ХАРКІВСЬКИЙ ПАТЕНТНО-КОМП’ЮТЕРНИЙ КОЛЕДЖ

Відділення «Інженерія програмного забезпечення»

Предметно-циклова комісія «Інженерія програмного забезпечення»

«Система сумісного перегляду відео з YouTube»

**Пояснювальна записка**

КП 121.6Х.20.81

|  |  |  |
| --- | --- | --- |
| Виконали: | Водянов | А.А |
|  | Якименко | Р.А |
| Керівник: | Чередніченко | С.Р. |

Харків – 2021 рік

Зміст

[ВСТУП 3](#_Toc66965072)

[1 ТЕОРЕТИЧНИЙ РОЗДІЛ 3](#_Toc66965073)

[1.1 Аналіз предметної області 3](#_Toc66965074)

[1.2 Аналіз та обгрунтування вибору технології розробки застосунку 3](#_Toc66965075)

[2 ПРОЕКТНИЙ РОЗДІЛ 3](#_Toc66965076)

[2.1 Проектування інформаційної системи 3](#_Toc66965077)

[2.2 Розробка структури інформмаційної системи 3](#_Toc66965078)

[2.3 Розробка бази даних 3](#_Toc66965079)

[3 Розробка застосунку 3](#_Toc66965080)

[3.1 Розробка екранних форм 3](#_Toc66965081)

[3.2 Розробка функціоналу інформаційної системи 3](#_Toc66965082)

[4 Висновки 3](#_Toc66965083)

[5 Перелік посилань 3](#_Toc66965084)

[6 ДОДАТОК А. Програмний код основної функції 3](#_Toc66965085)

# ВСТУП

Система сумісного перегляду відео – веб-сервіс, який надає можливість дивитися відео з YouTube разом з іншими людьми одночасно з різних куточках планети через інтернет.

Деякі означення:

* Блог - вебсайт, головний зміст якого є регулярно додавані записи, зображення чи мультимедіа. Для блогів характерні короткі записи тимчасової значущості.
* Блогер – людина, яка веде блог.
* Ютубер – людина, яка веде свій канал на YouTube. Зазвичай Блогер і Ютубер – синоніми.
* Плейліст – черга з якого-небуть медіа-контенту. Після закінчення одного медіа-контенту з черги запускається наступний у черзі. Наприклад: відео, музика і тд.

# ТЕОРЕТИЧНИЙ РОЗДІЛ

## 1.1 Аналіз предметної області

## 1.1.1 Характеристика об’єкта і результатів його функціонування

Існує відеохостин YouTube, який надає можливість завантажити, переглядати завантаженні відео інших людей, а також запуск прямої трансляції, до якої можна підключитись і дивитись її онлайн з іншими користувачами.

## 1.1.2 Опис існуючої інформаційної системи

Відеохостинг YouTube надає можливість:

* Знайти відео у пошуку та переглянути його.
* Написати коментар під відео, оцінити його, підписатися на автора відео.
* Створити чергу з відео, які бажаєте переглянути.
* Поділитися в соціальних мережах (Telegram, Twitter, Instagram і тд) з бажаним відео.
* Створити канал.
* Завантажити свої відео.
* Переглянути активність своїх відео.
* Відповідати на коментарі.
* Запуск прямої трансляції.

## 1.1.3 Опис недоліків існуючої інформаційної системи

В цілому існуюча система працює добре але не вистачає одної функції, яка була б у нагоді - сумісний перегляд вже завантажених відео.

Користувачі відеохостину для перегляду сумісного відео використовують так звані «милиці», а саме: на рахунок три запускають одне і теж саме відео на своїй приладах, за допомогою додатків-зв’язку, наприклад Discord, вмикають трансляцію екрану для інших людей у кімнаті зв’язку та демонструють відео.

## 1.1.4 Обґрунтування необхідності вдосконалення інформаційної системи об’єкта

Такі варіанти перегляду сумісних відео, як наведені вище не є ефективними та зручними:

* Якщо використовувати «милицю» одночасного запуску відео на рахунок 3:

1. Не має точної синхронізації
2. Потрібно мати додатковий зв’язок з людиною, з якою хочеш подивитися відео
3. Якщо хтось з людей поставить відео на паузу то інша людина продовжить дивитися відео і не буде знати, що хтось відійшов.
4. У когось може з’явитися реклама, яка зупиняє перегляд відео, інша людина може продовжити дивитися відео нічого не підозрюючі.

* Якщо використовувати додаток для зв’язку, у якому є демонстрація екрану (пряма трансляція і тд):

1. Не має точної синхронізації
2. У людини, яка демонструє екран, знижується продуктивність із-зі трансляції
3. Люди, які дивляться трансляцію, отримують зображення поганої якості.
4. Лічильник переглядів під відео зараховує тільки один перегляд, бо, формально, його переглядає одна людина і транслює іншим свій екран.

