
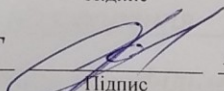
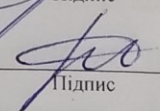


Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерних наук та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Веб-сервіс для обговорення кінофільмів

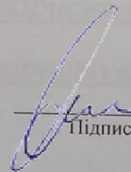
Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 4 курсу, група КН-17-1
Курс, група виконавця
 І.Я. Корольчук
Підпис Ініціали, прізвище
Керівник: к.ф.-м.н., доцент кафедри КНІТ
Науковий ступінь, посада
 В.Ц. Міхалевський
Підпис Ініціали, прізвище
Нормоконтроль: к.т.н., доцент кафедри КНІТ
Науковий ступінь, посада
 Р.О. Багрій
Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор

08 серпня 2021 р.


Підпис

О.В. Бармак
Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра комп'ютерних наук та інформаційних технологій

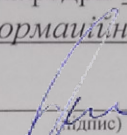
Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук та інформаційних технологій


д.т.н., професор О.В. Бармак

«08» лютого 2021 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Веб-сервіс для обговорення кінофільмів»


2. Завдання видано студенту Корольчуку Івану Яковичу
(прізвище, ім'я, по батькові)

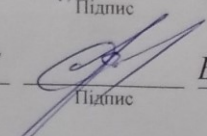
3. Керівник роботи доцент кафедри КНІТ Міхалевський Віталій Цезарійович
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «08» лютого 2021 р. № 11

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – розробка веб-сервісу для обговорення кінофільмів. Слід забезпечити виконання функцій: перегляд переліку кінофільмів на обговоренні, пошук кінофільму за назвою, роком прем'єри, жанром чи країною, перегляд докладного опису обраного кінофільму та вражень інших користувачів, долучення враження користувача про обраний кінофільм, оцінка обраного кінофільму, оцінка обраного враження іншого користувача, додавання нового кінофільму до веб-сервісу, перегляд переліку користувачів веб-сервісу та їх статистики.

Виконавець: студент 4 курсу, група КН-17-1  І.Я. Корольчук
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: к.ф.-м.н., доцент кафедри КНІТ  В.Ц. Міхалевський
Науковий ступінь, посада Підпис Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: «Веб-сервіс для обговорення кінофільмів»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-17-1
Корольчук Іван Якович

Керівник кваліфікаційної роботи бакалавра: к.ф.-м.н., доцент кафедри
КНІТ Міхалевський Віталій Цезарійович

Кваліфікаційна робота бакалавра містить:

| Пояснювальна записка | | | | Кількість додатків |
|----------------------|----------|---------|-------------------|--------------------|
| Сторінок | Рисунків | Таблиць | Джерел інформації | |
| 55 | 25 | 11 | 29 | 4 |

Метою кваліфікаційної роботи бакалавра є розробка веб-сервісу для обговорення кінофільмів. Для розробки веб-сервісу було використано платформу ASP.NET з використанням шаблону MVC, мову програмування C#, а також СКБД MS SQL Server.

Розроблена система призначена для любителів кінофільмів та адміністратора системи по збору відгуків та вражень про кінофільми. Автоматизовано функції незареєстрованого користувача, зареєстрованого користувача та адміністратора веб-сервісу. Напрямами практичного використання розробленої інформаційної системи визначено автоматизацію процесу перегляду та додавання вражень про той чи інший кінофільм. Також дана інформаційна система буде корисною для кінотеатрів та інших місць прокату кінофільмів.

Ключові слова: кінофільм, враження, обговорення, інформаційна система.

Виконавець: студент 4 курсу, група КН-17-1

Курс, група виконавця


Підпис

І.Я. Корольчук
Ініціали, прізвище

Зміст

| | |
|------------------------------------------------------------------------|----|
| Перелік скорочень | 3 |
| Вступ..... | 4 |
| Розділ 1 | |
| Характеристика предметної області та постановка задачі | 6 |
| 1.1 Аналіз предметної області | 6 |
| 1.2 Аналіз існуючого програмного забезпечення предметної області | 9 |
| 1.3 Аналіз сучасних засобів створення програмного забезпечення | 12 |
| 1.4 Постановка задачі та вимоги до розробки інформаційної системи..... | 17 |
| Розділ 2 | |
| Проектування інформаційної системи | 19 |
| 2.1 Функціональна структура та бізнес-процеси системи | 19 |
| 2.2 Інформаційна структура системи | 23 |
| 2.3 Вибір засобів розробки інформаційної системи | 29 |
| 2.3.1 Вибір мови програмування | 29 |
| 2.3.2 Вибір фреймворку | 31 |
| 2.3.3 Вибір редактора програмного коду..... | 32 |
| 2.3.4 Вибір СКБД | 34 |
| Розділ 3 | |
| Програмна реалізація інформаційної системи | 37 |
| 3.1 Структура та функціональне призначення складових системи | 37 |
| 3.2 Особливості реалізації складових системи | 39 |
| 3.3 Тестування інформаційної системи | 43 |
| 3.4 Інструкція користувача..... | 47 |
| 3.5 Вимоги до розгортання інформаційної системи..... | 50 |
| Висновки | 51 |
| Перелік посилань..... | 53 |
| Додатки | |

Перелік скорочень

| Скорочення, термін, позначення | Пояснення |
|-------------------------------------------|---------------------------------|
| БД | База даних |
| ІТ | Інформаційні технології |
| КРБ | Кваліфікаційна робота бакалавра |
| СКБД | Система керування базами даних |
| CLR | Common Language Runtime |
| MS | Microsoft |
| MVC | Model-View-Controller |

Вступ

Кінематограф був винайдений ще у XIX столітті і став вкрай популярний у XX столітті. У поняття кінематографа входять кіномистецтво – вид сучасного образотворчого мистецтва, твори якого створюються за допомогою рухомих зображень, і кіноіндустрія (кінопромисловість) – галузь економіки, яка виробляє кінофільми, спецефекти для кінофільмів, мультиплікацію і, що демонструє ці твори для глядачів. Твори кіномистецтва створюються за допомогою кілотехніки. Вивченням кінематографа займається наука кінознавство [1].

Вплив кінематографа на культуру і мистецтво безперечний, також величезний вплив кінематографа на політику і економіку. У багатьох країнах кіноіндустрія є значущою галуззю економіки. Виробництво кінофільмів зосереджено на кіностудіях. Фільми демонструються в кінотеатрах, по телебаченню, поширюються «на відео» в формі відеодисків.

На початковій стадії розвитку кіно, головним питанням було як показати новий вид мистецтва глядачеві. Стали з'являтися кінотеатри і люди потягнулися в темні зали, щоб разом з іншими незнайомцями вперше побачити фільм. Будь то боротьба за приз, похорон вбитого президента, танцююча балерина, – світ був підкорений мистецтвом кіно з самого початку. Знадобилося понад 30 років, щоб на горизонті стали з'являтися перші кінофестивалі.

На початку 1930-х років був заснований перший кінофестиваль [2]. Але в 1920-х роках у великих містах Європи, Південної Африки та Північної Америки стали виникати кіноклуби і кінематографічні суспільства. Клуби дали можливість режисерам і любителям кіно показувати свої фільми, а також ділитися враженнями і обговорювати нові ідеї.

Лондонське кінематографічне товариство було одним із перших кіноклубів [2]. Воно було засноване в 1925 році, щоб представити глядачеві фільми, заборонені для показу в багатьох європейських кінотеатрах через політичний змісту. На чолі з засновниками товариства письменником Гербертом

Джорджем Уеллсом і сценаристом Джорджем Бернардом Шоу в клубі обговорювали кінорежисуру, а також сучасні тенденції та проблеми кіно.

На сьогодні подивитися нещодавно вийшовший у прокат фільм, популярний серіал або мультфільм у себе вдома стає цілим ритуалом. Благо, що велика частина цього контенту коштує значно дешевше, ніж в класичних кінотеатрах, та й можливості тут значно ширші. Величезні бази з докладним описом і трейлерами допомагають підібрати потрібний варіант для різних випадків. Крім того, глядач сам визначає час, коли йому зручно подивитися той чи інший фільм / серіал, а також пристрій, на якому це можна зробити: телевізор, комп'ютер, монітор, ноутбук, планшет або смартфон. При цьому буде гарантовано максимальну якість картинки і стереозвук [3].

Інша справа, що сервісів для перегляду відео зараз безліч. Це і російські платформи, ну і загальносвітові сервіси онлайн-відео, які займаються виробництвом власного контенту. У топ 10 кращих сервісів 2020 року були включені (наведені в алфавітному порядку) такі платформи: «Амедіатека», «КиноПоиск», «ТНТ Premier», ivi.ru, Google Play, MEGOGO, Netflix, Okko, TVZavr і Wink [3].

Враховуючи розвиток кіноіндустрії та інформаційних технологій – робота у даному напрямку обговорення кінофільмів є актуальною.

Розділ 1

Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Сутності предметної області кінематографа, які підлягають організації в кваліфікаційній роботі, сформовані історично, причому з'являлись вони поступово із появою нових етапів та технологій в розвитку кінематографа.

Кінематограф – галузь людської діяльності, яка полягає у створенні рухомих зображень. Іноді також згадується як сінематограф (від фр. Cinematographe) і кінематографія. Кінематограф був винайдений в XIX столітті і став вкрай популярним у XX столітті [4]. У саме поняття кінематографа входять кіномистецтво, як вид сучасного образотворчого мистецтва, твори якого створюються за допомогою рухомих зображень, та кіноіндустрія (кінопромисловість) – галузь економіки, яка виробляє кінофільми, спецефекти для кінофільмів, мультиплікацію, та демонструє ці твори для глядачів. Твори кіномистецтва створюються за допомогою кінотехніки. Вивченням кінематографа займається наука кінознавства.

Історично кінематограф з'явився в результаті рішення задачі по закріпленню на матеріальному носії зображення безперервного руху об'єктів і проєкції цього руху на екран. Для вирішення цього завдання необхідно було створення відразу декількох технічних винаходів: гнучкої світлочутливої плівки, апарату хронікографічної зйомки, проєктора швидко змінюваних зображень. Перша гнучка світлочутлива негорюча плівка була винайдена російським фотографом І.В. Болдиревим (1878 – 1881), потім американськими винахідниками Г.Гудвіном в 1887 і Дж. Істменом в 1889 була створена, целулоїдна плівка. Перші ж апарати хронікофотографічної зйомки були сконструйовані в 80-х роках XIX століття. До них відносяться: «фоторушниця» французького фізіолога Е. Марєя (1882), апарат французького винахідника О. ле Пернса (1888), апарат англійських винахідників У. Фризе-Гріна і М. Еванса (1889), апарат російського фотографа В.А. Дюбюка (1891), «фоноскопія»

французького фізіолога Ж. Демени (1892). Піонерами в створенні апаратів для проєкції на екран швидко змінюваних зображень були: німецький та російський фотографи О. Анцюц і В.А. Дебюк, які створили відповідно в 1891 і 1892 роках проєкційні апарати різної конструкції, але з однаковою назвою – «Тахіскоп», французький винахідник Е. Рейно, який створив в 1892 році проєктор під назвою «Оптичний театр», і російські винахідники І.А. Тимченко і М.Ф. Фрейденберг (1893) [5].

