоління все частіше стало проводити час за комп'ютером і розважатися за допомогою відеоігор. Але час іде і те, що колись починалося як дружні змагання в іграх, переросло в велику індустрію. Індустрію, яка надає людям емоції та розваги. Зі зростанням популярності кіберспорту, збільшилася і аудиторія людей серед яких цей вид спорту став актуальним, а тому з'явилася потреба освічувати різного роду турніри по великій кількості дисциплін, про що свідчить розвиток таких медіаплатформ, як YouTube Gaming, Twitch, Justin.tv, Livestream, та інших. Також разом з цим з'явилося і багато інших платформ, які не орієнтовані на прямі трансляції, а просто показують новини, статистику команд, та певний аналіз кіберспортивних дисциплін. Отже створено достатньо сайтів, які надають дані з кіберспортивних матчів, але більшість з них орієнтовані тільки на свою дисципліну та мають велику затримку, адже вони заробляють за рахунок реклами букмекерських контор, проте не існує некомерційних проєктів, які у вигляді веб сервісу змогли б надавати ці дані будь якому користувачу.

Актуальність роботи та підстави для її написання. Індустрія кіберспорту росте щодня. Зростає як і кількість нових дисциплін, так і кількість людей, які звертають свою увагу в цю нішу. А тому з'являється необхідність у зручному та швидкому доступу до даних матчів з будь якої платформи та будь якого пристрою. І розвиток web технологій цьому сприяє, адже при наявному інтернет доступі, ми можемо отримувати необхідні дані з будь якого пристрою.

Мета й завдання роботи. Метою роботи є створення web-сервісу, який надав би можливість користувачу швидко та зручно отримувати дані кіберспортивних

матчів по своїй улюбленій дисципліні. Для досягнення цієї мети поставлено такі завдання:

- Дослідити існуючі додатки та сайти у сфері трансляції кіберспортивних подій
- Дослідити існуючі можливості отримання даних кіберспортивних подій
- Розробити технічне завдання для web-сервісу
- Розробити web-сервіс для отримання даних з різних дисциплін
- Структурувати проект для легкого інтегрування у різні системи та додатки інших розробників

Об'єкт та методи дослідження. Об'єктом розробки є web-сервіс для моніторингу даних кіберспортивних матчів.

Для створення даного проекту був використаний метод управління розробкою програмного забезпечення Канбан з використанням веб-сервісу Github для керування версіями файлів. В якості інструменту створення програмного засобу було обрано Visual Studio 2019 – інтегроване середовище розробки та мову програмування C#.

Можливі сфери застосування. Розроблений web-сервіс можна використовувати як web api для сайтів, які займаються моніторингом кіберспортивних матчів, або як бібліотеку класів для мови програмування C#, яку можна інтегрувати в будь яку програму в середовищі .NET.

РОЗДІЛ 1 КІБЕРСПОРТ

1.1 Перші кіберспортивні ігри та турніри

Кіберспорт – це вид спорту, який оснований на індивідуальних або командних змаганнях по комп'ютерним іграм [1]. За часту змагання проходять у режимі online або ж через локальну мережу. Проте у часи коли кіберспорт лише починав свій розвиток доступ до інтернету був не у кожного. А тому всі змагання полягали не в боротьбі один з одним через мережу, а в спробах поставити кращий результат в певній грі на одному пристрої. В сучасному світі прийнято говорити про змагання не як про гру, а як про кіберспортивну дисципліну, яка надає можливість змагання в ній гравців через інтернет або ж по локальній мережі. Але змагальна складова в іграх була ще присутня задовго до того, як інтернет став широко розповсюдженим. І першим турніром по відеоіграм вважається «Міжгалактична олімпіада по Spacemar» [2]. В 1972 році після робочого дня в лабораторії по дослідженні штучного інтелекту Стенфордського університету зібралося 25 працівників та студентів, які вирішили помірятися силами в грі Spacemar. Це гра в якій беруть участь дві людини, кожен з яких управляє космічним кораблем. Завдання гравців – збити корабель противника за допомогою торпеди. Оскільки ці змагання проводилися вперше, то для привертання уваги до цієї події студенти заманювали глядачів за допомогою безкоштовного пива. Переможцем в категорії FFA став студент-біолог Брюс Баумгарт і його нагородою була річна підписка на журнал Rolling Stone. Проте цей турнір не залишився без уваги і був освітлений спортивним журналістом з Stone Sports – Сьюартом Брандом. Його репортаж і став першою міткою журналіста про Кіберспорт [3].



Рисунок 1.1 Перший кіберспортивний турнір «Міжгалактична олімпіада по Spacewar» [4]

Наступним етапом у розвитку кіберспорту став чемпіонат світу по грі Space Invaders в 1981 році, організований компанією Atari, компанією по розробці і видавництві відеоігор, яка справила великий вплив на зародження індустрії відеоігор як перспективної сфери економіки. В турнірі брали участь більш ніж 10 тисяч людей з усієї території Сполучених Штатів Америки. На цей раз учасниками вже не були прості любителі відеоігор, а люди, які проводили більшу частину свого життя за тренуванням з ціллю стати кращим в своїй дисципліні. При чому в турнірі брали участь не тільки дорослі, але й діти.



Рисунок 1.2 Чемпіонат світу по Space Invaders [4]

1.2 Популяризація комп'ютерного спорту

Розвиток комп'ютерної техніки тісно пов'язаний з прогресом в суміжних областях. В кінці минулого тисячоліття почали з'являтися комп'ютери потужність яких стала спроможна обробляти не тільки гарну картинку, але і прораховувати штучний інтелект. І в той час найпопулярнішою відеогрою, яка використовувала ці потужності на максимум стала Quake. Загальний розвиток інтернету дозволяв все більшій кількості людей приймати участь в кіберспортивному русі. Тренування і участь в онлайн турнірах стали звичайною справою для кіберспортсменів. До цього часу турніри проходили або в широко розповсюджених на той час комп'ютерних клубах, або за рахунок видавця гри, але щоб організувати турніри регулярно необхідне спонсорство. Після проведення декількох великих турнірів під патронажем Microsoft до цього ринку почали підтягуватися і інші компанії, які займалися виробництвом комп'ютерного забезпечення та обладнання.

Проте не чекаючи нових інвесторів разом із Microsoft був проведений перший великий турнір по Quake, який називався Red Annihilation. Він проходив 21 першого червня в World Congress Center під час виставки E3. Головним ж призом став автомобіль ведучого розробника по Quake - Ferrari 328 GTS. Вартість автомобіля була більша ніж 60000 доларів. Що дало велику мотивацію для всіх гравців, адже улюблена гра могла принести їм ще й гроші. В результаті кваліфікацій з більш ніж 2000 людей до фінального етапу дійшли лиш 16 найкращих. Переможцем турніру став Денис «Tresh» Фонг. Це змагання на довгі роки стало найбільш щедрим на призові, які отримав переможець [5].

В подальшому розвитку індустрії проривом стало створення двох професіональних ліг кіберспортсменів. Перша була заснована Анхелем Муносом в 1997 році, яка спочатку починалася як сайт, який був посвячений ігровій індустрії, але згодом бізнесмен зрозумів, що організація турнірів набагато вигідніша справа, а тому почав організовувати такі великі турніри як Cyberathlete Professional League або The Foremost Roundup of Advanced Gamers.

Другою організацією, яка займалася розвитком кіберспорту та проведенням турнірів стала ліга PGL. Вона перша отримала спонсорство від великих компаній які займалися розробкою комп'ютерного обладнання. Серед спонсорів були відомі нам AMD, Logitech та інші. Ліга отримала 2 мільйони доларів і за допомогою цих інвестицій ліга змогла організовувати турніри по Quake, Warcraft, Starcraft та інших дисциплінах, а також вона існує і досі, але займається організацією турнірів по сучасних відеоіграх.

1.3 Сучасний кіберспорт

Комп'ютерний спорт стрімко розвивався протягом всієї своєї історії. Зростали кількість кравців, різного роду дисциплін, а також і суми призових фондів. Зараз кіберспортивні турніри проводяться на тих же самих стадіонах, що і найбільші спортивні змагання, в тому числі і Олімпійських об'єктах та стадіонах, побудованих для чемпіонату світу по футболу. Кінець першого десятку 2000-х років запам'ятав себе як ера спаду в індустрії комп'ютерного спорту. Епоха Warcraft 3 підходила до свого завершення, а схожі ігри які прийшли їй на заміну не так однозначно були сприйняті ігровою спільнотою. Також зменшилась і популярність на дуже популярні шутери від першого лиця, такі як легендарна Counter Strike 1.6. А тому була необхідність в чомусь новому, тому, що б дало новий етап розвитку індустрії. Першим до цього дійшла компанія Valve, яка зараз займає перші позиції серед популярності турнірів своїх ігор. Анонсувавши в 2011 році турнір, по Dota 2, грі у жанрі MOBA, яка на той час ще знаходилася на стадії бета тестування, компанія викликала величезний фурор, адже призовий фонд складав 1.6 мільйон доларів. Переможець «The International» отримував чек на один мільйон доларів і варто підмітити, що тріумфаторами цього турніру стала українська команда Natus Vincere [6].

Після успішного проведення такого масштабного турніра, вся кіберспортивна спільнота активізувалася, адже незважаючи на те, що затрати були надзвичайно великими, Valve отримали прибуток, який багатократно покривав всі ці витрати. Інші розробники навпаки були більш прихильними до стандартної моделі проведення турнірів і не вкладували всі фінанси в одне велике змагання, а розподіляли їх протягом різноманітних подій. Важливо зазначити, що і сама компанія Valve експериментує з форматами проведення турнірів, адже в дисципліні по Counter Strike Global Offensive вони надають перевагу саме другому формату

проведення. Але з кожним роком все продовжувало змінюватися, адже індустрія не стоїть на місці, бо інакше вона перестане приносити прибуток. Як вже зазначалося організаторами турнірів є не тільки компанії, які розробляли саму гру, але наймасштабніші турніри з найбільшим призовим фондом проводять саме вони. Хоча у Valve і є такі конкуренти як Riot Games, Blizzard чи Wargaming, але обігнати цього гіганта вони так і не змогли, адже призовий фонд TI, росте з кожним роком в неаби яких масштабах.