## 1.1.5 Цілі, критерії та обмеження створення АС

Ціль - забезпечити зручний перегляд сумісного відео на нашому сайті у створеній онлайн-кімнаті, зручну чергу з бажаними відео для перегляду, чат для обговорення відео чи просто спілкування.

## 1.1.6 Функції і завдання створюваної АС

1. Кожний користувач може створити нову кімнату або приєднатися вже до існуючої
2. У кімнаті можна додавати нові відео у чергу, видаляти відео з черги, перемикати відео з черги, ставити на плей/паузу відео, коментувати у чаті.
3. Головний у кімнаті може надавати права на використання функцій у пункті 2, змінити назву кімнати, вигнати користувача з кімнати.

## 1.1.7 Висновки і пропозиції

Система сумісного перегляду відео з YouTube – потрібний сервіс для дружньої компанії, або знайти нові знайомства підключившись до вже існуючої кімнати. Тут можна провести гарно і весело свій час.

Для створення такого сервісу буде використання спеціальних інструментів YouTube API .

## 1.2 Аналіз та обґрунтування вибору технології розробки застосунку

Перед початком розробки системи сумісного перегляду був проаналізований ринок технологій, які необхідні для реалізації цієї системи.

Для розробки Front-end частини було прийняте рішення розробити веб-сайт з використанням HTML5 + CSS3 для відображення їх у браузері, адже у кожного на комп’ютері, смартфоні встановлений браузер. Також такий підхід зробить систему сумісного перегляду кросплатформеною для всіх операційних систем, які мають браузер для виходу в Інтернет.

Back-end частину потрібно розглянути більш детальніше. Так як система сумісного перегляду має динамічну структуру роботи, наприклад синхронізація відео-потоку з іншими учасниками віртуальної кімнати, то необхідно підібрати фреймворк, який може працювати зі архітектурним підходом Rest, для передачі даних через мережу.

Нами був обраний фреймворк для розробки веб-застосунків Django. Він має зручний функціонал для роботи з REST API, а також дозволяє поділяти Back-end частину на підпрограми та реалізовувати їх окремо.

Для зберігання тимчасових даних і даних, до яких потрібен постійний, а головне швидкий доступ обрана NoSQL база даних Redis. Redis - резидентна система управління базами даних, орієнтована на досягнення максимальної продуктивності на атомарних операціях. Важливим фактором, який вплинув на вибір Redis, є її відкритий вихідний код на [github](https://github.com/redis/redis), аналіз якого дає нам можливість більш глибоко розуміти як працює технологія і виходячи з цього будувати максимально оптимальні рішення в нашому програмному продукті.

Для збереження даних кімнати була використана СУБД sqlite. У SQL базі ми зберігаємо дані, до яких не виконується частий доступ. Вибраний нами веб фреймворк Django реалізує шаблон MVC, який дає нам можливість в будь-який момент перейти на іншу більш потужнішу СУБД, але на даному етапі нам вистачає функціоналу sqlite.

Зв’язок між клієнтом і сервером здійснюється за допомогою технології WebSocket та REST API. WebSocket - протокол зв'язку поверх TCP-з'єднання, призначений для обміну повідомленнями між браузером і веб-сервером в режимі реального часу.

# ПРОЕКТНИЙ РОЗДІЛ

## 2.1 Проектування інформаційної системи

Проектування даної системи відбувалося за такими етапами: спочатку були визначені цілі, завдання, виявлені вимоги, визначено функціонал, складено технічне завдання Система складається з таких частин:

1. Front-end – набір сторінок, стилів та скриптів, для створення користувацького інтерфейсу, який дозволяє використовувати зручно систему сумісного перегляду. На рисунку 2.1.1 та 2.1.2 продемонстровано макети сторінок
2. Back-end – набір модулів для отримання, відправлення та обробки запитів від клієнта. Має в наявності локальну базу даних sqlite, та екземпляр розподіленого сховища Redis. На рисунку 2.1.3 продемонстровано послідовність дії чату.



