

Державний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Автоматизована система «Gaming Trends» підбору і
сортування ігрових трендів з платформи YouTube»**

Студента 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

підпис студента

Капусти Нікити
Вячеславовича

Науковий керівник
кандидат педагогічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Жирова Тетяна
Олександрівна

Гарант освітньої програми
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення
та кібербезпеки

підпис гаранта

Рзаєва Світлана
Леонідівна

КИЇВ – 2022

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

"09" листопада 2021 р.

Завдання

на випускню кваліфікаційну роботу студентіві

Капусті Никіті Вячеславовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube»

Затверджена наказом ректора від "29" листопада 2021 р. № 3928

2. Строк здачі студентом закінченої роботи «06» червня 2022 р.

3. Цільова установка та вихідні дані до роботи

Мета дослідження проаналізувати та дослідити актуальність і методи створення автоматизованої системи «Gaming Trends» для підбору і сортування ігрових трендів з платформи YouTube з подальшою її реалізацією.

Об'єкт дослідження проєктування та програмна реалізація автоматизованої системи «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube.

Предмет дослідження автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи(перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ, ПІДХОДІВ ДО СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

Аналіз предметної області та маркетингові дослідження

Виявлення проблеми та дослідження шляхів її вирішення

Вибір потенційних користувачів

Визначення основних функцій та принципу роботи системи

Технічне завдання проєкт

Загальні відомості

Мета та призначення

Вимоги до системи

Вибір середовища та інструментів розробки

Висновки до розділу 1

РОЗДІЛ 2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розробка архітектури програмного забезпечення

Діаграма прецедентів (use case diagram)

Діаграма класів (class diagram)

Діаграма кооперації (collaboration diagram)

Діаграма послідовностей (sequence diagram)

Діаграма стану (state diagram)

Діаграма діяльності (activity diagram)

Діаграма компонентів (component diagram)

Діаграма розгортання (deployment diagram)

Проектування архітектури бази даних

Створення концептуальної моделі

Створення логічної моделі

Створення фізичної моделі

Створення інтерфейсу і дизайну вебсайту

Висновки до розділу 2

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

Налаштування інструментів розробки

Створення та налаштування проєкту в середовищі розробки PyCharm

Налаштування системи контролю версій Git і GitHub

Створення бази даних

Розробка сторінок вебсайту

Підключення бази даних і створення моделей

Отримання трендових відео за допомогою YouTube API

Сортування та фільтрування списку відео

Реалізація CRUD (створення, зчитування, зміна і видалення)

Збереження списку відео

Розробка тестів

Створення REST та виведення веб-сторінок

Опис роботи програми

Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	
		За планом	Фактично
1.	Вибір теми випускної кваліфікаційної роботи	20.09.2021	20.09.2021
2.	Вступ та перелік літературних джерел	13.12.2021	13.12.2021
3.	Розділ 1. «ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ, ПІДХОДІВ ДО СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ»	18.02.2022	18.02.2022
4.	Розділ 2. «ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	04.03.2022	04.03.2022
5.	Розділ 3. «РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ»	11.04.2022	11.04.2022
6.	Висновки	25.04.2022	25.04.2022
7.	Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)	25.05.2022	25.05.2022
8.	Підготовка автореферату та презентації доповіді	26.05.2022	26.05.2022
9.	Попередній захист випускної кваліфікаційної роботи	30.05.2022 - 03.06.2022	01.06.2022
10.	Зовнішнє рецензування випускної кваліфікаційної роботи	04.06.2022	04.06.2022
11.	Здача прошитою випускної кваліфікаційної роботи	06.06.2022	06.06.2022
12.	Публічний захист випускної кваліфікаційної роботи		

7. Дата видачі завдання «09» листопада 2021 р.

8. Науковий керівник випускної кваліфікаційної роботи

Жирова Т. О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Рзасва С. Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Капуста Н. В.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

Дипломна робота на тему «Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube» містить:

1. Загальний обсяг сторінок: 47
2. Кількість рисунків: 17
3. Кількість додатків: 9
4. Кількість використаних джерел: 12

Мета дослідження: проаналізувати та дослідити актуальність і методи створення автоматизованої системи «Gaming Trends» для підбору і сортування ігрових трендів з платформи YouTube з подальшою її реалізацією.

Об'єкт дослідження: проектування та програмна реалізація автоматизованої системи «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube.

Предмет дослідження: автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube.

У першому розділі проведено аналіз предметної області та маркетингові дослідження, виявлено проблему та розглянуто шляхи її вирішення, визначено потенційних користувачів, основні функції та принципи роботи системи. Також було створено технічне завдання проекту і обрані основні інструменти для розробки.

Другий розділ включає в себе проектування системи за допомогою UML діаграм. Було розроблено наступні види діаграм: діаграма прецедентів (use case diagram), діаграма класів (class diagram), діаграма кооперації (collaboration diagram), діаграма послідовностей (sequence diagram), діаграма стану (state diagram), діаграма діяльності (activity diagram), діаграма компонентів (component diagram), діаграма розгортання (deployment diagram). Для розробки бази даних було спроектовано концептуальну, логічну та фізичну моделі бази даних. Також було створено інтерфейс та дизайн сторінок сайту.

У третьому розділі була проведена розробка програмного забезпечення, що включає: налаштування інструментів розробки, реалізація сторінок сайту, створення та підключення бази даних, розробка функціональної частини системи, тестів. Також було описано як працює програмне забезпечення, пояснено як працювати з інтерфейсом.



ANNOTATION

The thesis on the topic “Automated system ‘Gaming Trends’ to select and sort gaming trends from Youtube” consists:

1. Pages: 27
2. Drawings: 17
3. Attachments: 9
4. Used sources: 12

The aim of the study: analyze and explore the relevance and methods of creating the automated system "Gaming Trends" to select and sort gaming trends from YouTube with implementation.

Object of study: design and software implementation of the automated system "Gaming Trends" to select and sort gaming trends from YouTube.

Subject of study: the automated system "Gaming Trends" to select and sort gaming trends from YouTube.

The first section analyzes the subject area and marketing research, identifies the problem and ways to solve it, identifies potential users, the main functions and principles of the system. The terms of reference of the project have been also created and the main tools for development have been selected.

The second section includes designing the system using UML charts. The following types of diagrams have been developed: use case diagram, class diagram, collaboration diagram, sequence diagram, state diagram, activity diagram, activity diagram, component diagram, deployment diagram. Conceptual, logical and physical models of the database have been designed to develop the database. The interface and design of the site pages have also been created.

In the third section, software has been developed. Development included: setting up development tools, implementing site pages, creating and connecting a database, developing the functional part of the system, and tests. This part also describes how the software works, and explains how to work with the interface.

Зміст

ВСТУП.....	4
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ, ПІДХОДІВ ДО СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	7
1.1 Аналіз предметної області та маркетингові дослідження	7
1.2 Виявлення проблеми та дослідження шляхів її вирішення	9
1.3 Вибір потенційних користувачів	10
1.4 Визначення основних функцій та принципу роботи системи.....	11
1.4 Технічне завдання проєкту.....	13
1.4.1 Загальні відомості	13
1.4.2 Мета та призначення.....	14
1.4.3 Вимоги до системи.....	14
1.5 Вибір середовища та інструментів розробки	15
1.6 Висновки до розділу 1	18
РОЗДІЛ 2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	20
2.1 Розробка архітектури програмного забезпечення	20
2.1.1 Діаграма прецедентів (use case diagram).....	21
2.1.2 Діаграма класів (class diagram)	22
2.1.3 Діаграма кооперації (collaboration diagram)	23
2.1.4 Діаграма послідовностей (sequence diagram)	24
2.1.5 Діаграма стану (state diagram).....	25
2.1.6 Діаграма діяльності (activity diagram).....	26
2.1.7 Діаграма компонентів (component diagram)	28

					<i>ДТЕУ 121 06-09.БР</i>			
					<i>Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Стадія	Аркуш	Аркушів	
Зав. каф.		Криворучко О.В.		25.04.22	Зміст	2	46	
Керівник		Жирова Т. О.		25.04.22				
Гарант		Рзаєва С. Л.		25.04.22	<i>Зміст</i>			
Розроб		Капуста Н. В.		25.04.22	Факультет інформаційних технологій, 4 курс, 6 група			

2.1.8	Діаграма розгортання (deployment diagram).....	28
2.2	Проектування архітектури бази даних.....	29
2.2.1	Створення концептуальної моделі.....	29
2.2.2	Створення логічної моделі.....	30
2.2.3	Створення фізичної моделі.....	30
2.3	Створення інтерфейсу і дизайну вебсайту.....	31
2.4	Висновки до розділу 2.....	32
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ.....		33
3.1	Налаштування інструментів розробки.....	33
3.1.1	Створення та налаштування проєкту в середовищі розробки PyCharm.....	33
3.1.2	Налаштування системи контролю версій Git і GitHub.....	35
3.1.3	Створення бази даних.....	35
3.2	Розробка сторінок вебсайту.....	36
3.3	Підключення бази даних і створення моделей.....	36
3.4	Отримання трендових відео за допомогою YouTube API.....	37
3.4	Сортування та фільтрування списку відео.....	38
3.5	Реалізація CRUD (створення, зчитування, зміна і видалення).....	39
3.6	Збереження списку відео.....	39
3.7	Розробка тестів.....	40
3.8	Створення REST та виведення вебсторінок.....	40
3.9	Опис роботи програми.....	41
3.10	Висновки до розділу 3.....	42
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....		43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		45
ДОДАТКИ.....		47

ВСТУП

Розвиток і потенціал сучасних інформаційних технологій нині можна оцінити на високому рівні. Використовуючи різні інформаційні технології, можна створити і реалізувати безліч цікавих, корисних та прибуткових проєктів. Секрет успішного проєкта і стартапа – це вирішення проблем клієнта. Саме інформаційні технології дають можливість закрити якусь проблему користувача, автоматизувати її вирішення і, як наслідок, спростити роботу і зекономити час. Це можливе в основному завдяки різним методам програмування, алгоритмам, веб-програмуванню, базам даних (БД) і системам керування базами даних. Нині неможливо представити будь-який проєкт, якщо він хоча б якось пов'язаний з інформатизацією, без належного вебсайту, досконало робочого програмного забезпечення та правильно спроектованої, налаштованої бази даних.

У цій роботі буде розглянута розробка автоматизованої системи підбору і сортування ігрових трендів з платформи YouTube під назвою “Gaming Trends”. Для даної автоматизованої системи буде розроблено базу даних для зберігання списку трендів, вебсайт для зручного доступу користувачів з будь-якого девайсу.

На сьогоднішній день платформа YouTube є лідером серед відеохостингів. Мільйони користувачів активно використовують дану площадку: заходять, дивляться різні відео, підписуються на блогерів, ставлять лайки та коментують відео. Та безліч авторів знімають відео, публікують на своїх каналах і отримують дохід завдяки монетизації та рекламним інтеграціям. Чим більше переглядів, тим більше дохід. Отже, креатори максимально зацікавлені в розвитку своїх каналів і збільшенні кількості переглядів. Найкращий спосіб отримувати максимальну кількість переглядів – знімати відео на актуальні та трендові теми. Вибір вдалої ідеї для відео – це найбільша частина його успіху.

					<i>ДТЕУ 121 06-09.БР</i>			
					<i>Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube</i>	<i>Стадія</i> В	<i>Аркуш</i> 4	<i>Аркушів</i> 46
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			13.12.21				
Керівник	Жирова Т. О.			13.12.21				
Гарант	Рзаєва С. Л.			13.12.21				
Розроб	Капуста Н. В.			13.12.21	<i>Вступ</i>			
						Факультет інформаційних технологій, 4 курс, 6 група		

Спілкуючись з авторами геймерських каналів, було виявлено, що вони потребують більше ідей для своїх відео, а саме трендових і актуальних тем. Як правило, відмінним рішенням – є перегляд англomовних трендів в YouTube і адаптація їх для нашої аудиторії. Отже, була виявлена проблема і знайдено спосіб її вирішення. Щоб спростити і автоматизувати пошук трендових відео було вирішено розробити дане програмне забезпечення.

У цій роботі буде розроблена автоматизована-система “Gaming Trends”, що допоможе користувачу в зручному для нього вигляді отримати список останніх трендових відео з англomовного YouTube підібраних за певною грою. Додаток допоможе знизити витрати часу і зробити процес підбору ідей, аналізу трендів, значків та метаданих відео більш зручним для авторів YouTube каналів, менеджерів, сценаристів, аналітиків ігрового сегменту та маркетологів.

Крім того, створення автоматизованої системи підбору та сортування англomовних трендів з платформи YouTube є актуальним у наш час, адже конкурентного подібного програмного забезпечення не існує, даний проєкт буде передовим на ринку.

Проаналізувавши всі доступні фактори, актуальність проблеми і її затребуваність для клієнтів було визначено наступну мету, завдання та об'єкт дослідження.

Мета дипломної роботи: проаналізувати та дослідити актуальність і методи створення автоматизованої системи «Gaming Trends» для підбору і сортування ігрових трендів з платформи YouTube з подальшою її реалізацією.

Об'єкт дослідження: проєктування та програмна реалізація автоматизованої системи «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube.

Предмет дослідження: автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube.

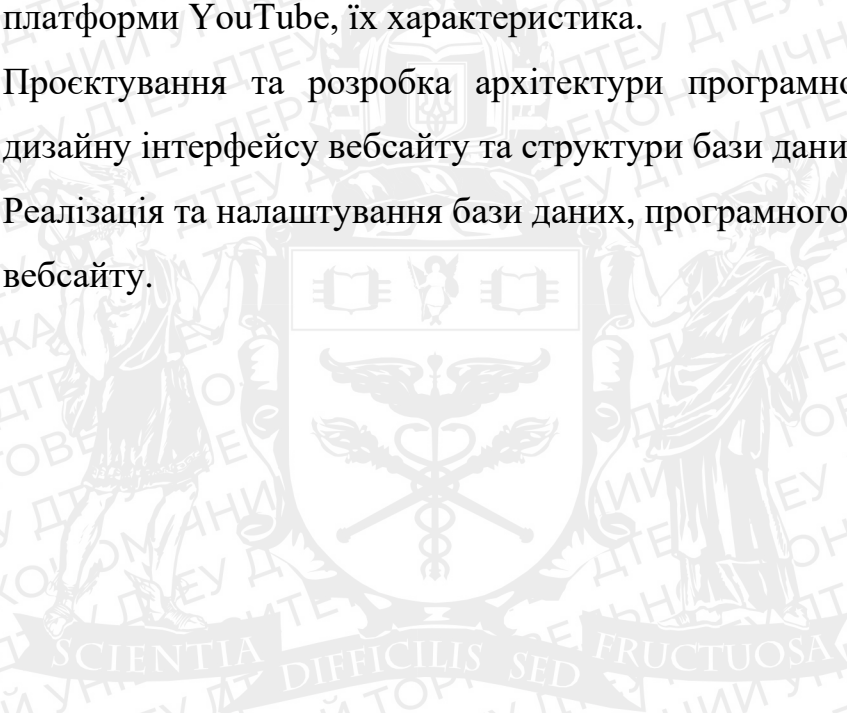
Завдання дослідження: створити автоматизовану систему підбору і сортування ігрових трендів з платформи YouTube за допомогою мови програмування Python і YouTube API, спроектувати та налаштувати базу даних,

Зм.	Аркуш	№ докум	Підпис	Дата

використовуючи PostgreSQL, розробити вебсайт і для отримання списку трендів в зручному для користувача вигляді завдяки мові програмування Python з розширенням Flask та візуалізації з HTML + CSS + Bootstrap у середовищі розробки PyCharm.