Турнір	Заявлені призові	Зібрані призові	Загальна сума
The International 2011	1 600 000 \$	1 600 000 \$	1 600 000 \$
The International 2012	1 600 000 \$	1 600 000 \$	1 600 000 \$
The International 2013	1 600 000 \$	1 274 380 \$	2 874 380 \$
The International 2014	1 600 000 \$	9 331 105 \$	10 931 105 \$
The International 2015	1 600 000 \$	16 829 613 \$	18 429 613 \$
The International 2016	1 600 000 \$	19 170 460 \$	20 770 460 \$
The International 2017	1 600 000 \$	23 187 916 \$	24 787 916 \$
The International 2018	1 600 000 \$	23 932 177 \$	25 532 177 \$
The International 2019	1 600 000 \$	32 730 068 \$	34 330 068 \$

Таблиця 1.1 Розподіл призових фондів The International [7]

Нажаль через глобальну пандемію, провести The International 2020 не вдалося, але весь призовий фонд перенесли на наступний рік, і наразі проведення цього турніру в 2021 році планується. І призовий фонд на даний момент складає 40 018 195 \$ і це ще не кінець, адже кожного року перед проведенням турніру компанія Valve випускає бойові перепустки 25% з продаж яких іде в фонд турніру.

Розмірковуючи про комп'ютерний спорт варто відмітити, що в кінцевому результаті, як і в класичному спорті, саме глядачі впливають на те, які дисципліни

будуть мати популярність. Адже вони не безмовні наглядачі, а активні учасники ігрової спільноти і можуть впливати не тільки на призовий фонд, як вже зазначалося, але навіть і на те, які дисципліни будуть представлені в тому чи іншому турнірі. Варто відмітити, що саме завдяки ігровій спільноті в вересні 2020 року кіберспорт був признаний офіційним видом спорту в Україні.

1.4 Трансляції кіберспортивних матчів

Популярність онлайн трансляцій відеоігор, а також турнірів по ним стрімко зростає з кожним роком, адже організатори турнірів зацікавлені в якісному освідченні подій, щоб залучити якомога більшої кількості глядачів. В цьому аспекті кіберспорт став ближче до традиційних видів спорту, у яких телевізійні рейтинги спортивних передач дуже високі. А тому за минуле десятиліття різко зросла популярність трансляційних сервісів, які не тільки показують турніри, але й прислухаються до кінцевого глядача. Першим сервісом, який почав отримувати на цьому прибуток став YouTube. Проте платформа не розроблялася для цих цілей а тому його досить швидко замінив сервіс, який з самого початку задумувався як платформа для прямих трансляцій різного роду ігор – Twitch. Цей сервіс був заснований у 2011 році, і був спочатку у вигляді нового функціоналу для Justin.tv, платформи на якій її розробник транслював своє життя 24/7. А оскільки ігрова індустрія тоді дуже швидко розвивалася, то головний розробник вирішив відділити цей функціонал в окремий сервіс. І він не помилився, адже лише після п'яти днів після запуску, Twitch мав 6.5 мільйонів переглядів з активною аудиторію біля 60-70 тисяч глядачів. Спільнота сервісу і його перегляди стрімко зростали, а тому й зростали інвестиції в цей проект і в 2014 році його викупили Amazon за 970 мільйонів доларів. І вони не прогадали, адже на даний момент Twitch створює близько 1.8% від загального інтернет трафіку, а це надзвичайно багато, адже більше

лише у таких гігантів як Netflix, Apple та Google. А також сервіс займає 43% від усього трафіку відеотрансляцій, що більше ніж у YouTube.

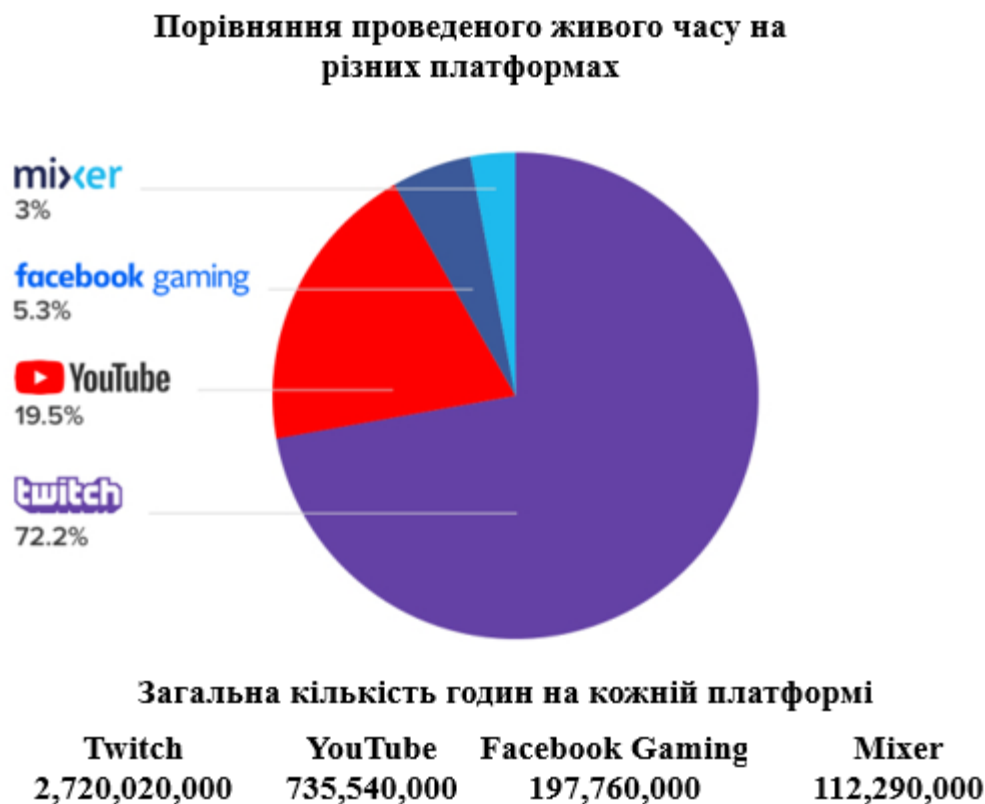


Рисунок 1.3 Діаграма порівняння живого часу на платформах [8]

Хоча на платформі Twitch транслюють не тільки ігри, але всі рекорди б'ють саме вони, особливо трансляції великих турнірів кіберспортивних дисциплін. Найпопулярнішими серед ігор є League of Legends, Counter Strike Global Offensive, Dota 2, Hearthstone та Fortnite. І з усіх цих дисциплін є свої турніри. Проте трансляція не зводиться лише до показу того як змагаються спортсмени. Для освідчення таких подій в кожній країні були створені свої студії, які займаються медіа сферою, щоб привернути до себе чим побільше глядачів. У студій є свої коментатори, вони проводять аналітику кіберспортивних матчів, роблять інтерв'ю з спортсменами, вигадують інтерактивну діяльність для глядачів, та багато іншого. В Україні найбільшою такою студією є Maincast, заснованою Віталієм «Vilat»

Волочай, який найперший в Україні почав коментувати кіберспортивні події і саме завдяки йому кіберспорт в нашій країні почав розвиватися.

Назва дисципліни	Кількість переглянутих годин за рік
League of Legends	1,701,952,797
Fortnite	1,214,851,116
Valorant	763,719,902
Counter Strike Global Offensive	746,590,961
Dota 2	490,057,057
Hearthstone	277,108,503

Таблиця 1.2 Кількість переглядів серед популярних кіберспортивних дисциплін на платформі Twitch [9]

РОЗДІЛ 2 ОПИС ПРОЄКТУ

2.1 Призначення проєкту

Разом із зростанням популярності кіберспорту збільшується і кількість дисциплін. При чому, так само як із класичним спортом де одна людина може, наприклад, слідкувати одночасно за футболом, баскетболом і хокеєм, так і у відносно новій для сучасного світу галузі кіберспорту є багато дисциплін, які можуть бути цікавими для користувача. Тому зростає необхідність у сайтах, порталах, чи додатках, за допомогою яких можна швидко та зручно дізнатися не тільки результати потрібного матчу, а й подивитися деяку статистику або ж просто розклад майбутніх матчів.

Оскільки більшість кіберспортивних турнірів транслюються з затримкою в декілька хвилин, то також постає необхідність в обході цієї затримки для того, щоб користувач міг отримувати актуальні дані, а не ті які йому показує трансляція. Це пов'язане з тим, що студії, які освідчують ці події співпрацюють з букмекерськими конторами як з спонсорами, а для них це не вигідно, адже це ж все ж таки комп'ютерний спорт у якому є запрограмована доля випадковості, яку складно, але частково можливо передбачити завдяки глибокому аналізу багатьох матчів. А така статистика є, адже ми і зараз можемо подивитися не тільки результати турнірів, які проходили десяток років назад, але і повністю передивитись матч та отримати необхідні дані з нього.

Головна ціль мого проєкту – розробити веб-сервіс за допомогою якого можна було б отримувати необхідні дані з мінімально можливою затримкою, який можна буде інтегрувати в будь-який додаток, оскільки сам сервіс реалізований у вигляді web-арі, то його можна використовувати з різними мовами програмування, на будь якій платформі. Що дає змогу іншим розробникам використовувати його у будь

яких своїх проєктах, будь то нативний додаток під операційну систему Windows, чи мобільний додаток на IOS.

