

# ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

## «Програмний продукт обміну відеоконтентом в реальному часі»

Студента 4 курсу, 7 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми «Інженерія  
програмного забезпечення»

Бортяного Максима  
Анатолійовича

\_\_\_\_\_

підпис студента

Науковий керівник  
асистент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

Хорольська Карина  
Вікторівна

\_\_\_\_\_

підпис керівника

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

Рзаєва Світлана  
Леонідівна

\_\_\_\_\_

підпис гаранта

# Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

## Затверджую

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

## Завдання

### на випускний кваліфікаційний проєкт студентів

Бортяному Максиму Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Програмний продукт обміну  
відеоконтентом в реальному часі»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту розробити та впровадити програмний продукт обміну  
відеоконтентом в реальному часі

Об'єкт дослідження обмін відеоконтентом у реальному часі

Предмет дослідження розробка механізму для обміну відеоконтентом в  
реальному часі

4. Консультанти проекту із зазначенням розділів, які консультують:

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

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ПРОГРАМНОГО ПРОДУКТА

1.1. Обмін відеоконтентом

1.2. Опис проблеми

1.3. Стан дослідження проблеми

1.4. Технічне завдання

1.5. Висновок до розділу 1

РОЗДІЛ 2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

2.1. Архітектура продукту

2.2 Інструменти розробки

2.3 Архітектура серверу

2.4. Проектування програмного продукту

2.5 Проектування бази даних

2.6. Висновок до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1. Розробка механізму синхронізації

3.2.

3.3. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## 6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ПРОГРАМНОГО ПРОДУКТУ</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проєкту на кафедру (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	29.05.2023 – 02.06.2023	
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	05.06.2023	05.06.2023
12.	<i>Здача прошого випускного кваліфікаційного проєкту на кафедру</i>	05.06.2023	05.06.2023
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту \_\_\_\_\_

Хорольська К.В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми \_\_\_\_\_

Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент \_\_\_\_\_

Бортяний М. А.

(прізвище, ініціали, підпис)



## АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці та реалізації програмного продукту для обміну відеоконтенту в реальному часі. Досліджено потреби користувачів у сфері відеоконтенту та визначено вимоги до функціональності та безпеки системи.

В результаті порівняльного аналізу аналогічних рішень визначено, що розроблений програмний продукт відповідає потребам користувачів і має конкурентні переваги, такі як синхронізація в реальному часі та зручний інтерфейс.

Розробка серверної частини виконана у середовищі ASP.NET з використанням Visual Studio та SignalR для реалізації механізму синхронізації. База даних Ms-SQL використовується для збереження інформації про користувачів та відеоматеріали.

Готовий програмний продукт було успішно протестовано з метою перевірки його функціональності, ефективності та надійності. Тести підтвердили правильну роботу синхронізації відеоконтенту та інші функціональні можливості продукту.

**Ключові слова:** відеоконтент, синхронізація, програмний продукт, ASP.NET, Visual Studio, SignalR, Ms-SQL, клас, хаб.

## ABSTRACT

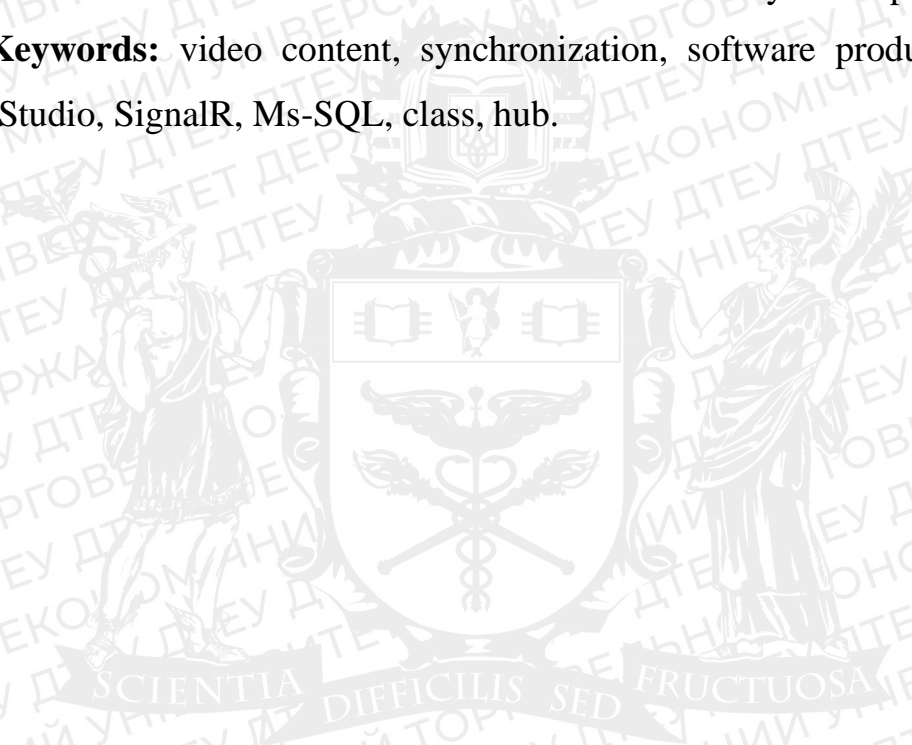
According to the purpose of the research, the work is devoted to the development and implementation of a software product for sharing video content in real time. The needs of users in the field of video content were studied and the requirements for the functionality and security of the system were determined.

As a result of a comparative analysis of similar solutions, it was determined that the developed software product meets the needs of users and has competitive advantages, such as real-time synchronization and a convenient interface.

The development of the server part was carried out in the ASP.NET environment using Visual Studio and SignalR to implement the synchronization mechanism. The Ms-SQL database is used to store information about users and video materials.

The finished software product has been successfully tested to verify its functionality, efficiency and reliability. The tests confirmed the correct operation of the synchronization of video content and other functionality of the product.

**Keywords:** video content, synchronization, software product, ASP.NET, Visual Studio, SignalR, Ms-SQL, class, hub.



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

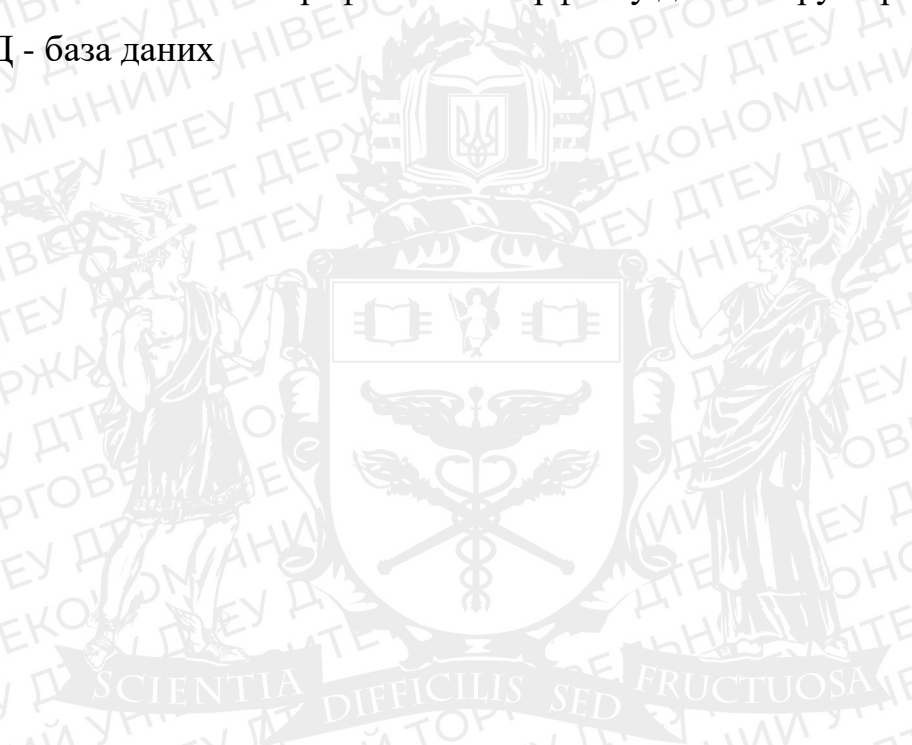
ПЗ – програмне забезпечення

MVC (англ. Model-View-Controller) – архітектурний патерн

MSSql – система управління базами даних

Чек-бокс – елемент графічного інтерфейсу для вибору параметрів

БД - база даних



<i>ДТЕУ 121 07-01.БР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		23.12.22
Керівник		Хорольська К.В.		23.12.22
Гарант		Рзаєва С.Л.		23.12.22
Розробив		Бортяний М.А.		23.12.22
<i>Зміст</i>				
			<i>Програмний продукт обміну відеоконтентом в реальному часі</i>	
			<i>Стадія</i>	<i>Арку</i>
			3	3
			<i>Аркушів</i>	
			43	
<i>Факультет інформаційних технологій 4 курс, 7 група</i>				



## ЗМІСТ

<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ПРОГРАМНОГО ПРОДУКТА .....</b>	<b>7</b>
1.1. Обмін відеоконтентом.....	7
1.2. Опис проблеми.....	8
1.3. Стан дослідження проблеми.....	10
1.4. Технічне завдання.....	11
1.5. Висновки до розділу 1.....	14
<b>РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....</b>	<b>15</b>
2.1. Архітектура продукту .....	15
2.2. Інструменти розробки .....	16
2.3. Архітектура серверу .....	18
2.4. Проектування програмного продукту .....	20
2.4. Проектування бази даних .....	26
2.5. Висновки до розділу 2.....	30
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ .....</b>	<b>31</b>
3.1. Розробка механізму синхронізації .....	31
3.2. Методи авторизації та аутентифікації.....	37
3.3. Висновок до розділу 3 .....	39
<b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....</b>	<b>40</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>42</b>
<b>ДОДАТКИ.....</b>	<b>44</b>

<i>ДТЕУ 121 07-01.БР</i>								
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		23.12.22	Програмний продукт обміну відеоконтентом в реальному часі	<i>Стадія</i>	<i>Арку</i>	<i>Аркушів</i>
Керівник		Хорольська К.В.		23.12.22		3	3	43
Гарант		Рзаєва С.Л.		23.12.22		<i>Факультет інформаційних технологій 4 курс, 7 група</i>		
Розробив		Бортяний М.А.		23.12.22				
<i>Зміст</i>								

## ВСТУП

В сучасному світі величезну увагу приділяють відеоконтенту, який стає все більш популярним та необхідним для бізнесу, освіти, розваг та багатьох інших галузей. З огляду на це, розробка програмного продукту обміну відеоконтенту стала актуальною та важливою задачею, що має великий потенціал для розвитку та успіху на ринку.

У даній дипломній роботі розглядається розробка програмного продукту обміну відеоконтенту, який дозволяє користувачам обмінюватися відео зі своїми контактами, а також зберігати та організовувати власний відеоконтент. Основними метою та завданнями дипломної роботи є аналіз ринку відеоконтенту, вивчення потреб та очікувань користувачів, розробка архітектури та функціональних можливостей програмного продукту, а також тестування та оцінка ефективності розробленого продукту.

Мета дослідження полягає в розробці та впровадженні програмного продукту обміну відеоконтентом в реальному часі.