Винаходами найбільш наближеними до кінематографа за своїми технічними характеристиками є: «кінетоскоп» Едісона, апарат І.А. Тимченко (1893), «хронофотограф» Ж. Демени (1893), проєктор американського винахідника Ж.А. Ле Роя (1894), проєктор «паноптіком» американського винахідника У. Латам (1895), «плеограф» польського винахідника К. Прушинського (1894) та ін. А вже в 1895-1896 роках були винайдені апарати, що поєднують в собі всі основні елементи кінематографа: у Франції – «сінематограф» братів Л. і О. Люм'єр (1895) і «хронофотограф» Ж. Демени (1895); в Німеччині – «Біоскоп» М. Складановський (1895) і кінопроєктор О. Местер (1896); в Англії – «Аніматограф» Р.У. Пола (1896); в Росії – «хронофотограф» А. Самарського (1896) і «стробограф» І. Акімова (1896), в США – «вітаскоп» Т. Армата (1896) [5].

Початок поширення кінематографа був покладений зйомкою і публічною демонстрацією перших короткометражних фільмів. 1 листопада 1895 року в Берліні М. Складановський продемонстрував свій «Біоскоп», а 28 грудня 1895 року в Парижі братами Люм'єр був продемонстрований їх «сінематограф». Протягом 1896 – 1897 років публічні демонстрації короткометражних фільмів були зроблені у всіх світових столицях.

Перші короткометражні фільми (15-20 метрів, приблизно 1,5 хвилини демонстрації) були здебільшого документальні, проте вже в комедійній інсценуванні братів Люм'єр «Политий поливальник» відображаються тенденції зародження ігрового кіно. Німі фільми – це ціла епоха, яка передувала кінематографу, відомому всім зараз. У картинах цього жанру знімалися такі

світові знаменитості, як Чарлі Чаплін, Бастер Кітон, Мері Пікфорд, Теда Бара, Макс Ліндер, Глорія Свенсон і ін [6]. Вони стали акторами-основоположниками кінематографа. Для того, щоб донести до глядачів сюжет картини, вони мали мінімальними засобами – тільки своєю мімікою. Невелика довжина перших фільмів була обумовлена технічною недосконалістю кіноапаратури, тим не менш, вже до 1900-х років довжина кінокартин збільшилася до 200-300 метрів (15-20 хвилин демонстрації). Удосконалення знімального і проєкційної техніки сприяло подальшому збільшенню довжини фільмів, якісного і кількісного збільшення художніх прийомів зйомки, акторської гри та режисури. А широке поширення кінематографа його популярність забезпечили його економічну вигідність, що, однак, не могло не позначитися на художній цінності кінокартин [5].

Ще до початку ХХ століття Томас Едісон намагався синхронізувати кінескоп с фонографом, але зазнав невдачі. Однак надалі Вільям Діксон – співавтор Едісона стверджував, що йому вже в 1889 році вдалося створити кінетофонограф (прилад, що відтворював звук і зображення одночасно). Однак не існує ніяких доказів, що підтверджують його слова.

У ранній період кінематографа звукове кіно намагалися створити в безлічі країн, але зіткнулися з двома основними проблемами: труднощі в синхронізації зображення і звуку і недостатня гучність останнього. Перша проблема була вирішена шляхом запису і звуку, і зображення на одному і тому ж носії, але для вирішення другої проблеми потрібно винахід підсилювача низької частоти, що сталося лише в 1912 році, коли кіномова розвинулась настільки, що відсутність звуку вже не сприймалася як серйозний недолік .

В результаті патент на ту систему звукового кінематографа, яка згодом зробила звукову революцію, був отриманий в 1919 році, але кінокомпанії не звернули ніякої уваги на можливість кіно заговорити, бажаючи уникнути подорожчання вартості виробництва і прокату кінофільмів і втрати іншомовних ринків. Поки в 1925 році компанія Warner Brothers, яка перебувала на межі банкрутства, не вклалася в ризикований звуковий проєкт.

Уже в 1926 році Warner Brothers випустила кілька звукових фільмів, що складаються в основному з музичних номерів, але особливого успіху у глядачів вони не мали. Успіх прийшов лише з фільмом "Співак джазу", в якому крім музичних номерів Ела Джалсона були присутні і його короткі репліки. 6 жовтня 1927 року – день прем'єри "Співака Джазу" – прийнято вважати днем народження звукового кіно.

Із розвитком кіно з'явилась потреба у фільтруванні вмісту фільмів, цензурі та введенні певних вікових обмежень. Одним із таких є рейтинг МРАА.

Система рейтингів Американської кіноасоціації (англ. МРАА film rating system) – прийнята в США система оцінки змісту фільму, введена Американською кіноасоціацією (МРАА). Залежно від отриманої оцінки, глядацька аудиторія картини може бути обмежена за рахунок виключення з неї дітей і підлітків. Рейтинг Американської кіноасоціації грає важливу роль у прокатній долі фільму [7].

Система введена в дію 1 листопада 1968 року і з деякими незначними змінами використовується до сих пір. Ця система не бере на себе функції критика, вона не визначає, погана чи хороша та чи інша картина. Вона не ставить її під цензуру, а лише роз'яснює потенційні небезпеки для дитячих очей. Причому, оцінки виставляють самі батьки – саме вони на ротаційній основі входять до правління адміністрації щодо класифікації та рейтингів.

Проте незважаючи на велику кількість об'єднань кінофанів у соціальних мережах, саме застосунків із повноцінним функціоналом і можливістю не тільки перегляду фільмів, а і їх обговорення практично немає. Тому робота у даному напрямку актуальна і потребує подальшого дослідження.

1.2 Аналіз існуючого програмного забезпечення предметної області

Окрім вікових обмежень, також формуються рейтинги по ТОП-фільмах. Формуються вони за певними алгоритмами, розглянемо деякі з них: IMDb, іві.рейтинг, Кіноафіша.

IMDb – список кращих художніх фільмів світу, який формується на основі оцінок, що виставляються картинам зареєстрованими відвідувачами сайту IMDb. Крім основного загального списку, на IMDb складаються окремі рейтинги за жанрами, за десятиліттям, і за віком і статтю тих, хто голосував. Існує також рейтинг «100 найгірших фільмів за версією IMDb» [8].

Рейтинг фільму проводиться на основі байесовської оцінки за такою формулою [8]:

$$W = \frac{RV + CM}{V + M}, \quad (1.1)$$

де W – остаточний рейтинг, V – число голосів, відданих за фільм, M – мінімальна кількість голосів для включення до рейтингу $M = 25\,000$, R – середня оцінка фільму (за десятибальною шкалою), C – середня оцінка серед усіх фільмів.

При цьому враховуються тільки голоси зареєстрованих відвідувачів сайту IMDb, які голосують постійно.

ivi.рейтинг – новий рейтинг розповідає про сильні сторони фільму, правильно формуючи очікування від перегляду. Економить час на вивченні відгуків та рецензій, а також показує різницю між фільмами з однаковими балами. На відміну від інших рейтингів, оцінку і думку про сильні сторони фільму можна поставити тільки після перегляду. Захищений від накрутки – враховуються оцінки тільки перевірених користувачів [9].

Рейтинг молодий і оцінки є ще не у всіх фільмів, але він завжди є у новинок.

Кіноафіша – рейтинг, що формується за оцінками користувачів сайту кіноафіші [10]. Також значною мірою на вподобання смаків користувачів впливають об'єднання фанатів кіно. На просторах інтернету є багато статей, на кшталт куди поїхати на відпочинок фанату, щоб побувати в місцях із відомих кінофільмів тощо [11].

Також створено багато різноманітних кінофорумів, з обговоренням різноманітних тем [12], як до прикладу, на рисунку 1.1.

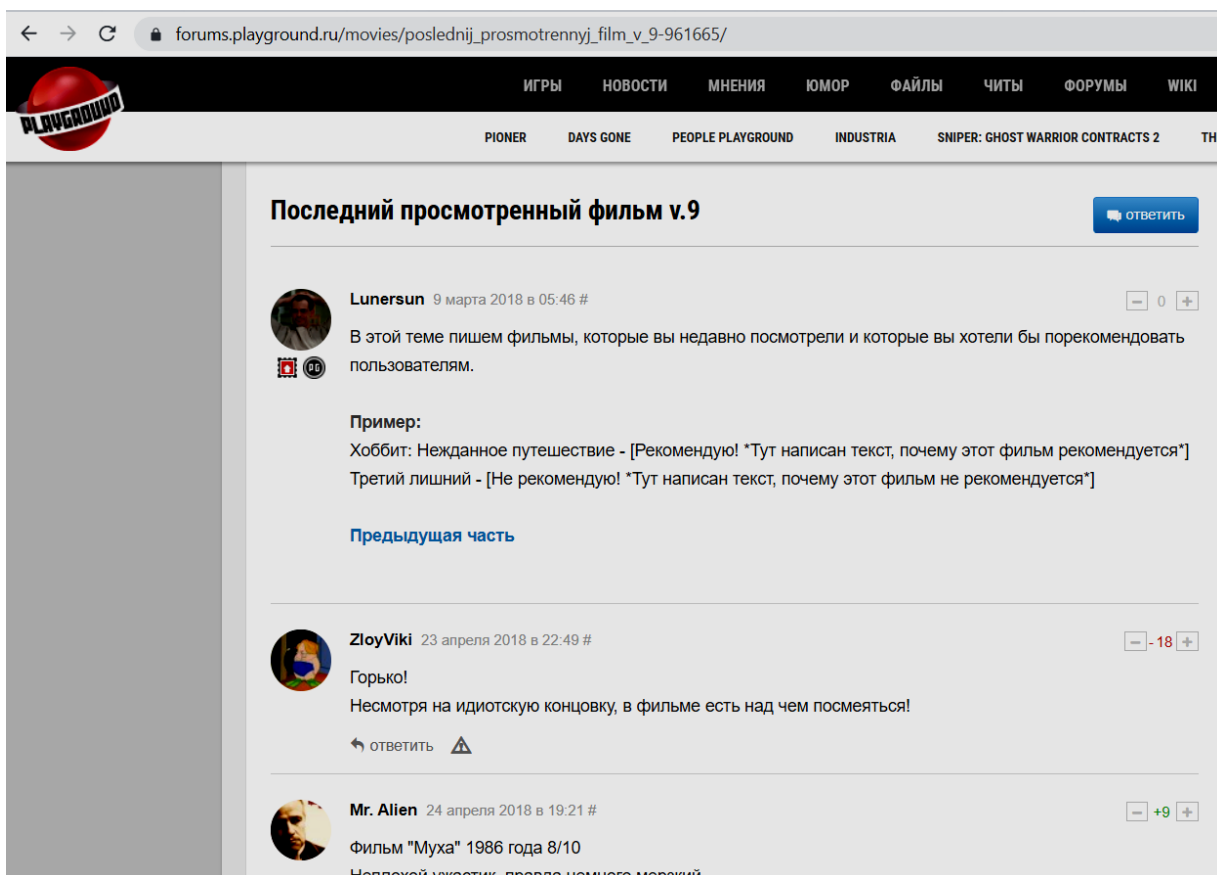


Рисунок 1.1 – Видяг кїнофоруму [12]