2.2 Порівняння існуючих альтернатив

Існує достатня кількість сайтів для того, щоб переглядати результати кіберспортивних матчів. Ці сайти зазвичай схожі між собою, в наповненні, і сенсі, адже вони відображають поточні матчі, майбутні і ті які вже пройшли, а також показують їхні результати, тобто сам рахунок. Ну і звісно не обходяться без реклами букмекерів, адже це їхній заробіток. Проте більшість з них або спеціалізовані під конкретну дисципліну, такі як: dota2.ru [10], valorantesports.com [11], lolesports.com, pro.eslgaming.com [12], або ж мають в собі велику кількість дисциплін, але при цьому не надають необхідну кількість даних, яку б хотілося отримати.

Звісно є і добре реалізовані сервіси багаті на дані, які надають їх у вільному доступі, такі як: game-tournaments.com [13] або ggscore.com [14]. Але і в них є декілька недоліків, перше це те, що вони обмежені веб платформою і ці дані неможливо більше ніде отримати крім їхнього сайту, а також, що безкоштовно вони надають невелику кількість даних, а для того щоб мати доступ до більш детальної статистики необхідно оформлювати підписку.

Зважаючи на те, що доступ до даних можна отримати лише через веб браузер з'явилися деякі сервіси, які надають свої арі, для того, щоб розробники могли інтегрувати їх в свої додатки і використовувати у своїх цілях. Вони надають тестову версію своїх сервісів, які є обмеженими в кількості запитів на добу, та мають середню затримку, що не є найкращим результатом. У ході розробки даного проєкту я звертався до тих хто створив gamescorekeeper.com. Це саме такий сервіс який

надає API для отримання необхідних і при тому досить детальних даних з декількох дисциплін. Вони погодилися надати мені безкоштовний доступ до свого сервісу, але розкривати секрет реалізації не стали. Але мені прийшлося відмовитися, адже ціллю проєкту є розробити власний сервіс, а не використати вже реалізований.

Тому проаналізувавши наявні сайти та сервіси я дійшов до висновку, що наразі немає open-source проєкту, який міг би задовільнити поставлені цілі.

2.3 Технічні вимоги до проєкту

Даний веб сервіс повинен мати змогу діставати актуальні дані кіберспортивних матчів з різних дисциплін. Серед них обов'язково має бути список, поточних і майбутніх матчів, а також достатня статистика для відображенню користувачу.

Додаток має бути реалізований у вигляді web-API, який можна буде запустити у хмарі, наприклад в Azure, щоб користувачі могли мати доступ до необхідних даних у будь який момент часу. Архітектурна додатку має бути слабко зв'язною, щоб весь основний функціонал отримання даних можна було винести в окрему бібліотеку на мові програмування C#, і з легкістю інтегрувати в інші додатки розроблювані на платформі .NET. Також на web-API має бути реалізована авторизація, щоб контролювати кількість запитів від кожного користувача і прибрати можливість перенавантаження сервера.

2.4 Перспективи розвитку

Тенденція розвитку кіберспорту показує, що кількість дисциплін буде зростати з кожним роком, адже це дуже прибуткова справа. А разом із ростом кіберспортивних дисциплін буде і збільшуватися кількість турнірних матчів, а тому

даний проєкт зможе розвиватися паралельно із ними. Отримані дані можна буде не тільки використовувати для того, щоб користувачі могли тримати руку на пульсі своїх улюблених команд, але також їх можна буде зберігати у базах даних для глибокого аналізу матчів, а також для того, щоб завжди можна було мати доступ до детальної статистики будь якої команди з будь якої дисципліни. Мала затримка забезпечить обхідний шлях від букмекерських контор, які співпрацюють з організаторами турнірів і мають ці дані раніше ніж звичайні глядачі і тим самим вміло маніпулюють ними.

Збираючи інформації з матчів і надаючи її користувачу, можна також вести статистику і в іншу сторону та слідкувати за тим, чим цікавляться користувачі та паралельно передбачувати тенденції розвитку і популярності дисциплін. А ця інформація є доволі цінною на сьогоднішній час, адже більшість гігантів таких як Google, Facebook і Amazon збирають дані з своїх користувачів у рекламних цілях.

Окрім цього зі збільшенням користувачів, можна буде збільшити і кількість розробників і за рахунок цього знайти кращі способи отримання даних і також збільшити саму кількість даних. Вже зараз в деяких дисциплінах існує штучний інтелект, якому не складно переграти професійних кіберспортсменів. А тому не буде дивним, якщо в майбутньому ми будемо слідкувати і за турнірами в яких будуть брати участь запрограмовані боти. А за рахунок цих даних можна буде відтворювати майже всі дії гравців і тим самим створювати свій штучний інтелект.

РОЗДІЛ 3 РОЗРОБКА WEB-СЕРВІСУ

3.1 Аналіз існуючих можливостей отримання даних

Оскільки на сьогоднішній день існує велика кількість дисциплін, то для розробки даного проєкту було обрано три найпопулярніші, а саме: Counter Strike Global Offensive, Dota 2 та League Of Legends. Перші дві дисципліни розроблялися однією компанією - Valve, а тому очікувалося, що процес отримання даних буде схожим. А третя компанією Riot Games, яка займає високу позицію в кіберспортивній галузі, а також позиціонує себе як дружня до кінцевого користувача.

Counter Strike Global Offensive – це багатокористувацька гра у жанрі First Player Shooter, яка була випущена в 2012 році та стала безкоштовною лише в грудні 2018. Як і в інших іграх цієї серії, гравці діляться на дві команди, після чого вони змагаються між собою в серії раундів, по результатам яких, обирається переможець. Саме в цій версії гри був введений режим змагання, який використовується для проведення турнірів – перемагає команда, яка перша виграє 16 раундів.

Існує багато способів інтегрування своїх сервісів з грою. Є можливості активувати освітлення сцени або піротехніку під час раундів, активувати освітлення або гаптичні елементи в периферійних пристроях, підключених до ПК, доповняти трансляцію гри за допомогою власних графічних ефектів, отримувати всі зміни стану гри, тощо. Сам клієнт гри за замовчуванням може надавати всі дані про стан гри, а також надсилати сповіщення про його зміну, до будь-якої віддаленої або локальної точки через http post запити, використовуючи формат JSON для представлення своїх даних. Тобто компанія Valve за замовчуванням надала

можливість отримувати всі необхідні дані через свій арі. Деякі ентузіасти вже написали обгортку для нього у вигляді SDK для платформи .NET, за допомогою якого можна підключитися до клієнта гри без необхідності написання власного додаткового коду. Проте в даній роботі ми не можемо використовувати наведені вище методи, адже для підключення до ігрового серверу необхідно знати його ір адресу, а організатори турнірів їх не тільки не надають, а навпаки заховують від звичайних користувачів, адже інакше кожен міг би за допомогою невеликих зусиль перенавантажити сервер і зірвати кіберспортивний матч. Тому потрібно шукати саме спосіб, який не дістає дані напряму з гри, а методи які дозволяють отримувати дані з конкретних матчів.

Серед усіх сайтів і ресурсів які надають статистику по дисципліні Counter Strike Global Offensive найбільш виділяється hltv.org [15]. HLTV – акронім від Half-Life Television, є найбільшим веб сайтом і форумом серед усієї спільноти в цій дисципліні, який покриває усі кіберспортивні новини, маті та статистику. На сьогоднішній день на даний портал заходять більш ніж 5 мільйонів унікальних відвідувачів за місяць. Сайт набув велику популярність і має свій власний рейтинг гравців, який є найпопулярнішим і найвпливовішим серед усіх інших, адже ресурс існує з 2002 року і має надзвичайно велику базу даних статистики на кожен кіберспортивну команду і спортсмена у своїй дисципліні. Оскільки дана платформа покриває всі турніри які проводяться, навіть найменші, дозволяє переглядати дані та статистику поточних матчів з мінімальною затримкою, яка складає близько 30 секунд, а також дозволяє побачити статистику на кожного конкретного гравця в будь який момент часу, то для отримання даних було обрано саме її. Але нажаль сам сайт не надає свого власного арі для отримання інформації. А, оскільки на інших ресурсах які можуть надавати розклад турнірів та деяку мінімальну статистику по матчах затримка набагато більша ніж на даному ресурсі, то для

отримання необхідних даних було вирішено парсити цей сайт для отримання необхідної інформації. А оскільки на цьому сайті немає ніяких обмежень на кількість переглядів за короткий проміжок часу, то дані можна отримувати хоч кожні 2 секунди, що повністю нас задовольняє.

Другою грою в нашому списку є Dota 2 – гра у жанрі МОВА, яка розроблена компанією Valve. З самого початку ця гра була користувацькою модифікацією для вже старої кіберспортивної дисципліни Warcraft III, але в 2009 році почалася розробка її як окремої гри. Офіційний реліз був у червні 2013 року, хоча до цього ще два роки була доступна всім будучи на стадії бета тестування. Адже, як вже зазначалося, перший турнір по даній кіберспортивній дисципліні відбувся в 2011 році. Сама суть гри складається в змаганнях на карті особливого вигляду, в кожній команді беруть участь по п'ять гравців, кожен з яких управляє своїм героєм з унікальними наборами здібностей. Щоб перемогти одній команді потрібно знищити особливий об'єкт-споруду, якою володіє сторона противника і при цьому захистити від знищення свою.

Оскільки розробкою гри займалася компанія Valve, то як і в попередній дисципліні вона надала достатню кількість інструментів для модифікування гри, розробки нових плагінів, предметів, або кастомізацій, а також і для отримання ігрових даних з матчів [16]. Серед набору інструментів можна відмітити такі:

- Items Instruments: Створення нових предметів для магазину
- Addons Instruments: Створення та публікування власних плагінів для гри
- Level Design: Створення нових карт, та режимів
- Scripting: Створення власних скриптів, для визначення ігрових правил та поведінки під час гри.

- **UI Customization:** Створення та модифікування власного ігрового інтерфейсу під час гри, дозволяє модифікувати моделі, звуки, панорами та багато іншого.
- **Third-Party Tools:** Інструменти для інтеграції з іншими системами.