Об'єктом дослідження є обмін відеоконтентом у реальному часі. Дослідження спрямоване на вивчення та аналіз технологій, платформ, алгоритмів і рішень, які використовуються для забезпечення ефективного та безперебійного обміну відеоданими між користувачами в режимі реального часу. Основною метою дослідження є створення програмного продукту, який забезпечує зручну та надійну платформу для обміну відеоконтентом між користувачами, забезпечує швидку передачу даних, якість відео та аудіо, інформаційну безпеку та конфіденційність, а також різноманітні функції.

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 07-01.БР			
Зав. каф.	Криворучко О.В.			23.12.22	Програмний продукт обміну відеоконтентом в реальному часі	Стадія	Аркуш	Аркушів
Керівник	Хорольська К.В.			23.12.22		В	4	43
Гарант	Рзаєва С.Л.			23.12.22		Факультет інформаційних технологій		
Розробив	Бортяний М.А.			23.12.22		4 курс, 7 група		
					Вступ			

У відповідності з метою дослідження поставлені наступні завдання:

- Аналіз існуючої платформи та технологій: детальний аналіз існуючих платформ і технологій для обміну відеовмістом у реальному часі. Дослідити їх функціональність, можливості, архітектуру, алгоритми передачі даних і масштабованість.
  - Визначення вимог до програмного продукту: установіть вимоги до програмного продукту для обміну відеовмістом у реальному часі. Розгляньте функціональні, нефункціональні, технічні та інші аспекти, необхідні для забезпечення конкретних цілей і потреб користувачів.
  - Розробка архітектури програмного продукту: розробка архітектури програмного продукту, яка відповідає вимогам і цілям дослідження. Визначте компоненти системи, їхні зв'язки, інтерфейси, протоколи зв'язку, алгоритми обробки даних та безпеку.
  - Розробляйте логічні та фізичні моделі баз даних: створюйте логічні та фізичні моделі баз даних для зберігання та керування інформацією про користувачів, відеодані, сеанси та інші пов'язані дані.
- Методи дослідження:
- Аналіз вимог користувачів: Виконання детального аналізу вимог і потреб користувачів, що включає збір і аналіз вимог, розуміння їхніх очікувань та специфікацію функціональності, яка повинна бути реалізована у програмному продукті.
  - Проектування архітектури системи: Створення архітектури програмного продукту, включаючи визначення компонентів системи, їх взаємозв'язків, інтерфейсів та взаємодій. Мета

						Аркуш
					ДТЕУ 121 07-01.БР	5
Зм.	Аркуш	№ докум	Підпис	Дата		

полягає в створенні структурованої та масштабованої архітектури, яка задовольняє вимоги проекту.

- Розробка функціональності: Реалізація функціональних можливостей програмного продукту на основі встановлених вимог. Це включає створення модулів для реєстрації користувачів, управління контентом, пошуку та відтворення відеоконтенту, а також забезпечення безпеки даних.
- Тестування та верифікація: Проведення тестування програмного продукту для перевірки його працездатності, надійності та відповідності вимогам. Це може включати функціональне та навантажувальне тестування, а також верифікацію правильності виконання функцій.

Наукова новизна дослідження полягає в розробці нового алгоритму обміну відеоконтентом в реальному часі, що забезпечує ефективну та швидку передачу даних.

Практичне значення дослідження полягає в створенні програмного продукту для обміну відеоконтентом в реальному часі, який може знайти широке застосування в різних галузях. Цей продукт може бути використаний для відеоконференцій, стрімінгових платформ, віртуального навчання, онлайн трансляцій подій та багатьох інших сфер.

						Аркуш
					ДТЕУ 121 07-01.БР	6
Зм.	Аркуш	№ докум	Підпис	Дата		

## РОЗДІЛ 1

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ПРОГРАМНОГО ПРОДУКТА

### 1.1.Обмін відеоконтентом

Відеоконтент - це мультимедійний матеріал у вигляді відеофайлу, який містить зображення та звукову інформацію. Він може включати в себе різноманітні типи відео, такі як фільми, серіали, відеокліпи, рекламні ролики, відеоуроки, стріми тощо.

Відеоконтент є динамічним засобом передачі інформації, який здатний ефективно комунікувати повідомлення та сприйматися аудиторією. Він може бути використаний для розваг, освіти, реклами, засобу комунікації та багатьох інших цілей.

Завдяки поширенню високошвидкісного Інтернету та розвитку технологій, відеоконтент став доступним для перегляду та обміну в реальному часі. Платформи для стрімінгу відео, відеоконференції, соціальні мережі з відеофункціями та інші програмні продукти дозволяють користувачам створювати, спільно використовувати, коментувати та взаємодіяти з відеоконтентом.

Відеоконтент володіє великим потенціалом в електронній комунікації, маркетингу, освіті та інших галузях. Його використання дозволяє створювати захоплюючі візуальні історії, передавати емоції, донести повідомлення та сприяти взаємодії з аудиторією.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 07-01.БР</i>			
Зав. каф.		Криворучко О.В.		27.01.23	Програмний продукт обміну відеоконтентом в реальному часі	Стадія	Аркуш	Аркушів
Керівник		Хорольська К.В.		27.01.23		РІ	7	43
Гарант		Рзаєва С.Л.		27.01.23		Факультет інформаційних технологій 4 курс, 7 група		
Розробив		Бортяний М.А.		27.01.23				
					Аналіз предметної області застосування програмного продукту			

## 1.2.Опис проблеми

Поява різноманітних платформ для обміну відеоконтентом, таких як Twitch, YouTube, Facebook Live, Instagram Live та незліченна кількість інших, підкреслює проблему обміну візуальним контентом у режимі реального часу, що впливає із зростання переваги відео як основного каналу для передачі та споживання інформації. Зі швидким розвитком технологій захоплення, передачі та редагування відео вирішення цієї скрутної ситуації є більш важливим, ніж будь-коли раніше.

Коли справа доходить до збільшення використання та оцінки відеоконтенту, неминуче виникають проблеми, які потребують уваги, щоб їх пом'якшити. Однією з таких проблем є якість передачі відео. Для забезпечення потокового відеовмісту в реальному часі стабільна та безперебійна передача даних є обов'язковою, щоб забезпечити якість відео. Проблема полягає в обмеженій пропускну здатності мережі та неналежній якості інтернет-з'єднань, що може призвести до збоїв або перебоїв, які погіршують враження від глядача.

Під час масових заходів або популярних трансляцій мережева інфраструктура стикається зі значними труднощами через великі обсяги відеоданих, які потрібно обробляти. Збільшене навантаження на мережеві ресурси, спричинене величезним споживанням відео, може призвести до затримок. Щоб вирішити цю проблему, ми повинні вжити низку заходів. Одним із таких кроків є вдосконалення нашої мережевої інфраструктури, тоді як іншим є розробка інноваційних алгоритмів керування трафіком, які оптимізують передачу відео. Окрім цього, ми також повинні досліджувати нові технології, які можуть допомогти нам безперешкодно передавати величезні обсяги відеоданих. Одним із таких підходів було б покладатися на

					ДТЕУ 121 07-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

CDN (Мережа доставки вмісту) або ми могли б проекспериментувати з поступовим завантаженням відео.

Платформи обміну відеовмістом мають дилему конфіденційності та безпеки. Захист конфіденційності користувачів, ідентифікація та фільтрація небажаного або шкідливого вмісту – ось питання, які виникають через велику кількість контенту, який завантажується та публікується.

Збереження та тривалий доступ до відеовмісту стало проблематичним через зростання відеоданих і різноманітність доступних платформ для розміщення. Для боротьби з цією проблемою необхідно розробити ефективні стратегії, зосередивши дослідження на методах стиснення даних, архівації та організації великих медіа-архівів. Це забезпечує збереження відеоконтенту протягом тривалого часу та забезпечує зручний доступ до нього.

Коли йдеться про використання відеовмісту, виникають етичні проблеми щодо конфіденційності, дотримання авторських прав і моральних норм. Оскільки все більше відеовмісту публікується в Інтернеті, стає надзвичайно важливим захистити особисту інформацію та конфіденційність користувачів. Тому дослідження зосереджені на створенні ефективних методів відсіву небажаного вмісту, виявленні порушень авторських прав і встановленні етичних принципів використання відеоконтенту.

Усі ці аспекти досліджень показують, що проблема обміну відеоконтентом в реальному часі є комплексною і має багато вимірів. Хоча існують різні платформи та технології, які намагаються вирішити ці проблеми, багато аспектів все ще потребують додаткового дослідження та вдосконалення.

						ДТЕУ 121 07-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			9

### 1.3. Стан дослідження проблеми

Еволюція досліджень обміну відео в реальному часі постійно активна і триває. Останніми роками це було гарячою темою, яка привертала увагу дослідників у різних галузях. Були досліджені соціальні, безпекові та технічні аспекти цієї теми. Зараз у центрі уваги технології передачі відео в реальному часі – розробка швидких і ефективних алгоритмів стиснення відео. Метою цих алгоритмів є надання високоякісних зображень навіть при роботі в межах обмеженої пропускної здатності мережі. Для досягнення цієї мети створюються нові кодеки, такі як AV1, VP9 і H.265. Ці кодеки стискають відео з більшою ефективністю та мінімальною втратою якості. Залучення є ключовим напрямком досліджень, коли йдеться про платформи обміну відео та їхній вплив на соціальну динаміку. Оптимізація мережевого трафіку та зменшення затримки передавання також є важливими факторами, які вивчаються, оскільки вони можуть значно покращити передачу відео в реальному часі.

На задоволеність користувачів, залученість і взаємодію з ними впливають взаємодія, механізми спільноти, спільний перегляд і коментування, і ці компоненти ретельно досліджуються в нашому аналізі. Крім того, наше дослідження заглиблюється в алгоритми рекомендацій і персоналізований вміст, які допомагають надавати користувачам відповідний і захоплюючий відеовміст.

У ході досліджень розробляються алгоритми машинного навчання та штучного інтелекту, щоб розпізнавати та фільтрувати шкідливий або небажаний вміст. Також досліджуються методи захисту конфіденційності користувачів, а також механізми шифрування та автентифікації для забезпечення безпеки даних під час передачі відео. Дослідження спрямоване

						ДТЕУ 121 07-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			10



на виявлення та запобігання порушенням авторських прав, а також на виявлення та фільтрацію насильницького, образливого та спаму.

Однак, варто відзначити, що деякі аспекти проблеми обміну відеоконтентом в реальному часі залишаються відкритими та потребують подальшого дослідження. Один з таких аспектів - це масштабованість систем обміну відеоконтентом. З ростом популярності цих платформ та збільшенням кількості користувачів, виникають виклики, пов'язані зі збереженням якості передачі відео та забезпеченням стабільної роботи систем при великому навантаженні. Дослідження в цій сфері спрямовані на розробку масштабованих архітектур та розподілених систем, які можуть ефективно працювати з великою кількістю користувачів і забезпечувати стабільне з'єднання та передачу відео.

Крім того, іншим важливим напрямком досліджень є покращення взаємодії та соціальної активності на платформах обміну відеоконтентом. Дослідження спрямовані на розуміння способів стимулювання активності користувачів, залучення аудиторії та створення позитивного та взаємовигідного середовища для спілкування.