Завдання включає в себе наступні пункти:

1. Дослідження та аналіз предметної області, підходів до створення автоматизованої системи підбору і сортування ігрових трендів з платформи YouTube, їх характеристика.
2. Проектування та розробка архітектури програмного забезпечення, дизайну інтерфейсу вебсайту та структури бази даних.
3. Реалізація та налаштування бази даних, програмного забезпечення, вебсайту.



Зм.	Аркуш	№ докум	Підпис	Дата

ДТЕУ 121 06-09.БР

Аркуш

6

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ, ПІДХОДІВ ДО СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

1.1 Аналіз предметної області та маркетингові дослідження

На сьогодні Youtube - це найпопулярніша платформа для публікації і перегляду відео. Якщо порівнювати кількість користувачів платформи з другими соціальними мережами, то згідно з даними, що опубліковані на сайті www.statista.com, YouTube займає друге місце станом на жовтень 2021 року (рис. 1.1). Кількість користувачів платформи дорівнює 2,291 мільйонам.

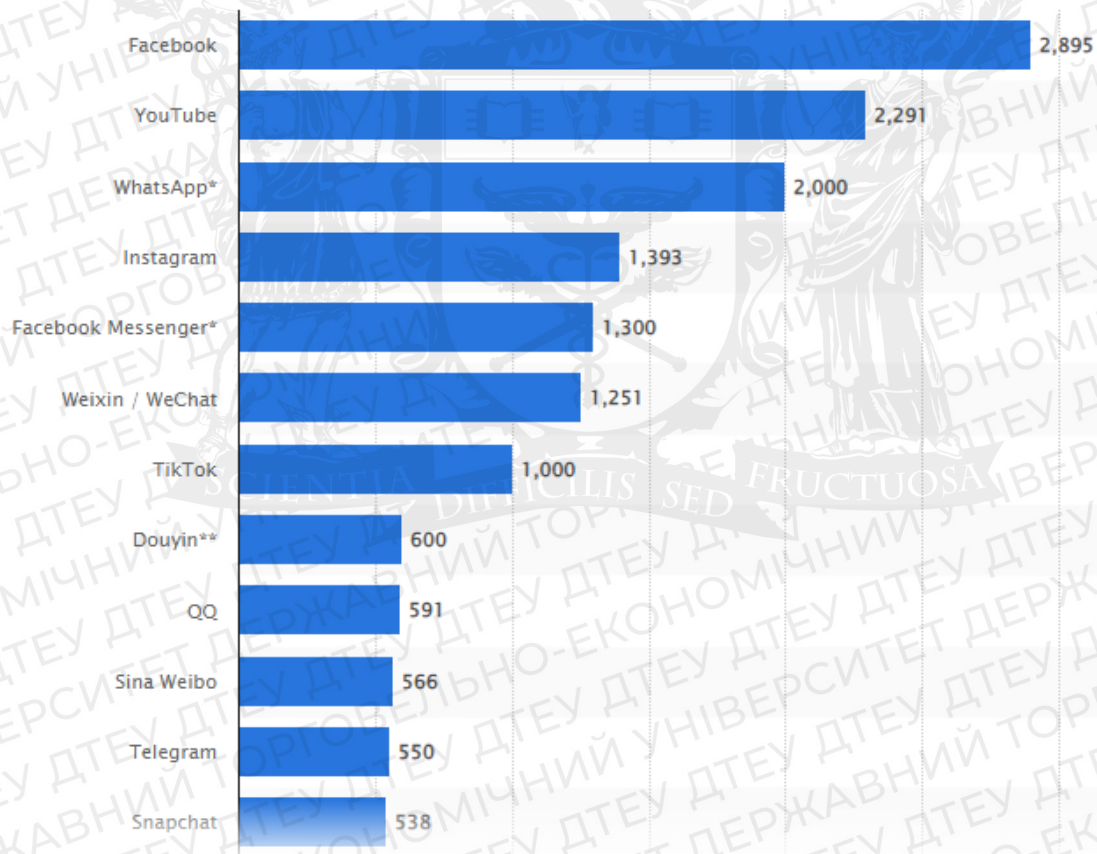


Рисунок 1.1 Найпопулярніші соціальні мережі в жовтні 2021

					<i>ДТЕУ 121 06-09.БР</i>			
					<i>Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube</i>	<i>Стадія</i> P1	<i>Аркуш</i> 7	<i>Аркушів</i> 46
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		18.02.22	<i>Дослідження предметної області, підходів до створення автоматизованої системи</i>			
Керівник		Жирова Т. О.		18.02.22				
Гарант		Рзаєва С. Л.		18.02.22				
Розроб		Капуста Н. В.		18.02.22				
					<i>Факультет інформаційних технологій, 4 курс, 6 група</i>			

При цьому можна зазначити, що платформа розвивається і з кожним роком її популярність зростає, що можна дослідити порівнявши кількість користувачів з минулими роками (рис. 1.2). Тенденція росту дуже позитивна, і на майбутнє вона повинна зберігатись.

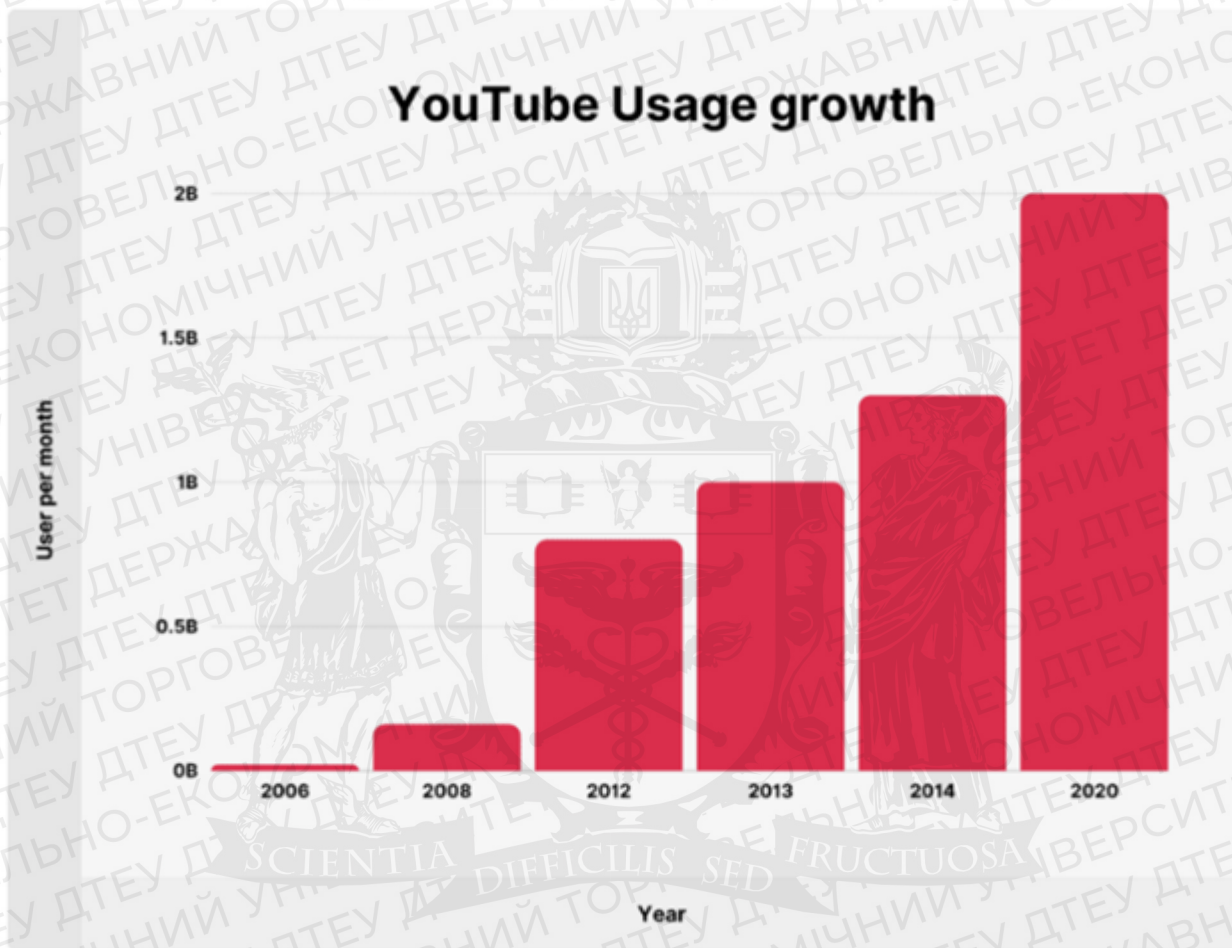


Рисунок 1.2 Порівняння кількості користувачів YouTube

Проаналізувавши ці дані, можна з упевненістю стверджувати, що реалізація проєкта, який пов'язаний з YouTube, буде актуальною сьогодні і в майбутньому.

На платформі існує безліч різного контенту, кожен глядач може обрати відео згідно зі своїми уподобаннями. Слід визначити, спрос на які тематики є найбільш затребуваний. Згідно з особистими спостереженнями, можна зазначити, що ігровий контент є одним із найпопулярніших серед користувачів. При цьому, слід звернути увагу, що дохід від відеоігор зростає з кожним роком і ця сфера діяльності активно розвивається. Це доводить графік прогнозу доходу від відеоігор з сайту www.statista.com (рис. 1.3). А, отже, геймінг сегментом все більше цікавляться з кожним роком, на YouTube в тому числі.

Зм.	Аркуш	№ докум	Підпис	Дата

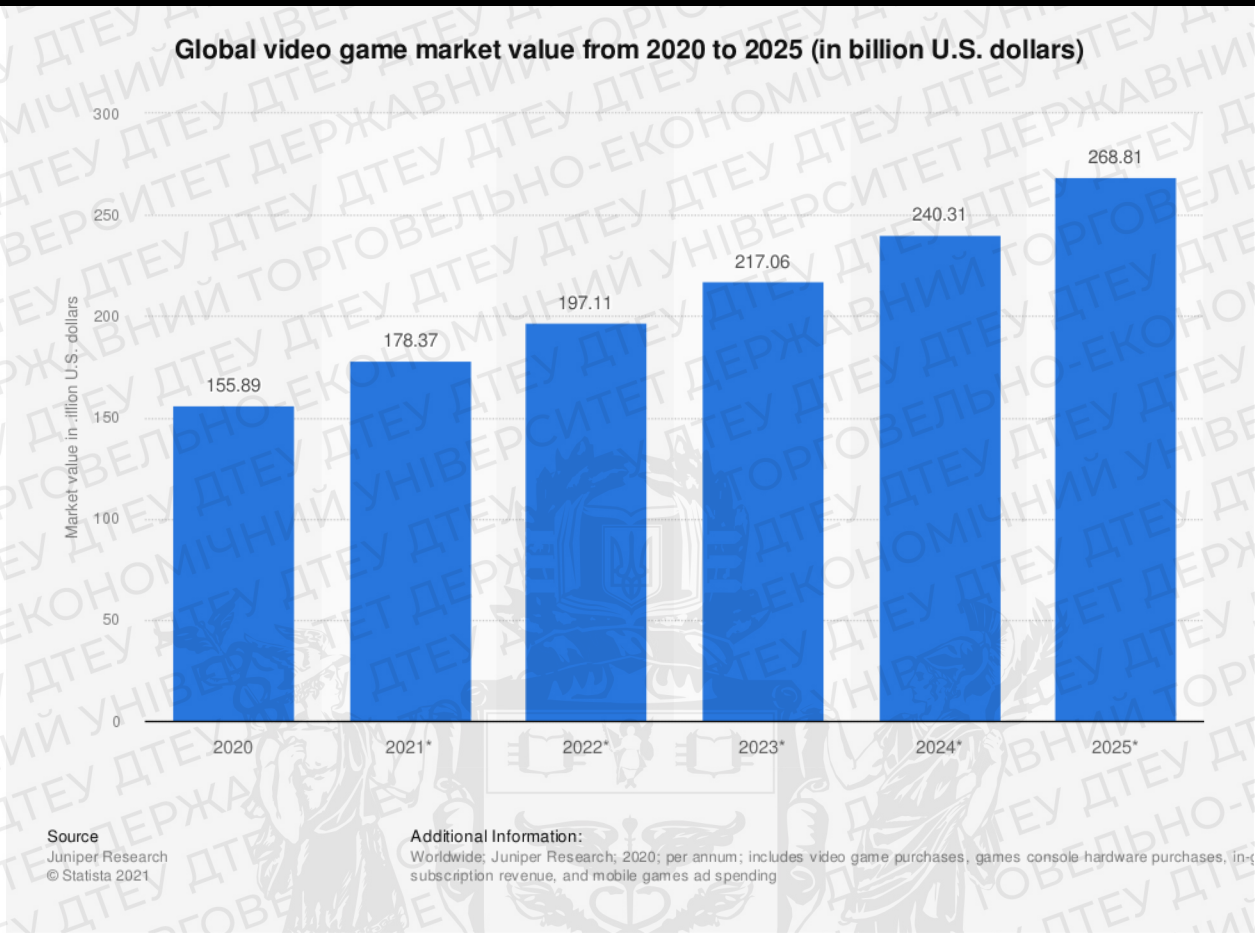


Рисунок 1.3 Прогноз доходу від сфери відеоігор

1.2 Виявлення проблеми та дослідження шляхів її вирішення

Власники ютуб каналів максимально зацікавлені в збільшенні кількості переглядів і розвитку своїх каналів, так як чим більше переглядів, тим більше дохід. Найкращий спосіб отримувати максимальну кількість переглядів – знімати відео на актуальні та трендові теми. У авторів є проблема, що пов’язана з відслідковуванням і вибором таких тем. А також, вони потребують більше ідей для своїх відео. Для цього вони навіть можуть наймати спеціального працівника, що буде виконувати роботу з підбору та генерації ідей. Один з найкращих варіантів для вирішення цієї проблеми, не звертаючись за допомогою до додаткового працівника, є перегляд англомовних трендів і адаптація їх для нашої аудиторії.

На платформі YouTube існує спеціальний розділ з трендовими відео різних категорій. Алгоритми платформи самі відслідковують найпопулярніші відеоролики і додають їх в цей розділ. Для кожної країни система підбирає свої відео. А отже, можна переключити країну в налаштуваннях та переглядати її

Зм.	Аркуш	№ докум	Підпис	Дата

тренди. Як правило, спочатку найкращі ідеї та актуальні формати з'являються в англomовних країнах. Таким чином, потрібно стежити і переглядати тренди наступних країн: США, Великобританія, Канада, Австралія.

Додатково розділ трендових відео поділяється на певні категорії: зараз(новини), музика, ігри, фільми (рис. 1.4). Кожна категорія включає в себе 30 найпопулярніших відео за допомогою яких можна проаналізувати успішні формати і використовувати їх для подальшого створення контенту.