Нас саме цікавить останній пункт, адже цей інструментарій має свій арі для отримання необхідних даних стану конкретного матчу. Але головним недоліком цих інструментів є те, що можна отримати дані тільки з не відкритих лобі, а турнірні матчі завжди проводяться в закритих, щоб запобігти перенавантаженню сервера, а також захиститися від зловмисників які хочуть нашкодити процесу гри. А також сама гра і відповідно її функціонал не дозволяє відстежувати які кіберспортивні матчі з цієї дисципліни проходять на даний момент.

Тому як і з попередньою грою було вирішено обрати метод синтаксичного аналізу сайтів, а для цього необхідно було обрати ресурс, який задовольняв би такі умови:

- Мінімальна затримка
- Є історія минулих матчів, та розклад для майбутніх
- Статистика для кожного з матчів
- Детальна статистика ходу гри поточних кіберспортивних матчів

Серед таких ресурсів першим в голову приходить найбільший сайт для детальної статистики пройдених матчів - dotabuff.com [17]. Це багатоцільовий сайт, на якому зібрана детальна статистика ігор і гравців, а також останні новини з світу кіберспорту в даній дисципліні. Хоча сайт надає досить детальну статистику по вже закінченим матчам, на ньому немає можливості переглядати майбутні та поточні

змагання, а також в даного ресурсу немає свого арі, для того щоб спростити сам процес отримання даних.

Наступним розглянутим ресурсом став сайт game-tournaments.com. Він займається трансляцією поточних кіберспортивних матчів з різних дисциплін на ньому можна переглянути розклад матчів по будь яких турнірах, навіть найменших, а також подивитися результати минувших матчів, і актуальні дані з поточних ігор ну і звісно подивитися саму трансляцію гри. Проте на цьому сайті не можна подивитися більш детальну статистику по поточним іграм, а також оскільки сайт співпрацює з різного роду букмекерськими конторами, то затримка між результатами і самою грою досить велика.

Між балансом серед всіх вимог до ресурсів я зупинився на сайті cybersport.ru [18]. Це наразі головний портал для кіберспортивних новин в нашому регіоні і був заснований в далекому 2001 році. Основною темою сайту є саме новини кіберспорту, але на ньому також і присутня статистика по різним дисциплінам, минулі, активні та майбутні матчі, а також деяка статистика по ним і, що найбільш важливо для поточних матчів доступні актуальні дані ігрового стану. Затримка на даному ресурсі є також відносно невеликою, адже вона складає лише 30 секунд, не беручи до уваги, що для всіх ресурсів існує стандартна затримка в 2 хвилини, яку для безпеки зробили самі Valve і яку неможливо обійти. Тому для синтаксичного аналізу було обрано саме цей ресурс.

Третьою дисципліною виступає гра [League Of Legends](http://LeagueOfLegends.com), яка розроблена компанією [Riot Games](http://RiotGames.com) і є найпопулярнішою кіберспортивною грою серед усіх інших, але така популярність зумовлена тим, що більшість гравців це азіати, звісно є багато і європейських або американських команд, але вони не мають таких успіхів. Це стратегічна кооперативна гра в жанрі МОВА, яка є схожою до [Dota 2](http://Dota2.com), по стилю та цілям гри, адже так само команди складаються з 5 людей, кожен з яких

управляє своїм персонажем і по аналогії головним критерієм перемоги є знищити головну споруду противника при цьому захищаючи свою.

Riot Games пішла далі своїх конкурентів і створила досить зручне арі не тільки для отримання даних гри в реальному часі, а й надала можливість повністю автоматизувати створення невеликих турнірів за допомогою своїх сервісів. Також є можливість підписуватися на події під час гри і при зміні хоч якогось стану буде отримане сповіщення разом з актуальними даними. Проте як і у всіх попередніх дисциплінах постає проблема у тому, що у цілях безпеки неможливо отримати адреси сервера на якому розміщується кіберспортивний матч, а також сама арі не надає жодних даних про кіберспортивні події, тому незважаючи на зручність у даному проєкті її не можливо використовувати.

Ми знову повертаємося до синтаксичного аналізу вже існуючих ресурсів, які є в договорі з організаторами кіберспортивних турнірів і мають необхідні дані. Першим таким сайтом є lolesports.com, це офіційний ресурс від Riot Games, який займається трансляцією турнірних матчів. Він надає змогу дивитися розклад обраних турнірів, а також результати поточних і минувших матчів. Але не надає жодної статистики по ходу гри, хоча і є дуже зручним.

По даній дисципліні немає більше спеціалізованих сайтів, а тому ми повертаємося до пошуку сайтів які надають інформацію по декільком дисциплінам а не для конкретних. З ще не розглянутих нами залишаються egamersworld.com та ggscore.com. На першому можна подивитися розклад всіх матчів, а також можна отримати розклад всіх матчів, а також деяке порівняння двох команд, яке ґрунтується на попередніх результатах команди, а також на особистих зустрічах. Необхідні дані, але їх замало. Другий ж ресурс є пішов трішки далі і крім всього того, що є у першого, цей сайт надає можливість переглянути статистику по кожному гравцю окремо і порівнює їх за рахунок особистих результатів відносно

ролі яку вони виконують. Але є також і мінуси, адже на сайті досить велика затримка і він не надає даних які ми потребуємо, а саме поточний ігровий стан. Тому було вирішено діставати дані вже з знайомого нам сайту cybersport.ru, адже він найкраще підпадає під поставлені нами критерії.

3.2 Вибір інструментів розробки

Даний сервіс повинен бути швидким та зручним для використання. Оскільки реалізація сервісу буде у вигляді Web-апі, то всі запити повинні бути асинхронними, щоб забезпечити можливість обробки одних і тих же запитів паралельно і незалежно один від одного. Також основна частина функціоналу повинна бути винесена в окремі сервіси у бібліотеці класів, яку в майбутньому можна буде поширювати як пакет для синтаксичного аналізу даних на платформі .NET. Виходячи з вище зазначених умов було обрано мову програмування C#.

C# - об'єктно орієнтована мова програмування, а такий підхід дозволяє вирішувати задачі по створенню великих і в той же час гнучких, масштабованих і розширюваних додатків. На етапі розробки проекту актуальною версією мови є C# 9.0, яка вийшла в листопаді 2020 року разом із релізом платформи .NET 5 [19].

Даний проект розроблявся, використовуючи саме цю платформу.

Ключовими перевагами цієї платформи можна виділити наступні:

- Підтримка декількох мов. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує декілька мов: наряду з C#, також VB.NET, C++, F#.
- Кроссплатформеність. Платформа .NET підтримується на більшості сучасних операційних системах: Windows, MacOS, Linux та інших, а тому

використовуючи різні технології цієї платформи можна розробляти додатки, які зможуть без проблем запускатися на кожній з них.

- Потужна бібліотека класів. .NET надає єдину для всіх підтримуваних мов бібліотеку класів.
- Різноманітність технологій. Загальномовне середовище виконання CLR і базова бібліотека класів стали основою для великої кількості різного роду технологій, які розробники можуть використати при розробці своїх додатків. Так платформа .NET має інструменти для роботи з базами даних, для розробки графічних додатків з багатим і насиченим інтерфейсом, для розробки мобільних додатків, створення веб-сайтів, і веб-додатків та багато іншого.
- Продуктивність. Згідно з тестами веб-додатки на платформі .NET 5 сильно випереджають по швидкодії інші веб-додатки, які побудовані за допомогою інших технологій.

Для реалізації поставленої задачі використовувалася одна з технологій платформи .NET, а саме ASP.NET, яка дозволяє створювати широкий спектр веб-додатків: від невеликих сайтів, до масштабних веб-арі та веб-сервісів. Також з переваг можна відмітити, що це повністю open-source фреймворк і всі вихідні файли проекту знаходяться на GitHub.

Середовищем розробки було обрано Visual Studio 2019 . Вона включає в себе багато інструментів для зручної розробки на мові C#, зручний дебагер (програма для тестування та виправлення вад коду), вбудований менеджер пакетів за допомогою якого можна встановлювати необхідні бібліотеки, а також надає змогу прямої роботи зі сервісом для контролю версій GitHub.

Для отримання необхідних даних з сайтів потрібно використовувати бібліотеку, яка швидко та якісно зможе парсити ресурси. І, що не менш важливо, вона має бути кросплатформеною і підтримуватися різними мовами програмування, адже в майбутньому планується використовувати основні сервіси як окрему бібліотеку класів для платформи .NET. Саме тому було обрано безкоштовну бібліотеку Selenium WebDriver версії 4.0.0 [20].

Selenium – це інструмент для автоматизованого управління браузерами. Найчастіше він використовується для автоматизації тестування великих веб-додатків, але завдяки Selenium, можна запрограмувати будь які рутинні дії, які виконуються через веб-браузер.

Розробку Selenium підтримують розробники веб-браузерів, вони адаптують свої додатки для більш тісної взаємодії з Selenium, а інколи навіть реалізують вбудовану підтримку даного інструменту. Selenium дозволяє писати сценарії на великій кількості мов програмування і є ключовим компонентом багатьох методів автоматизації. Selenium дає змогу управляти браузером віддалено, завдяки чому є можливість запускати браузери в хмарі.

Список браузерів, які підтримуються розробниками Selenium:

- Chromium/Google Chrome. На платформах Windows, MacOS, Linux
- Firefox. На платформах Windows, MacOS, Linux
- Edge. На платформі Windows 10
- Safari На платформі MacOS
- Opera На платформах Windows, MacOS, Linux

Виходячи з підтримуваних браузерів було обрано саме Google Chrome, адже він має змогу запускатися майже на будь якій платформі. Для даного проєкту це не грає ролі, адже розроблюваний веб-сервіс можна запустити у хмарі і використовувати потрібний браузер, але для того, щоб поставляти основний функціонал як бібліотеку, потрібно мати підтримку найбільшої кількості платформ.