В цілому, дослідження проблеми обміну відеоконтентом в реальному часі є широким і різноманітним полем досліджень. Інноваційні технології, соціальні взаємодії та безпекові аспекти продовжують залучати увагу дослідників та практиків у багатьох галузях.

#### 1.4. Технічне завдання

Технічне завдання (ТЗ) - це документ, який містить опис вимог і вимог до програмного продукту, який планується розробити. У дипломній роботі технічне завдання відіграє дуже важливу роль. Воно виконує кілька функцій:

					ДТЕУ 121 07-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

- Фіксує обсяг роботи: Технічне завдання визначає, які функції повинен виконувати програмний продукт, які компоненти мають бути розроблені і які вимоги повинні бути задоволені.
- Визначає вимоги користувачів: Технічне завдання містить вимоги, що висуваються користувачами до програмного продукту. Це можуть бути функціональні вимоги, які описують, які задачі має вирішувати програмний продукт, або нефункціональні вимоги, такі як швидкодія, безпека, надійність тощо.
- Визначає архітектурні рішення: Технічне завдання може містити опис архітектури програмного продукту, включаючи структуру бази даних, інтерфейси користувача, компоненти системи та залежності між ними.
- Встановлює критерії успіху: Технічне завдання визначає критерії, за якими буде оцінюватись успішність розробки програмного продукту. Це можуть бути вимоги до продуктивності, якості коду, відповідність стандартам тощо

Для дипломного проєкту було створено технічне завдання, яке визначає вимоги до програмного продукту.

Загальні відомості:

1.1. Найменування системи PoruchTv:

1.1.1. Повне найменування системи: Система обміну відеоконтентом «PoruchTv».

1.1.2. Скорочене найменування системи: COB.

1.2. Планові терміни початку та закінчення робіт: 01.09.2023 - 01.06.2023.

1.3. Порядок оформлення і пред'явлення результатів робіт: Програмний продукт та документація

									Аркуш
									12
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 07-01.БР</i>				

1.4. Головний бенефіціар та потенційні користувачі системи: Компанії, що займаються створенням та розповсюдженням відеоконтенту, інтернет-магазини, соціальні мережі.

Мета та призначення створення системи:

2.1. Призначення системи: Забезпечення ефективного та безпечного обміну відеоконтентом між користувачами.

2.2. Мета створення системи: Зниження витрат часу та коштів на обмін відеоконтентом, покращення якості відеоконтенту, забезпечення конфіденційності та безпеки користувачів.

Вимоги до системи:

3.1. Вимоги до системи в цілому:

Система повинна забезпечувати швидкий та надійний обмін відеоконтентом між користувачами.

Система повинна бути простою та зручною у використанні.

Система повинна бути масштабованою та здатною працювати з великою кількістю користувачів.

3.1.1. Вимоги до структури та функціонування системи, перелік підсистем: Система повинна забезпечувати можливість завантаження, зберігання та відтворення відеоконтенту, а також пошук відео за ключовими словами та фільтрацію відповідно до заданих критеріїв.

Система повинна містити сервер, який забезпечує збереження та передачу відеоконтенту.

Система повинна містити додатки для користувачів, які дозволяють завантажувати та переглядати відео.

Підсистеми: система авторизації, система збереження даних, система пошуку та фільтрації, система відтворення відео, система забезпечення безпеки.

					ДТЕУ 121 07-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

3.1.2. Вимоги до способів і засобів інформаційного обміну між компонентами системи:

Система повинна використовувати протоколи безпеки та шифрування для передачі даних між компонентами.

Система повинна мати можливість передачі відеоконтенту різного формату та якості.

### 1.5. Висновки до розділу 1.

У першому розділі «Аналіз предметної області застосування програмного продукту» було проведено детальний аналіз предметної області, в якій планується застосування програмного продукту. В ході дослідження було вивчено основні аспекти, вимоги та потреби користувачів, аналізовані існуючі рішення та проблеми, що виникають у даній області.

Аналіз предметної області дозволив нам отримати глибше розуміння сутності та контексту, в якому буде використовуватись наш програмний продукт. Ми оцінили потреби користувачів, їх вимоги та очікування щодо функціональності та якості продукту.

					ДТЕУ 121 07-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

## РОЗДІЛ 2

### ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

#### 2.1. Архітектура продукту

Метою проєкту було створення функціоналу, який дозволяє користувачам обмінюватись відео контентом в реальному часі. Для досягнення цієї мети була використана архітектура MVC (Model-View-Controller).

Model-View-Controller (MVC) — це архітектурний шаблон, який розділяє програму на три основні логічні компоненти: Model, View і Controller. Звідси і скорочення MVC. Кожен компонент архітектури створено для обробки певного аспекту розробки програми. MVC відокремлює бізнес-логіку та рівень презентації один від одного. Він традиційно використовувався для настільних графічних інтерфейсів користувача (GUI). Сьогодні архітектура MVC у веб-технологіях стала популярною для розробки веб-додатків, а також мобільних додатків[1].

Основними перевагами використання MVC є:

- Легка роздільність логіки додатку на три окремі компоненти: моделі (Models), представлення (Views) та контролери (Controllers). Це дозволяє розподіляти функціонал між різними командами розробників та уникнути конфліктів при змінах коду.
- Моделі в MVC є відповідальними за логіку та роботу з даними, а представлення - за відображення цих даних користувачам. Контролери виконують посередницьку роль між моделями та представленнями. Це дозволяє розподілити функціонал між різними частинами коду та дозволяє

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 07-01.БР</i>			
Зав. каф.		Криворучко О.В.		03.03.23	Програмний продукт обміну відеоконтентом в реальному часі	Стадія	Аркуш	Аркушів
Керівник		Хорольська К.В.		03.03.23		P2	15	43
Гарант		Котенко Н.О.		03.03.23		Факультет інформаційних технологій		
Розробив		Бортяний М.А.		03.03.23		4 курс, 7 група		

більш ефективно здійснювати розробку та тестування додатку.

- MVC використовує розділення на клієнтську та серверну частини, що дозволяє зосередитись на різних аспектах додатку окремо. Крім того, це дозволяє зменшити кількість запитів до сервера та сприяє поліпшенню швидкості роботи додатку.

На рисунку 2.1 зображена схема роботи архітектурного патерну MVC, що вказує на основні шари та методи комунікації між ними.

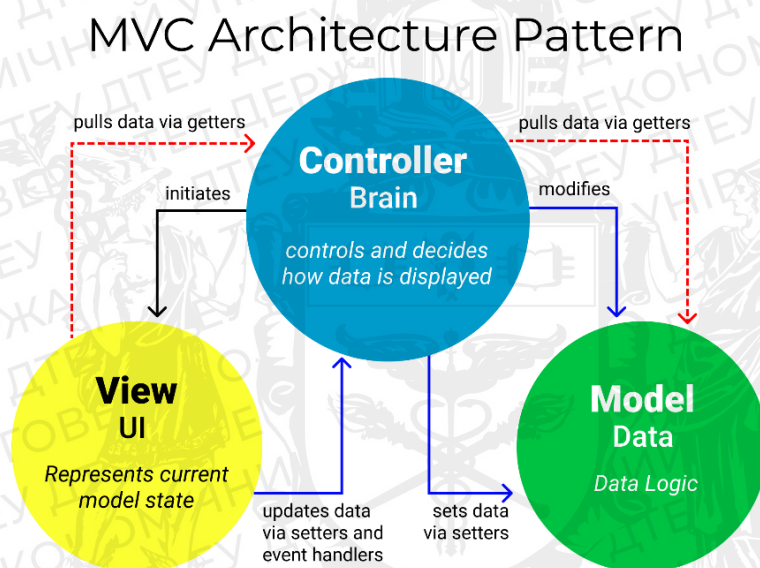


Рис. 2.1. Схема архітектурного патерну MVC.

Джерело: [2].

## 2.2. Інструменти розробки

Для розробки програмної частини продукту в якості середовища розробки обрано мову C# та Visual Studio.

**Visual Studio** — це потужне інтегроване середовище розробки (IDE), яке надає багатий набір інструментів, які допомагають розробникам писати, налагоджувати та тестувати свій код. Це робить розробку швидшою та ефективнішою, оскільки всі необхідні інструменти знаходяться в одному місці[8].

							Аркуш
							16
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-01.БР		

**C#** – проста, потужна, статично типізована, об’єктно орієнтована мова програмування від компанії Microsoft. C# входить до сімейства мов програмування C, синтаксис мови буде знайомим програмістам, що працювали з C, C++, Java та JavaScript[3]. Це також мова загального призначення, яку можна використовувати для широкого спектру завдань розробки, від настільних програм до веб-сайтів і мобільних програм.

Серверна частина сайту буде написана на ASP.Net Core. **ASP.NET Core** — це відкрите кросплатформне програмне забезпечення для розробки веб-додатків, розроблене Microsoft[9]. Це мінімалістична модульна структура, яка дозволяє розробляти продуктивні, масштабовані та безпечні веб-додатки на різних платформах, включаючи Windows, Linux і macOS.

ASP.NET Core має вбудовану підтримку архітектури MVC (Model-View-Controller), а також Web API і SignalR, що дозволяє одночасно розробляти веб-додатки та веб-сервіси. Він підтримує різноманітні способи розгортання та використання баз даних, включаючи Entity Framework Core.

ASP.NET Core також має високу продуктивність і вбудовану підтримку обробки запитів у реальному часі, наприклад WebSockets. Крім того, він має вбудовану підтримку Docker, що дозволяє легко розгорнути веб-додатки в контейнерах.

Visual Studio інтегрується з інфраструктурою Microsoft ASP.NET, що полегшує розробку та розгортання веб-сайтів за допомогою C#. ASP.NET надає повний набір інструментів і бібліотек, які допомагають розробникам створювати динамічні веб-сайти, керовані даними.

Для збереження даних було обрана база даних MS SQL. **Microsoft SQL (Special Query Language)** - система управління базами даних, побудована мовою запитів. Вона називається Transact-SQL, розроблена компанією Microsoft та Sybase у 1989 році[4]. MS SQL є однією з найбільш широко використовуваних систем керування базами даних у всьому світі, надійна та

						Аркуш
					ДТЕУ 121 07-01.БР	17
Зм.	Аркуш	№ докум	Підпис	Дата		

потужна платформа для керування даними. Використовуючи ASP.NET Core, веб-розробники можуть швидко розгорнути та створювати гнучкі та швидкі програми. Переваги інтеграції MS SQL із веб-сайтом ASP.NET Core включають швидкий доступ до даних, надання даних і підтримку транзакцій, ефективний пошук і вибір за допомогою індексів, а також можливість масштабування для збільшення навантаження. Завдяки величезній спільноті користувачів і багатьом ресурсам розробники можуть швидко вирішувати свої запитання та проблеми, використовуючи MS SQL.

Для обміну відеоконтентом у реальному часі ми вирішили використовувати SignalR, бібліотеку .NET, яка допомагає розробляти веб-додатки, які забезпечують двосторонній зв'язок між клієнтом і сервером. Використовуючи SignalR, ми забезпечуємо синхронізацію всіх команд, тим самим покращуючи досвід веб-перегляду користувача.