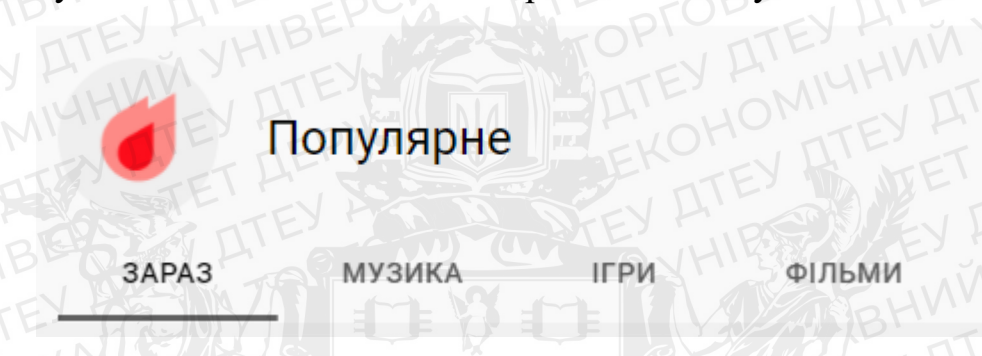


Рисунок 1.4 Розділ тренди на платформі YouTube

Досліджуючи геймінг сегмент і категорію ігри в розділі трендів, було виявлено, що саме ця категорія є найбільш вдалою і корисною для підбору ідей, оскільки кожне ігрове відео поділяється ще по назві гри. Як наслідок, можна виокремити і відсортувати всі відео з розділу за певною грою, що дозволяє зробити вибірку найбільш вдалою для різних геймерських каналів. Слід зазначити, що кожен автор свого каналу, зазвичай, прив'язаний до однієї гри і знімає саме її. Отже, йому буде корисно отримувати список відео саме зі своєю тематикою (грою). Найчастіше в трендах сьогодні з'являються наступні найпопулярніші ігри: Minecraft, GTA 5, Roblox і Fortnite. Саме на них програмне забезпечення буде орієнтуватись і сортувати відео за цими іграми.

1.3 Вибір потенційних користувачів

Щоб визначити потенційних користувачів для розроблювальної системи, перш за все, потрібно визначити мету проекту. Мету можна сформулювати наступним чином: знизити витрати часу і зробити процес підбору ідей, аналізу трендів, значків та метаданих відео більш зручним. Тобто дане програмне забезпечення буде доцільне і корисне всім, кому потрібно підбирати ідеї для

Зм.	Аркуш	№ докум	Підпис	Дата

контенту, створювати значки або формувати для їх створення технічне завдання, аналізувати метадані та генерувати назву відео.

Як уже було зазначено, автоматизована система підбору та сортування ігрових трендів “Gaming Trends” буде затребувана, насамперед, для власників ютуб каналів. Але, крім того, існує інша цільова аудиторія, якій буде зручно і корисно працювати з даним програмним забезпеченням. До такої цільової аудиторії можна віднести співробітників, що працюють безпосередньо з платформою YouTube і авторами каналів, зацікавлених у дослідженні тенденцій і трендів геймінг сегменту. Можна виділити основні ролі: менеджери ютуб каналів, сценаристи, аналітики та маркетологи. Звісно, кожен з них повинен працювати саме з ігровою тематикою та геймінг каналами.

1.4 Визначення основних функцій та принципу роботи системи

Основна функція системи - це можливість клієнту отримати список трендових відео з платформи YouTube у зручному для нього вигляді. Вона включає в себе: самостійне відслідковування ігрових трендів, їх отримання, зберігання, сортування та відправлення користувачеві. У користувача буде можливість отримати список відео за допомогою сайту.

Завдяки аналізу проблеми та функцій системи було сформовано наступний список технічних вимог до розроблювального програмного забезпечення:

- відслідковування трендових відео в ігровому сегменті на платформі YouTube;
- отримання списку трендових відео з платформи YouTube;
- сортування відео та внесення в базу даних;
- зберігання списку відео в базі даних;
- сортування відео за певною грою;
- отримання відсортованого списку відео користувачем, використовуючи сайт.

Кожен з пунктів потрібно окремо дослідити і визначити шлях вирішення, принцип роботи. Для відслідковування популярних відео в ігровому сегменті використовується розділ трендів і категорія ігри на платформі YouTube. Для вибору

Зм.	Аркуш	№ докум	Підпис	Дата

саме англомовних відео слід використовувати список трендів в наступних країнах: США, Великобританія, Канада, Австралія.

Щоб отримувати список відео та автоматизувати процес відслідковування програмне забезпечення буде використовувати YouTube Data API. Завдяки йому можна запросити і отримати список трендових відео будь-якої категорії та країни.

Після отримання списку трендових відео потрібно його відсортувати і внести до бази даних. Сортування буде відбуватись за визначеним списком актуальних відеоігор, серед яких на даний момент наступні: Minecraft, GTA 5, Roblox і Fortnite. В майбутньому цей список може коригуватися в залежності від актуальності цих ігор та ситуації в ігровому сегменті. Після чого тільки відсортовані відео будуть вноситись в базу даних.

Так як в YouTube API не передбачено отримувати інформацію щодо назви гри по якій відео, слід визначити як саме система буде виконувати пошук відео по заданій грі та фільтрувати їх. На самій платформі під кожним ігровим відео зазначена гра, що вибирається автоматично алгоритмами або автором відео (рис. 1.5). Це можна використати і отримати назву гри за допомогою парсинга сторінки.

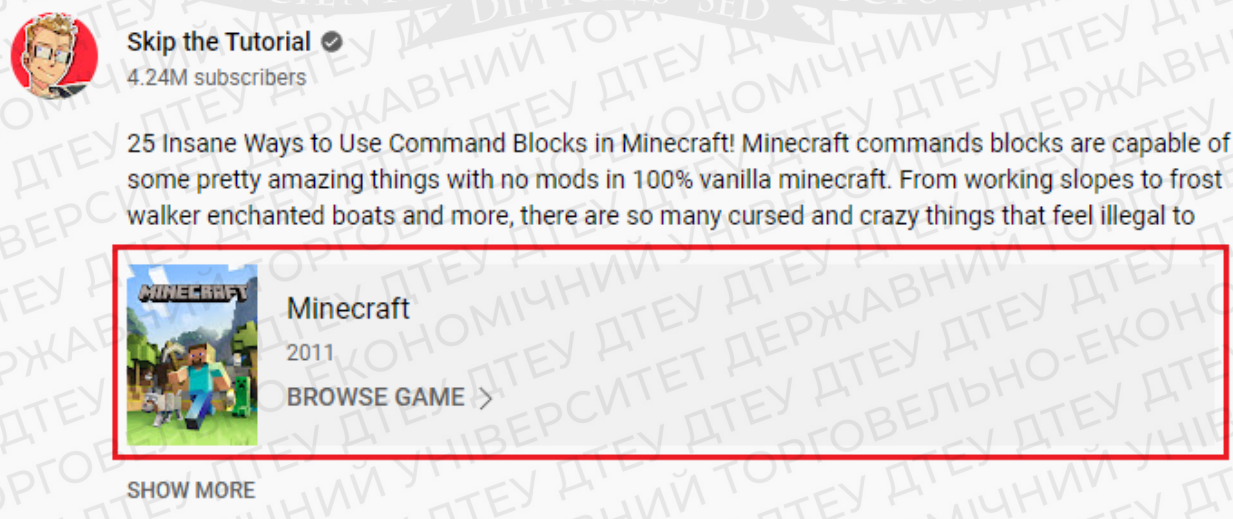


Рисунок 1.5 Приклад ігрового відео із зазначеною грою

Для користувача найбільш важливо отримувати саме актуальні і нові списки популярних відео. Водночас треба розуміти, що гарні ідеї з'являються не кожен день і відео можуть триматися в трендах декілька днів. Отже, користувачу було б

Зм.	Аркуш	№ докум	Підпис	Дата

зручно отримувати список відео за весь тиждень, а вже потім його аналізувати. Тобто відео повинні зберігатися протягом тижня щоб в будь-який день можна було їх отримати. Для цього буде використовуватися база даних.

Алгоритм роботи системи і зберігання списку відео в базі даних буде організовано наступним чином: кожні 12 годин програма буде отримувати трендові відео з платформи YouTube, обирати з них унікальні, так як часто тренди в англійськомовних країнах можуть повторюватись, відсортовувати та фільтрувати їх за назвою відеоігор, що були передбачені, та вносити їх до бази даних, перевіряючи при цьому унікальність, щоб відео не повторювались, так як популярні відео можуть триматися в трендах декілька днів. Раз в тиждень база даних буде очищатися щоб зберігати лише актуальні і нові відео.

Наступним важливим пунктом в роботі системи є отримання відео користувачем. Для цього потрібно, перш за все, отримати відео з бази даних, відсортованих лише за вибраною користувачем грою. Користувач зможе обрати, яка гра його цікавить за допомогою вебсайту, після чого отримати список трендових відео по цій грі.

1.4 Технічне завдання проєкту

На основі проведених досліджень можна сформулювати технічне завдання проєкту.

1.4.1 Загальні відомості

Повне найменування системи: Автоматизована система “Gaming Trends” підбору і сортування ігрових трендів з платформи YouTube.

Коротке найменування системи: “Gaming Trends”.

Початок роботи: листопад 2021 р.

Закінчення роботи: квітень 2022 р.

Програмне забезпечення створюється у вигляді вебсайту з реалізацією REST API. Роботи зі створення системи «Gaming Trends» виконуються поетапно в зазначеній послідовності:

- проведення маркетингових досліджень;
- аналіз предметної області;

Зм.	Аркуш	№ докум	Підпис	Дата

- створення технічного завдання;
- вибір інструментів розробки;
- проектування архітектури програмного забезпечення;
- розробка архітектури бази даних;
- проектування інтерфесу та дизайну вебсайту;
- налаштування інструментів розробки;
- створення бази даних, вебсторінок;
- розробка програмної функціональної частини;
- розробка тестів та тестування;

Після реалізації цих пунктів, результат проведених робіт пред'являється за допомогою презентації.

Потенційні користувачі системи: власники ютуб каналів, менеджери ютуб каналів, сценаристи, аналітики та маркетологи.

1.4.2 Мета та призначення

Мета: знизити витрати часу і зробити процес підбору ідей, аналізу трендів, значків та метаданих відео більш зручним для авторів YouTube каналів, менеджерів, сценаристів, аналітиків ігрового сегменту та маркетологів

Програмне забезпечення підбору ігрових трендів «Gaming Trends» призначене для збору списку відео з вкладки тренди розділу відеоігор на платформи YouTube, а також їх аналізу, сортування за назвою гри і їх відображення для користувача в зручному режимі з можливістю завантажити.

1.4.3 Вимоги до системи

Система повинна надавати доступ користувачу завдяки вебсайту з використанням зрозумілого інтерфейсу і дизайну. Для збереження інформації повинна використовуватись база даних. У склад програмного продукту повинні входити наступні модулі: управління базою даних, отримання відео з YouTube, сортування відео, відображення та збереження списку відео.

Вимоги до програмного продукту були розписані за стандартом ISO/IEC 9126 (2001).

Загальні вимоги до програмного продукту:

					<i>ДТЕУ 121 06-09.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

найбільше підходить для створення наведеного додатку. Розглядаємо три основних варіанти: Java, C#, Python.

Java – мова бізнесу і корпоративних додатків. Java більш орієнтована на масштабні проекти для великих корпорацій, програмне забезпечення яких має неосяжний функціонал, створюється протягом значимого періоду та є результатом величезних витрат. Також Java часто використовується для мобільної розробки, а саме під систему Android.

C# - головна мова платформи .NET, на ній можна писати що угодно: від маленьких комп'ютерних програм до величезних веб-сервісів. Сама ж бібліотека .NET має все необхідне для комфортної роботи і створення дійсно якісних додатків. Зараз C# також активно використовується для розробки ігор.

Python - швидка, багатofункціональна, проста в розгортанні і використанні мова програмування. Це потужна скриптова мова з величезною кількістю модулів і бібліотек. В наш час Python активно розвивається і використовується в провідних галузях IT: математичні розрахунки, data science, веб-застосунки, DevOps. З його допомогою можна створювати різні за складністю веб-проекти. А також Telegram-боти створюються саме завдяки Python.

Проаналізувавши варіанти, було встановлено, що Python найкраще підходить для вирішення поставлених завдань. Дуже приваблива, ефективна і потужна мова в наш час. Модулі і бібліотеки, що встановлюються за допомогою pip, допоможуть не гаяти час, а зосередитися на розробці і вирішенню задачі. Як середовище розробки слід розглянути PyCharm – це найкраще IDE для Python. Тим більше, воно є безкоштовним для студентів. Також, слід зазначити, що гарним варіантом є Visual Code, але було обрано саме PyCharm, отже, він буде використовуватися при розробці.

Наступним кроком є вибір СУБД для бази даних. Існують SQL бази даних і NoSQL бази даних. SQL – структурована мова запитів управління реляційними базами даних. SQL бази даних більш функціональні і мають ширше коло використання. Серед них слід виділити: Oracle, MySQL, Microsoft SQL Server,

Зм.	Аркуш	№ докум	Підпис	Дата

PostgreSQL – 4 лідера серед SQL баз даних. Для даного проєкту підходить будь-яка з них, але було обрано PostgreSQL.

Для створення сервісу знадобиться один з веб-фреймворків для Python. Два найбільш розповсюджені – Django і Flask. Основна їх відмінність в тому, що Flask реалізується з мінімальними надбудовами, які можна розширяти доповненнями або розробником, а Django слідує філософії «все з коробки» і надає великий асортимент для роботи. Вибирати слід на підставі наступної інформації. Flask надає простоту, гнучкість і акуратність в роботі, дозволяючи користувачеві самому вибирати, як реалізувати ті чи інші речі. Django надає пакет «все включено»: панель адміністратора, інтерфейси баз даних, ORM, і структура каталогів для додатків і проєктів [1]. Також слід зазначити, що Flask реалізує паттерн MVC, в той же час Django – MVT. Ці паттерни дозволяють якнайкраще спроектувати сервіс та структурувати його. Ці два фреймворки дуже потужні, але вибір впав саме на Flask, отже, він буде використовуватися для розробки веб-частини проєкту.

Реалізовувати веб-служби потрібно дотримуючись архітектурного стиля REST, який являє собою узгоджений набір обмежень, що враховуються при проєктуванні розподіленої гіпермедіа-системи. У більшості випадків це приводе до підвищення швидкості роботи і спрощення архітектури.

Щоб поєднати Python і Flask з базою даних найзручніше та найефективніше використовувати ORM (англ. object-relational mapping, укр. об'єктно-реляційне відображення). Цей підхід дозволяє створити класи рівноцінні таблицям бази даних і відображати дані у вигляді об'єктів. Для створення ORM підходить модуль SQLAlchemy, так що саме він буде використовуватись.

Першим кроком створення автоматизованої системи є проєктування архітектури програмного забезпечення. Проєктування відбувається у вигляді розробки UML діаграм. Для зручності розробки діаграм існує достатньо спеціальних сервісів. В даній роботі буде використовуватись сайт www.diagrams.net, завдяки якому буде створено різні UML діаграми.