Для синтаксичного аналізу самих веб-сторінок і отримання необхідних даних було використано XPath (XML Path Language) – мова запитів до елементів XML або XHTML документів. Вона була розроблена для організації швидкого і зручного доступу до частин XML документу. XPath покликаний для того, щоб реалізувати навігацію по DOM (Document Object Model).

XPath використовує вирази шляхів, для того щоб обирати окремі вузли в xhtml документах. Ці вирази дуже схожі на ті, які ми використовуємо працюючи з традиційною файловою системою комп'ютера. XPath включає в себе більше сотні вбудованих функцій. Є функції для рядкових і числових значень, порівняння вузлів, управління послідовностями та багато іншого.

Використання цих технологій дозволить забезпечити вище зазначені потреби.

3.3 Розробка програмного коду

3.3.1 Процес синтаксичного аналізу сайтів

Синтаксичний аналіз – це процес аналізу вхідної послідовності символів, з ціллю виділення необхідних граматичних структур, які задаються граматиною. Основним завданням цього методу в даному проєкті є аналіз сайтів і знаходження необхідних елементів серед DOM необхідного сайту. При чому потрібно

забезпечити оптимізований спосіб аналізу, щоб не робити зайвих дій, адже ми потребуємо найбільшої швидкості з ціллю мінімізувати затримку.

Першим для синтаксичного аналізу будемо розглядати сайт hltv.org. Якщо переглянути його сторінку з розкладом матчів (див. рис. 3.1), то можна замітити, що він ділиться на дві секції – матчі, які йдуть на даний момент, а також заплановані події. З секції з поточними кіберспортивними матчами. Ми можемо дістати усі потрібні нам дані, які містять коротку інформацію про матч, а саме: формат проведення, назву команд, поточний рахунок по карті і всій грі, а також назву турніру. А з секції для майбутніх матчів, ми також можемо отримати всі ті самі дані, окрім поточного результату, бо його ще немає, але до них додається час та дата проведення матчу.

LIVE ***** bo3	Copenhagen Flames Lyngby Vikings	3 (0) 1 (0) Expand	LOOT.BET Season 9	Odds
LIVE ***** bo3	Dire Wolves VERTEX	10 (0) 3 (0) Expand	ESL Australia & NZ Championship Season 12	Odds
Upcoming CS:GO matches				
Saturday - 2021-05-08				
11:00 ***** bo3	ttc Case		Funspark ULTI 2021 Europe Regional Series 2021	Odds
11:00 ***** bo3	Nexus eXploit		Funspark ULTI 2021 Europe Regional Series 2021	Odds
12:30 ***** bo3	ORDER Paradox		ESL Australia & NZ Championship Season 12	Odds
13:00 ***** bo3	Spirit Academy ALTERNATE aTTaX		BLAST Rising 2021	Odds

Рисунок 3.1 Сторінка з розкладом матчів сайту hltv.org [21]

Натиснувши на будь який з матчів у списку ми перейдемо на сторінку з деталями самого матчу, які будуть схожими для поточного і майбутнього матчів. Однаковими для обох випадків будуть:

- Деталі матчу (див. рис. 3.2). Більш детальні відомості про сам матч, а також які карти будуть гратися по ходу матчу.
- Порівняння гравців (див. рис. 3.3). Наведена детальна статистика по кожному з гравцю, за останні 3 місяця.
- Статистика по картам. Дані про відсоток перемоги на кожній з карт за останні 3 місяці на кожній з яких зіграно мінімум по 3 рази.



Рисунок 3.2 Деталі майбутнього матчу

З деталей майбутнього матчу ми можемо дістати необхідні дані про саму подію, тобто назви команд і з яких вони країн, назву турніру, час та дату проведення, а також заявлені карти на матч, які з'являються незадовго до початку матчу.

Секція, яка містить порівняння гравців (див. рис. 3.3) містить детально інформацію по кожному з гравців, яка основана на його персональних результатах за останні три місяці, до неї входить:

- Rating 2.0. Показник, який формується на основі всіх позитивних та негативних дій гравця під час його матчів. Є найголовнішою ознакою якості гравця.

- Kills Per Round. Середнє значення вбивств за раунд.
- Death Per Round. Середнє значення смертей за раунд.
- Kast. Відсоток раундів у яких гравець зробив вбивство, допомогу, вижив, або був розмінний на гравця ворожої команди.
- Impact. Перша версія рейтингу 2.0, Показник, який формується на основі кількості вбивств за раунд, статистику виживання і кількості мультивбивств.
- Average Damage Round. Середня кількість шкоди, яку наносить гравець за раунд.

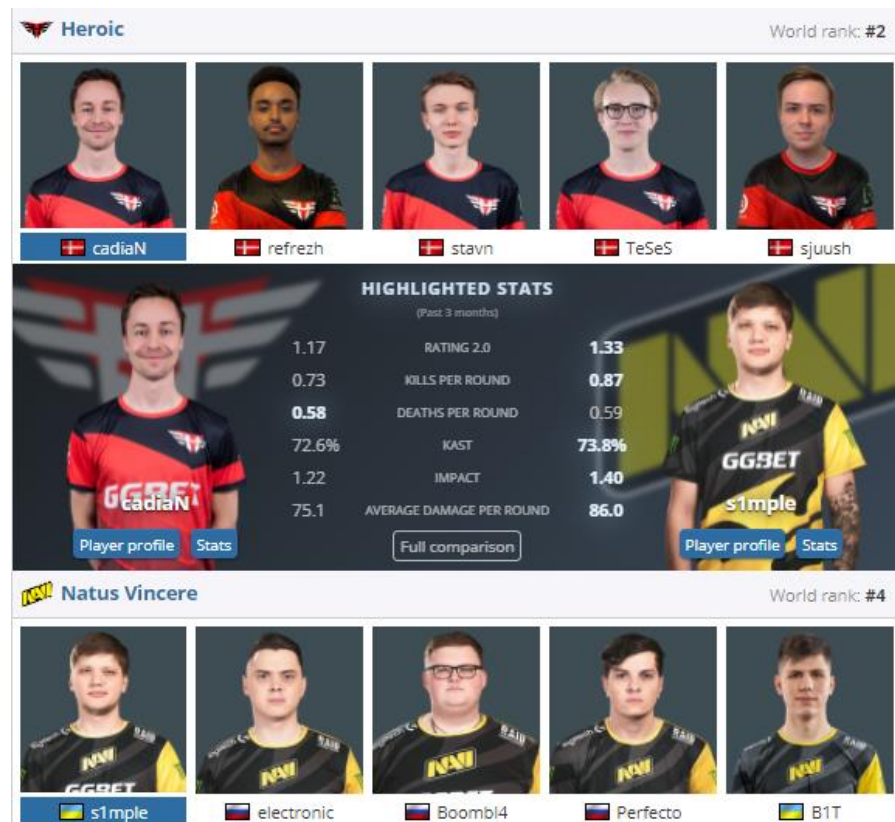


Рисунок 3.3 Секція з порівнянням статистики гравців [22]

Наступною корисною інформацією про матч є статистика перемоги по кожній з карт (див. рис. 3.4). Всього в Counter Strike Global Offensive є 13 змагальних карт, але в кіберспортивних турнірах обирається лише 7 з них, які змінюються з часом, а тим самим інші 6 дороблюються та оновлюються, щоб гра не стояла на місці, а розвивалася. На більшості турнірів перші матчі проводяться в форматі Best Of 1, тобто грається тільки одна карта і переможцем в матчі стає переможець цієї карти. Наступні ігри аж до самого фіналу проводяться у форматі Best Of 3, тобто перемагає та команда, яка першою отримає дві перемоги з трьох можливих карт. А також в деяких турнірах фінал проходить у форматі Best Of 5, де відповідно перемогу здобуває команда, яка першою переграє противника на трьох картах з п'яти можливих. Оскільки під час матчу граються не всі карти з можливого вибору, то ця статистика є важливою, для обрання фавориту матчу та створенню якихось прогнозів на гру.

Map stats Match lineup core winrate, past 3 months, min. 3 maps played








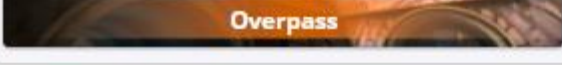

	 Heroic	 Natus Vincere
 Dust2	-	80% 10 maps
 Mirage	75% 8 maps	50% 8 maps
 Inferno	73% 11 maps	50% 4 maps
 Nuke	88% 8 maps	71% 7 maps
 Train	57% 7 maps	60% 5 maps
 Overpass	75% 4 maps	33% 6 maps
 Vertigo	83% 6 maps	- 0 maps

Рисунок 3.4 Статистика перемоги команд по кожній з карт за останні 3 місяці [22]

Дана статистика буде однаковою для минулих, поточних і майбутніх матчів, а тому і DOM сторінки відрізнятися не буде. І за допомогою синтаксичного аналізу можна отримати кожен з наведених статистик. Звісно є можливість отримати їх всі відразу, але оскільки це більш затратно і займає більше часу, то було вирішено створити окремий метод для отримання кожної з них.

Для матчів які проходять в даний момент до всієї вище описаної статистики додаються дані про поточний результат матчу (див. рис. 3.5). У даній секції є всі необхідні дані для отримання ігрового стану матчу, а також вона постійно оновлюється по ходу зміни цього стану, тобто інформація оновлюється динамічно. Але при обраному методі синтаксичному аналізу ми можемо дістати дані з DOM елементу тільки в певний момент часу. Проте це не проблема, адже аналіз проходить дуже швидко а тому ми можемо отримувати необхідні дані кожні 2 секунди, або більше в залежності від потреб.