SignalR працює на основі технології WebSocket, яка забезпечує бідирекційний зв'язок між клієнтом та сервером з мінімальними затримками та швидкістю обміну даними. Таким чином, SignalR дозволить нам забезпечити швидкий та безперебійний обмін даними між клієнтами та сервером, що є ключовим для забезпечення високоякісного відтворення відеоконтенту в реальному часі.

### 2.3. Архітектура серверу

Використовуючи архітектуру клієнт-сервер, проект буде заснований на веб-архітектурі. Підтримкою на стороні сервера буде ASP.NET Core, який є технологічною крос-платформною для розробки веб-додатків і дозволить вам без проблем розміщувати сайти на серверах Linux.

Щоб зменшити навантаження на сервер і оптимізувати роботу сайту, ми запровадимо Nginx як проксі-сервер. Nginx - це спритний посередник, який забезпечує рівень шифрування даних і балансування навантаження, щоб забезпечити рівномірний розподіл запитів клієнтів.

						Аркуш
					ДТЕУ 121 07-01.БР	18
Зм.	Аркуш	№ докум	Підпис	Дата		



Нашою системою керування базою даних буде MSSql, система реляційної бази даних, яка є одночасно відкритою та безкоштовною. Ідеально підходить для нашого проекту, MSSql можна використовувати в будь-якій операційній системі, навіть у Linux.

Конструкцію проекту можна розбити на три частини: базу даних, серверну частину та клієнтську частину. Коди HTML, CSS і JavaScript будуть зібрані разом на стороні клієнта та доступні через проксі-сервер Nginx.

Загальна архітектура проекту буде модульною, що дозволить розширювати та модифікувати проєкт у майбутньому.

Щоб зрозуміти, як сайт і база даних будуть розміщені на сервері, як вони будуть взаємодіяти один з одним і яку роль в архітектурі системи буде відігравати проксі-сервер, створюється діаграма розгортання. Діаграма розгортання — це тип діаграми UML, який дозволяє описати архітектуру системи та зв'язки між її компонентами на рівні фізичного розгортання.

Ця діаграма корисна при проектуванні архітектури системи, оскільки вона дозволяє візуалізувати, як компоненти системи будуть розміщені на сервері та як вони взаємодіятимуть один з одним. Зокрема, діаграми розгортання допомагають визначити кількість і типи серверів, необхідних для розгортання системи, а також компоненти системи, які будуть розміщені на кожному сервері.

Рисунок 2.2 пояснює основні моменти архітектури системи. На діаграмі розгортання є три основні шари: SQL server, Web-server та Nginx. Кожен з цих шарів архітектури виконує важливу функцію та працює з іншими шарами. Web-server – це програмна частина продукту, що виконує основні функції на сайті та контактує з SQL-server в обміні даними. SQL-server – це база даних, яка зберігає данні про відеоконтент та данні користувачів. Без неї неможливо створити багатофункціональний програмний продукт, що забезпечить усі потреби користувачів.

						Аркуш
					ДТЕУ 121 07-01.БР	19
Зм.	Аркуш	№ докум	Підпис	Дата		

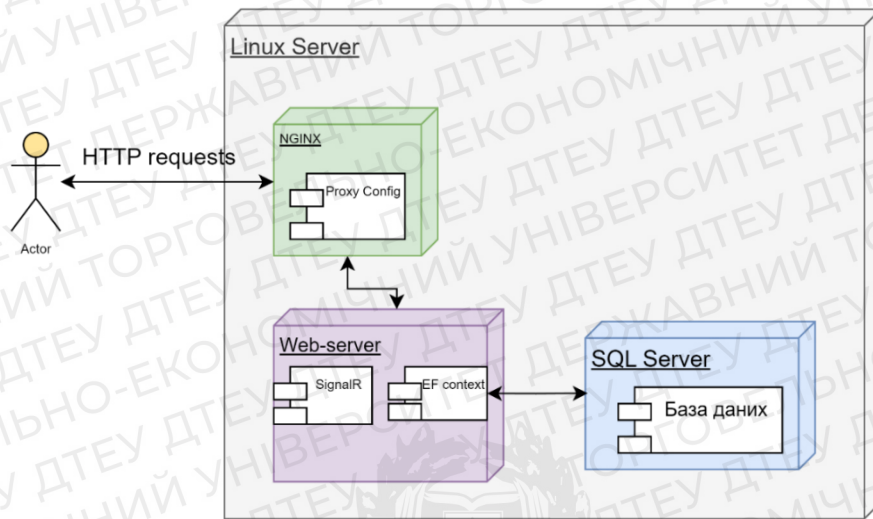


Рис. 2.2. Діаграма розгортання

Джерело: побудовано автором

## 2.4. Проектування програмного продукту

Створення діаграм є важливою частиною проектування програмного продукту і має кілька цілей:

- Візуалізація архітектури: Діаграми допомагають візуалізувати структуру та взаємозв'язки компонентів програмного продукту. Вони надають зрозумілий та зручний спосіб представлення архітектурних концепцій, що дозволяє команді розробників та зацікавленим сторонам легше розуміти взаємодію між компонентами та модулями системи.
- Виявлення проблем та помилок: Діаграми допомагають виявити потенційні проблеми або недоліки в архітектурі програмного продукту. Вони дозволяють перевірити сумісність компонентів, ідентифікувати можливі конфлікти або неясності щодо функціональності та взаємодії.
- Комунікація та співпраця: Діаграми є потужним засобом комунікації між розробниками, дизайнерами, тестувальниками та іншими учасниками проекту. Вони допомагають узгоджувати

						Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-01.БР	

розуміння між усіма сторонами, уникати непорозумінь та забезпечувати спільне бачення архітектурних рішень.

- Документування: Діаграми є важливим елементом документації проекту. Вони служать як посилання на розроблену архітектуру та структуру системи, допомагаючи зберегти знання про продукт та спростити майбутні розробки, тестування та підтримку.

Діаграма варіантів використання створена з метою візуалізації та опису можливих варіантів використання системи з точки зору користувачів. Вона допомагає зрозуміти, які функції та операції можуть бути виконані користувачами, як вони взаємодіють з системою та які результати можна отримати.

**Діаграма варіантів використання** (use case diagram, діаграма сценаріїв) - це вихідна концептуальна модель системи в процесі її проектування і розробки, на якій зображуються відношення між зовнішніми агентами і варіантами використання. Use case діаграма є описом загальних функцій модельованої системи без розгляду внутрішньої структури цієї системи.

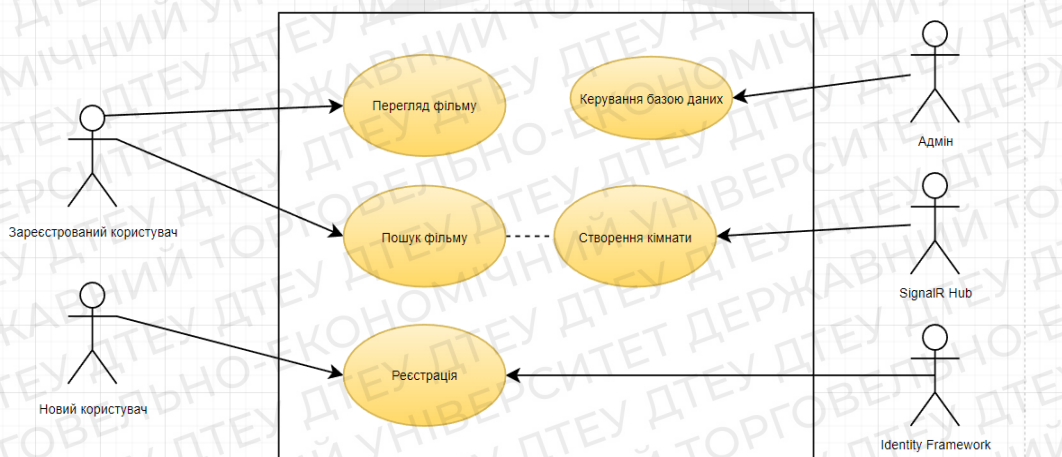


Рис. 2.3. Use case діаграма

Джерело: побудовано автором

На рисунку 2.3 показано діаграму варіантів використання, яка показує зовнішніх агентів, таких як нові користувачі, зареєстровані користувачі, адміністратори сайту, SignalR Hub і Identity Framework. SignalR Hub є лідером у доставці та синхронізації відеоконтенту. Структура ідентифікації керує авторизацією та автентифікацією. Важливу роль у підтримці веб-сайту відіграють адміністратори.

Діаграми станів є важливим інструментом проектування системи, який дозволяє аналізувати різні стани системи та зв'язки між ними. **Діаграма Statechart (діаграма станів)** – діаграма, яка призначена для опису станів об'єкта й умов переходу між ними. Опис станів дозволяє точно описати модель поведінки об'єкта при одержанні різних повідомлень і взаємодії з іншими об'єктами. У цьому випадку була створена діаграма стану для аналізу компонентів і технологій, необхідних для створення механізму обміну відеоконтентом у реальному часі.

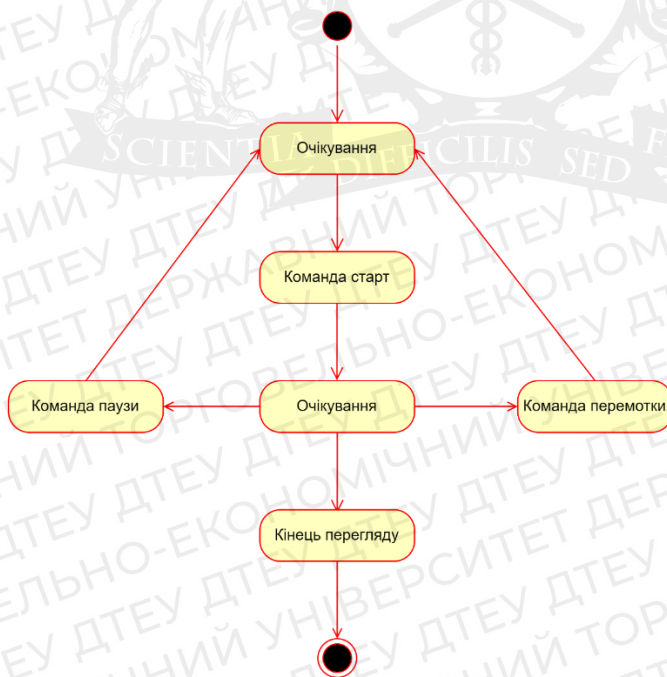


Рис. 2.4. Діаграма станів

*Джерело: побудовано автором*

						Аркуш
					ДТЕУ 121 07-01.БР	22
Зм.	Аркуш	№ докум	Підпис	Дата		

На рисунку 2.4 зображено роботу синхронізації відеоконтенту. Після запуску відеоплеєру програма очікує на команди від користувача, першою основною командою є команда старт, після якої починається відтворення відеоконтенту. В цей час програма очікує на інші можливі команди або на завершення перегляду.

Була створена блок-схема з метою візуалізації алгоритму та послідовності дій, які необхідно виконати в проекті. Вона використовується для графічного представлення кроків, умов, розгалужень та інших складових частин алгоритму, що допомагає легше розуміти та аналізувати його.