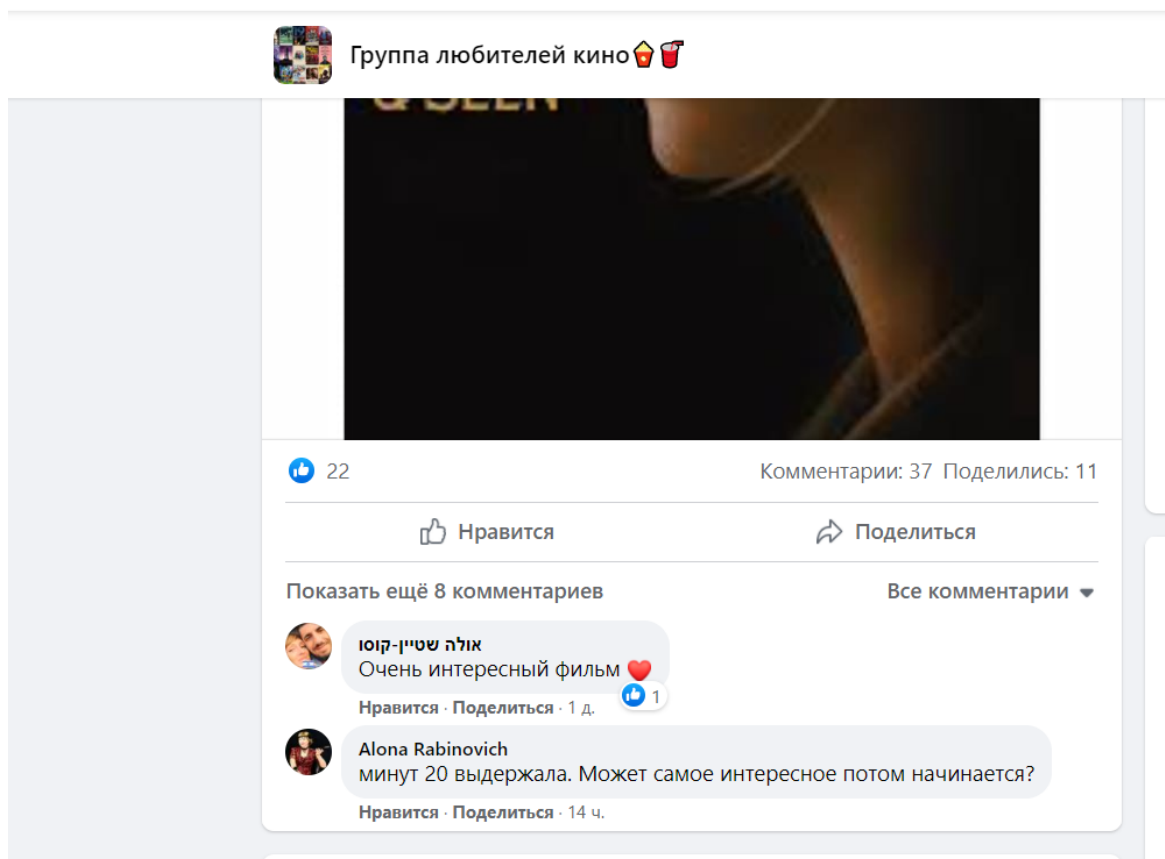


Рисунок 1.2 – Видяг спїльноти кїнофанатїв Facebook [13]

Окрім форумів є також тематичні об'єднання у соціальних мережах, де кінофанати можуть обговорювати різноманітні теми кіномистецтва, телесеріали та інше [13], рисунок 1.2. Більшість таких спільнот – закриті і потребують підтвердження членства у групах (рисунок 1.3).

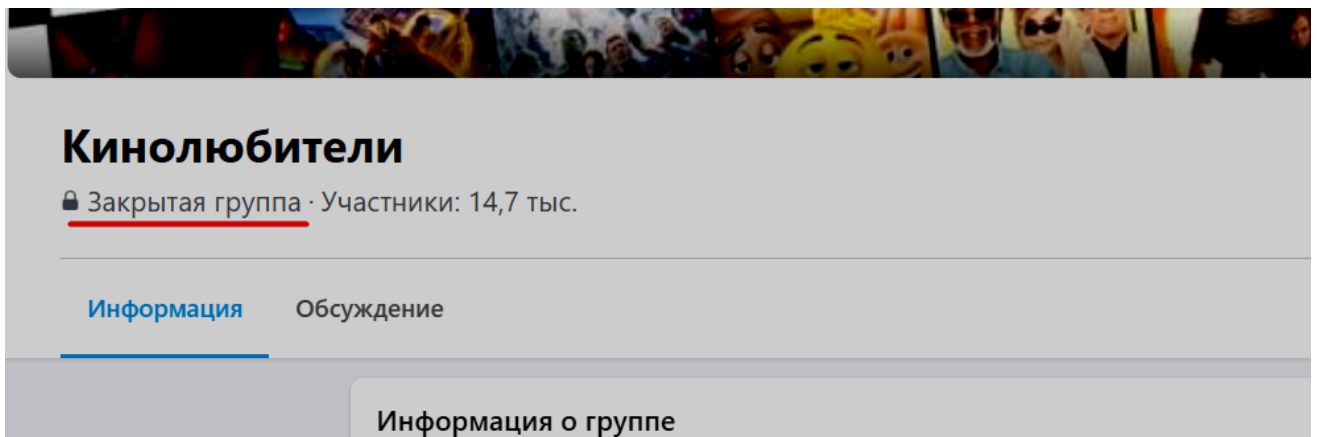


Рисунок 1.3 – Вигляд закритої групи Facebook [13]

Отже, окремо існує багато сервісів оцінок кіно, також існують групи обговорення, проте програм де був би доступний одночасно функціонал перегляд переліку кінофільмів на обговоренні, пошуку кінофільму за назвою, роком прем'єри, жанром чи країною, перегляду докладного опису обраного кінофільму та вражень інших користувачів, долучення власного враження про обраний кінофільм, оцінки обраного кінофільму, оцінки обраного враження іншого користувача, а також додавання нового кінофільму до веб-сервісу та перегляду переліку користувачів веб-сервісу та їх статистики немає. Тому є потреба в автоматизації процесів засобами інформаційних технологій.

1.3 Аналіз сучасних засобів створення програмного забезпечення

Перш за все потрібно визначитись із типом розроблюваного застосунку. Враховуючи предметну область, застосунок повинен бути доступним з інтернету і бажано з будь-якого пристрою, бути легким у доступі (не потребувати

установлення). Тому застосування під десктоп та застосування виключно під мобільні додатки не підходять, оскільки десктопні програми потребують для їх роботи необхідні достатні апаратні ресурси комп'ютера, сам застосунок і набір бібліотек, що містять функції для роботи з застосунком [14]. Що стосується мобільного застосування – то це програмне забезпечення, створене для мобільних пристроїв (смартфонів, планшетів і т.п.) і адаптоване під певну платформу (iOS, Android, Windows). На відміну від веб-ресурсів, воно працює без доступу до мережі та потребує установки [15].

Отже, найкращим рішенням для нашого застосування буде саме веб-версія. Веб-застосування – це прикладне програмне забезпечення, яке працює на веб-сервері, на відміну від комп'ютерних програм, що запускаються локально в операційній системі пристрою. Доступ до веб-програм користувач здійснює через веб-браузер із активним мережевим підключенням. Ці програми програмуються за допомогою змодельованої структури клієнт-сервер – користувачеві ("клієнту") надаються послуги через зовнішній сервер, розміщений сторонньою стороною [16].

Далі потрібно визначитись із платформою, на якій є можливість створити веб-застосування для любителів кіномистецтва.

Обираємо серед платформ .NET, PHP та Java.

PHP. Фреймворки PHP – це програмні платформи, які значно полегшують і прискорюють розробку сайтів, web- і мобільних додатків. Вони частково автоматизують процес написання коду спрощують формування структури майбутнього програмного рішення і спрощують об'єднання програмних компонентів проєкту. Також у багатьох з них передбачена можливість швидкої інтеграції та налаштування взаємодії програмного продукту зі сторонніми сервісами [17].

Однак, використання PHP 7 має недоліки [18].

– Відсутність лідера – багато технологій і мов мають лідера, архітектора, який визначає вигляд технології, задає вектор розвитку, приймає рішення про те, що повинно бути обов'язково, а чого не буде ніколи (Linux, Python, Ruby і т. П). У PHP лідера немає, багато рішень і конструкцій – це компроміс зацікавлених груп і історично сформованих реалій.

– Непослідовний синтаксис – при вивченні мови PHP, особливо старої частини заснованої на функціях, можна помітити, що частина функцій має префікси `array_`, `str_`, частина не має. Параметри функцій можуть бути розташовані не зовсім логічно і не так, як в іншій функції цієї ж групи.

– PHP – вже досить довго існуюча мова. Коли вона тільки з'являється, була досить елегантна і внутрішньо узгоджена. У міру життєвого циклу мова обростає додатковими ключовими словами, артефактами, застарілими конструкціями, які начебто є, працюють, але якими не рекомендується користуватися. У PHP була досить бурхлива молодість, в ході якої була скасована маса директив і прийомів, які на перший погляд повинні були полегшувати розробку, а на практиці оберталися серйозними проблемами безпеки. Сам PHP, що стартував як НЕОБ'ЄКТНО-орієнтована мова, в даний момент стала повноцінною об'єктно-орієнтованою мовою. Однак в ньому повно старих процедурних артефактів, якими доведеться користуватися.

– Спільнота PHP-розробників велика і роз'єднане, оскільки PHP – це одна з перших технологій для розробки веб-проектів, половина інтернету створена з його участю. В PHP-розробку одночасно було залучено величезну кількість програмістів по всьому світу. Все це породило велике число самих різних підходів, фреймворків і несумісних один з одним екосистем.

Java – мова, орієнтована на об'єкти, що характеризується строгою типізацією. Це універсальна програмна платформа, за допомогою якої розробляють настільні і мережеві, а також мобільні додатки. Її застосовують для сайтів уряду, наукових проєктів, технологій Apache Storm, Hadoop. Додатки

Android також працюють на Java. Вказуючи, що пишуть на Java, обов'язково слід згадати комп'ютерні ігри, десктопний софт, масштабні бізнес-проекти.

При безлічі плюсів мову програмування Java має деякі мінуси [19]:

– Низька швидкість і продуктивність, якщо порівнювати з C і C ++, викликана компіляцією і абстракцією віртуальної машини. До інших проблем, що уповільнює швидкість, можна віднести софт, очищаючий пам'ять, який займає більше 20% продуктивності процесора і негативно впливає на загальну продуктивність. Невірні налаштування кешування нерідко призводять до перевантаження пам'яті. При взаємній блокуванні потоків також виникають помилки, пов'язані з нестачею пам'яті.