Наступним кроком є розробка схематичного макету (вигляду) інтерфейсу сайту та його дизайну. Для реалізації цієї мети можна використовувати різні

Зм.	Аркуш	№ докум	Підпис	Дата

графічні редактори, один з яких Figma. Саме Figma буде використовуватися для розробки макету та дизайну інтерфейсу вебсайту.

Також, слід зазначити, що увесь проєкт буде розроблятися з використанням розподіленої системи керування версіями файлів Git. Його використання допоможе контролювати процес написання окремих частин проєкту для врахування допущення помилки. Тобто, якщо в якийсь момент написання програми буде допущена помилка і все перестане працювати, то можна буде “відкотитись” до минулої версії коду, яка була збережена за допомогою Git. Для доступу до файлів програми з будь-якого пристрою буде використовуватися веб-сервіс GitHub, що створений на основі системи контролю версій Git.

Для реалізації інтерфейсу вебсайту буде використовуватися HTML і CSS, щоб додаток підтримувався будь-якими браузерами. Окрім цього, для веб-додатку буде передбачена кросплатформність, що передбачає коректне відображення вебсайту на будь-яких пристроях, завдяки інструменту Bootstrap. Bootstrap включає в себе готові HTML та CSS оформлення веб-інтерфейса, які заздалегідь створені з підтримкою різних пристроїв і будь-якої роздільної здатності екранів.

1.6 Висновки до розділу 1

Після проведення маркетингових досліджень, аналізу предметної області, було виявлено, що Youtube – це найпопулярніша платформа для публікації і перегляду відео. Власники ютуб каналів максимально зацікавлені в збільшенні кількості переглядів і розвитку своїх каналів. Найкращий спосіб отримувати максимальну кількість переглядів – знімати відео на актуальні та трендові теми. Щоб відслідковувати та отримувати вдалі ідеї, як правило, потрібно переглядати англійські тренди в YouTube і адаптувати їх для нашої аудиторії.

Була виявлена проблема, що пов’язана з підбором популярних тем та ідей для авторів. Щоб спростити цей процес і зробити його більш зручним, було вирішено розробити автоматизовану систему підбору та сортування ігрових трендів під назвою “Gaming Trends”. Основна мета проєкту – знизити витрати часу і зробити процес підбору ідей, аналізу трендів, значків та метаданих відео більш зручним. Основна функція системи – це можливість клієнту отримати список трендових

відео з платформи YouTube у зручному для нього вигляді. Система “Gaming Trends” буде затребувана серед авторів, менеджерів ютуб каналів, сценаристів, аналітиків та маркетологів.

Завдяки аналізу проблеми та функцій системи було сформовано наступний список технічних вимог до розроблювального програмного забезпечення:

- відслідковування трендових відео в ігровому сегменті на платформі YouTube;
- отримання списку трендових відео з платформи YouTube;
- сортування відео та внесення в базу даних;
- зберігання списку відео в базі даних;
- сортування відео за певною грою;
- отримання відсортованого списку відео користувачем, використовуючи сайт;
- завантаження списку відео користувачем.

Після визначення всіх основних вимог системи та принципу її роботи, було обрано інструменти, що будуть використані в процесі створення автоматизованої системи:

- мова програмування Python;
- середовище розробки PyCharm;
- СУБД PostgreSQL;
- Веб-фреймворк Flask;
- Архітектура REST;
- ORM SQLAlchemy;
- GitHub;
- Figma;
- HTML, CSS, Bootstrap.

Зм.	Аркуш	№ докум	Підпис	Дата

РОЗДІЛ 2

ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Розробка архітектури програмного забезпечення

Першим підготовчим кроком для розробки програмного забезпечення є проєктування і розробка його архітектури. Проєктування архітектури ПЗ – це процес розроблення, що виконується після етапу аналізу і формулювання вимог. Задача такого проєктування — перетворення вимог до системи у вимоги до програмного забезпечення і побудова на їхній основі архітектури системи [2]. Побудова архітектури системи здійснюється шляхом визначення цілей системи, її вхідних і вихідних даних, декомпозиції системи на підсистеми, компоненти або модулі та розроблення її загальної структури. Дослідження архітектури програмного забезпечення намагається визначити як найкраще розбити систему на частини, як ці частини визначаються та взаємодіють одна з одною, як між ними передається інформація.

Проєктування системи може здійснюватися на основі об'єктно-орієнтованого моделювання за допомогою UML діаграм, який дозволяє враховувати аспекти, властиві діючим особам (акторам) системи, створювати сценарії виконання системи тощо. Саме цей підхід і буде використовуватись для проєктування архітектури системи. Буде створено основні UML діаграми для програмного забезпечення:

- діаграма прецедентів (use case diagram);
- діаграма класів (class diagram);
- діаграма кооперації (collaboration diagram);
- діаграма послідовностей (sequence diagram);
- діаграма стану (state diagram);
- діаграма діяльності (activity diagram);

					<i>ДТЕУ 121 06-09.БР</i>			
					<i>Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube</i>	<i>Стадія</i> <i>P2</i>	<i>Аркуш</i> <i>20</i>	<i>Аркушів</i> <i>46</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		14.03.22				
Керівник		Жирова Т. О.		14.03.22				
Гарант		Рзаєва С. Л.		14.03.22				
Розроб		Капуста Н. В.		14.03.22	<i>Проєктування програмного забезпечення</i>		Факультет інформаційних технологій, 4 курс, 6 група	

діаграма компонентів (component diagram);

діаграма розгортання (deployment diagram);

2.1.1 Діаграма прецедентів (use case diagram)

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених межею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами [3]. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть діаграми прецедентів полягає в тому, що проєктована система подається у вигляді множини сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система під час діалогу з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів із системою.

Для розробки даної діаграми потрібно визначити назву системи, акторів і моделі варіантів використання. Система називається “Вебсайт YouTube Trends”. Серед акторів слід виділити користувача і сам сервіс, що обробляє запити користувача і взаємодіє з системою. Також були визначені наступні прецеденти (варіанти використання): перегляд інформації про сайт і можливості сайту, перегляд списку трендових відео за вибраною грою, що розширюється (extends) можливістю скачати список, і включає (include) фільтрування трендових відео за вибраною грою, яке включає (include) в себе отримання списку трендових відео з бази даних, що включає (include) збереження відео в базі даних, яке включає (include) отримання списку трендових відео з Youtube. В результаті було створено uml діаграму (рис. 2.1).

Зм.	Аркуш	№ докум	Підпис	Дата

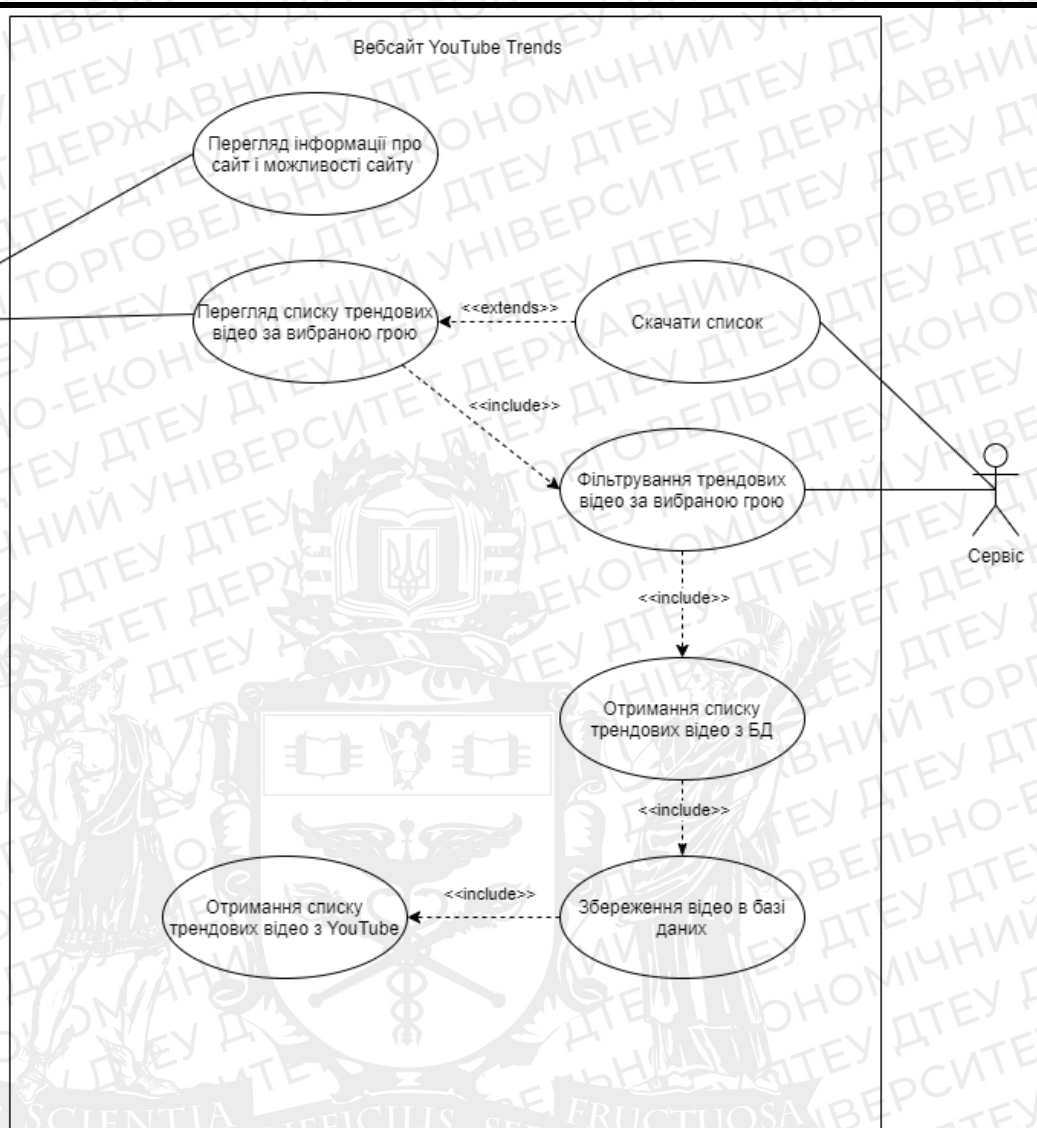


Рисунок 2.1 Use case diagram

2.1.2 Діаграма класів (class diagram)

Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини. В UML існують різні типи зв'язків для визначення відносин між класами: асоціація, агрегація, композиція, наслідування і реалізація.

Для даної системи було визначено класи, інтерфейси, функції, змінні, типи даних змінних, їх доступність, та відношення між класами. На основі цих даних було створено діаграму класів (рис. 2.2).

Зм.	Аркуш	№ докум	Підпис	Дата

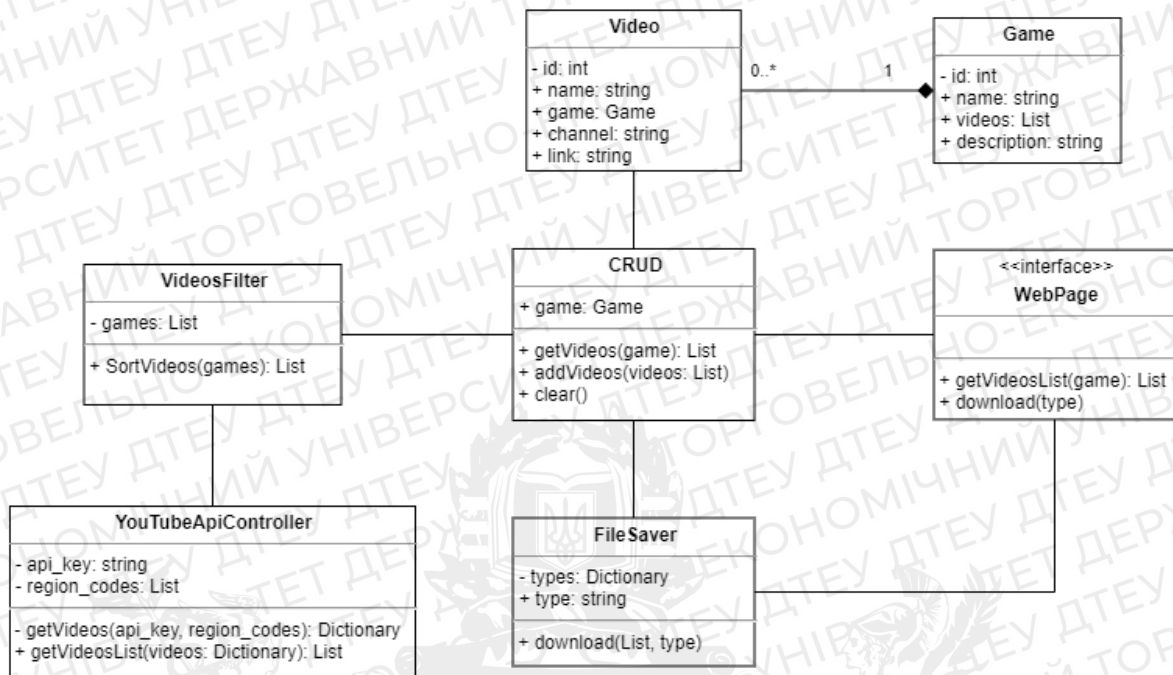


Рисунок 2.2 Class diagram

Програма поділяється на частини саме завдяки поділу на класи і об'єктно-орієнтованому підходу програмування. Кожен клас відповідає за свій функціонал у системі. Для даного програмного забезпечення було визначено наступні класи:

- Video - зберігає інформацію про відео;
- Game - зберігає інформацію про гру;
- YouTubeApiController - взаємодіє з Youtube API;
- VideosFilter - використовується для сортування відео;
- CRUD - взаємодіє з базою даних;
- FileSaver - використовується для зберігання списку у вигляді файлу.

2.1.3 Діаграма кооперації (collaboration diagram)

Діаграма кооперації служить для того, щоб специфікувати особливості реалізації окремих найбільш значущих операцій в системі. Кооперація визначає структуру поведінки системи в термінах взаємодії учасників цієї кооперації. Головна особливість діаграми кооперації полягає в можливості графічно представити не тільки послідовність взаємодії, але й усі структурні відносини між об'єктами, які беруть участь у цій взаємодії. Вона відображає взаємодію між класами системи в певному порядку дій, які пронумеровані (рис. 2.3).