Створення блок-схеми має декілька цілей. По-перше, вона допомагає розробнику чітко спланувати послідовність виконання завдань та визначити логіку проекту. Вона дозволяє виділити основні кроки, врахувати умови, цикли та інші елементи, необхідні для виконання процесу.

По-друге, блок-схема сприяє комунікації між розробниками та іншими учасниками проекту. Вона створює єдину мову, яку можуть розуміти всі учасники, незалежно від їхнього технічного рівня. Це полегшує обговорення, аналіз та спільне розуміння роботи, що допомагає уникнути недорозуміння та помилок.

По-третє, блок-схема допомагає при відладці програми. Вона може бути використана для виявлення та виправлення помилок шляхом систематичного аналізу послідовності кроків. Розробник може відслідковувати шлях виконання програми, встановлювати умови та перевіряти правильність кожного кроку.

На рисунку 2.5 зображено блок-схему роботи програмного продукту обміну відеоконтентом, де вказано послідовність роботи програми та можливі варіації в залежності від того чи авторизований користувач.

						Аркуш
					ДТЕУ 121 07-01.БР	23
Зм.	Аркуш	№ докум	Підпис	Дата		

Початок роботи програми – це вибір контенту, після чого система автоматично повинна створити кімнату для перегляду. Для обміну контентом в реальному часі потрібно поділитися посиланням в кімнату для перегляду. Для цього є два можливих варіанти, які залежать від стану користувача. Перший варіант – це авторизований користувач запрошує друзів через систему дружніх відносин на сайті, яка автоматично відправить запрошення в системі. Інший варіант – це власноруч відправити посилання через соціальні мережі.

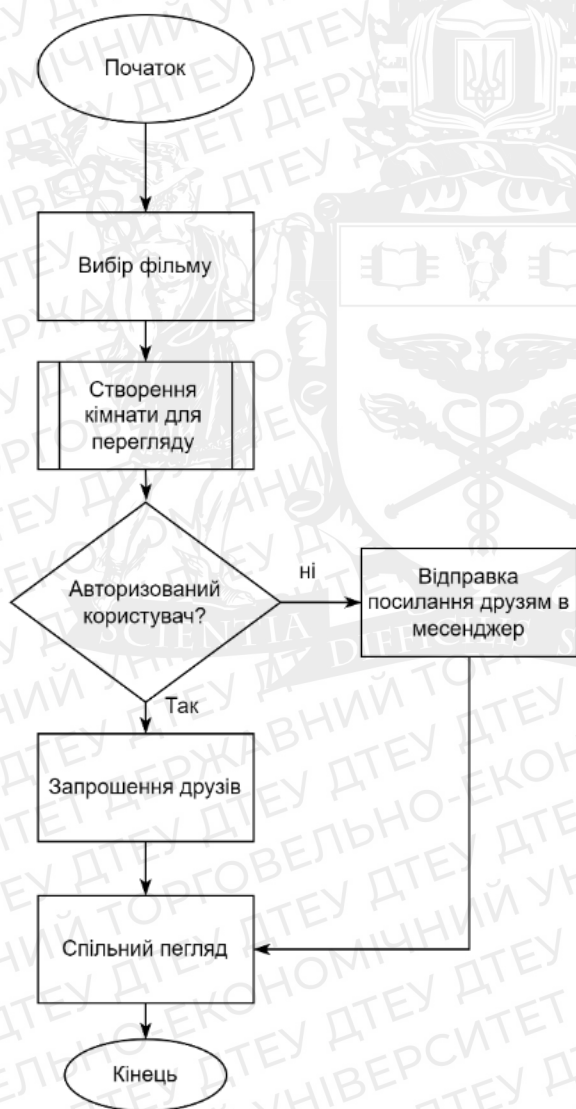


Рис. 2.5. Блок схема  
*Джерело: побудовано автором*

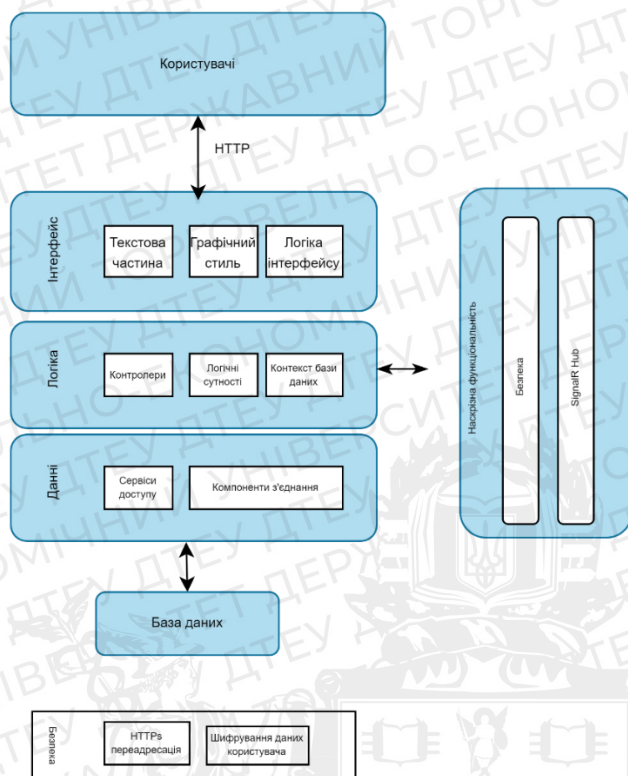


Рис. 2.6. Еталона модель програмного продукту

*Джерело: побудовано автором*

Еталонна модель сайту представлена на рисунку 2.6. Ця модель демонструє характеристики архітектури веб-сервісів. Користувач взаємодіє з візуальною частиною веб-сайту – інтерфейсом і пов’язаною з ним логікою. Цей сайт має основні принципи забезпечення безпеки користувачів та їхніх даних. Браузери перевіряють наявність сертифікатів SSL під час входу на будь-який сайт, тому використання протоколу HTTPS є обов’язковою частиною безпеки сайту. Щоб надійно зберігати паролі користувачів, вони повинні бути зашифровані.

Для спрощення розуміння структури системи було вирішено створити діаграму класів. Діаграма класів – це тип діаграм, які частіше за все використовуються при моделюванні об’єктно-орієнтованих систем. Елементи діаграм класів пов’язані різними структурними зв’язками. Такі діаграми

використовують для проектування словника системи чи кооперацій та систем[5].

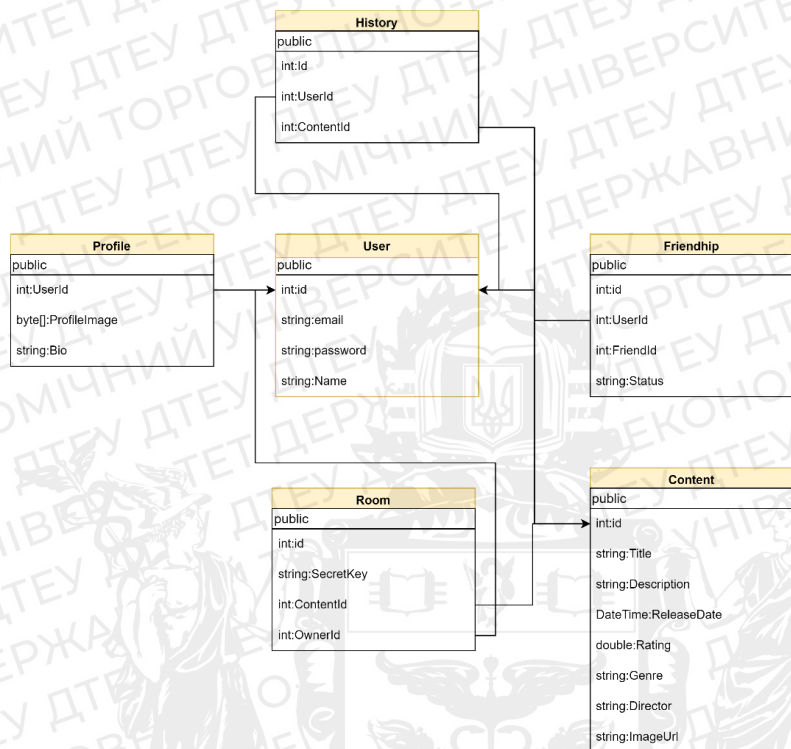


Рис. 2.7. Діаграма класів

*Джерело: побудовано автором*

На рисунку 2.7 зображено діаграму класів котра відображає основні класи сайту. На діаграмі зображені зв'язки між класами та їх внутрішні властивості с типами даних та модифікаторами доступу. Завдяки чому можна з легкістю зрозуміти основну концепцію внутрішньої роботи програми.

#### 2.4. Проектування бази даних

Вже існуюча діаграма класів спонукає до розробки проекту бази даних, а саме логічної та фізичної моделі БД. У результаті проектування має бути визначена структура бази, тобто склад таблиць, їхня структура та логічні зв'язки. Структура реляційної таблиці визначається складом стовпців, їхньою



послідовністю, типом даних кожного стовпця та їхнім розміром, а також ключем таблиці[6].

На рисунку 2.8 зображені таблиці в логічній моделі бази даних. На логічній моделі можна побачити два основних блока табличних відносин: відеоконтент, користувачі. Таблиці відеоконтенту зберігають інформацію про відеоконтент, його опис та знаходження. Таблиці користувача зберігають приватну інформацію: логін, пароль, соціальні мережі.

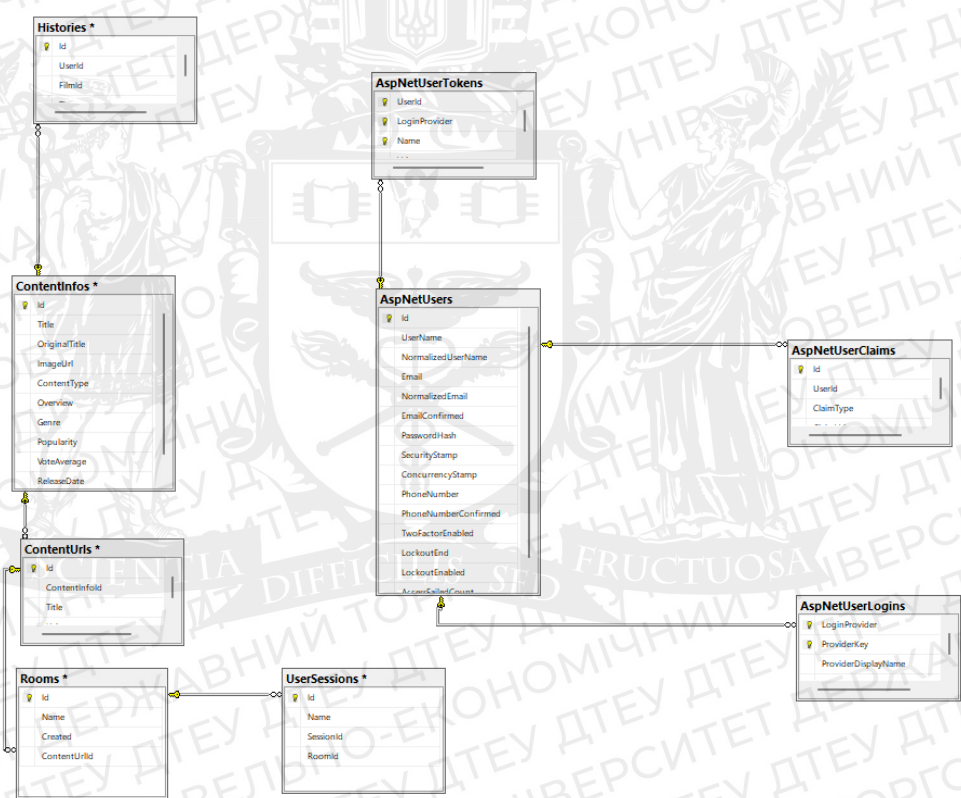


Рис. 2.8. Логічна модель бази даних

Джерело: побудовано автором

Фізична модель бази даних містить усі деталі, необхідні конкретній СУБД для створення бази даних (імена таблиць і стовпців, типи даних, визначення первинних і зовнішніх ключів тощо).