Дана інформація містить в собі набір даних, який складається з:

- Псевдоніму гравця
- Кількість поточного здоров'я у відсотках
- Поточна кількість грошей
- Кількість вбивств
- Кількість допомог
- Кількість смертей
- Середню шкоду за раунд на основі вже зіграних

Scoreboard Normal Advanced BETA

R: 16 - nuke 4:11 1:26

Lyngby Vikings				K	A	D	ADR
hns	100	\$150	11	2	13	77.1	
birdfromsky	100	\$0	9	1	11	55.6	
raalz	100	\$150	7	3	11	55.5	
FASHR	100	\$150	7	1	12	42.4	
anarkez	0	\$150	5	0	14	36.6	

Copenhagen Flames				K	A	D	ADR
jabbi	100	\$150	16	2	9	104.3	
MODDII	100	\$0	13	3	8	86.4	
nicoodoz	100	\$450	13	2	7	77.3	
AcilloN	100	\$0	9	5	8	70.5	
Nodios	100	\$150	9	4	7	61.0	

Рисунок 3.5 Інформація про поточний стан гри [22]

Таким чином ми можемо отримати всі необхідні дані для моніторингу кіберспортивних матчів по дисципліні Counter Strike Global Offensive. А оскільки ми користуємося методами синтаксичного аналізу, то завдяки широкому спектру статистики ресурсу hltv.org в майбутньому проєкт може розвиватися і надавати не тільки дані про матчі, а також і детальну статистику по кожному з гравців, за всі його зіграні матчі у кожній з команд.

Другим ресурсом, який був використаний для моніторингу кіберспортивних матчів з дисциплін Dota 2 та League Of Legends став сайт cybersport.ru. Оскільки сайт має однакову структуру для кожної з дисциплін, то отримання даних буде доволі схожим, а сильно відрізняться буде лише поточний ігровий стан матчу. Тому для прикладу наведемо лише синтаксичний аналіз для отримання даних для Dota 2.

Сторінка з матчами по кожній з дисциплін містить в собі три секції з списком матчів: минулими, активними та майбутніми. Кожна з цих секцій містить в собі список матчів з короткою інформацією про кожен з них (див. рис. 3.6). З даних секцій можна отримати необхідні дані, які будуть містити в собі дату проведення матчу, назви команд та назву турніру з його стадією. Для поточних та минулих матчів буде також наявний результат поєдинку команд.

08.05.2021, 12:00	Team MagMa	VS	Aster	i-League 2021 Групповий етап
08.05.2021, 13:00	Fnatic	VS	Lilgun	Dota Pro Circuit 2021: Season 2 – ЮВА 1-й дивізіон Сезон
08.05.2021, 14:00	Unique	VS	HR	Pinnacle Cup Групповий етап
08.05.2021, 16:00	Into The Breach	VS	Hippomaniacs	Dota Pro Circuit 2021: Season 2 – Європа 2-й дивізіон Сезон

Рисунок 3.6 Секція з розкладом майбутніх матчів [23]

На відмінну від попереднього ресурсу, де всі матчі містилися на одній сторінці, тут список матчів є досить великим, а тому розділений по сторінкам, список яких відображається внизу сторінки. А тому ми можемо мати до нього доступ і завдяки інструментам Selenium які дозволяють автоматизувати роботу в браузері ми можемо перейти по кожній з сторінок і отримати новий список кіберспортивних матчів для синтаксичного аналізу.

Перейшовши на будь яку сторінку майбутнього матчу, ми зможемо отримати більш детальну інформацію про сам матч, а також деяку статистику по командам. Серед необхідних нам даних в першу чергу нас цікавить інформація про деталі матчу, а також про команди які беруть участь в ньому і їхні склади (див. рис. 3.7). Оскільки у складі команди може бути більше ніж 5 гравців, а у матчі може брати участь лише п'ять, то ці дані нам необхідні. Ми маємо змогу отримати назви команд, країни за які вони виступають, а також псевдоніми та справжні імена учасників команд.

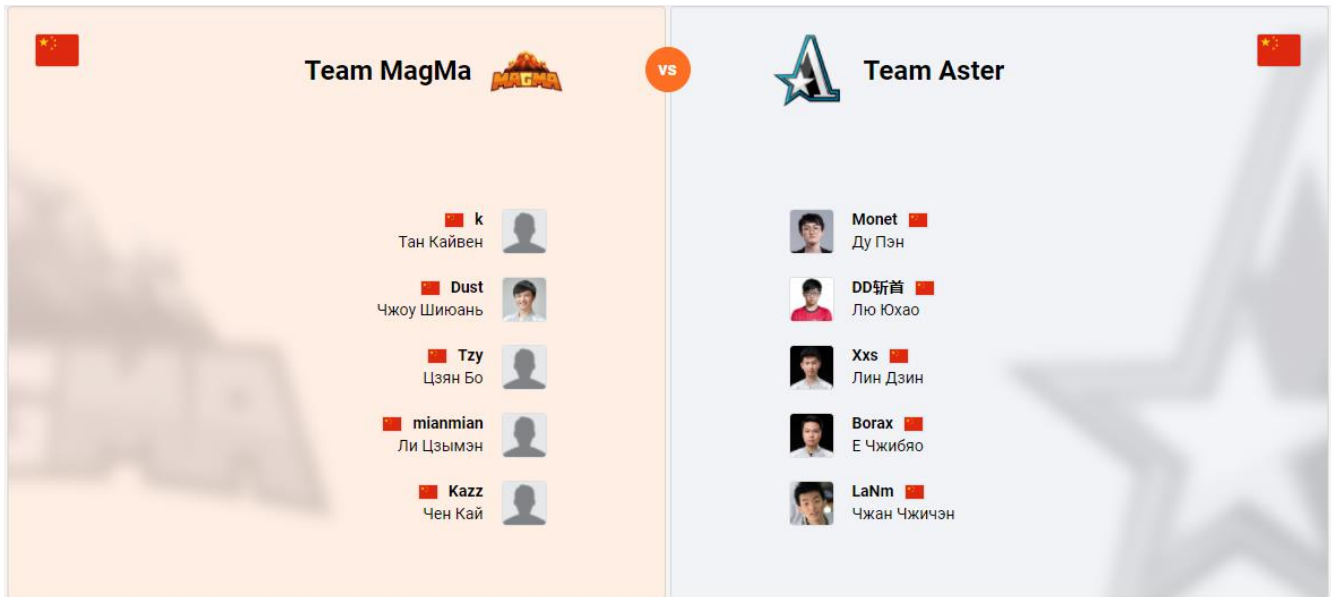


Рисунок 3.7 Деталі матчу [23]

Ці дані будуть також відображатися не тільки для майбутніх матчів, але також і для минулих та поточних, тому їхній синтаксичний аналіз не буде відрізнятися.

Також по кожному з матчів буде відображатися статистика результатів минулих матчів команд (див. рис. 3.8), які ми можемо отримати. Їхня структура та дані ідентичні до тих, які ми отримали дістаючи інформацію про вже пройдені матчі. А тому ми можемо використати той ж самий метод синтаксичного аналізу, що і раніше, але до іншого елемента DOM. І таким чином отримати результати кожної з команд.

TEAM MAGMA				
Дата	Матч		Турнір	
07.05.2021	Team MagMa	0 : 2	PSG.LGD	i-League 2021 Груповий етап
04.05.2021	Team MagMa	2 : 1	IG	i-League 2021 Груповий етап
30.04.2021	Team MagMa	0 : 2	RNG	i-League 2021 Груповий етап
29.04.2021	Team MagMa	0 : 2	VG	i-League 2021 Груповий етап

Рисунок 3.8 Результати минулих матчів однієї з команд [23]

Аналогічно до деталей матчу дана інформація буде присутня на сторінці будь якого матчу не залежно від дати його проведення.

Як було зазначено раніше основною відмінністю між дисциплінами буде інформація про поточний стан гри (див. рис. 3.9). А ще складністю є те, що сайт по дивному підтягує дані про поточні результати і їх досить складно дістати за допомогою синтаксичного аналізу. А тому в даному випадку було використано комбінація аналізу для отримання необхідних елементів DOM сторінки і регулярних виразів для діставання самих даних. Регулярні вирази – це формальна мова пошуку і здійснення необхідних маніпуляцій з підрядками в тексті. Вони основуються на використанні метасимволів і для пошуку використовується рядок-зразок, який складається з цих символів і задає правила пошуку. Завдяки цим інструментам ми можемо отримати такі дані про поточний стан гри:

- Псевдонім гравця для його ідентифікації
- Герой, яким управляє гравець
- Кількість вбивств
- Кількість смертей
- Кількість допомог
- Кількість добитих крипів, персонажів гри за вбивство яких гравець отримує золото для покупки предметів
- Кількість золота за хвилину
- Кількість досвіду за хвилину
- Нанесена шкода героям
- Кількість вилікуваного здоров'я союзникам

- Нанесена шкода спорудам
- Статистика по поставленим вардам, предмети який дають огляд карти
- Предмети гравця












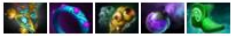



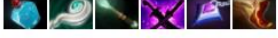




Світ: T1											
	Kuku	4	1	21	191 / 30	497 / 647	24101	-	1994	-/-	
	Karl	7	3	19	209 / 11	697 / 677	28759	-	888	-/-	
	Whitemon	2	3	22	73 / 3	340 / 549	9548	17683	989	-/-	
	Xepher	5	3	19	26 / 2	301 / 434	10454	392	609	-/-	
	23savage	11	0	7	444 / 12	901 / 752	27750	-	24812	-/-	
Тьма: Excretion											
	RR	0	7	7	30 / 9	183 / 254	9585	5603	-	-/-	
	Nikko	2	7	6	95 / 6	285 / 356	10174	-	-	-/-	
	Shaun Murphy	3	8	5	228 / 9	458 / 540	28204	115	138	-/-	
	BDz	2	4	6	10 / 4	177 / 314	1789	14951	69	-/-	
	o p s	3	4	4	302 / 13	563 / 473	30356	-	548	-/-	

Рисунок 3.9 Інформація про поточний стан гри [23]

Це всі дані які необхідні для того, щоб оцінити поточний стан гри , а також для змоги зробити певний її аналіз. На самому сайті вони оновлюються динамічно. Але ми можемо запускати наш синтаксичний аналіз кожні декілька секунд і тому інформація про стан гри буде завжди актуальна.