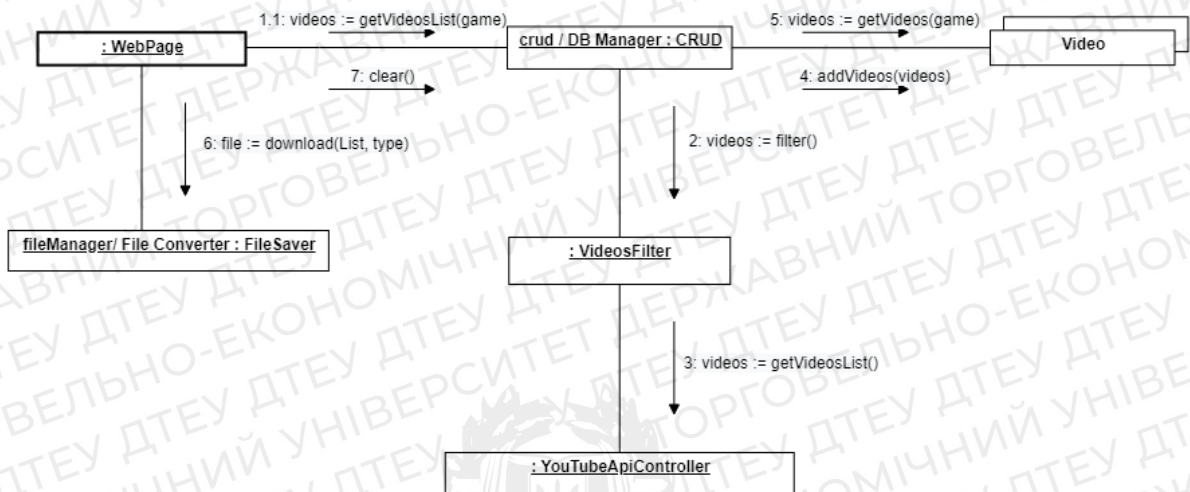


Рисунок 2.3 Collaboration diagram

2.1.4 Діаграма послідовностей (sequence diagram)

Діаграма послідовності – відображає взаємодії об'єктів впорядкованих за часом. Діаграми послідовностей використовується для більш детального опису логіки сценаріїв використання. Це відмінний засіб документування проекту з точки зору сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються, і які повертаються результати, які пов'язані з повідомленнями.

Для створення діаграми послідовності потрібно використати існуючі класи, додати для кожного лінію життя і його стан. Лінія життя служить для позначення періоду часу, протягом якого об'єкт існує в системі і, отже, може потенційно брати участь у всіх її взаємодіях. Об'єкти можуть перебувати в активному стані, безпосередньо виконуючи певні дії, або стані пасивного очікування повідомлень від інших об'єктів. Для зображення лінії життя об'єкта використовується пунктирна лінія. Для зображення стану використовується фокус управління, що зображується у формі витягнутого вузького прямокутника. Визначивши акторів (користувачів), об'єкти, їх лінію життя, стан та взаємодію між собою (повідомлення), була створена діаграма послідовностей (рис. 2.4).

Зм.	Аркуш	№ докум	Підпис	Дата

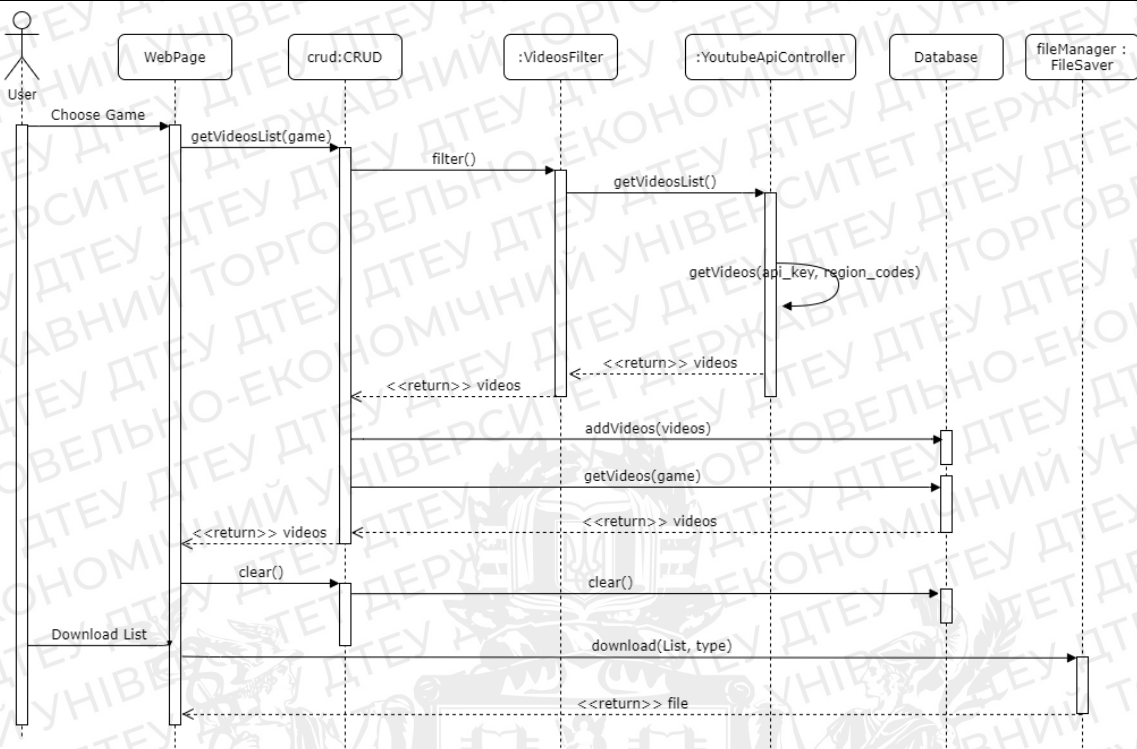


Рисунок 2.4 Sequence diagram

2.1.5 Діаграма стану (state diagram)

Діаграма станів використовуються для опису поведінки окремих об'єктів системи. Вона застосовуються для того, щоб пояснити, яким чином працюють складні об'єкти. Діаграма станів показує, як об'єкт переходить з одного стану в інший.

Головний об'єкт в розроблювальній системі — це сторінка вебсайту з якою взаємодіє користувач, змінюючи її стан. Також є окремий процес отримання відео.

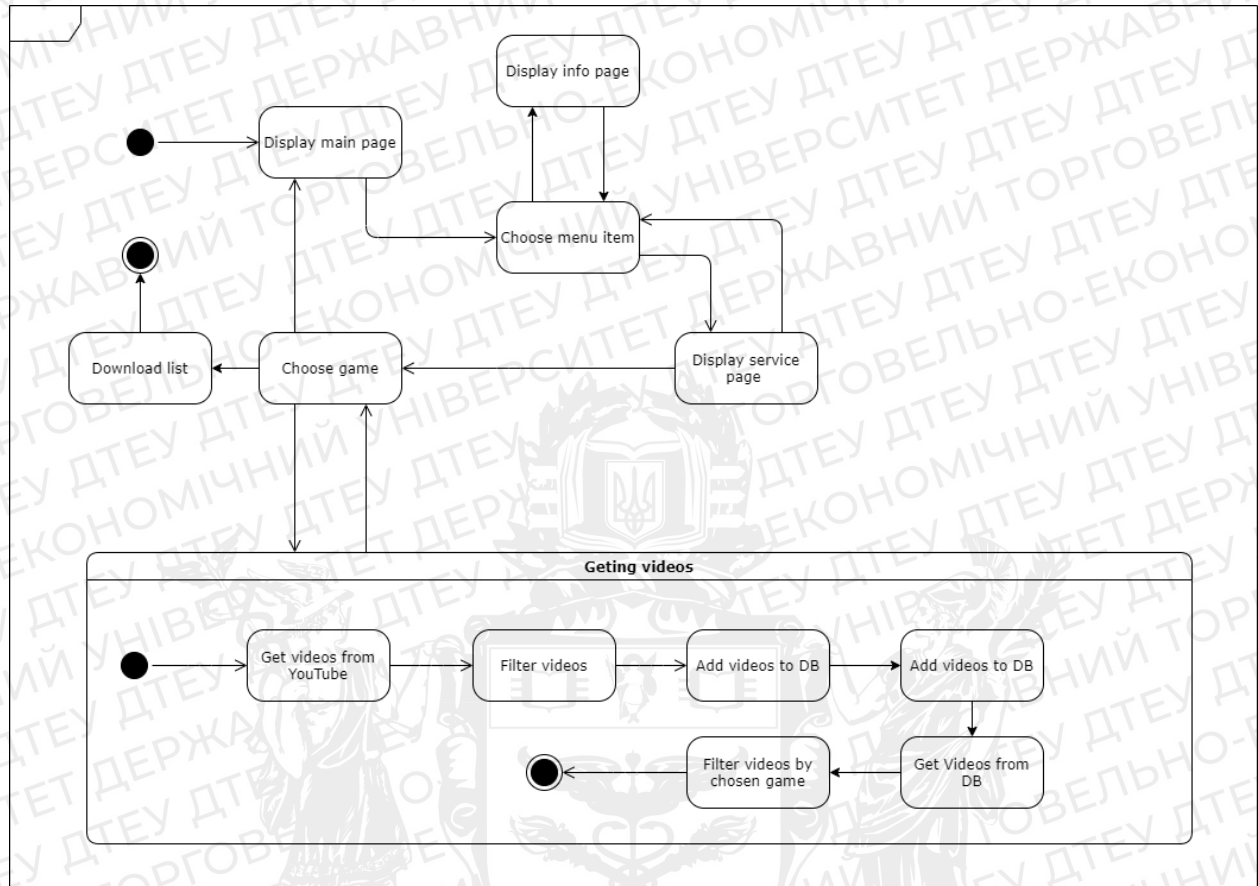


Рисунок 2.5 State diagram

2.1.6 Діаграма діяльності (activity diagram)

При моделюванні поведінки проєктованої системи виникає необхідність не лише представити процес зміни її станів, але і деталізувати особливості алгоритмічної і логічної реалізації виконуваних системою операцій. Для цієї мети може використовуватися діаграма діяльності, яка акцентують увагу на послідовності виконання певних дій, що призводять до бажаного результату і напоминає алгоритм роботи системи (рис 2.6).

Зм.	Аркуш	№ докум	Підпис	Дата
-----	-------	---------	--------	------

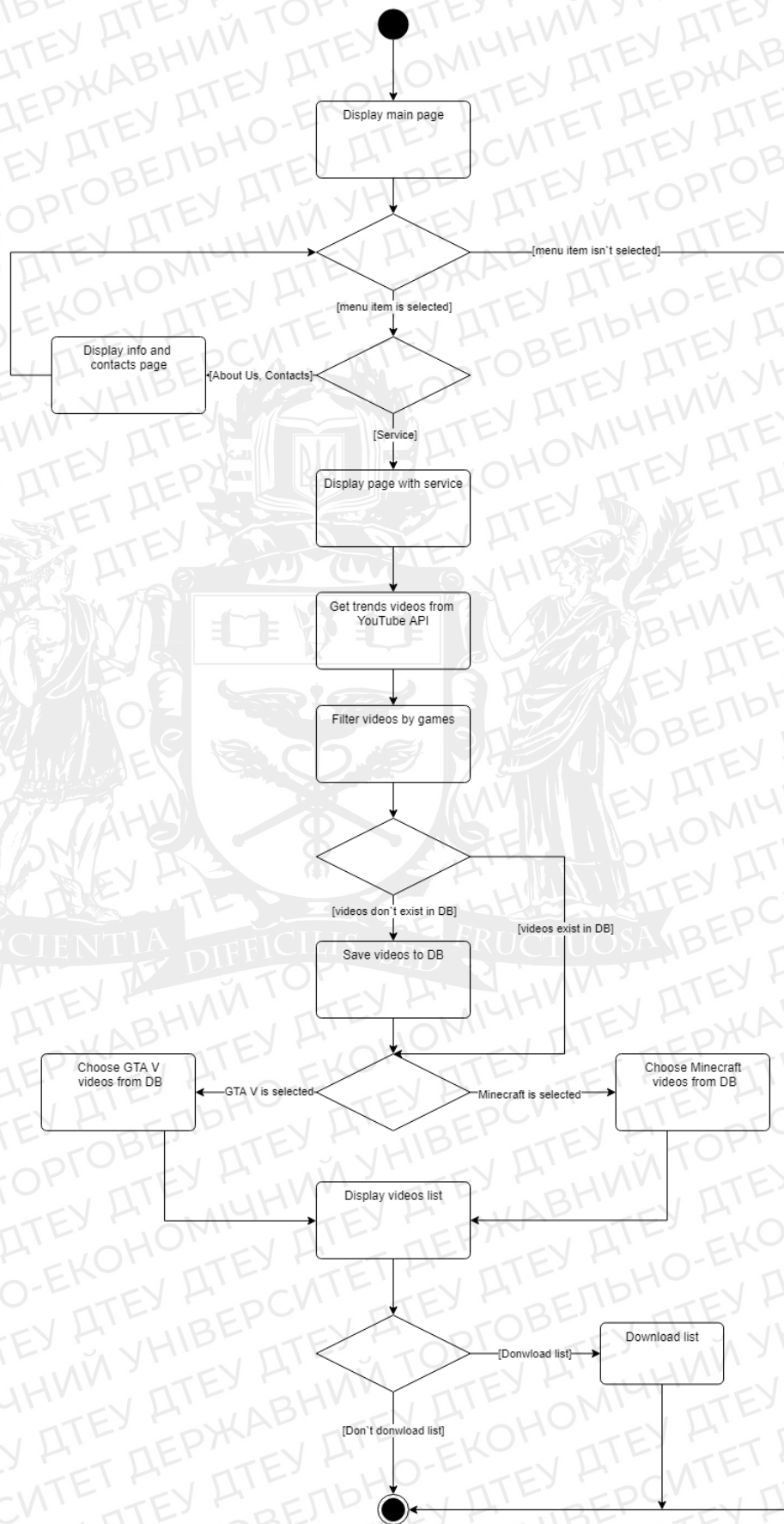


Рисунок 2.6 Activity diagram

Зм.	Аркуш	№ докум	Підпис	Дата

2.1.7 Діаграма компонентів (component diagram)

Діаграма компонентів відображає залежності між компонентами програмного забезпечення. До компонентів даної системи можна віднести усі модулі програми (python файли), базу даних, файли із веб-сторінками (html і css файли), які будуть створені в процесі розробки (рис. 2.7).