Фізична модель будується на логічній моделі з урахуванням обмежень, що накладаються можливостями обраної СУБД (у нашому випадку MS SQL Server 2019): імена, назви сутностей і поля (таблиці та стовпці) повинні бути написані латинським алфавітом Write, без пробілів, починається з літери; тип даних

Для кожного поля потрібно визначити, який тип даних воно містить у формі

Таблиця 2.1

**Таблиця Users**

Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
<i>UserId</i>	Primary key	Так	int	Не допустимі
<i>UserName</i>			nvarchar(256)	Не допустимі
<i>Email</i>			nvarchar(256)	Не допустимі
<i>EmailConfirmed</i>			bit	Не допустимі
<i>PasswordHash</i>			nvarchar(MAX)	
<i>SecurityStamp</i>			nvarchar(MAX)	
<i>ConcurrencyStamp</i>			nvarchar(MAX)	

Джерело: побудовано автором

Таблиця 2.2

**Таблиця UserSessions**

Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
<i>UserId</i>	Foreign Key		int	Не допустимі
<i>SessionKey</i>	Primary key		nvarchar(MAX)	Не допустимі
<i>RoomId</i>	Foreign Key		int	Не допустимі

Джерело: побудовано автором

Таблиця 2.3

Таблиця UserLogins

Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
UserLoginId	Primary key	Так	int	Не допустимі
UserId	Foreign Key		int	Не допустимі
LoginProvider			nvarchar(MAX)	Не допустимі
ProviderKey			nvarchar(MAX)	Не допустимі

Джерело: побудовано автором

Таблиця 2.4

Таблиця ContentInfos

Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
ContentInfoId	Primary key	Так	int	Не допустимі
Title			nvarchar(MAX)	Не допустимі
OriginalTitle			nvarchar(MAX)	Не допустимі
ImageUrl			nvarchar(MAX)	
ContentType			nvarchar(MAX)	
Overview			nvarchar(MAX)	
GenreId	Foreign Key		nvarchar(MAX)	
ReleaseDate			nvarchar(MAX)	

Джерело: побудовано автором

Таблиця 2.5

Таблиця ContentUrls

Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
ContentUrlId	Primary key	Так	int	Не допустимі
ContentInfoId	Foreign Key		int	Не допустимі
Name			nvarchar(MAX)	Не допустимі
Url			nvarchar(MAX)	Не допустимі

Джерело: побудовано автором

					Аркуш
					29
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-01.БР

Таблиця 2.6

<b>Таблиця Rooms</b>				
Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
RoomId	Primary key	Так	int	Не допустимі
ContentUrlId	Foreign Key		int	Не допустимі
Key			nvarchar(MAX)	Не допустимі

Джерело: побудовано автором

Таблиця 2.7

<b>Таблиця Histories</b>				
Назва поля	Ключ	Індексоване поле	Тип даних	Null-значення
HistoryId	Primary key	Так	int	Не допустимі
UserId	Foreign Key		int	Не допустимі
ContentInfolId	Foreign Key		int	Не допустимі
Time			datetime2(7)	

Джерело: побудовано автором

Таблиці 2.1 – 2.6 зображують фізичну модель бази даних. Кожна таблиця вказує на конкретні типи даних, які використовують та зв'язки між ними. Для будівництва відносин між таблицями використані Foreign Key, що посилається на Primary key, таким чином таблиці мають залежності одна від одної та утворюють загальну модел БД.

## 2.5. Висновки до розділу 2

Отже, підсумовуючи вище викладене, можна визначити, що для розробки програмного продукту найкраще використовувати мову С# в поєднанні з ASP.Net Core та MSSql. Для якісної розробки програми були створенні діаграми та схеми, що відображають потреби в розробці. Створені логічна та фізична моделі бази даних, дозволили побудувати швидку базу даних без помилок.

						Аркуш
						30
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-01.БР	

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

#### 3.1. Розробка механізму синхронізації

Основною частиною розробки програмних продуктів для обміну відеоконтентом є створення механізмів синхронізації. Щоб об'єднати всіх користувачів під одним спільним переглядом, було розроблено хаб із використанням SignalR.

Для початку були створені методи: Play, Pause, SetTime, Send.

```
public async Task Play(string roomId, double seek)
{
    await Clients.OthersInGroup(roomId).SendAsync("Play", roomId, seek);
}

public async Task Pause(string roomId, double seek)
{
    await Clients.OthersInGroup(roomId).SendAsync("Pause", roomId, seek);
}

public async Task SetTime(string roomId, double time)
{
    await Clients.OthersInGroup(roomId).SendAsync("SetTime", time);
}

public async Task Send(string message, string roomId, string username)
{
    await Clients.Group(roomId).SendAsync("Send", message, username);
}
```

Рис.3.1. Методи роботи синхронізації

*Джерело: побудовано автором*

Усі методи концентратора зображені на рисунку 3.1, де зображено, що вони покладаються на вхідний параметр roomId для розрізнення клієнтів за кімнатами. Цей ідентифікатор, створений автоматично і складається з 10 символів, служить ключем для доступу до кожної окремої кімнати. Функції SetTime, Play і Pause дотримуються спільного принципу, розподіляючи завдання трансляції кодів відео та часу відтворення для всіх глядачів. Водночас метод «Send» дозволяє користувачам ділитися індивідуальними повідомленнями в чаті

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 07-01.БР			
Зав. каф.		Криворучко О.В.		14.04.23	Програмний продукт обміну відеоконтентом в реальному часі	Стадія	Аркуш	Аркушів
Керівник		Хорольська К..		14.04.23		РЗ	31	43
Гарант		Рзаєва С.Л.		14.04.23	Реалізація програмного продукту	Факультет інформаційних технологій		
Розробив		Бортяний М.А.		14.04.23		4 курс, 7 група		

Додавання користувача в чат і відповідну кімнату здійснюється методом Enter. Що стосується параметрів, то вони складаються з часу відтворення відео (seek), посилання на відео (link), ідентифікатора кімнати (roomId) і імені користувача (username).

Клієнт отримує повідомлення про введення свого імені користувача, якщо метод визначає, що воно відсутнє. Цей початковий крок підтверджує введення імені користувача для перегляду.

```
public async Task Enter(string roomId, string username, double seek, string link)
{
    if (String.IsNullOrEmpty(username))
    {
        await Clients.Caller.SendAsync("Notify", "Для перегляду введіть ім'я користувача");
    }
    else
    {
        if (Context.User.Identity.IsAuthenticated)
        {
            var user = db.Users.FirstOrDefaultAsync(x => x.UserName == Context.User.Identity.Name);
        }
        var room = await db.Rooms.FirstOrDefaultAsync(x => x.Name == roomId);
        if (room == null)
        {
            await db.Rooms.AddAsync(new Room() { Created = DateTime.Now, Name = roomId, FilmUrl = link });
            await db.SaveChangesAsync();
        }
        var connectionId = Context.ConnectionId;
        await db.UserSessions.AddAsync(new UserSession() { Name = username, RoomId = roomId, SessionId = connectionId, Url = link });
        await db.SaveChangesAsync();
        await Groups.AddToGroupAsync(Context.ConnectionId, roomId);
        Users.TryAdd(username, seek);
        var users = await db.UserSessions.Where(x => x.RoomId == roomId).Select(x => x.Name).ToListAsync();
        await Clients.Group(roomId).SendAsync("Notify", users);
    }
}
```

Рис.3.2. Метод Enter

*Джерело: побудовано автором*

Метод, що зображений на рисунку 3.2 перевіряє автентифікацію користувача після того, як переконається, що ім'я користувача не пусте. Щоб отримати дані користувача, виконується перевірка завдяки параметру «IsAuthenticated», що повертає значення true. Після перевірки дані користувача шукають в базі даних методом «FirstOrDefault» в який передається унікальне ім'я. За допомогою ідентифікатора кімнати в базі даних шукаються будь-які доступні кімнати, і, якщо їх не знайдено, у таблиці кімнат створюється новий запис, який містить ідентифікатор кімнати, час створення та відеозв'язок.

						Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-01.БР	

ConcurrentDictionary зберігає активних користувачів під час відтворення відео. Це допомагає створити новий запис користувача в таблиці UserSessions, перераховуючи ім'я особи, ідентифікатор кімнати та ідентифікатор сеансу підключення до сокета SignalR. Потім цей запис зберігається в базі даних.

```
public override async Task OnDisconnectedAsync(Exception? exception)
{
    var user = await db.UserSessions.FirstOrDefaultAsync(x => x.SessionId == Context.ConnectionId);
    if (user != null) db.UserSessions.Remove(user);
    await db.SaveChangesAsync();
    if (user != null)
    {
        await Groups.RemoveFromGroupAsync(user.RoomId, "SignalR Users");
        var users = await db.UserSessions.Where(x => x.RoomId == user.RoomId).Select(x => x.Name).ToListAsync();
        await Clients.Group(user.RoomId).SendAsync("Notify", users);
    }

    await base.OnDisconnectedAsync(exception);
}
```

Рис.3.3. Метод від'єднання користувача

*Джерело: побудовано автором*

На рисунку 3.3 зображено метод, який оброблює вихід користувача з кімнати перегляду. Коли користувач від'єднується від концентратора, метод OnDisconnectedAsync() із SignalR Hub викликається автоматично. Де він видаляється зі списку глядачів за певною кімнатою і всі глядачі отримують сповіщення про це, та видаляють його ім'я зі списку глядачів.

Розроблений хаб виконує функції комутатора між різними користувачами та надає можливість синхронно обмінюватись відеоконтентом. Але для повноцінної роботи цього замало і було створено скрипт, який оброблює команди від хабу. Скрипт написаний на мові JavaScript та використовує заготовлену бібліотеку від SignalR. Цей скрипт розміщений на сторінці з відеоплеєром та автоматично виконує запити до хабу.

Для налаштування підключення до хабу, треба отримати параметри кімнати та відео, з адреси сторінки, для цього був створений метод «setSettings()», що зображений на рисунку 3.4. За допомогою інформації зі сторінки заповнює заготовленні змінні: url, name, roomId, link, seekMain, isPlay. Після чого цей метод автоматично визивається при заході на сторінку.