3.3.2 Архітектура проєкту

Для розробки даного проєкту використовувалася багаторівнева архітектура проєктування [24]. Кожен з рівнів є окремим проєктом і представляє собою функціонал, який відповідає за свою логіку і в правильному порядку посилаються один на одного. Таким чином у розробленому проєкті є три рівні:

- **Application layer.** Рівень бізнес-логіки, на якому розміщені всі моделі представлення даних, а також методи для отримання даних від користувача, її обробці за допомогою сервісів, і повернення необхідної інформації отримувачу.
- **Infrastructure layer.** Рівень сервісів, на якому розміщені самі сервіси для функціоналу синтаксичного аналізу сайтів для моніторингу кіберспортивних матчів, оскільки цей рівень це бібліотека класів на платформі .NET 5, то його з легкістю можна винести в окремий пакет, який можна використовувати в інших додатках.
- **Presentation layer.** Рівень представлення, який реалізований у вигляді веб-сервісу написаного з використанням технології ASP.NET. Це той рівень з яким і буде взаємодіяти користувач через web-арі. Він містить контролери, які обробляють запит користувача і відправляють його на Application рівень для їх подальшої обробки.

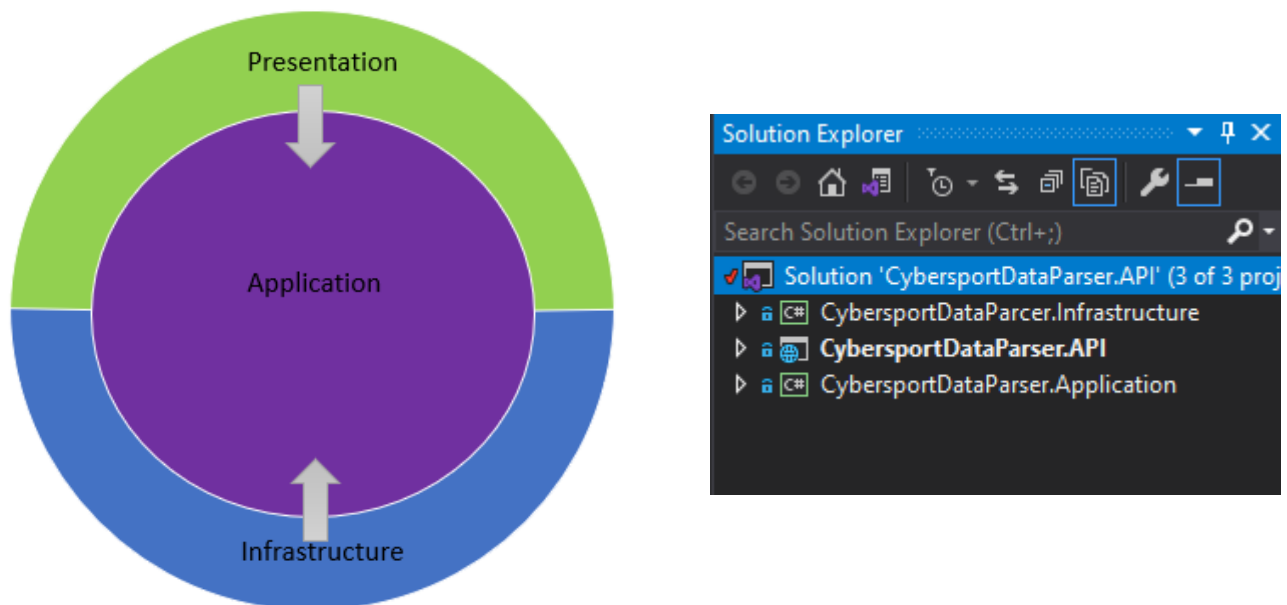


Рисунок 3.10 Архітектура проекту: схематичне представлення та сама структура

Компоненти в даній архітектурі мусять бути слабо зв'язними, а тому невід'ємною складовою проєкту є впровадження залежностей, яке для збереження термінології далі будемо називати *dependency injection*.

Dependency injection (DI) – представляє собою механізм, який дозволяє зробити слабо зв'язними об'єкти, які взаємодіють між собою в додатку. Такі об'єкти пов'язані між собою за допомогою абстракцій, наприклад, через інтерфейси, які роблять всю систему більш адаптивною, гнучкою і розширюваною. Зазвичай для встановлення залежностей в таких системах використовуються спеціальні IoC (*Inversion of Control*) контейнери, які по суті є фабриками, які займаються встановленням залежності між абстракціями і конкретно реалізованими об'єктами, а також управляють створенням цих об'єктів.

В попередніх версіях ASP.NET для встановлення залежностей потрібно було використовувати зовнішні IoC контейнери такі як: Ninject, Autofac, Unity, Windsor Castle, StructureMap та інші. Але нова версія надає власний механізм для виконання цих функцій, який представляється інтерфейсом *IServiceProvider*. Самі залежності дуже часто називають сервісами, а тому даний механізм і має назву провайдер сервісів. Цей контейнер відповідає за встановлення залежностей з конкретними типами, а також, за встановлення залежностей в різного роду об'єктів.

Таким чином, ми можемо оголосити абстракцію у вигляді інтерфейсів на *Application* рівні, реалізувати сервіси в *Infrastructure* рівні і в ньому ж реалізувати метод розширення для *IServiceProvider*, щоб зробити залежність між абстракцією, та конкретною реалізацією сервісу. І таким чином ми можемо працювати з нашими сервісами через інтерфейси у будь якому об'єкті програми, просто передавши залежність в конструктор об'єкту.

Оскільки рівень представлення відповідає за отримання та віддачу даних, в ньому не мусить міститися жодної логіки, а тому ми не повинні працювати напяму з реалізованими сервісами. В цьому нам допоможе об'єкт посередник. Посередник - це шаблон проектування, який відповідає за інкапсуляцію та роз'єднання взаємодіючих об'єктів в системі. Це шаблон поведінкового типу, визначення якого говорить, що слід уникати з'єднання двох компонентів, що взаємодіють між собою, а також має бути можливість динамічно змінювати взаємодію між компонентами, не порушуючи роботи системи [25].

Також для забезпечення розширюваності системи було вирішено використати CQRS (Command Query Responsibility Segregation). CQRS – це шаблон архітектурного проектування, який відповідає за розділення або групування методів на основі того, як вони взаємодіють з даними, та проектують їх залежно від вимог. Цей шаблон класифікує дві різних поведінки: команди та запити. Команди – це методи, які модифікують хоч якусь частину набору даних, а запити це ті методи, які не впливають на набір даних, а тільки отримують їх.

Шаблон посередника створює новий проміжний шар між компонентами, у якому і будуть зібрані усі необхідні залежності за допомогою Dependency Injection. Компоненти в системі делегують виклики своїх функції посереднику і він вирішує який компонент для чого потрібно викликати. Цей спосіб гарантує ту необхідну для розширюваності слабку зв'язність. А також це створює можливість замінювати реалізацію компонента без впливу на роботу системи. У CQRS за запити та команди відповідають спеціальні методи обробники і при розробці стає складно зв'язати ці методи з їхніми обробниками, ось чому посередник дуже корисний, адже він інкапсулює ці взаємодії та турбується про зв'язування за нас. Таким чином всі об'єкти в системі стають слабо зв'язними і тим самим проект легко тестувати,

досить просто розширювати, а також, при необхідності, реалізації сервісів можна легко замінити не порушуючи роботу системи.

3.3.3 Авторизація

Для забезпечення безпеки веб-сервісу, а також контролю перенавантажень і збирання статистики запитів серед користувачів було вирішено додати авторизацію для реалізованого web-арі. Авторизація основана на так званих JWT (JSON Web Token) токенах. JWT-токен представляє собою веб-стандарт, який надає спосіб передачі даних про користувача у зашифрованому варіанті.

JWT-токен складається з трьох частин:

- Header – JSON об’єкт, який містить в собі інформацію про тип токена, і алгоритмі його шифрування.
- Payload – JSON об’єкт, який містить в собі дані які необхідні для авторизації користувача.
- Signature – рядок, який створюється за допомогою секретного коду, Headera і Payload. Цей рядок використовується для того, щоб верифікувати токен.

Додатки зазвичай використовують два фундаментальних шляхи для використання різного роду захищених арі – використання ідентичності додатку, або ж управління ідентичності користувача. OAuth2 – це протокол, який дозволяє програмам отримувати токени доступу в єдиного централізованого сервера, який відповідає за видачу і використовувати ці токени для забезпечення зв’язку з арі.

Для реалізації авторизації на проекті було використано IdentityServer версії 4. IdentityServer це middleware який додає кінцеві точки запитів для отримання JWT-

токенів, які створюються відповідно від користувача. Після встановлення пакету нам необхідно його налаштувати. Для цього потрібно створити загальний набір прав доступу. А також додати самих клієнтів. Оскільки користувачами нашого веб-сервісу будуть додатки, то клієнти мусять мати такі необхідні дані:

- Унікальний ідентифікатор клієнта
- Ім'я клієнта
- Секретний ключ
- Список можливих методів отримання токенів
- Список доступних прав доступу

Ці дані будуть зберігатися на сервері авторизації, а також надаватися кожному окремому клієнту. І кінцевий користувач маючи ці дані, зможе відправити їх на спеціальну кінцеву точку та обміняти їх на персональний JWT-токен, який йому необхідно буде відправляти разом з кожним запитом на розроблену арі.

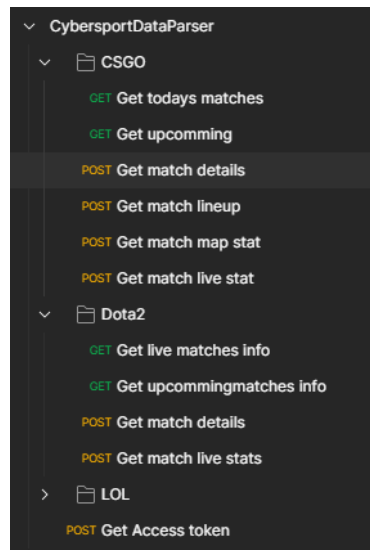
Зазвичай великі веб сервіси мають декілька арі і сервер авторизації розміщений окремо, але в нашому випадку є лише одне арі, а також один метод авторизації, тому вся робота з авторизацією реалізована в одному сервісі.