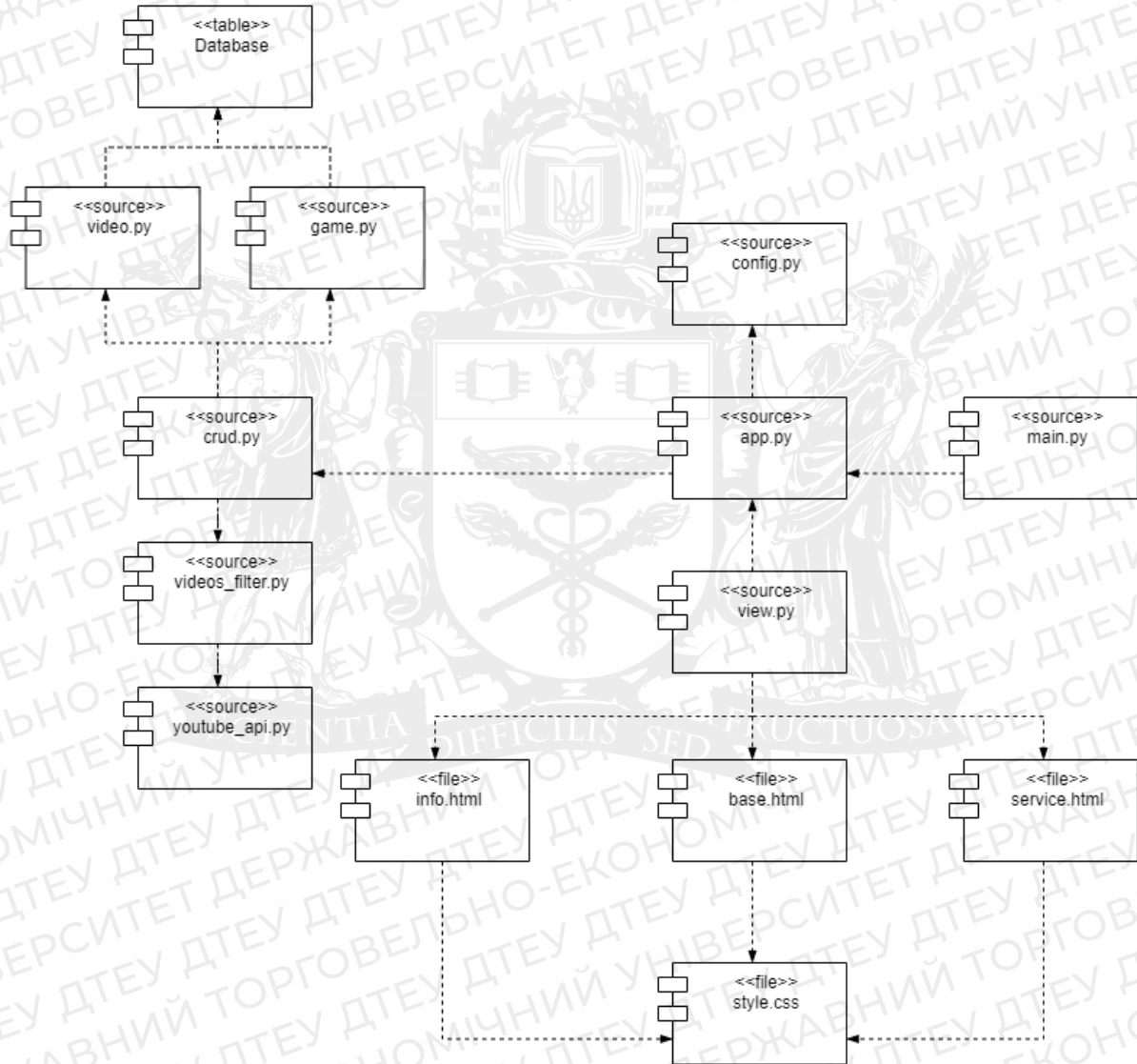


Рисунок 2.7 Component diagram

2.1.8 Діаграма розгортання (deployment diagram)

Діаграма розгортання – діаграма на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Для даної системи було розроблено спрощену діаграму розгортання, що

Зм.	Аркуш	№ докум	Підпис	Дата

відображає основні вузли: Users device (комп'ютер, телефон), Web server і Database server.



Рисунок 2.8 Deployment diagram

2.2 Проектування архітектури бази даних

Система буде використовувати базу даних для зберігання інформації про ігри і трендові відео. Для реалізації бази даних потрібно перш за все її спроектувати, а саме: розробити концептуальну, логічну та фізичну моделі БД. Перелічені етапи допоможуть визначити основну структуру бази даних, потрібні таблиці, стовбці та типи даних.

2.2.1 Створення концептуальної моделі

Концептуальне проектування – побудова семантичної моделі предметної області, тобто інформаційної моделі найбільш високого рівня абстракції. Така модель створюється без орієнтації на якусь конкретну СУБД і модель даних. Вона представляє з себе лиш абстрактні елементи, що розкривають суть основної теми бази даних. Для реалізації концептуальної моделі потрібно сформулювати основні об'єкти та встановити між ними зв'язки. Для даної системи було визначено два об'єкти: Гра (Game) і Відео (Video) і встановлено зв'язок між ними один до багатьох (рис 2.9). Тобто кожне відео відноситься тільки до однієї гри, при цьому одна гра може бути в різних відео.

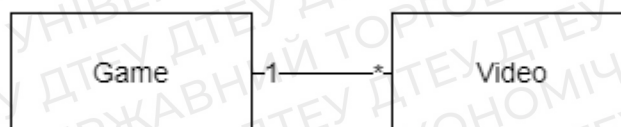


Рисунок 2.9 Концептуальна модель

2.2.2 Створення логічної моделі

Логічне проектування – створення логічної схеми бази даних на основі конкретної інформаційної моделі даних, предметної області. Логічна модель створюється незалежно від її майбутньої реалізації, тобто модель не прив'язується до конкретних типів даних або СУБД.

Для створення системи необхідно дві таблиці, які будуть пов'язані між собою та взаємодіяти одна з одною. Відносини між таблицями досягаються зазвичай із зазначенням первинних ключів, а також зв'язками між відносинами, що представляють собою зовнішні ключі.

Перша таблиця під назвою “Гра” (Game) включає назву гри.

Друга таблиця називатиметься “Відео” (Video) і містить в собі назву відео, назву каналу автора, посилання на відео та значок відео. Вона пов'язана з грою завдяки ключу з ID гри.

Визначивши всі таблиці потрібно їх з'єднати завдяки ключів і логічна модель спроектована (рис. 2.10).

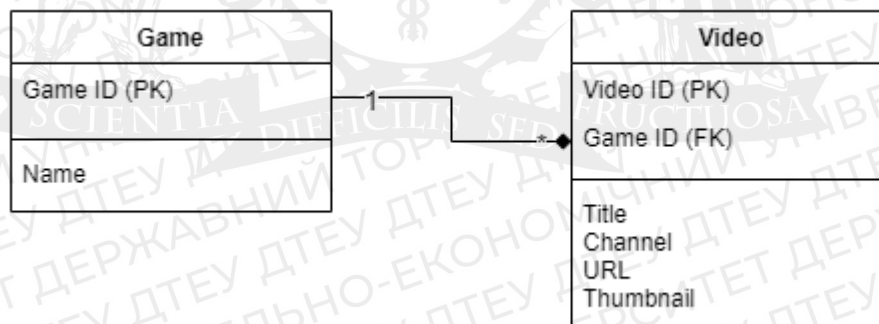


Рисунок 2.10 Логічна модель

2.2.3 Створення фізичної моделі

Останній етап проектування бази даних полягає в створенні фізичної моделі, що являє собою логічну модель заточену під реалізацію в конкретній СУБД. Тобто логічна модель перетворюється в фізичну завдяки зазначенню конкретних назв об'єктів, полів і їх типів даних.

Найбільш важливою деталлю проектування саме фізичної моделі – є зазначення типу даних кожному елементу таблиці. В даному проекті для реалізації була вибрана СУБД PostgreSQL. Отже, всі типи даних орієнтовані саме для неї. Для

Зм.	Аркуш	№ докум	Підпис	Дата

елементів з числовими даними використовується тип даних INTEGER, а для елементів з текстовими даними – тип даних CHARACTER, де в дужках вказується максимально можлива кількість символів.

Згрупувавши всі ці дані та підставивши відповідні типи даних для кожного елемента можливо спроектувати фізичну модель (рис. 2.11)

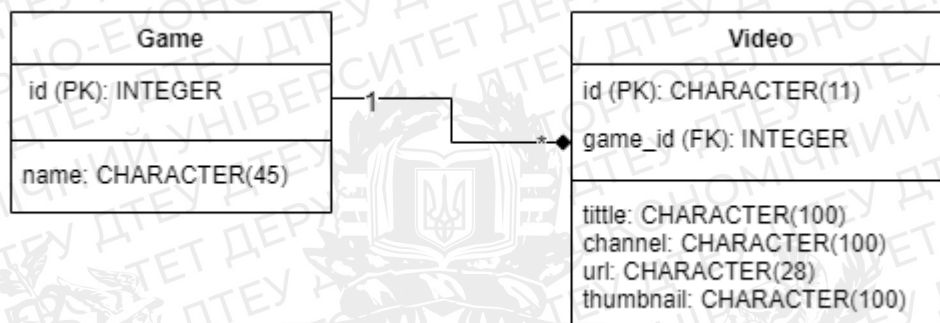


Рисунок 2.11 Фізична модель

2.3 Створення інтерфейсу і дизайну вебсайту

Для проектування інтерфейсу і розробки дизайну буде використаний сервіс Figma. За його допомогою можна зручно виконати поставлені задачі. Серед задач слід виділити: розробка інтерфейсу та дизайну трьох сторінок сайту, їх представлення на мобільному телефоні.

Перша сторінка сайту є головною і повинна містити інформацію про сайт і систему, загальний алгоритм роботи. Друга сторінка включає детальну інформацію про сервіс, а також інформацію про розробника і всі контактні дані для зв'язку. Третя сторінка – це сам сервіс. На цій сторінці можливо отримати список трендових відео за зазначеною грою.

Кожна сторінка сайту у верхній частині містить шапку з логотипом системи і різним текстом в залежності від сторінки. Між сторінками можлива навігація завдяки меню, яке також знаходиться у шапці. В нижній частині на кожній сторінці знаходиться футер, що являє собою блок з контактною інформацією.

Дизайн буде представлено в червоному кольорі в зв'язку з тим, що основний колір на YouTube – червоний. Також буде використано різні візуальні доповнення у вигляді картинок, геометричних фігур і значків.

Отже, в результаті був розроблен інтерфейс і його дизайн з яким можливо ознайомитись в додатку А.

2.4 Висновки до розділу 2

Після проведення аналізу вимог та завдань системи, було спроектовано архітектуру програмного забезпечення на основі об'єктно-орієнтованого моделювання за допомогою UML діаграм. Для подальшої розробки програмного забезпечення було розроблено наступні види UML діаграм:

- діаграма прецедентів (use case diagram);
- діаграма класів (class diagram);
- діаграма кооперації (collaboration diagram);
- діаграма послідовностей (sequence diagram);
- діаграма стану (state diagram);
- діаграма діяльності (activity diagram);
- діаграма компонентів (component diagram);
- діаграма розгортання (deployment diagram);

Кожна діаграма відповідає за свою область діяльності та призначена для подальшого розуміння структури проекту, його розробки.

Окрім цього, система в процесі роботи використовуватиме базу даних. Для її створення було розроблено концептуальну, логічну та фізичну модель бази даних.

Також було створено інтерфейс користувача та його дизайн завдяки сервісу Figma, а саме: головну сторінку, сторінку з інформацією про сервіс Youtube Trends з контактною інформацією та сторінку для вибору гри і отримання на її основі списку відео.

Зм.	Аркуш	№ докум	Підпис	Дата

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1 Налаштування інструментів розробки

Першим підготовчим етапом для розробки програмного забезпечення є налаштування інструментів для розробки. Серед інструментів, що будуть використовуватись слід виділити: середовище розробки PyCharm, система контролю версій GitHub і базу даних PostgreSQL.

3.1.1 Створення та налаштування проєкту в середовищі розробки PyCharm

Для початку створення програмного додатку потрібно створити та налаштувати проєкт в середовищі розробки. Середовищем розробки було обрано PyCharm, так як саме він найбільш доцільний і зручний для написання проєктів на мові програмування Python.

Першим і важливим кроком при створенні нового проєкту на мові Python – є налаштування віртуального середовища та завантаження всіх необхідних модулів, які будуть використовуватися при написанні проєкту. Віртуальне середовище представляє собою ізольоване середовище для встановлення бібліотек (модулів) під окремий проєкт. Тобто кожен проєкт, маючи своє середовище, є цілком незалежним і містить свій перелік модулів, незалежно від того, які в іншого проєкту.

Заходимо в PyCharm і створюємо новий проєкт (File/New Project) серед списку вибираємо Python або Flask, вказуємо назву проєкту та його місцезнаходження. У цьому ж вікні доступне налаштування віртуального середовища (Project Interpreter). Там можна створити нове або вибрати вже існуюче віртуальне середовище. Так як воно ще не було сформоване потрібно створити нове для даного проєкту. Вибирається модуль (за замовчуванням Virtualenv) та вказується шлях, де його потрібно створити (рис. 3.1).

					<i>ДТЕУ 121 06-09.БР</i>			
					<i>Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube</i>	<i>Стадія</i> РЗ	<i>Аркуш</i> 33	<i>Аркушів</i> 46
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		11.04.22				
Керівник		Жирова Т. О.		11.04.22				
Гарант		Рзаєва С. Л.		11.04.22	<i>Розробка програмного продукту</i>	Факультет інформаційних технологій, 4 курс, 6 група		
Розроб		Капуста Н. В.		11.04.22				

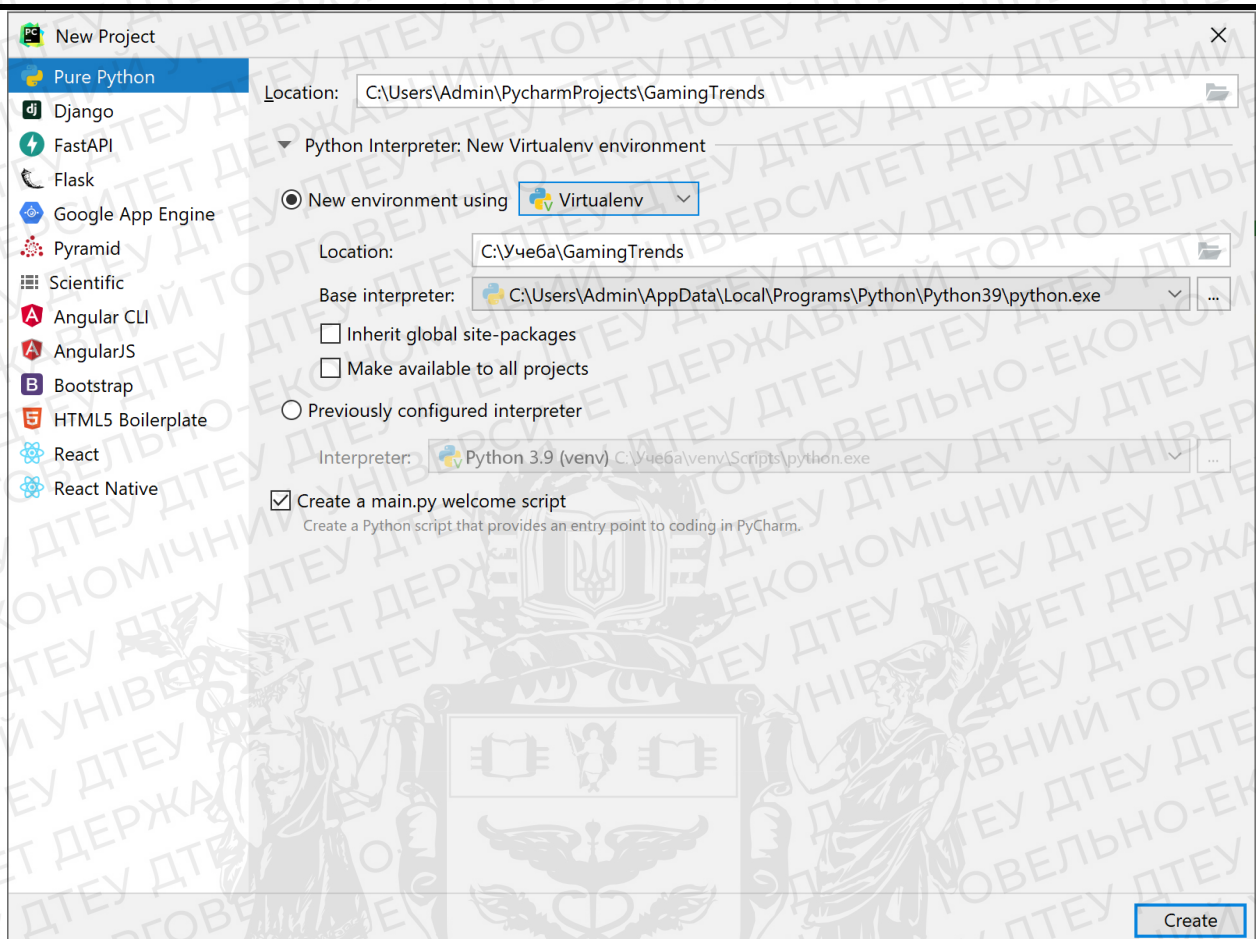


Рисунок 3.1 Створення проекту в PyCharm

Після створення віртуального середовища потрібно його налаштувати шляхом встановлення відповідних модулів, що будуть використовуватися при написанні додатку. Зробити це можна відкривши термінал(командний рядок) та вписавши команду ***pip install назва_модуля***, передусім активувавши віртуальне середовище(за замовчуванням активоване, якщо запускати термінал в PyCharm). Всі модулі можна зазначити в файлі “requirements.txt” для подальшого встановлення на будь-якій робочій машині, де буде розроблятися, розширюватися або запускатися програмний продукт. В файл записуємо модулі, що на потрібні: Flask, Flask-RESTful, requests, Flask-SQLAlchemy, Selenium, Schedule, Webdriver-manager. Звісно, потім можна буде додати ще якісь модулі, якщо виникне потреба. Останнім кроком є встановлення всіх модулів використовуючи команду ***pip install -r requirements.txt***, але не забуваючи, що віртуальне середовище повинно бути активним.