– Плата за комерційне використання. Організації, що використовують Java Standard Edition 8, при отриманні оновлень і усунення помилок з 2019 року мають вносити оплату.

– Не передбачено нативний дизайн. Щоб розробляти софт з нативним дизайном, на Андроїд передбачений інструмент Android Studio. Але для розробки інтерфейсу користувача на ПК необхідний інструментарій відсутній, тому розбираючись, в чому недоліки Java, слід згадати невідповідності інтерфейсів.

– Складні коди. Багатослівні коди спрощують вивчення мови. Але довгі і складні речення часто викликають труднощі при читанні і перегляді кодів.

.NET Framework і ASP.NET, як платформи для веб-розробки.

.NET Framework – це програмна платформа, випущена компанією Microsoft, яка підходить для різних мов програмування. ASP.NET – технологія створення веб-додатків і веб-сервісів від компанії Microsoft. Вона є складовою частиною платформи Microsoft .NET і розвитком старішої технології Microsoft ASP [20].

Вважається, що платформа .NET Framework стала відповіддю компанії Microsoft на набрала на той час велику популярність платформи Java. ASP.NET

ґрунтується на Common Language Runtime: розробники можуть писати код для ASP.NET, використовуючи практично будь-які мови програмування, деякі з яких входять в комплект .NET Framework (C #, Visual Basic.NET і JScript .NET), а інші можуть бути встановлені додатково (IronRuby, IronPython, PHP, Perl, Smalltalk, Haskell і ін.). Проте, нерідко використовується не цілком коректний термін «.NET-програмист», який, мабуть, характеризує розробника на будь-якій мові програмування, який може бути використаний в .NET-проект.

Деякі особливості ASP.NET [20]:

- Компільований код виконується швидше, а більшість помилок отлавлюється ще на стадії розробки.
- Розширюваний набір елементів управління і бібліотек класів, що прискорює розробку.
- Можливість кешування всієї сторінки, її частин або даних, що використовуються на сторінці.
- Можливість поділу візуальної частини та бізнес-логіки по різних файлах, є можливість виділяти часто використовувані шаблони для користувача елементів управління, таких як меню сайту, наявність master-сторінок для завдання шаблонів оформлення, підтримка AJAX (розширення ASP.NET AJAX).
- Розширювані моделі подій, обробки запитів і серверних елементів управління.
- Підтримка CRUD-операцій при роботі з таблицями через GridView.
- Можливе створення веб-додатків, які реалізують шаблон Model-View-Controller (ASP.NET MVC Framework).

.NET досить широко поширений в сфері розробки внутрішньокорпоративних застосувань, але в веб-розробці все ж зустрічається відносно рідко, як і інші програмні продукти корпорації Microsoft. Тому знайти розробників для веб-проекту буває досить непросто. Використання .NET «тягне» за собою покупку і іншого програмного забезпечення від корпорації Microsoft

(серверної ОС, СКБД і т.п.). Технологія досить дорога в розробці і супроводі: крім витрат на покупку ліцензій на необхідне програмне забезпечення істотний внесок до бюджету проєктів вносять високі зарплати розробників.

Отже, враховуючи вищевикладений матеріал, було прийняте рішення розробки веб-застосування на платформі ASP.NET.

1.4 Постановка задачі та вимоги до розробки інформаційної системи

Метою кваліфікаційної роботи бакалавра є розробка веб-сервісу для обговорення кінофільмів на платформі ASP.NET, що призначена для трьох категорій користувачів із відповідними функціями:

1. Незареєстрований користувач:

- перегляд переліку кінофільмів на обговоренні;
- перегляд докладного опису обраного кінофільму;
- подання заявки на реєстрацію в системі.

2. Зареєстрований користувач:

- авторизація в системі;
- перегляд переліку кінофільмів на обговоренні;
- пошук кінофільму за назвою, роком прем'єри, жанром чи країною;
- перегляд докладного опису обраного кінофільму та вражень інших користувачів;

- долучення власного враження про обраний кінофільм;

- оцінка обраного кінофільму;

- оцінка обраного враження іншого користувача;

- додавання нового кінофільму до веб-сервісу;

- перегляд переліку користувачів веб-сервісу та їх статистики.

3. Адміністратор веб-сервісу:

- авторизація в системі;

- затвердження та відхилення заявок на реєстрацію у веб-сервісі;

- перегляд переліку кінофільмів на обговоренні з можливістю їх блокування;
- перегляд докладного опису обраного кінофільму та вражень інших користувачів з можливістю блокування обраних вражень;
- додавання нового кінофільму до веб-сервісу;
- перегляд переліку користувачів веб-сервісу з можливістю їх блокування.

Розділ 2

Проектування інформаційної системи

2.1 Функціональна структура та бізнес-процеси системи

При розробці веб-сервісу обговорення кінофільмів необхідно автоматизувати наступні бізнес-процеси:

- бізнес-процес «Робота з переліком кінофільмів на обговоренні»;
- бізнес-процес «Робота з поданням заявки на реєстрацію в системі»;
- бізнес-процес «Робота з пошуком кінофільму»;
- бізнес-процес «Робота з докладним описом обраного кінофільму»;
- бізнес-процес «Робота з користувачами веб-сервісу»;

Бізнес-процес «Робота з переліком кінофільмів на обговоренні». Система повинна надавати доступ до переліку кінофільмів, що є на обговоренні.

Доступ користувача до цієї інформації відбувається наступним чином:

1. Користувач відкриває сторінку веб-сайту.
2. Користувач переглядає на головній сторінку перелік фільмів, які є на обговоренні.

Діаграму дій для даного бізнес-процесу зображено на рисунку 2.1.

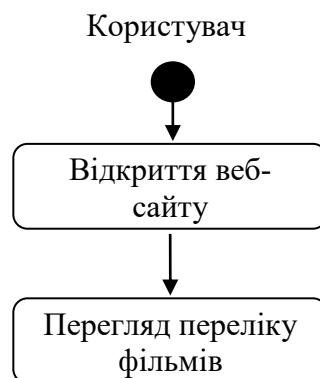


Рисунок 2.1 – Діаграма дій бізнес-процесу «Робота з переліком кінофільмів на обговоренні».

Бізнес-процес «Робота з поданням заявки на реєстрацію в системі».

Система повинна надавати доступ незареєстрованому користувачу подати заяву на реєстрацію, яка далі відправляється адміністратору на підтвердження.

Робота користувача з заявою на реєстрацію відбувається наступним чином:

1. Користувач відкриває головну сторінку веб-сервісу та бачить там кнопку «Подати заяву на реєстрацію».
2. Користувач натискає на кнопку та переходить на сторінку подання заявки, де має ввести необхідні дані для реєстрації.
3. Для завершення подання заявки користувач натискає кнопку «Подати заяву».

Діаграму дій для даного бізнес-процесу зображено на рисунку 2.2.



Рисунок 2.2 – Діаграма дій бізнес-процесу «Робота з поданням заявки на реєстрацію в системі».

Бізнес-процес «Робота з пошуком кінофільму». Система повинна надавати доступ до кінофільмів, які шукає зареєстрований користувач, якщо вони є в базі даних.

Робота зареєстрованого користувача з пошуком кінофільмів відбувається наступним чином:

1. Користувач відкриває головну сторінку веб-сайту.
2. Користувач виконує авторизацію в системі.
3. Користувач натискає на кнопку «Пошук».
4. Користувач вводить слово для пошуку та обирає по чому буде відбуватись пошук.
5. Користувач натискає на кнопку «Шукати фільм».
6. Користувач бачить знайдені за його запитом фільми, якщо нічого не знайдено, то виводиться відповідне повідомлення.

Діаграму дій для даного бізнес-процесу зображено на рисунку 2.3.

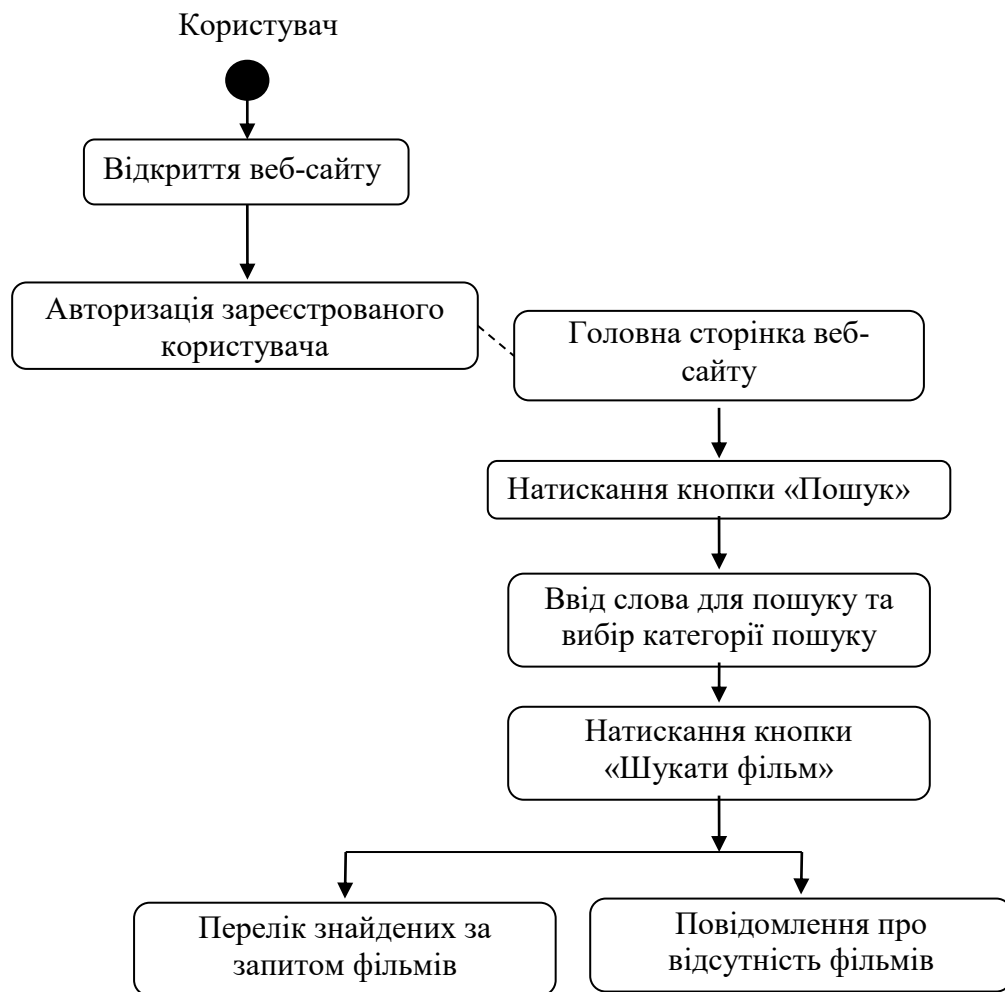


Рисунок 2.3 – Діаграма дій бізнес-процесу «Робота з пошуком кінофільму»

Бізнес-процес «Робота з докладним описом обраного кінофільму».