						Аркуш
						33
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-01.БР	

```

9   async function setSettings() {
10      url = document.URL;
11      name = "";
12      roomId = url.split('?key=')[1].split("&")[0];
13      link = url.split('url=')[1].replaceAll("%2F", "/");
14      link = link.replaceAll("%3F", "?");
15      link = link.replaceAll("%3D", "=");
16      isPlay = false;
17      seekMain = 0;
18      document.getElementById("if").src = link;
19   }
20
21   setSettings();
22   var connection = new signalR.HubConnectionBuilder()
23     .withUrl('/chat')
24     .build();
25
26   async function start() {
27     try {
28       await connection.start();
29       name = document.getElementById("userName").innerText;
30       if (name === "") {
31         do {
32           input = prompt("Введіть ім'я");
33           } while (input == null || input == "");
34           name = input;
35         }
36       Enter();
37       console.log("SignalR Connected.");
38     } catch (err) {
39       console.log(err);
40       setTimeout(start, 5000);
41     }
42   }

```

Рис.3.4. Методи налаштування хабу

*Джерело: побудовано автором*

Після налаштування даних про кімнату та користувача, створюється з'єднання з SignalR хабом. Для цього створюється змінна connection, якій присвоюється новий екземпляр об'єкту HubConnectionBuilder з параметром URL /chat.

Після створення змінної connection потрібно відкрити з'єднання з веб-сокетом, для цього створений метод start. Він перевіряє чи зареєстрований користувач та в залежності від цього запитує ім'я на сторінці, або використовує ім'я акаунту.

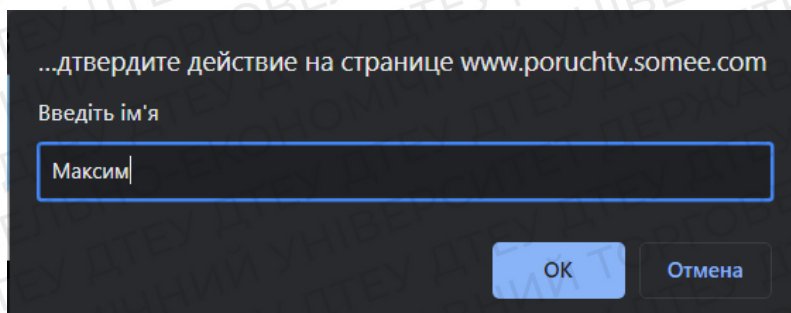


Рис.3.5. Запит ім'я користувача

*Джерело: побудовано автором*





```

connection.on('Play', function (command, seek) {
    seekMain = seek;
    document.getElementById("if").contentWindow.postMessage({ "api": "seek", "set": seek }, "*");
    isPlay = true;
    document.getElementById("if").contentWindow.postMessage({ "api": "play" }, "*");
});
connection.on('SetTime', function (command, time) {
    document.getElementById("if").contentWindow.postMessage({ "api": "seek", "set": time }, "*");
});
connection.on('Pause', function (command, time) {
    seekMain = time;
    document.getElementById("if").contentWindow.postMessage({ "api": "seek", "set": time }, "*");
    document.getElementById("if").contentWindow.postMessage({ "api": "pause" }, "*");
});
connection.on('Send', function (message, username) {
    let userNameElem = document.createElement("b");
    userNameElem.appendChild(document.createTextNode(username + ': '));
    let elem = document.createElement("p");
    elem.appendChild(userNameElem);
    elem.appendChild(document.createTextNode(message));

    var firstElem = document.getElementById("chat").firstChild;
    document.getElementById("chat").insertBefore(elem, firstElem);
});

```

Рис.3.7. Методи отримання команд

*Джерело: побудовано автором*

На рисунку 3.7 зображені методи, які привязані до обробників подій. Такі методи як Play, SetTime, Pause, отримують данні від хабу та починають їх оброблювати для коректної синхронізації. Кожен з цих методів отримує за айді параметром HTML, плеєр що знаходиться в Iframe та відсилає певну команду, що відповідає методу, та час який прийшов від хабу.

Обробник подій з методом Send, виконує роботу по додаванню нового повідомлення в чат з користувачами. Для цього створюється новий HTML тег «b» в якого поміщається нік користувача від якого повідомлення та саме повідомлення. Після чого цей тег додають до списку тегів чату.

Глядачі на сайті можуть з легкістю обмінюватись враженнями, або обговорювати інформацію вкладенну у відео в зручному та локалічному чаті поряд с відеоплеєром. Реалізація цього чату зображена на рисунку 3.8. Чат складається з 3 блоків. Перший блок - це список активних глядачів, які знаходяться в кімнаті. Другий блок – це список повідомлень, що вже отримав та отримав користувач чату. Третій блок – це текстовий блок, що приймає в

						Аркуш
						36
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-01.БР	

себе вхідне повідомлення та завдяки кнопці «Відправити» надає можливість відправити повідомлення в чат кімнати.



Рис.3.8. Чат для спілкування

*Джерело: побудовано автором*

### 3.2. Методи авторизації та аутентифікації

Кожен сайт або програма має в собі систему, що дозволяє організувати роботу з користувачами. Методи авторизації та аутентифікації дозволяють, як і обмежувати права деяких користувачів так і відкривають новий функціонал. Програмний продукт обміну відеоконтентом не є виключенням з правил, тому було створенно якісну та безпечну систему авторизації та аутентифікації.

					ДТЕУ 121 07-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		37

Головним питанням у вирішенні цієї задачі було створити механізм, що зберігатиме данні користувачів та не дозволить злоумисникам вкрасти їх. Одним з найпопулярніших способів запобігти викраденню паролів є шифрування їх або є хешування. При реєстрації, користувач надає пароль у вигляді звичайного рядка з текстів та символів, але зберігати пароль у такому вигляді - це небезпечно, при будь-якому витіку даних, злоумисники з легкістю викрадуть всі акаунти завдяки паролю. Тому пароль потрібно захистити за допомогою хешування. Хешування – це перетворення масиву вхідних даних довільної довжини в (вихідний) бітовий рядок встановленої довжини, що виконується певним алгоритмом[7].

Після виконання певних операцій над паролем ми отримуємо набір символів, які неможливо повернути до попереднього стану. Такий набір символів отримується кожного разу коли хтось вводить пароль, та для підтвердження особи, хеш порівнюється один з одним.

В свою чергу користувач бачить лише, два рядки для вводу даних. Такий спосіб для заходу на сайт є найпростішим з точки зору клієнта. Вхідна форма зображена на рисунку 3.9. Вона має два поля для вводу та чек-бокс для збереження авторизації в системі.

Рис.3.9. Вхідна форма

*Джерело: побудовано автором*

						Аркуш
					ДТЕУ 121 07-01.БР	38
Зм.	Аркуш	№ докум	Підпис	Дата		

### 3.3. Висновок до розділу 3

Отже, поєднання роботи серверної частини синхронізації на основі SignalR та клієнтської на мові JavaScript, дозволили створити програмний продукт, що надає змогу обмінюватись відеоконтентом в реальному часі, швидко та без помилок.

Завершення реалізації програмного продукту підтверджує, що наша розробка була успішною. Ми досягли мети створення функціонального та якісного програмного продукту, який відповідає потребам та очікуванням користувачів.



						ДТЕУ 121 07-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			39

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З проведених досліджень можна зробити наступні висновки:

Проводячи аналіз, ми занурилися в предметну область і заглибилися в уподобання користувачів у сфері відеовмісту. Ми ретельно вивчили різні платформи для потокового відео та вивчили їхні функції та інтерфейс. Ретельно досліджуючи технології синхронізації відеовмісту, ми також досліджували важливе питання безпеки та захисту інформації.

Враховуючи специфіку та потреби проекту, як засоби розробки були обрані ASP.NET, Visual Studio, SignalR та MSSql. ASP.NET був основним вибором для створення веб-додатків, тоді як Visual Studio забезпечувала ідеальне інтегроване середовище розробки. SignalR використовувався для виконання синхронізації відеовмісту, а для запису даних користувача та відеоматеріалів використовували Ms-SQL.

В процесі виконання проекту було розроблено: програмний продукт, що виконує функції обміну відеоконтентом в реальному часі, має широкий асортимент відеоконтенту та відповідає всім вимогам. Для зручності користування було розроблено методи аутентифікації та авторизації, можливість додавати друзів та переписуватись в чаті під час перегляду.

Результатом роботи є програмний продукт, який повністю відповідає вимогам і потребам користувачів у сфері відеоконтенту. Він забезпечує зручний доступ до відеоматеріалів, має ефективний механізм синхронізації та надійний механізм захисту інформації. Користувачі можуть насолоджуватися переглядом відео, спілкуватися з іншими користувачами та отримувати персоналізовані рекомендації.

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 07-01.БР			
Зав. каф.		Криворучко О.В.		28.04.23	Програмний продукт обміну відеоконтентом в реальному часі	Стадія	Аркуш	Аркушів
Керівник		Хорольська К.В.		28.04.23		ВП	40	43
Гарант		Рзаєва С.Л.		28.04.23		Факультет інформаційних технологій		
Розробив		Бортяний М.А.		28.04.23		4 курс, 7 група		
					Висновки та пропозиції			



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MVC Framework Tutorial for Beginners: What is, Architecture & Example [Електронний ресурс] // Guru99 – Режим доступу до ресурсу: <https://www.guru99.com/mvc-tutorial.html>.
2. The Model View Controller Pattern – MVC Architecture and Frameworks Explained [Електронний ресурс] // freeCodeCamp – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>.
3. Вступ в С# [Електронний ресурс] // programm.top – Режим доступу до ресурсу: <https://programm.top/uk/c-sharp/tutorial/introduction/>.
4. Microsoft SQL [Електронний ресурс] // itglobal.com – Режим доступу до ресурсу: <https://itglobal.com/ru-ru/company/glossary/microsoft-sql/>.
5. Марголін О. UML для бізнес-моделювання: для чого потрібні діаграми процесів [Електронний ресурс] / Олександр Марголін // evergreens – Режим доступу до ресурсу: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>.
6. Проектування баз даних [Електронний ресурс] // pidru4niki.com – Режим доступу до ресурсу: [https://pidru4niki.com/11570718/bankivska\\_sprava/proektuvannya\\_baz\\_danih](https://pidru4niki.com/11570718/bankivska_sprava/proektuvannya_baz_danih).
7. Коли хешування інформації може бути неефективним або небезпечним [Електронний ресурс] // Kharkiv National University of

<i>ДТЕУ 121 07-01.БР</i>				
Зм.	Аркуш	№ докум.	Підпис	Дата
Зав. каф.		Криворучко О.В.		23.12.22
Керівник		Хорольська К.В.		23.12.22
Гарант		Рзаєва С.Л.		23.12.22
Розробив		Бортяний М.А.		23.12.22
Програмний продукт обміну відеоконтентом в реальному часі				
Список використаних джерел				
Стадія		Аркуш	Аркушів	
СВД		42	43	
Факультет інформаційних технологій 4 курс, 7 група				



Radio Electronics. – 2812. – Режим доступу до ресурсу:

<https://www.ukrlogos.in.ua/10.11232-2663-4139.04.39.html>.

8. Visual Studio 2022 [Електронний ресурс] – Режим доступу до ресурсу:

<https://visualstudio.microsoft.com/>.

9. Overview of ASP.NET Core [Електронний ресурс] – Режим доступу до

ресурсу: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>.