Рисунок 2.1.1 – Макет сторінки віртуальної кімнати



Рисунок 2.1.2 – Макет сторінки пошуку віртуальної кімнати

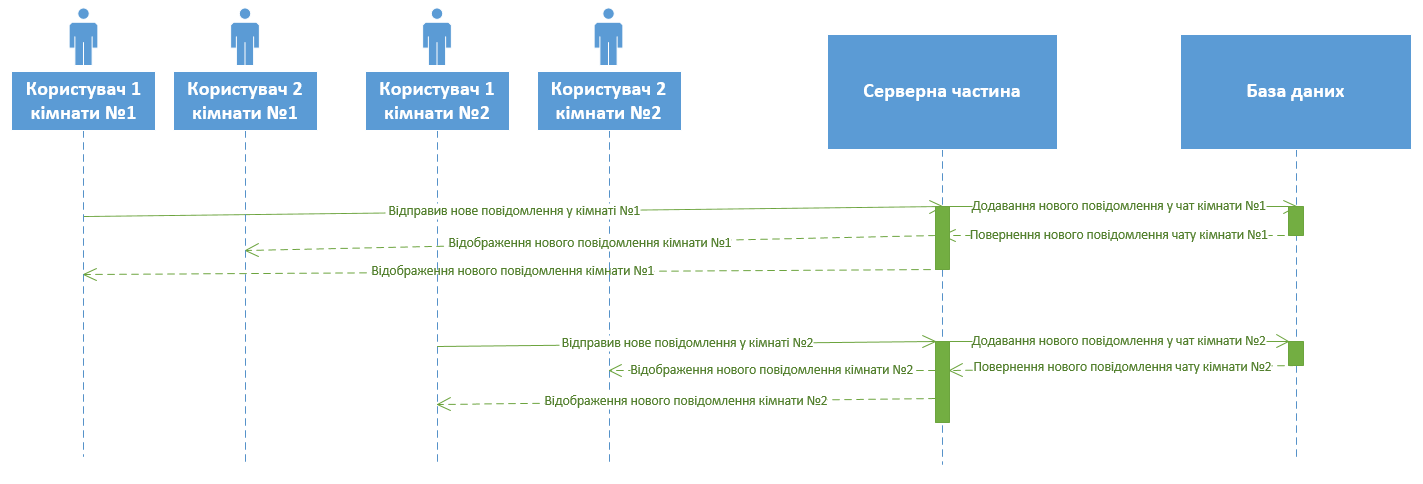


Рисунок 2.1.3 - Діаграма послідовності для чату

2.2 Розробка структури інформаційної системи

Розробку даної інформаційної системи було поділено на функції цієї системи, а саме:

1. Створення віртуальної кімнати – на серверній частині буде виділено ресурси (пам’ять, процесорний час), завдяки цим ресурсам буде створена нова кімната на сервері.
2. Пошук віртуальної кімнати – спочатку на серверну частину буде відправлений запит з усіма існуючими кімнатами на цей час, після отримання результату запита, інформація буде відображена на сторінці користувача.
3. Приєднання до віртуальної кімнати – користувач може знайти існуючу кімнату в пошуку або отримати посилання на кімнату одним з учасників. При вході у кімнату на сервер відправляється запит з отриманням поточних даних кімнати: учасники кімнати, плейліст з відео, поточне відео та його тайм-код, повідомлення у чаті.
4. Додавання відео в плейліст – користувач бере посилання на відео з YouTube та вставляє у спеціальний елемент на сторінці кімнати. Буде зроблена перевірка на існування цього відео за допомогою API YouTube (відправка запиту на сервера YouTube з пошуком відео за посиланням), якщо таке відео існує – воно буде додане.
5. Зміна порядку черги – за допомогою спеціальних кнопок можна змінити порядок відео у черзі.
6. Надіслання повідомлення у чат – у спеціальному полі на робочій поверхні можна вписати повідомлення, натиснути на кнопку «відправити», після буде створений запит, який відправить повідомлення користувача на сервер, де за допомогою Redis pub/sub повідомлення пошириться на всіх користувачів у кімнати.
7. Надання прав користувачам головним у кімнаті – володар віртуальної кімнати може надавати у робочій поверхні.

## 2.3 Розробка бази даних

Під час розробки бази даних для системи сумісного перегляду відео було дотримано такі правила:

* База даних буде працювати поруч з головним сервером. Вона повинна бути завжди доступна та узгоджена (CAP-теорема CA). Зберігати в собі список кімнат, список користувальницьких сесій та інше.
* Усі дані будуть синхронізовані з користувачами при вході у віртуальну кімнату чи при пошуку віртуальних кімнат.

Тому були розроблені запити для отримання даних з бази даних, мовою SQL.

# Розробка застосунку

## 3.1 Розробка екранних форм

При розробці користувацького інтерфейс, а саме веб-сторінок, було дотримано декілька парадигм:

* Веб-сторінки повинні буди адаптовані – підлаштовуватися під різні розміри дисплея, тач-скріна, екрану і так далі.
* Веб-сторінка не повинна бути статичною – деякі дані на сторінці будуть змінюватися за допомогою запитів до серверною частини, які надаватимуть необхідну інформацію, дані, метрики і т.д.

Зовнішній вигляд сторінок має мінімалістичний стиль з темними кольорами для зручності у використані як вдень так і вночі. Всі сторінки веб-застосунку дотримуються єдиного стилю.

## 3.2 Розробка функціоналу інформаційної системи

# Висновки

# Перелік посилань

# ДОДАТОК А. Програмний код основної функції