Система повинна надавати користувачу можливість переглядати детальну інформацію про кінофільм.

Робота користувача з докладним описом обраного кінофільму відбувається наступним чином:

1. Користувач відкриває головну сторінку веб-сервісу.
2. Користувач обирає фільм та натискає на кнопку «Детальніше» під ним.
3. Користувач переглядає детальну інформацію про обраний кінофільм.

Діаграму дій для даного бізнес-процесу зображено на рисунку 2.4.

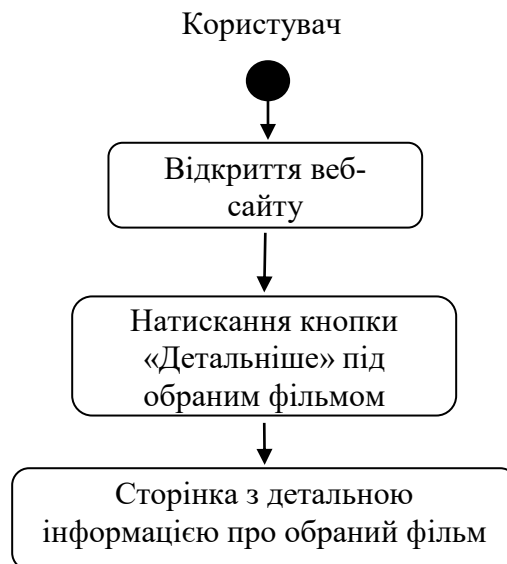


Рисунок 2.4 – Діаграма дій бізнес-процесу «Робота з докладним описом обраного кінофільму».

Бізнес-процес «Робота з користувачами веб-сервісу». Система повинна надавати зареєстрованому користувачу можливість переглядати детальну інформацію про інших користувачів веб-сервісу та їх статистику, яка зберігається в базі даних системи.

Робота користувача з інформацією про інших користувачів відбувається наступним чином:

1. Користувач відкриває головну сторінку веб-сервісу.
2. Користувач виконує авторизацію в системі.

3. Користувач натискає на кнопку «Користувачі веб-сервісу».

4. Користувач переглядає детальну інформацію про користувачів та може переглянути статистику кожного з них.

Діаграму дій для даного бізнес-процесу зображено на рисунку 2.5.



Рисунок 2.5 – Діаграма дій бізнес-процесу «Робота з користувачами веб-сервісу».

Таким чином бізнес-процеси реалізують функції користувачів, що були описані в постановці завдання.

2.2 Інформаційна структура системи

Для структурованого збереження даних було створено базу даних системи. Опираючись на завдання кваліфікаційної роботи розроблена БД була спроектована таким чином, щоб під час її використання можна було з легкістю додавати, видаляти та редагувати дані. Даталогічна модель створеної бази даних зображена на рисунку 2.6.

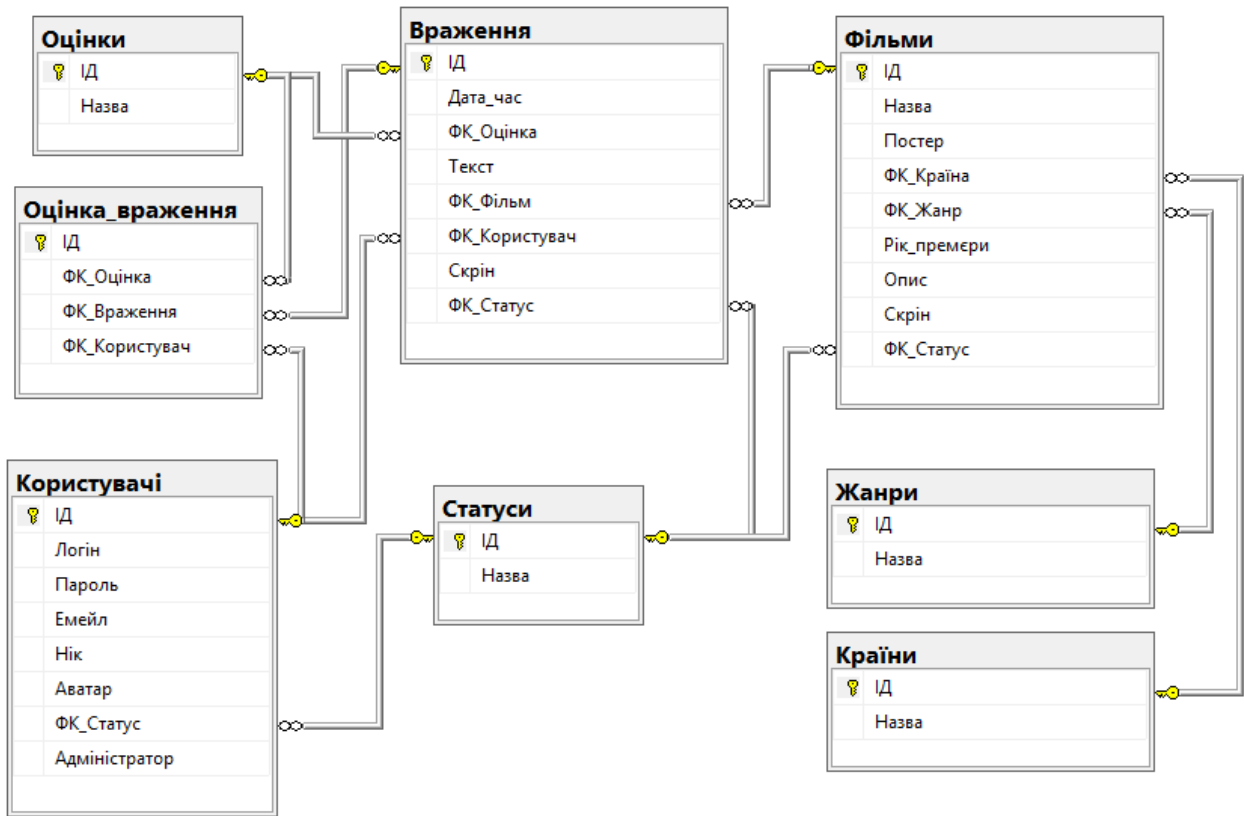


Рисунок 2.6. – Даталогічна модель БД веб-сервісу обговорення кінофільмів

Таблиця «Враження» є одною з головних таблиць, що використовується для збереження даних про враження, які залишають користувачі (таблиця 2.1). Поле «ID» це первинний ключ, який має цілочисельний тип даних. Поле «Дата_час» має тип даних `datetime` та зберігає інформацію про дату і час в який було залишено враження. Поле «ФК_Оцінка» це вторинний ключ, який з'єднує таблицю «Враження» з таблицею «Оцінки» та має цілочисельний тип даних. Поле «Текст» зберігає дані про текст враження, яке залишив користувач. Поле «ФК_Фільм» є вторинним ключем та поєднує таблиці «Враження» та «Фільми». Поле «ФК_Користувач» зберігає інформацію про те який користувач залишив враження, та поєднує дану таблицю з таблицею Користувачі. Поле «Скрін» зберігає інформацію про назву скріншоту враження. Останнім є поле «ФК_Статус» яке зберігає інформацію про те який статус має залишене враження (новий, активний, заблокований) та поєднує таблиці «Фільми» і «Статуси».

Таблиця 2.1 – Атрибути таблиці БД «Враження»

| № п/п | Назва | Тип поля | Опис |
|-------|---------------|--------------|-----------------------------------------------------------|
| 1 | ІД | int | Первинний ключ призначений для ідентифікації записів |
| 2 | Дата_час | datetime | Дата та час коли було залишено враження |
| 3 | ФК_Оцінка | int | Вторинний ключ, що пов'язується із таблицею «Оцінки» |
| 4 | Текст | text | Текст |
| 5 | ФК_Фільм | int | Вторинний ключ, що пов'язується із таблицею «Фільми» |
| 6 | ФК_Користувач | int | Вторинний ключ, що пов'язується із таблицею «Користувачі» |
| 7 | Скрін | nvarchar(50) | Скріншот враження |
| 8 | ФК_Статус | int | Вторинний ключ, що пов'язується із таблицею «Статуси» |

Для збереження даних про фільми буде використовуватись таблиця «Фільми» (таблиця 2.2). Поле «ІД» це первинний ключ, який має цілочисельний тип даних. Поле «Назва» зберігає інформацію про назву фільмів. Поле «Постер» зберігає інформацію про назву фото, яке є головним при відображенні даних про фільми. Поле «ФК_Країна» вторинний ключ, який зберігає інформацію про країну в якій було випущено фільм та з'єднує таблиці «Фільми» та «Країни». Поле «ФК_Жанр» вторинний ключ, що зберігає в БД дані про жанр фільму, та поєднує таблиці «Фільми» та «Жанри». В полі «Рік_премери описується рік прем'єри фільму. Поле «Опис» включає в себе дані про опис фільму. Поле «Скрін» відповідає за додаткові фото фільму. Останнім є поле ФК_Статус – вторинний ключ, який надає інформацію про статус фільму та поєднує таблиці «Фільми» і «Статуси».

Таблиця 2.2 – Атрибути таблиці БД «Фільми»

| № п/п | Назва | Тип поля | Опис |
|-------|-------------|----------------|-------------------------------------------------------|
| 1 | ІД | int | Первинний ключ призначений для ідентифікації записів |
| 2 | Назва | nvarchar(1000) | Назва фільму |
| 3 | Постер | nvarchar(50) | Текст |
| 4 | ФК_Країна | int | Вторинний ключ, що пов'язується із таблицею «Країни» |
| 5 | ФК_Жанр | int | Вторинний ключ, що пов'язується із таблицею «Жанри» |
| 6 | Рік_премери | int | Рік прем'єри фільму |
| 7 | Опис | text | Опис фільму |
| 8 | Скрін | nchar(50) | Скріншот враження |
| 9 | ФК_Статус | int | Вторинний ключ, що пов'язується із таблицею «Статуси» |

Для збереження даних про користувачів використовується таблиця «Користувачі». Поле «ІД» це первинний ключ, який має цілочисельний тип даних. Поле «Логін» має строковий тип даних та зберігає дані про логіни користувачів, які використовуються при авторизації користувача. Поле «Пароль», аналогічним чином, відповідає за збереження даних про паролі користувачів та використовується при авторизації. Поле «Емейл» зберігає дані про e-mail користувача. Поле «Нік» відповідає за збереження даних про ніки користувачів. Поле «Аватар» текстовий тип даних який зберігає назву файлу аватару користувача. Поле «ФК_Статус», яке надає інформацію про статус користувача. Поле «Адміністратор» служить вказівником на те чи є користувач адміністратором чи ні.