					<i>ДТЕУ 121 07-01.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		23.12.22	Програмний продукт обміну відеоконтентом в реальному часі	Стадія	Аркуш	Аркушів
Керівник		Хорольська К.В.		23.12.22		СВД	43	43
Гарант		Рзаєва С.Л.		23.12.22		Факультет інформаційних технологій 4 курс, 7 група		
Розробив		Бортяний М.А.		23.12.22				
					<i>Список використаних джерел</i>			

## ДОДАТКИ

### ДОДАТОК А

#### Код класу HomeController

```
namespace poruchTv.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;
        private UserContext _db;
        public HomeController(UserContext context, ILogger<HomeController> logger)
        {
            _db = context;
            _logger = logger;
        }

        public async Task<IActionResult> Index()
        {
            var mainPosters = await _db.MainPosters.ToListAsync();
            ViewBag.MainPosters = mainPosters;

            var otherPosters = await _db.PopularContent.Include(x => x.ContentInfoApi).ToListAsync();
            ViewBag.OtherPosters = otherPosters;

            return View();
        }

        public IActionResult Privacy()
        {
            return View();
        }

        [HttpGet]
        public IActionResult GetFilm(Content movie)
        {
            return RedirectToAction("Index", "Movie", movie);
        }
    }
}
```

```
[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
```

```
public IActionResult Error()
```

```
{
```

```
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
```

```
}
```

```
}
```



**Код класу ContentController**

```

namespace poruchTv.Controllers
{
    public class ContentController : Controller
    {
        private UserContext db;
        public ContentController(UserContext context)
        {
            db = context;
        }

        public async Task<ActionResult> Index(string contentType = "movie", int pageNumber = 1)
        {
            if (!new[] { "movie", "tv-series" }.Contains(contentType))
                return NotFound();
            ViewData["contentType"] = contentType;
            ViewData["yearStart"] = 1910;
            ViewData["yearEnd"] = 2022;
            ViewData["ratingLow"] = 0;
            ViewData["ratingHigh"] = 10;
            var result = db.ContentInfosApi.OrderByDescending(x => x.Popularity).Where(x => x.ContentType == contentType);
            return View(await PaginatedList<ContentInfoApi>.CreateAsync(result, pageNumber, 15));
        }

        [HttpGet]
        public async Task<ActionResult> GetFilms(int yearStart, int yearEnd, double ratingLow, double ratingHigh, Genres input,
            int pageNumber = 1, string contentType = "movie")
        {
            if (!new[] { "movie", "tv-series" }.Contains(contentType))
                return NotFound();
            ViewData["yearStart"] = 1910;
            ViewData["yearEnd"] = 2022;
            ViewData["ratingLow"] = 0;
            ViewData["ratingHigh"] = 10;

            if (yearStart != 0) ViewData["yearStart"] = yearStart;
            if (yearEnd != 0) ViewData["yearEnd"] = yearEnd;
            if (ratingLow >= 0) ViewData["ratingLow"] = ratingLow;
            if (ratingHigh != 0) ViewData["ratingHigh"] = ratingHigh;
        }
    }
}

```

## Продовження дод. Б

```
var b = input.GenresInput.Where(x => x.Value == true).Select(x => x.Key).ToList();
ViewData["contentType"] = contentType;
var result = db.ContentInfosApi.Where(x=>x.ContentType == contentType).Where(x =>
    (int) (object) x.ReleaseDate.Substring(0, 4) >= yearStart &&
    (int) (object) x.ReleaseDate.Substring(0, 4) <= yearEnd && x.VoteAverage >= ratingLow &&
    x.VoteAverage <= ratingHigh);
// result.Join(b, x=>x.genre_ids.Split(','), p=>p, {});
foreach (var genre in b)
{
    result = result.Where(x => x.Genre.Contains(genre.ToString()));
}
//result.Where(x=>Array.ConvertAll(x.genre_ids.Split(','), b=>int.Parse(b)))
return View("Index", await PaginatedList<ContentInfoApi>.CreateAsync(result.OrderByDescending(x => x.Popularity),
pageNumber, 15));
}

private bool IsInRange(int check, int start, int end)
{
    if (check >= start && check <= end)
    {
        return true;
    }
    else
    {
        return false;
    }
}
}
```

**Код класу RoomController**

```

namespace poruchTv.Controllers
{
    public class RoomController : Controller
    {
        private UserContext db;
        private IOnlineService service;
        private IHubContext<NotificationHub> hub;
        public RoomController(UserContext context, IOnlineService srv, IHubContext<NotificationHub> notificationHub)
        {
            db = context;
            service = srv;
            hub = notificationHub;
        }
        public async Task<IActionResult> Index(string key, string url)
        {
            if (string.IsNullOrEmpty(url))
            {
                return StatusCode(400);
            }
            var id = key.Substring(key.IndexOf("tt"), key.Length - key.IndexOf("tt"));
            var filmId = db.ContentInfosApi.FirstOrDefault(x => x.imdb_id == id);
            ViewBag.FilmTitle = filmId?.Title;
            if (User?.Identity.IsAuthenticated == true)
            {
                //var id = await db.ContentUrls.FirstOrDefaultAsync(x => x.Url == url);
                //var filmId = db.ContentInfos.FirstOrDefault(x => x.Id == id.ContentInfoId);
                await db.Histories.AddAsync(new History() { Time = DateTime.Now, UserId = User.Identity.Name, FilmId = filmId!.Id });
                await db.SaveChangesAsync();
            }
            if (HttpContext.User.Identity.IsAuthenticated)
            {
                var userName = HttpContext.User.Identity.Name;
                var friends = db.Friendships.Where(x => x.IsConfirm == true &
                    x.FirstUserName == userName ||
                    x.SecondUserName == userName);
                var query1 = friends.Select(x => x.FirstUserName);
                var query2 = friends.Select(x => x.SecondUserName);
                var query3 = query1.Where(x => x != userName);
                query3 = query3.Concat(query2.Where(x => x != userName));
                var onlineUsers = service.GetOnlineUsers();

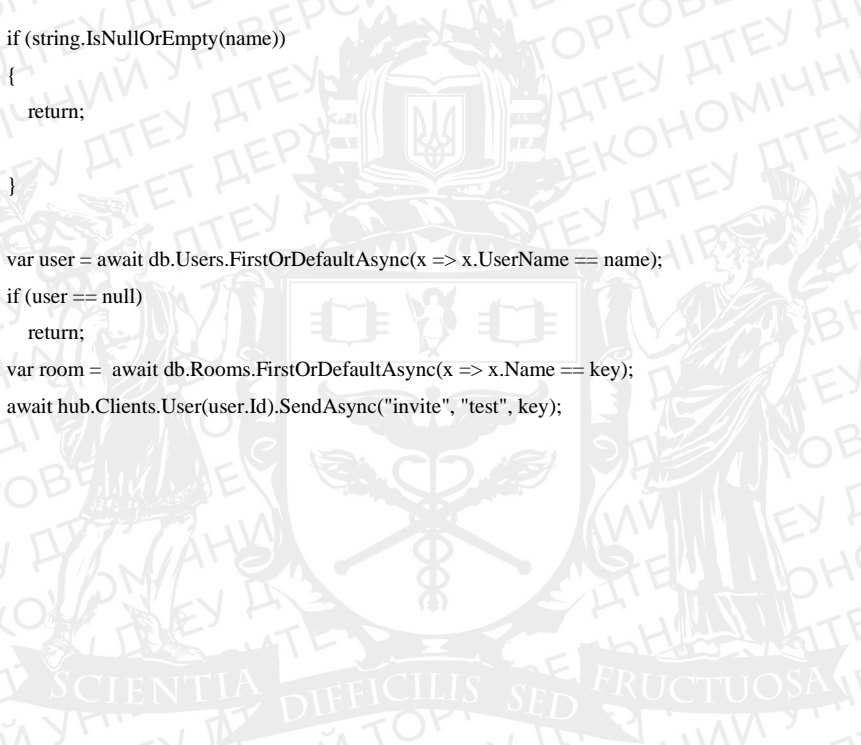
                return View(onlineUsers.Intersect(query3));
            }
            return View();
        }
        [HttpGet]
        public async Task<IActionResult> Enter(string key)
    }
}

```

```
{
    var room = await db.Rooms.FirstOrDefaultAsync(x => x.Name == key);
    if (room == null)
        return NotFound();
    return Redirect($"{Room?key={room.Name}&url={room.FilmUrl}");
}

[HttpPost]
public async Task Invite(string name, string key)
{
    if (string.IsNullOrEmpty(name))
    {
        return;
    }

    var user = await db.Users.FirstOrDefaultAsync(x => x.UserName == name);
    if (user == null)
        return;
    var room = await db.Rooms.FirstOrDefaultAsync(x => x.Name == key);
    await hub.Clients.User(user.Id).SendAsync("invite", "test", key);
}
}
```



**Код класу UserController**

```

namespace poruchTv.Controllers
{
    public class UserController : Controller
    {
        private UserContext db;
        public UserController(UserContext context)
        {
            db = context;
        }
        public async Task<ActionResult> Index(string name)
        {
            if (string.IsNullOrEmpty(name))
            {
                return NotFound();
            }
            var user = await db.Users.FirstOrDefaultAsync(x => x.UserName == name);
            if (user == null)
            {
                return NotFound();
            }
            return View(user);
        }
        [HttpPost]
        public async Task<ActionResult> SendFriendRequest(string name)
        {
            if (string.IsNullOrEmpty(name))
            {
                return Redirect("/Home");
            }
            if (!HttpContext.User.Identity.IsAuthenticated)
            {
                return Redirect("/Identity/Account/Login");
            }
            var userName = HttpContext.User.Identity.Name;
            if (db.Friendships.Any(x=>(x.FirstUserName == userName | x.FirstUserName == name) & (x.SecondUserName == userName
            | x.SecondUserName == name) ))
            {
                return Redirect("/Home");
            }
            await db.Friendships.AddAsync(new Friendship()
            {FirstUserName = userName, SecondUserName = name, IsConfirm = false});
            await db.SaveChangesAsync();
            return await Index(name);
        }
    }
}

```



**Код класу Room**

namespace poruchTv.Models.Rooms

```
{  
    public class Room  
    {  
        public int Id { get; set; }  
        public string Name { get; set; }  
        public DateTime Created { get; set; }  
        public string FilmUrl { get; set; }  
    }  
}
```

**Код класу Content**

```
namespace poruchTv.Models.Video
{
    public class Content
    {
        [Key]
        [field: NonSerialized]
        public int id { get; set; }
        public string ru_title { get; set; }

        public string? kinopoisk_id { get; set; }
        public string? imdb_id { get; set; }
        public string year { get; set; }
        public string orig_title { get; set; }

        [field: NonSerialized]
        public string imgUrl { get; set; }

        public string iframe_src { get; set; }
        public string content_type { get; set; }
        [field: NonSerialized]
        public string overview { get; set; }
        [field: NonSerialized]
        public double popularity { get; set; }
        [field: NonSerialized]
        public double vote_average { get; set; }
        [field: NonSerialized]
        public string genre_ids { get; set; }

        [NotMapped]
        public List<ContentUrls> translations { get; set; }
    }
}
```