3.3.4 Тестування веб-сервісу

Для забезпечення коректної роботи веб-сервісу потрібно налаштувати покриття реалізованого web-арі тестами, як автоматичними так і ручними. Для автоматичного тестування було використано Postman. Postman – це платформа для розробки арі. Він надає інструменти для спрощення розробки веб сервісів на

кожному кроці, а також функціонал зручного тестування розроблених проєктів. Сам Postman має графічний інтерфейс, а тому для тестування необхідно створити окрему колекцію для арі (див. рис. 3.11), описати всі необхідні запити які ми хочемо протестувати і запустити самі тести. У разі успішного проходження повернеться відповідь з тілом і успішним кодом відповіді, а якщо тест звалиться, то повернеться помилка з текстом помилки.

Рисунок 3.11 Колекція тестів розробленого арі



Для забезпечення ручного тестування було обрано інструмент Swagger. Swagger – це фреймворк, який перш за все створений для специфікації арі. Його головною перевагою є те, що він дає можливість не тільки інтерактивно переглядати програмну специфікацію, а і відправляти необхідні запити вручну за допомогою так званого Swagger UI (див. рис. 3.12). Тобто тестування це не основна функція, але завдяки використанні цього інструменту нам не потрібно писати документацію вручну. Всі необхідні деталі можна запрограмувати і Swagger автоматично згенерує специфікації з усіма присутніми кінцевими точками, а також з моделями даних, які необхідно відправити і які повертаються. Також даний

інструмент підтримує авторизацію і не дозволяє вручну тестувати арі без попередньо отриманого JWT-токена доступу.

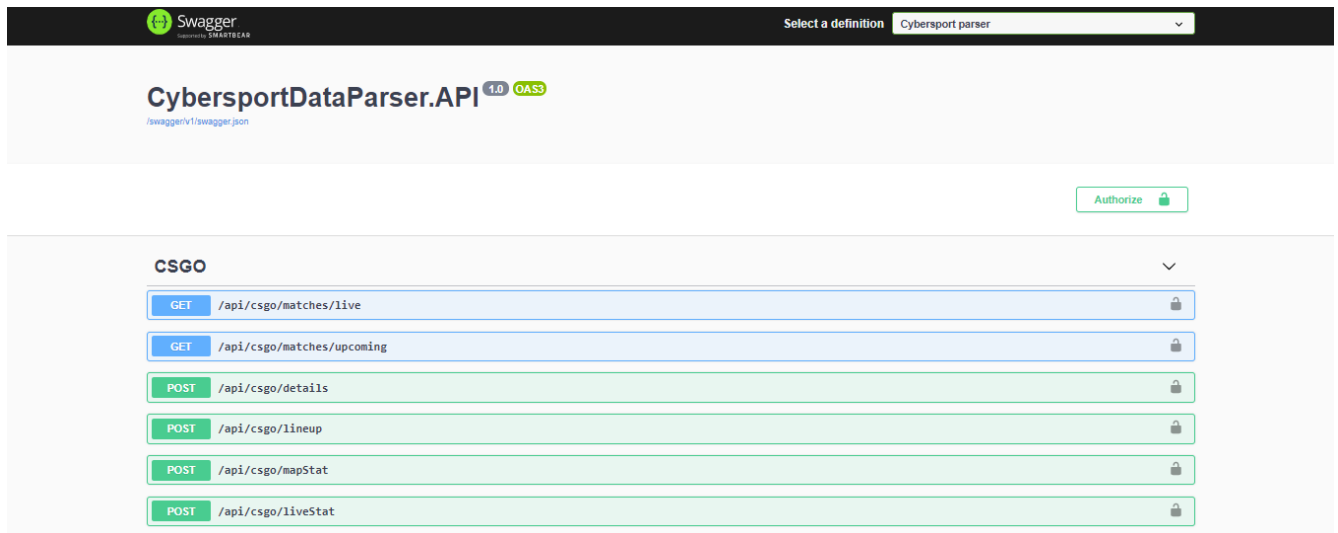


Рисунок 3.12 Частина специфікації згенерованої за допомогою Swagger

ВИСНОВКИ

В результаті виконання даної кваліфікаційної роботи був розроблений веб-сервіс, який вирішує проблему отримання даних кіберспортивних матчів з мінімально можливою затримкою. Було проведено аналіз функціоналу вже існуючих на сьогоднішньому ринку ресурсів та сайтів, які займаються наданням статистики та новин з різного роду кіберспортивних дисциплін. Створено веб-арі для отримання актуальних даних з турнірних матчів по комп'ютерним іграм з використанням Open Source бібліотеки Selenium мовою програмування С# для легкої інтеграції з будь якими додатками на різних платформах, а також розроблено якісну архітектуру завдяки якій основний функціонал синтаксичного аналізу кіберспортивних ресурсів можна винести в окремий пакет для подальшого інтегрування в у додатки на платформі .NET.

При виконанні кваліфікаційної роботи був проведений аналіз існуючих способів, для того, щоб отримувати необхідні дані напряму з гри за допомогою арі, яку надають самі розробники. А також бібліотеки та фреймворки для синтаксичного аналізу DOM сторінок сайтів. Беручи до уваги кросплатформенісь мови програмування С# та платформи .NET, було обрано бібліотеку Selenium, яка підтримується багатьма операційними системами та містить рішення для багатьох проблем синтаксичного аналізу веб-сторінок. Були досліджені методи бібліотеки, які дозволяють вирішити поставлену задачу, а також способи їх оптимізації для забезпечення мінімально можливої затримки між відправленням запитів і отриманням необхідних даних.

У ході виконання кваліфікаційної було створено багатопроєктний веб сервіс, з сучасною архітектурою, який поставляється у вигляді web-арі, для моніторингу даних кіберспортивних матчів у режимі реального часу з мінімально можливою

затримкою. Розроблена архітектура забезпечує слабку зв'язність між об'єктами системи та високий рівень абстракції. Завдяки чому система є адаптивною, гнучкою, та легко розширюваною. При необхідності можливо змінити реалізацію методів синтаксичного аналізу без впливу на роботу системи. Також для забезпечення безпеки та контролю перенавантаження було додано авторизацію за допомогою JWT-токенів. Веб-сервіс повністю покритий автоматичними і ручними тестами для забезпечення коректної роботи, а також наявна програмно згенерована документація по кінцевим точкам web-арі.

Під час написання дипломної роботи я набув практичних навичок програмування, застосував теоретичні знання на практиці. Навчився вирішувати різноманітні завдання синтаксичного аналізу веб-сторінок та використав отримані знання для створення веб-сервісу. Поставлені задачі були виконані у повному обсязі.

Надалі планується розширення функціоналу для додавання нових кіберспортивних дисциплін. А також розробка алгоритмів синтаксичного аналізу для зменшення затримки між відправленням даних та їх отриманням. А також повністю винести основний рівень з сервісами в окремий пакет для синтаксичного аналізу потрібних ресурсів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Esports [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Esports>
2. Spacewar! [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Spacewar!>
3. S P A C E W A R [Електронний ресурс] – Режим доступу до ресурсу: https://www.wheels.org/spacewar/stone/rolling_stone.html
4. First Spacewar Tournament [Зображення] – Режим доступу до ресурсу: <https://www.esports.net/wiki/esports-history/>
5. Red Annihilation [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Red_Annihilation
6. The International 2011 [Електронний ресурс] – Режим доступу до ресурсу: https://liquipedia.net/dota2/The_International/2011
7. Dota 2 Tournaments Prize Pool [Електронний ресурс] – Режим доступу до ресурсу: <https://dota2.prizetrac.kr/>
8. Streaming Services Comparing [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pcmag.com/picks/best-video-game-live-streaming-services>
9. Twitch Games Statistic [Електронний ресурс] – Режим доступу до ресурсу: <https://twitchtracker.com/statistics/games>
10. Dota 2 [Електронний ресурс] – Режим доступу до ресурсу: <https://dota2.ru/>

11. Valorant Esports [Электронный ресурс] – Режим доступа до ресурсу:
<https://valorantesports.com/>
12. EsL Gaming [Электронный ресурс] – Режим доступа до ресурсу:
<https://pro.eslgaming.com/>
13. Games Tournament [Электронный ресурс] – Режим доступа до ресурсу:
<https://game-tournaments.com/>
14. GGScore [Электронный ресурс] – Режим доступа до ресурсу:
<https://ggscore.com/>
15. HLTV [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.hltv.org/>
16. Dota Workshop Tools [Электронный ресурс] – Режим доступа до ресурсу:
https://developer.valvesoftware.com/wiki/Dota_2_Workshop_Tools
17. Dotabuff [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.dotabuff.com/>
18. Cybersport news [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.cybersport.ru/>
19. .NET 5 [Электронный ресурс] – Режим доступа до ресурсу:
<https://docs.microsoft.com/en-us/dotnet/core/dotnet-five>
20. Selenium [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.selenium.dev/>
21. HLTV Matches [Зображення] – Режим доступа до ресурсу:
<https://www.hltv.org/matches>

22. NAVI vs HEROIC [Зображення] – Режим доступу до ресурсу:
<https://www.hltv.org/matches/2348012/heroic-vs-natus-vincere-dreamhack-masters-spring-2021>
23. Cybersport matches [Зображення] – Режим доступу до ресурсу:
<https://www.cybersport.ru/base/match>
24. Clean Architecture [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.youtube.com/watch?v=5OtUm1BLmG0>
25. Head First. Паттерны проектирования. Фріман Є., Сьерра К., Бейтс Б., Робсон Є. 2018