Таблиця «Оцінки» зберігає дані про оцінки які користувач може поставити фільму (таблиця 2.4). Наприклад, найгірша оцінка – 1 це «Жах», а найкраща оцінка – 5 це «Круто». Поле «ІД» це первинний ключ, який має цілочисельний тип даних. Поле «Назва» зберігає інформацію про назву оцінки.

Таблиця 2.3 – Атрибути таблиці БД «Користувачі»

| № п/п | Назва | Тип поля | Опис |
|-------|---------------|--------------|-------------------------------------------------------|
| 1 | ІД | int | Первинний ключ призначений для ідентифікації записів |
| 2 | Логін | nvarchar(50) | Логін користувача |
| 3 | Пароль | nvarchar(50) | Пароль |
| 4 | Емейл | nvarchar(50) | Адреса електронна |
| 5 | Нік | nvarchar(50) | Назва ніку користувача |
| 6 | Аватар | nvarchar(50) | Назва фото аватарів користувачів |
| 7 | ФК_Статус | int | Вторинний ключ, що пов'язується із таблицею «Статуси» |
| 8 | Адміністратор | bit | Приналежність до адміністратора |

Таблиця 2.4 – Атрибути таблиці БД «Оцінки»

| № п/п | Назва | Тип поля | Опис |
|-------|-------|--------------|------------------------------------------------------|
| 1 | ІД | int | Первинний ключ призначений для ідентифікації записів |
| 2 | Назва | nvarchar(50) | Назва оцінки |

Таблиця «Оцінка_враження» зберігає дані про оцінку враження користувача, яку залишив інший користувач (таблиця 2.5). Поле «ІД» це первинний ключ, який має цілочисельний тип даних. Поле «ФК_Оцінка» це вторинний ключ, який поєднує таблиці «Оцінка_враження» та «Оцінки», та зберігає оцінку яку залишив користувач. Поле «ФК_Враження» це вторинний ключ, який поєднує таблиці «Оцінка_враження» та «Враження», зберігає дані про враження на яке було залишено оцінку. Поле «ФК_користувач» це вторинний ключ, який поєднує таблиці «Оцінка_враження» та «Користувачі», зберігає дані про користувача, який залишив оцінку.

Таблиця 2.5 – Атрибути таблиці БД «Оцінка_враження»

| № п/п | Назва | Тип поля | Опис |
|-------|---------------|----------|------------------------------------------------------|
| 1 | ІД | int | Первинний ключ призначений для ідентифікації записів |
| 2 | ФК_Оцінка | int | Оцінка враження |
| 3 | ФК_Враження | int | Враження на яке було залишено оцінку |
| 4 | ФК_Користувач | int | Користувач який залишив оцінку |

Таблиця «Статуси» зберігає дані про статуси користувачів та фільмів (таблиця 2.6). Поле «ІД» це первинний ключ, який має цілочисельний тип даних. Поле «Назва» зберігає інформацію про назву статусу (новий, активний, заблокований).

Таблиця 2.6 – Атрибути таблиці БД «Статуси»

| № п/п | Назва | Тип поля | Опис |
|-------|-------|--------------|------------------------------------------------------|
| 1 | ІД | int | Первинний ключ призначений для ідентифікації записів |
| 2 | Назва | nvarchar(50) | Назва статусу |

Таблиця «Жанри» зберігає дані про жанр фільмів (таблиця 2.7). Поле «ІД» це первинний ключ, який має цілочисельний тип даних. Поле «Назва» зберігає інформацію про назву жанру.

Таблиця 2.7 – Атрибути таблиці БД «Жанри»

| № п/п | Назва | Тип поля | Опис |
|-------|-------|--------------|------------------------------------------------------|
| 1 | ІД | int | Первинний ключ призначений для ідентифікації записів |
| 2 | Назва | nvarchar(50) | Назва жанру |

Таблиця «Країни» зберігає дані про країну фільмів (таблиця 2.8). Поле «ІД» це первинний ключ, який має цілочисельний тип даних. Поле «Назва» зберігає інформацію про назву країни.

Таблиця 2.8 – Атрибути таблиці БД «Країни»

| № п/п | Назва | Тип поля | Опис |
|-------|-------|--------------|------------------------------------------------------|
| 1 | ІД | int | Первинний ключ призначений для ідентифікації записів |
| 2 | Назва | nvarchar(50) | Назва країни |

Таким чином, в створеній базі даних, можна зберігати, додавати, видаляти та редагувати всю інформацію, яка потрібна для роботи веб-сервісу обговорення кінофільмів.

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

Враховуючи обрану платформу, мову будемо обирати з таких мов програмування, як: C#, VB.NET, JScript і C++ з керованими розширеннями. При цьому розроблений код є компільованим, тому код інтерфейсу може бути відділений від бізнес-логіки і поміщений в окремий файл [21].

VB.NET розшифровується як Visual Basic.NET, це мова програмування, яка була розроблена компанією Microsoft. Вперше дана мова була випущена у 2002 році, замість Visual Basic 6.

VB.NET є об'єктно-орієнтованою мовою програмування. Це означає, що вона підтримує функції об'єктно-орієнтованого програмування, які включають в себе інкапсуляцію, поліморфізм, абстракцію та успадкування [22].

Visual Basic .ASP NET працює на платформі .NET, що означає, що вона має повний доступ до бібліотек .NET. Така мова є продуктивним інструментом для швидкого створення досить великого спектра веб-додатків, додатків для Windows, Office і мобільних додатків, які створені на платформі .NET

Проте, як і кожна мова, дана має ряд недоліків [23]:

1. Підтримка операційних систем, лише Windows і Mac OS X.
2. Відсутнім є механізм який успадковує реалізації об'єктів. Тобто, існуюче в мові успадкування дозволяє успадковувати тільки інтерфейси програми, але не їх реалізацію.
3. Для роботи програми потребує встановити бібліотеку `msvbvmXX.dll`.
4. Повільна швидкість роботи. Пояснюється тим, що практично всі вбудовані функції мови реалізуються через бібліотеку часу виконання (runtime library), натомість бібліотека, в свою чергу, виробляє багато «зайвої» роботи з перевірки та / або перетворення типів.

Мова програмування JScript дозволяє зробити сторінку веб-сайту інтерактивною, тобто таку щоб при діях користувача сторінка змінювалась, або змінювались певні дані на ній [24]. Сама мова є об'єктно-орієнтованою, скриптовою і є діалектом мови ECMAScript. Дану мову можна використовувати на веб-сторінках та безпосередньо розташовувати всередині сторінки. Щоб додати код на сторінку необхідно написати тег `<script></script>`.

При використанні ідентифікаторів, в даній мові, необхідно уважно слідкувати за реєстром, оскільки мова є реєстрозалежною. Використання цифр на початку назви змінних заборонено.

Недоліком є те, що захист сторінок при використанні такої мови досить слабкий, оскільки, код виконується на стороні клієнта, і тому деякі програмісти можуть скористатись цим в зловмисних цілях.

До переваг даної мови можна віднести:

- керування пам'яттю відбувається автоматично;
- досить висока швидкість роботи;
- простота синтаксису;
- популярність.

Для створення додатків використовується і мова програмування C++. Мова є об'єктно-орієнтовною з підтримкою ще таких парадигм, як: узагальнена та процедурна. При створенні даної мови розробники хотіли зберегти сумісність з мовою попередником – С. До мови C++ було додано: обробку винятків, простори імен, перевантаження, додаткові типи даних та ін [25].

Мова C# має статичну типізацію підтримує багато функцій, зокрема, перевантаження операторів, атрибути, події, винятки та ін [26]. На відмінну від мови C++, дана мова не підтримує множинну спадковість, тобто за один раз може бути успадкований лише один клас.

Отже, підсумувавши всі переваги та недоліки розглянутих мов програмування, для розробки було обрано мову програмування C#.

2.3.2 Вибір фреймворку

Для створення додатку розглядалися такі фреймворки: MVC 4, MVC 5 та ASP.NET.

ASP.NET візуально нагадує старішу версію ASP, це дало розробникам можливість з легкістю перейти на більш нову версію технології. Даний фреймворк має ряд переваг, зокрема:

- швидкість;
- розділення візуальної частини та бізнес-логіки;
- багатомовність;
- різноманітність бібліотек та елементів розробки.

Розширенням ASP.NET стала версія ASP.NET MVC. Фреймворк MVC реалізує шаблон Model-View-Controller, тобто архітектура проекту поділяється на три складових: модель даних, вигляд, модуль керування – контролер. Саме завдяки такому шаблону при розробці ІС забезпечується гнучкий дизайн, та швидке і просте повторне використання окремих частин програми.

2013 року вийшла нова версія ASP.NET MVC – MVC 5, та Visual Studio 2013, яка надає інструменти та бібліотеки для роботи з MVC5 [27].

В більшості моментах MVC 5 не дуже сильно відрізняється від MVC 4, більшість з попередньої версії можна використати та застосувати і в новій, проте існує і ряд відмінностей:

- зміна концепції аутентифікації і авторизації, замість SimpleMembershipProvider у нову версію було впроваджено систему ASP.NET Identity, дана система застосовує компоненти OWIN і Katana;

- щоб створити адаптивний та розширюваний інтерфейс в MVC 5 вже використовується css-фреймворк Bootstrap;

- було додано нові фільтри аутентифікації та функціональність перевизначення фільтрів;

- було додано атрибути маршрутизації.

Що для однієї версії, що для іншої є багато документації, за допомогою якої розробник з легкістю може створити необхідний йому програмний продукт.

Отже, для розробки веб-сервісу було обрано фреймворк MVC 5.

2.3.3 Вибір редактора програмного коду

Одною з головних переваг ASP.NET в порівнянні з іншими мовами і платформами під час створення веб-застосунків є безкоштовна доступність повноцінних інструментів програмування. Жодний безкоштовний застосунок

для інших веб-технологій не зрівняється з можливостями і зручністю роботи з інструментами для ASP.NET [28]. Для розробки та запису програмного коду необхідно обрати редактор програмного коду, вибір буде зроблено серед таких редакторів: Microsoft Visual Studio Community, Visual Studio Code та Microsoft Visual Studio 2019 Professional.

Microsoft Visual Studio Community – безкоштовне, повнофункціональне і розширюване інтегроване середовище розробки, яке використовується для створення сучасних застосунків для Windows, Android і iOS, а також веб-застосунків і хмарних служб. Visual Studio Community була створена для індивідуальних розробників, проектів які мають відкритий код, наукових досліджень і невеликих груп фахівців. Даний редактор багатомовний, тобто, підтримує програмування на різних мовах: C #, Visual Basic, F #, C ++, HTML, JavaScript, Python та ін.

Visual Studio Code – безкоштовний редактор коду, який був створений на відкритому вихідному коді. Також, він є багатомовним, та підтримує такі мови програмування, як: C #, Razor, PHP, HTML, XML, CSS, JavaScript, Sass, Python, Perl, F #, C ++ та ін.

Даний редактор дає можливість виконувати налагодження коду прямо з редактора. Це стає можливим за допомогою точок зупини, стеків викликів і інтерактивної консолі. Visual Studio Code є розширюваним та налаштовуваним.