3.1.2 Налаштування системи контролю версій Git і GitHub

Наступним важливим етапом розробки є підключення до системи управління версіями. Система управління версіями (СУВ) — це система, що зберігає зміни в одному або декількох файлах так, щоб потім, за потреби, можна було відновити відповідні старі версії. Цей підхід замінює копіювання проєкту і допомагає запобігти помилкам. За допомогою систем управління версіями можна зберігати минулі версії програм і завантажувати їх у разі потреби [10].

Серед усіх аналогів СУВ для розробки проєкту було обрано розподілену систему управління версіями GIT. Саме завдяки ній існує сервіс GitHub, що додатково дозволяє завантажувати збережені версії проєкту на хмару для подальшого завантаження на іншому пристрої або для спільної роботи над проєктами.

Для початку роботи з GIT потрібно встановити його на робочий комп'ютер. Після чого відкрити папку з проєктами за допомогою спеціальної консолі і налаштувати, використовуючи команди. Так само відбувається і підключення проєкта до GitHub, передусім створивши його на сайті. Після чого можна зберігати кожен версію проєкту за допомогою спеціальних команд в консолі або за допомогою середовища розробки. Рекомендується зберігати будь-які суттєві зміни.

3.1.3 Створення бази даних

Для взаємодії з системою була обрана система управління базами даних PostgreSQL. Для написання запитів до бази даних використовується мова SQL. Але також було вирішено використовувати ORM SQLAlchemy.

ORM (англ. Object-relational mapping, Об'єктно-реляційна проєкція) — технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи “віртуальну об'єктну базу даних”. Такий підхід допоможе перетворити систему в сховище об'єктів з якими можна працювати, а вони автоматично зберігатимуться в реляційній базі даних. ORM позбавляє від написання великої кількості коду, часто одноманітного і схильного до помилок, тим самим значно підвищуючи швидкість розробки.

Зм.	Аркуш	№ докум	Підпис	Дата

Для реалізації бази даних потрібно встановити СУБД PostgreSQL на робочу машину, налаштувати дані для входу та створити базу даних, використовуючи команду **CREATE SCHEMA [НАЗВА]**. Відразу було вирішено створити дві бази даних. Першу для тестів під назвою “trends” і другу основну для роботи системи під назвою “trends_tests”. Після чого за допомогою цих даних можна підключити бази даних до програмного забезпечення, що розробляється.

3.2 Розробка сторінок вебсайту

Перед розробкою функціональної частини продукту потрібно створити інтерфейс з яким буде працювати користувач. Макети сторінок вже створені, залишилось їх реалізувати завдяки HTML, CSS, Bootstrap.

Створюється папка “templates”, саме тут будуть зберігатися всі html файли, що буде використовувати Flask. Всередині папки потрібно додати новий файл під назвою «base.html», що буде основою сторінок. У “base.html” пишеться структура HTML файлу, підключається Bootstrap, який перед цим треба скачати і встановити в папку “static”, а також підключається CSS файл зі стилями, який потрібно створити в папці “static”.

Далі створюється ще 3 файли, що являються сторінками вебсайту: “main.html”, “info.html” і “service.html”. Всі файли наслідуються від “base.html” завдяки спеціальним блокам команд на мові Jinja, що підтримується Flask. Далі за допомогою HTML і CSS потрібно створити інтерфейс і дизайн.

Готові сторінки виглядають добре завдяки розробленому дизайну і ними можна користуватися на будь-якому пристрої, з будь-яким екраном завдяки кросплатформності, що надає Bootstrap.

Після того, як буде розроблена функціональна частина, залишиться лише зв'язати інтерфейс з програмною частиною та вивести всю потрібну інформацію за допомогою Jinja.

3.3 Підключення бази даних і створення моделей

Визначивши структуру таблиць бази даних, потрібно створити моделі, що представляються у вигляді об'єктів та створюються написанням класу за допомогою ORM підходу і модулю SQLAlchemy. Для цього додається папка

Зм.	Аркуш	№ докум	Підпис	Дата

“models”, де будуть зберігатися всі моделі проекту. В папці створюється два файли “game.py” і “video.py”. У файлах потрібно написати однойменні класи, які і будуть моделями.

Кожен клас, що являє собою модель, повинен унаслідуватись від *Model* з модуля flask_sqlalchemy. Але передусім потрібно його ініціалізувати, робиться це в окремому файлі під назвою “ari.py”, який розміщений в головній папці проекту. Ініціалізація бази даних проводиться за наступними кроками:

- 1) Імпортуються всі необхідні модулі.
- 2) Створюється екземпляр класу Flask: ***app = Flask(__name__)***
- 3) Вказуються дані для підключення та налаштування бази даних через об’єкт, який можна ініціалізувати в фалі “config.py” для зручності:
app.config.from_object(AppConfiguration)
- 4) Далі потрібно створити екземпляр класу *SQLAlchemy*, що буде об’єктом бази даних: ***db = SQLAlchemy(app)***

Екземпляр *SQLAlchemy* створений під назвою *db* імпортується в кожний файл з моделей. У класах моделей записуються відповідні поля, які були визначення при проектуванні моделей бази даних, що будуть стовпчиками таблиць. Також після створення моделей необхідно створити саму базу даних завдяки команді в коді ***db.create_all()***. У результаті було написано наступні два класи (Додаток Б).

3.4 Отримання трендових відео за допомогою YouTube API

Для отримання трендових відео з платформи YouTube буде використовуватись YouTube API. Перш за все, потрібно мати Google аккаунт та отримати YouTube API ключ на вебсайті Google Cloud Platform, коректно його налаштувати.

Ключ для зручності можна зберігати в файлі “config.py”, який вже було створено. Далі створюється окремий модуль “youtube_ari.py” для роботи з YouTube API в новій папці під назвою “service”, де будуть зберігатись подібні функціональні модулі для роботи системи. В новоствореному файлі модуля слід імпортувати модуль requests для створення запитів по API, а також сам ключ з файлу “config.py”

та список кодів країн, який можна прописати в тому ж “config.py”. Після чого створити клас *YouTubeApiController*, який буде відповідати за отримання трендів. В класі потрібно написати два статичних методи. Перший відповідатиме за отримання списку ID 30 найпопулярніших трендових відео з кожної із зазначених країн, відправляючи запит через YouTube API. Другий буде отримувати всю інформацію про відео за допомогою посилань, що будуть створюватись на основі цих ID (Додаток В).

3.4 Сортування та фільтрування списку відео

Однією із основних задач системи після отримання списку трендових відео з YouTube є їх фільтрування за певною грою. Для реалізації цієї задачі створюється новий модуль під назвою “videos_filter.py” в папці “service”, в якому слід написати клас *VideosFilter*, який буде відповідати за процес фільтрування.

З’являється проблема, так як використання YouTube API не дозволяє переглянути інформацію про назву гри будь-якого ігрового відео. Одним із варіантів для вирішення цієї проблеми є скрапінг, тобто отримання інформації з вебсторінки. Для реалізації скрапінгу в даній системі краще використати Selenium, так як інформацію про гру підгружається динамічно. Selenium WebDriver — інструмент для автоматизації дій браузера. У більшості випадків використовується для тестування вебдодатків, але в даному випадку за його допомогою можна отримати назву гри, яка відображається при відкриванні сторінки з відео в браузері.

Для використання Selenium імпортуються потрібні об’єкти з однойменного модуля, а також ChromeDriverManager з модуля webdriver-manager. Саме він буде використовуватись для запуску браузера Chrome. Після чого створюються потрібні функції для роботи класу і реалізації поставлених задач. Функція *get_sorted_videos* відповідає за отримання списку відео, звертаючись до створеного класу *YouTubeApiController*, та його фільтрування за списком ігр, що передаються в змінну. Функція *check_game* використовується минулою функцією для фільтрування відео. Вона вертає назву гри, отримавши аргументом ID відео. Також була написана додаткова функція *prepare_link*, що потрібна для підготовки посилання на відео завдяки його ID (Додаток Г).

3.5 Реалізація CRUD (створення, зчитування, зміна і видалення)

CRUD – акронім, означаючий чотири базові функції, які використовуються при роботі з базами даних: створення, зчитування, зміна і видалення. Розроблюваний сервіс повинен підтримувати деякі з цих функції при роботі з базою даних, які потрібні системі для коректного функціонування. Для їх реалізації створюється модуль “crud.py” в папці “service”. У новоствореному модулі пишуться окремі класи *Games* та *Videos*, які будуть включати в себе ці функції.

Перед написанням класів потрібно імпортувати моделі і екземпляр бази даних. Кожен з методів, що буде реалізувати якусь функцію, краще писати з декоратором *@staticmethod*, роблячи його статичним. Всі методи оновлюють базу даних завдяки об’єкту *db*. Також, слід написати коментарі до кожного методу.

Методи класу *Games*:

- `get_all()` – отримати список всіх ігор;
- `get(game_id)` – отримати об’єкт гри по `id`;
- `get_by_name(game_name)` – отримати об’єкт гри по її назві;
- `add(name)` – додати нову гру;
- `remove(game_id)` – видалити гру.

Методи класу *Videos*:

- `get_all()` – отримати всі відео;
- `get_sorted_videos(game_id)` – отримати відфільтрований список відео за грою;
- `add_videos()` – додати відео з трендів;
- `remove_all()` – видалити всі відео.

3.6 Збереження списку відео

Для реалізації можливості збереження списку відео у форматі файлу створюється окремий модуль “file_saver.py” і клас *FileSaver*. Було вирішено, що досить буде зберігати список відео у форматі pdf. Отже, створюється спеціальна функція для вирішення цього завдання (Додаток Д). Вона приймає список відео, обробляє його і повертає файл в байтовому представленні. Для генерації pdf файлу використовується модуль `fpdf`. В майбутньому можна буде за необхідності додати ще методів для генерації інших форматів файлів.

3.7 Розробка тестів

Тестування – важлива частина розробки програмного забезпечення. Вона полягає в тому, щоб виявити коректність роботи програми і можливі помилки. Є багато видів тестування і навіть окрема професія пов’язана з цим, але актуальне і потрібне в даному проєкті саме модульне або юніт-тестування. Так як написання юніт-тестів – це безпосередня робота розробника.

Модульне тестування, або юніт-тестування (англ. Unit testing) – процес в програмуванні, що дозволяє перевірити на коректність окремі модулі вихідного коду програми. Ідея полягає в тому, щоб писати тести для кожної нетривіальною функції або методу. Це дозволяє досить швидко перевірити, чи не призвело чергову зміну коду до регресії, тобто до появи помилок в уже протестованих місцях програми, а також полегшує виявлення і усунення таких помилок [11].

Для написання юніт-тестів в Python існує два потужних модуля: unittest і pytest. У даній роботі використовуватиметься unittest. Ідея полягає в написанні тесту до кожного класу і методів класу, використовуючи різні конструкції з модуля. Тобто для кожного вже написаного модуля потрібно створити окремі файли з назвами «test_модуль» в папці “tests”. Після чого реалізувати тестові класи для кожного класу модулів і їх функцій. При тестуванні модуля “crud.py” потрібно попередньо налаштувати тестову базу даних і заповнити її даними для тестування, це можна зробити за допомогою спеціальних функцій в класі *BaseTestCase*, який буде наслідуватись тестовими класами, що взаємодіють з базою даних (Додаток Е).

3.8 Створення REST та виведення вебсторінок

Для функціонування системи було вирішено створити вебсервіс з дотриманням REST. Цей підхід дозволяє відправляти запити до системи через спеціальні посилання з будь-якого пристрою та отримувати у відповідь дані в JSON форматі. Завдяки цьому будь-який додаток або інший сервіс з доступом зможе отримувати дані і користуватися ними. Таким же чином вебсайт, що розробляється, буде отримувати інформацію, а саме список трендових відео.

Зм.	Аркуш	№ докум	Підпис	Дата

Наступним кроком є формування функцій, що будуть обробляти запити до відповідних вебсторінок і виводити її користувачеві. Для цього функції позначаються декоратором `@app.route('/')`, де вказується адреса після «/», при переході на яку буде виконуватись написана функція і відобразатися сторінка з даними і інтерфейсом. Сторінки інтерфейсу уже готові, щоб їх відобразити потрібно повернути `render_template` з назвою файлу сторінки та можливими параметрами в кінці функції, якщо потрібно передати дані для відображення на сторінці. Наприклад, функція `trends_video` повертає `render_template('service.html', videos=videos)`, що виводить сторінку “service.html” і передає список відео на цю сторінку. Аналогічно створюються і інші функції для виводу сторінок. Ці функції прийнято формувати в папці “views” і модулі “view.py”.

Для вибору гри на сторінці `service` було створено спеціальне поле і кнопка для підтвердження, що відправляє вибір з поля запитом `POST` в відповідну функцію. Для обробки запита додається параметр `methods = 'GET', 'POST'` в декоратор функції, а в тілі функції через умову визначається який запит надійшов і в залежності від цього координується робота функції.

Після написання функції `trends_video`, що відправляє список відео на сторінку `service`, потрібно написати обробку даних і відображення цього списку. Для цього існує мова `Jinja`, яка записується прямо в `HTML` структурі завдяки спеціальним позначенням `{% ... %}`, а після рендера виводиться готова сторінка. `Jinja` підтримує весь стандартний функціонал мов програмування таких як: цикли, умови. Цього досить, аби відтворити дані в найкращому вигляді.

Запуск програми відбувається запуском файлу “main.py”, який потрібно створити в кореневій папці проєкту і імпортувати туди модуль `view` для старту вебсайту через об’єкт `app`.