Для збільшення можливостей редактора встановлюються розширення, які додають нові мови, теми, відлагоджувальників і можливість підключення до додаткових сервісів. Розширення виконуються в окремих процесах, саме тому вони не уповільнюють роботу редактора. Visual Studio Code працює з такими операційними системами, як: Windows, Linux, Mac.

Microsoft Visual Studio 2019 є комерційним середовищем розробки. Воно використовується для індивідуальних розробників і підприємств. Має розширені

можливості налагодження, діагностики, тестування і кроссплатформеної розробки. Для студентів деяких університетів видається ліцензія на користування даним продуктом, саме тому є можливість безкоштовно користуватись.

Підсумувавши все, для створення програмного продукту було обрано редактор Microsoft Visual Studio 2019 Professional.

2.3.4 Вибір СКБД

Для правильного збереження даних інформаційної системи необхідно використовувати базу даних, тому потрібно обрати СКБД. Для платформи ASP.NET доцільно буде розглянути такі системи як: MS SQL Server та MySQL.

Як одна так і інша СКБД є популярними серед розробників, тому для того щоб зробити вибір на користь тієї чи іншої системи необхідно розглянути їхні подібні та відмінні характеристики [29].

Спільним для двох систем керування можна визначити такі характеристики:

- стандартна модель таблиць, що використовуються для зберігання даних у стовпцях та рядках;
- швидкість збереження та обробки даних, оскільки дані зберігаються в таблиці, то СКБД забезпечують швидку взаємодію з ними;
- висока продуктивність;
- первинні та зовнішні ключі, використовуються для поєднання таблиць бази даних;
- популярність, а отже велика кількість документації, різних форумів та спільнот, які зможуть допомогти в ознайомленні та використанні систем керування;

- масштабованість, обидві системи можуть масштабуватись як під малі, так і під великі проекти та можуть виконувати до мільйона транзакцій щодня;
- подібний синтаксис;
- підтримка різними програмними редакторами.

Серед ряду подібного є і відмінні характеристики, оскільки, більшість процесів відбувається у фоновому режимі, розробник може навіть не звернути на них уваги.

Однією з відмінностей є те, що MySQL може працювати на різних операційних системах, таких як: Linux, Mac OS X та Windows, на відмінну від MS SQL Server, який розроблявся для користувачів операційної системи Windows. Хоча зараз працівники Microsoft реалізують використання СКБД і для інших операційних систем.

MS SQL дозволяє налаштовувати свої класи сутності в .NET, тобто можна працювати з запитамі LINQ, чого не можна зробити в MySQL, оскільки, тут доведеться завантажувати сторонні інструменти для використання .NET.

Для того, щоб зберегти дані, можна робити резервні копії, у цьому напрямку SQL Server більш простіший. Під час копіювання в MySQL виконується блокування бази даних, це зроблено для того, щоб мінімізувати ризики пошкодження даних під час перемикання між версіями.

MySQL не дає користувачу можливості редагувати або скасовувати запит після його запуску. Щоб зупинити виконання запиту користувачу потрібно зупинити весь процес. Натомість у MS SQL користувачі можуть змінити або скасувати запит під час роботи, не зупиняючи повний процес. Більше того, MS SQL для стабільного стану, використовує механізм транзакцій. Це надає MS SQL Server досить значну перевагу над MySQL.

В більшій кількості розробники використовують MS SQL Server для розробки з .NET, а MySQL для розробки з PHP, тому підсумувавши відмінності

та подібності двох СКБД для розробки веб-сервісу обговорення кінофільмів було обрано СКБД MS SQL Server 2012.

Отже, для створення програмного продукту було обрано такий набір засобів розробки: мову програмування C#, фреймворк MVC 5, редактор Microsoft Visual Studio 2019 Professional та СКБД SQL Server 2012.

Розділ 3

Програмна реалізація інформаційної системи

3.1 Структура та функціональне призначення складових системи

Беручи за основу розроблену структуру веб-сервісу для обговорення кінофільмів було створено діаграму класів, яка буде застосовуватись при розробці проекту. Розроблена діаграма зображена на рисунку 3.1.

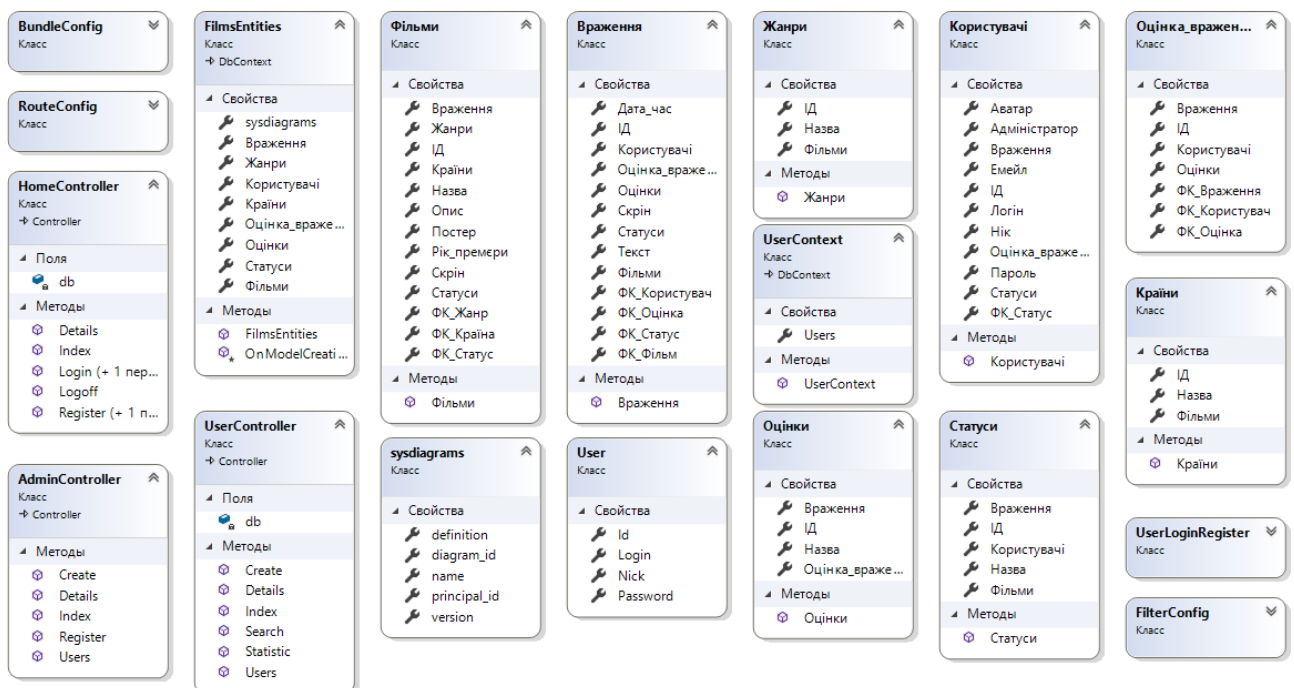


Рисунок 3.1 – Діаграма класів веб-сервісу

Для створення веб-додатку було обрано фреймворк ASP.NET MVC, тому для таблиць бази даних, які використовуються в додатку як основні було створено моделі, представлення та контролери: Index, Details, Create та ін.

Представлення Index виводить на сторінку перелік даних відповідної моделі. Представлення Details відображає детальні дані про обраний запис, у даному випадку про обраний кінофільм. Представлення Create надає можливість створити та додати в базу даних новий запис, у даному випадку додати новий кінофільм для обговорення. Представлення Register надає можливість незареєстрованому користувачу подати заяву на реєстрацію у веб-сервісі.

Класи Фільми, Враження, Жанри, Оцінки, Користувачі, Статуси, Оцінка_враження та Країни відповідають аналогічним таблицям бази даних та є їхніми моделями. Класи мають свої контролери, які забезпечують їх відображення та пов'язують моделі з їх представленнями.

Програма розпочинає свою роботу з початкового контролера – HomeController. В даному контролері за відображення головної сторінки веб-сервісу відповідає представлення Index. Також, в даному контролері реалізовані методи, що відповідають за відображення даних. Метод Details дозволяє переглянути детальну інформацію про фільм, який обирається на головній сторінці. Метод Login() дає можливість авторизуватись зареєстрованому раніше користувачу, або авторизуватись адміністратору сайту. Метод Logout надає можливість вийти з авторизованого акаунта.

Реєстрація користувача реалізована за допомогою подачі заяви незареєстрованого користувача, заяву може підтвердити або відхилити лише адміністратор сайту на відповідному представленні.

Клас User відповідає за перегляд інформації зареєстрованим користувачем про фільми, враження, інших користувачів та їх статистику. З даним класом взаємодіє контролер UserController. Метод Create надає можливість додати в базу даних новий кінофільм для обговорення. Метод Details дозволяє переглядати детальну інформацію про фільм, який обирається на головній сторінці. Метод Users надає доступ до перегляду інформації про інших зареєстрованих користувачів та їх статистику. Метод Statistic відображає статистику обраного користувача. Окрім основних методів відображення даних, в даному контролері реалізовано метод пошуку – Search.

Контролер AdminController відповідає за роботу адміністратора з системою. В цьому контролері існують такі представлення, як: Index, Details, Create, Register та Users. Метод Register відображає подані заяви незареєстрованих користувачів, адміністратор веб-сайту може їх підтвердити або відхилити. Метод Users надає доступ до перегляду інформації про інших

зареєстрованих користувачів з можливістю їх блокування. Адміністратор має можливість блокувати не тільки користувачів, а ще й обрані враження та фільми.

Таким чином, розроблена структура повністю реалізовує основні необхідні функції веб-сервісу для обговорення кінофільмів.

3.2 Особливості реалізації складових системи

Система розроблена для трьох категорій користувачів: незареєстрований користувач, зареєстрований користувач та адміністратор.

Основними функціями незареєстрованого користувача є: перегляд переліку кінофільмів на обговоренні, перегляд докладного опису обраного кінофільму та подання заявки на реєстрацію в системі. Для кожної функції яка має бути реалізована було створено відповідні модулі: Index, Details, Register. Модуль Index відображає кінофільми які є на обговоренні (рисунок 3.2).