3.9 Опис роботи програми

Після запуску додатку користувач може перейти на вебсайт. Його зустрічає головна сторінка сайту зі своєю шапкою, коротким описом про сервіс та візуальним поясненням принципу роботи (Додаток Ж).

У користувача є можливість за допомогою меню обрати інші 2 сторінки. Одна з них об'єднує в собі два пункти меню: About Us і Contacts. На цій сторінці більш детальний опис сервісу, принципу його роботи та інша корисна інформація про сервіс. Також на цій сторінці розміщена контактна інформація про розробника з контактами (Додаток І).

Останньою і найважливішою сторінкою є сама сторінка з сервісом. Завдяки спеціальному полю для вибору користувач може обрати гру за якою йому цікаво отримати список відео. Підтвердити свій вибір за допомогою кнопки "Show". В результаті на сторінці виводиться список найпопулярніших відео, що в цей момент знаходяться на платформі YouTube в трендах розділу відеоігри, відфільтрованих за потрібною грою (Додаток К). Також у користувача існує можливість завантажити список цих відео у форматі pdf, натиснувши на кнопку "Download" (Додаток Л).

Система налаштована таким чином, що кожні 12 годин алгоритми отримують список відео з YouTube, після чого додають нові відео в базу даних. База даних очищається раз в тиждень по вівторках. Це зроблено спеціально щоб користувачі могли в понеділок отримати список популярних відео за весь тиждень.

3.10 Висновки до розділу 3

У написанні додатку "Gaming Trends" було використано найсучасніші методи та інструменти розробки, знання мови програмування Python, його фреймворків, модулів, включаючи вебфреймворк Flask і ORM SQLAlchemy, досвід роботи в середовищі PyCharm і з СУБД PostgreSQL.

Було реалізовано підбір найпопулярніших відео на YouTube з вкладки тренди розділу відеоігор завдяки YouTube API, а також реалізовано їх сортування за певними іграми завдяки спеціальному унікальному алгоритму з використанням Selenium.

Було створено REST API та вебсайт для використання сервісом на будь-якому пристрої, зручний і приємний інтерфейс для користувачів, що дозволяє ознайомитись з інформацією про систему та переглянути список трендових відео, відсортований за вибраною грою, можливість його завантажити у форматі pdf.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У результаті виконання роботи було досліджено підходи до створення і побудови інформаційної системи підбору та сортування ігрових трендових відео з платформи YouTube. Було розроблено вебдодаток “Gaming Trends”, який дозволяє користувачу ознайомитись з інформацією про систему та переглянути список трендових відео, відсортований за вибраною грою, можливість його завантажити у форматі pdf.

Мета і завдання курсової роботи було виконано на сто відсотків. Було проаналізовано та досліджено актуальність і методи створення автоматизованої системи “Gaming Trends” для підбору і сортування ігрових трендів з платформи YouTube. Для реалізації системи була розроблена архітектура програмного забезпечення в форматі UML діаграм, спроектована концептуальна, логічна та фізична моделі бази даних, створений дизайн та інтерфейс вебсайту. Завдяки архітектурі була розроблена система, база даних і створений вебсайт для доступу користувачів.

Під час виконання роботи було використано безліч сучасних технологій: Python, Flask, REST, PostgreSQL, SQLAlchemy, Bootstrap та інші. Усі ці технології активно використовуються в практичній діяльності при створенні програмного забезпечення.

Звісно, додаток має потенціал подальшого масштабування та розробки, з додаванням більш широкого функціоналу, іншого інноваційного дизайну та розширенням можливостей. Все залежить від виявлених проблем та необхідності їх вирішити завдяки розширенню можливостей системи.

					<i>ДТЕУ 121 06-09.БР</i>			
					<i>Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube</i>	<i>Стадія</i> ВП	<i>Аркуш</i> 43	<i>Аркушів</i> 46
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		25.04.22				
Керівник		Жирова Т. О.		25.04.22				
Гарант		Рзаєва С. Л.		25.04.22				
Розроб		Капуста Н. В.		25.04.22	<i>Висновки та пропозиції</i>	Факультет інформаційних технологій, 4 курс, 6 група		

Створений вебсайт надає можливість користувачу отримати та завантажити список трендових відео з платформи YouTube, який можливо відфільтрувати за вибраною грою, з будь-якого пристрою.



									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 06-09.БР</i>				44

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dwyer G. Flask vs. Django: Why Flask Might Be Better [Електронний ресурс] / Gareth Dwyer. – 2017. – Режим доступу до ресурсу: <https://www.codementor.io/@garethdwyer/flask-vs-django-why-flask-might-be-better-4xs7mdf8y>.
2. Software architecture - Wikipedia [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Software_architecture.
3. Rumbaugh J. The unified modeling language reference manual / J. Rumbaugh, G. Booch, I. Jacobson., 1999. – 568 с.
4. Beazley D. Python Cookbook: Recipes for Mastering Python 3 / D. Beazley, B. Jones.. – 708 с. – (3).
5. Aggarwal A. Flask Framework Cookbook Second Edition / Aggarwal Aggarwal. – Birmingham: Packt Publishing, 2019. – 258 с
6. Python documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/>.
7. Flask’s documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://flask.palletsprojects.com/en/1.1.x/>
8. YouTube Data API | Google Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/youtube/v3>.
9. UML Diagram [Електронний ресурс] – Режим доступу до ресурсу: <https://www.smartdraw.com/uml-diagram/>.
10. Гриценко В.Г. Використання системи управління версіями Git для організації командної роботи над ІТ проектом / В.Г.Гриценко, О.М.Подольн // Інформаційні технології і засоби навчання. – 2014. – Том 39. – № 1. – С.251-252.

					<i>ДТЕУ 121 06-09.БР</i>			
					<i>Автоматизована система «Gaming Trends» підбору і сортування ігрових трендів з платформи YouTube</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>СВД</i>	45	46
Зав. каф.		Криворучко О.В.		13.12.21	<i>Список використаних джерел</i>	Факультет інформаційних технологій, 4 курс, 6 група		
Керівник		Жирова Т. О.		13.12.21				
Гарант		Рзаєва С. Л.		13.12.21				
Розроб		Капуста Н. В.		13.12.21				

11. Unit testing - Wikipedia [Електронний ресурс] – Режим доступу до ресурсу:

https://en.wikipedia.org/wiki/Unit_testing

12. Database design [Електронний ресурс] – Режим доступу до ресурсу:

https://en.wikipedia.org/wiki/Database_design

13. Documentation PostgreSQL 14 Chapter 8. Data Types [Електронний ресурс] –

Режим доступу до ресурсу: <https://www.postgresql.org/docs/14/datatype.html>.

14. Bootstrap documentation [Електронний ресурс] – Режим доступу до ресурсу:

<https://getbootstrap.com/docs/5.0>



Зм.	Аркуш	№ докум	Підпис	Дата

ДТЕУ 121 06-09.БР

Аркуш

46

ДОДАТКИ

Додаток А

Макети інтерфейсу і дизайну веб-сторінок

Youtube_Trends_Designs.pdf

Додаток Б

Об'єкти-моделі бази даних

```
class Game(db.Model):
    """Game model for SQL Alchemy"""
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(45))
    videos = db.relationship('Video', backref='game')

    def __eq__(self, other):
        if (self.id == other.id) and (self.name == self.name):
            return True
        return False

class Video(db.Model):
    """Video model for SQLAlchemy"""
    id = db.Column(db.String(11), primary_key=True)
    title = db.Column(db.String(100))
    channel = db.Column(db.String(100))
    url = db.Column(db.String(28))
    thumbnail = db.Column(db.String(100))
    game_id = db.Column(db.Integer, db.ForeignKey('game.id', onupdate='CASCADE',
ondelete='CASCADE'))

    def __eq__(self, other):
        if self.id == other.id:
            return True
        return False
```

Додаток В

Клас YouTubeApiController для роботи з YouTube API

```
class YouTubeApiController:
    """Class for work with Youtube Api"""
    @staticmethod
    def get_trending_videos(region_codes=region_codes_cfg):
        """
        Get games trending videos from Youtube
        :param region_codes: list with region codes ['en', 'us', ...]
        :return: list with videos ids
        """
        videos_ids = []
        for code in region_codes:
            url =
f'https://youtube.googleapis.com/youtube/v3/videos?chart=mostPopular&maxResults=30&regionCode={code}&videoCategoryId=20&key={api_key}'
            r = requests.get(url)
            videos = dict(r.json())['items']
            for video in videos:
```

```

        videos_ids.append(video['id'])
    return videos_ids

    @staticmethod
    def get_videos_info(videos_ids):
        """
        Get all videos' info from ids
        :param videos_ids: List with videos ids
        :return: videos list with full info about them
        """
        url = 'https://youtube.googleapis.com/youtube/v3/videos?part=snippet'
        for video_id in videos_ids:
            url += f'&id={video_id}'
        url += f'&key={api_key}'
        r = requests.get(url)
        try:
            videos = r.json()['items']
        except:
            videos = []
        return videos

```

Додаток Г

Модуль videos_filter

```

from service.youtube_api import YouTubeApiController
from selenium.webdriver import Chrome
from selenium.webdriver import ChromeOptions
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager

class VideosFilter:
    """Class for work and filter all trending videos"""

    @staticmethod
    def get_sorted_videos(games, videos_ids=None):
        """
        Get all trending videos and filter it by games
        :param videos_ids: if soft need definitely ids and for testing
        :param games: games names that uses for filtering
        :return: filtered list of trending videos
        """
        if videos_ids is None: # for test case
            videos_ids = set(YouTubeApiController.get_trending_videos()) #
            without duplicates
        videos_ids = list(videos_ids)
        sorted_videos_ids = []
        sorted_games = []
        for video_id in videos_ids:
            game = VideosFilter.check_game(video_id)
            if game in games:
                sorted_videos_ids.append(video_id)
                sorted_games.append(game)
        sorted_videos =
        YouTubeApiController.get_videos_info(sorted_videos_ids)
        for i in range(0, len(sorted_videos)):
            sorted_videos[i]['game'] = sorted_games[i]
        return sorted_videos

    @staticmethod
    def check_game(video_id):

```



```

"""
Check game by link uses scraping
:return str: game's name
"""
url = VideosFilter.prepare_link(video_id)
# use scraping(парсинг) with selenium
data = []
op = ChromeOptions()
op.add_argument('--headless')
s = Service(ChromeDriverManager().install())
try: # if game doesnt selected in video
    with Chrome(service=s, options=op) as driver:
        wait = WebDriverWait(driver, 5)
        driver.get(url)

        for info in wait.until(
            EC.presence_of_all_elements_located((By.CSS_SELECTOR, "#title.ytd-rich-metadata-renderer"))):
                data.append(info.text)
            game = data[0]
        except:
            game = 'None'
        return game

    @staticmethod
    def prepare_link(video_id):
        """
        Use video's id and create url
        :param video_id: video's id
        :return: video's url
        """
        return 'https://youtu.be/' + video_id

```

Додаток Д

Функція для створення pdf файлу зі списком відео

```

def generate_pdf(videos):
    """
    Generates pdf file with videos list
    :param videos: list of videos
    :return: pdf file in bytes string
    """
    pdf = FPDF()
    pdf.add_page()
    pdf.set_font("Arial", size=14)
    for i in range(0, len(videos)):
        pdf.cell(200, 10, videos[i].title, link=f'{videos[i].url}', ln=i)
    return pdf.output(dest='S').encode('latin-1')

```

Додаток Е

Клас для налаштування і заповнення тестової бази даних

```

class BaseTestCase(unittest.TestCase):
    def setUp(self):
        app.config.from_object(TestConfiguration)
        self.app = app.test_client()

```

```

db.create_all()
db.session.add(Game(name='Minecraft'))
db.session.add(Video(id='M8Kx63nCIXo', game_id=1,
url='https://youtu.be/M8Kx63nCIXo',
title='Minecraft Siren Head vs GTA 5 Siren Head',
channel='Никита Золотаревич'))
db.session.add(Video(id='4eKXwKDDxYA', game_id=1,
url='https://youtu.be/4eKXwKDDxYA',
title='100 Days - [Minecraft Superflat]', channel='Luke
TheNotable'))
db.session.commit()

def tearDown(self):
db.session.remove()
db.drop_all()

```

Додаток Ж

Головна сторінка вебсайту



Who we are?

Youtube Gaming Trends - це сервіс для підбору та сортування ігрових трендів з платформи YouTube. Це унікальний сервіс, що дозволить економити час в пошуках ідей, слідувати за трендами в зручному вигляді та можливість обирати тематику(гру). Алгоритми кожного дня прослідковують і аналізують відео з вкладки тренди на YouTube, завдяки чому користувачі отримують список найактуальніших і популярних відео на платформі.

[READ MORE](#)



How we work



Get in touch



Email

nikita@yoola.com



Phone

+38 099 063 55 14

Copyright © 2022

Додаток II

Сторінка About Us і Contacts

YOUTUBE TRENDS

Home About Us Service Contacts

About Us

who we are

Сервіс Gaming Trends дає можливість отримати список останніх трендових відео з англійського YouTube підібраних за певною грою в зручному для вас вигляді. Додаток допомагає знизити витрати часу і зробити процес підбору ідей, аналізу трендів, значків та метаданих відео більш зручним для авторів YouTube каналів, менеджерів, сценаристів, аналітиків ігрового сегменту та маркетингові.

Алгоритми кожного дня прослідковують і аналізують відео з вкладки тренди на YouTube, завдяки чому користувачі отримують список найактуальніших і популярних відео на платформі. Система збирає трендові відео протягом тижня, після чого по вівторкам дані удаляються. Так що для перегляду трендів за весь тиждень рекомендується переглядати їх в понеділок.

Список відео зі значками відео можна переглядати на сайті, обравши в меню сторінку Service. Також список відео можна завантажувати у форматі pdf. Використання сервісу є безкоштовним. Знімайте трендові відео і отримуйте мільйони переглядів! Приємного користування!

Contacts



Nikita Kapusta

Email: nikita@yoola.com

Phone: +38 099 063 55 14

Instagram: [nik.zolo](#)

Get in touch



Email

nikita@yoola.com



Phone

+38 099 063 55 14

Додаток К

Сторінка зі списком трендових відео

YOUTUBE TRENDS

[Home](#) [About Us](#) [Service](#) [Contacts](#)

Service

Game

All

Show

Download



Hermitcraft 9: Episode 3 - MY STARTER HOUSE!

Grian
Minecraft



Minecraft but I beat it Backwards

Craftee
Minecraft



Minecraft But There's Custom Bosses

PrestonPlayz
Minecraft



From APHMAU To ANIMATED In Minecraft!

Aphmau
Minecraft



Aphmau's 100% IMPOSSIBLE Minecraft Prison!

Aphmau
Minecraft



We Survived Eating Minecraft Food for 24 Hours

The Boys
Minecraft



NAZ KICKED SABRE OUT OF MINECRAFT Gaming w/ The Norris Nuts

Norris Nuts Gaming
Minecraft



If I Touch Grass Minecraft Gets More Realistic

Shark
Minecraft



Get in touch



Email

nikita@yoola.com



Phone

+38 099 063 55 14

Copyright © 2022

Додаток Л

Список відео в форматі pdf
trends_list.pdf