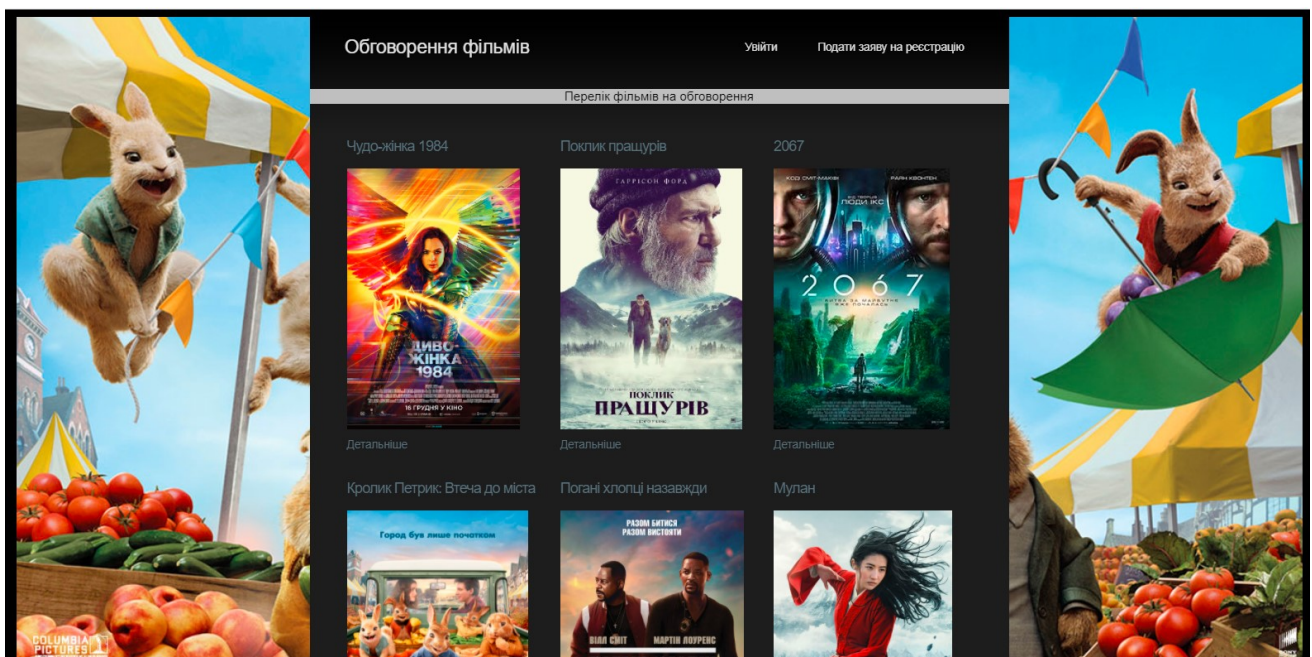


Рисунок 3.2 – Сторінка веб сайту

Передавання даних в представлення відбувається в контролері HomeController та реалізується за допомогою наступного програмного коду:

```
private FilmsEntities db = new FilmsEntities();
public ActionResult Index()
{
    return View(db.Фільми.ToList());
}
```

Наступним було реалізовано модуль Details, він відображає детальну інформацію про обраний на головній сторінці фільм. Наприклад, було обрано фільм «Поклик пращурів», для цього необхідно було натиснути кнопку «Детальніше» під зображенням фільму, результат зображено на рисунку 3.3.

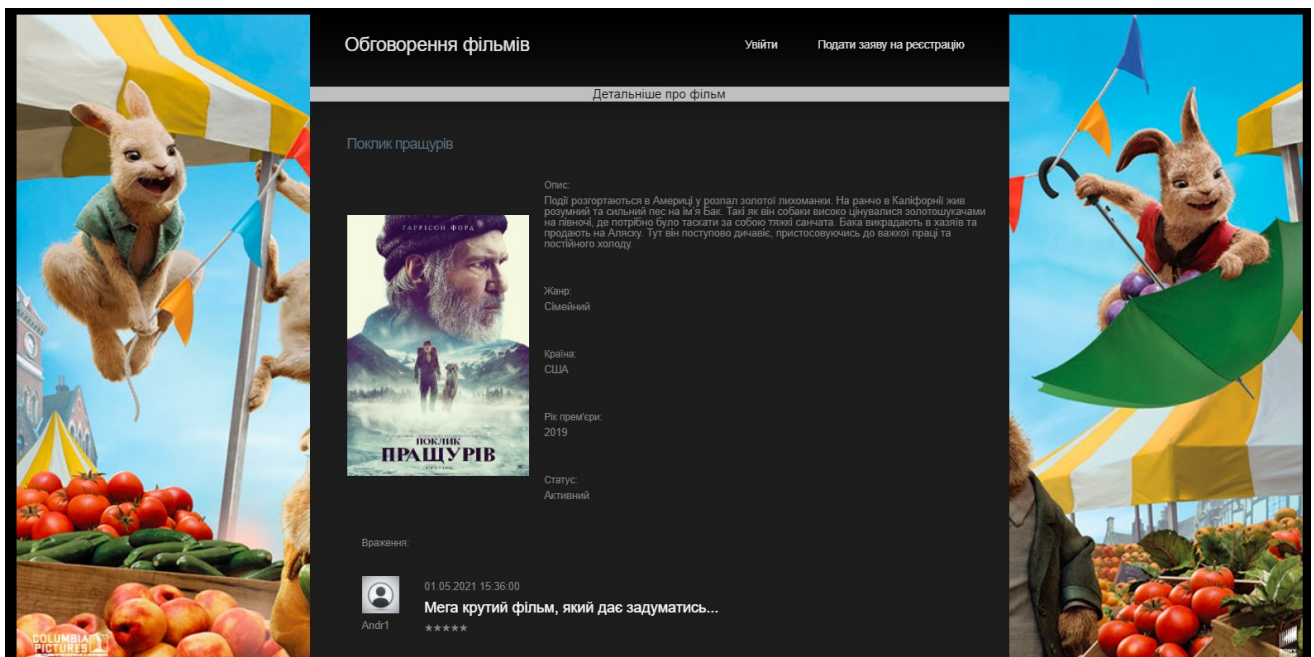


Рисунок 3.3 – Сторінка детальніше про фільм

Робота з відбором даних з БД відбувається за допомогою коду:

```
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Фільми films = db.Фільми.Find(id);
    if (films == null)
    {
        return HttpNotFound();
    }
    ViewData["Враження"] = db.Враження.ToList();
    return View(films);
}
```

Модуль Register надає можливість незареєстрованому користувачу подавати заяву на реєстрацію, яка потім потрапляє до адміністратора. Форма заяви на реєстрацію зображена на рисунку 3.4.

The screenshot shows a web interface for a movie review site. The main header is 'Обговорення фільмів' with a 'Вийти' (Logout) link on the right. Below the header is a sub-header 'Заява на реєстрації у веб-сервісі'. The form contains four input fields: 'Логін' (Login), 'Пароль' (Password), 'Нік' (Nickname), and 'Емейл' (Email). A 'Подати заяву' (Submit) button is located at the bottom of the form. The page is flanked by two images of a rabbit character holding flags. A small copyright notice '© 2011 - Обговорення фільмів' is visible in the bottom left corner.

Рисунок 3.4 – Форма заяви на реєстрацію у веб-сервісі

Для зареєстрованого користувача було реалізовано такі основні модулі: Create, Search та Users. Модуль Create надає користувачу можливість додавати новий фільм до обговорення. Форма додавання нового фільму зображена на рисунку 3.5.

The screenshot shows a web interface for adding a new movie. The main header is 'Обговорення фільмів' with 'Пошук' (Search) and 'Вийти' (Logout) links on the right. Below the header is a sub-header 'Додавання нового фільму'. The form contains five input fields: 'Назва' (Name), 'Постер' (Poster) with a file selection button 'Виберіть файл' and 'Файл ... вибран', 'Жанр' (Genre), 'Країна' (Country), and 'Рік прем'єри' (Release Year). A 'Додати фільм' (Add movie) button is located at the bottom of the form. The page is flanked by two images of a rabbit character holding flags.

Рисунок 3.5 – Форма додавання нового фільму

Передавання даних з форми заяви на сторінку адміністратора, а після підтвердження в базу даних системи відбувається за допомогою наступного програмного коду:

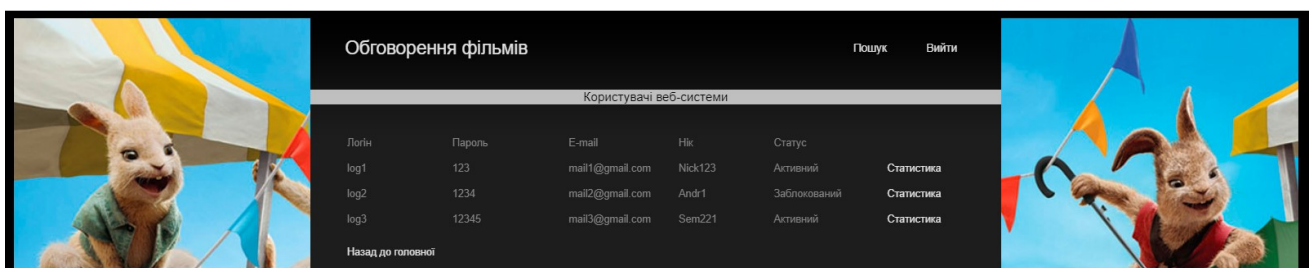
```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Register(UserLoginRegister.UserRegister model)
{
    if (ModelState.IsValid)
    {
        User user = null;
        using (UserContext db = new UserContext())
        {
            user = db.Users.FirstOrDefault(u => u.Login == model.Login);
        }
        if (user == null)
        {
            using (UserContext db = new UserContext())
            {
                db.Users.Add(new User { Login = model.Login, Password =
model.Password, Nick = model.Nick });
                db.SaveChanges();
                user = db.Users.Where(u => u.Login == model.Login &&
u.Password == model.Password).FirstOrDefault();
            }
            if (user != null)
            {
                FormsAuthentication.SetAuthCookie(model.Login, true);
                return RedirectToAction("Index", "Home");
            }
        }
        else
        {
            ModelState.AddModelError("", "Користувача з таким логіном вже
існує");
        }
    }

    return View(model);
}

```

Модуль Search надає користувачу можливість виконувати пошук по кінофільмах. Зокрема, виконувати пошук по назві, жанрах, року прем'єри та країні. Модуль Users виводить дані про усіх користувачів окрім авторизованого, також можна переглянути статистику кожного з користувачів. Сторінка з даними про користувачів зображена на рисунку 3.6.



| Користувачі веб-системи | | | | | |
|-------------------------|--------|-----------------|---------|--------------|------------|
| Логін | Пароль | E-mail | Нік | Статус | Статистика |
| log1 | 123 | mail1@gmail.com | Nick123 | Активний | Статистика |
| log2 | 1234 | mail2@gmail.com | Andr1 | Заблокований | Статистика |
| log3 | 12345 | mail3@gmail.com | Sem221 | Активний | Статистика |

Назад до головної

Рисунок 3.6 – Користувачі веб-сервісу

Аналогічним чином реалізовані функції адміністратора системи, з можливістю блокування кінофільмів, вражень та користувачів. Таким чином, реалізовані модулі виконують функції які були поставлені в завданні.

3.3 Тестування інформаційної системи

Для перевірки коректності роботи програми та виконання необхідних функцій було розроблено 3 тестових випадки (тест-кейси).

Перший тестовий випадок (таблиця 3.1) перевіряє вивід переліку фільмів для незареєстрованого користувача. Після запуску веб-сервісу має відобразитись головна сторінка сайту на якій буде відображений перелік фільмів (рисунок 3.7), які є на обговоренні (вони зберігаються в базі даних в окремій таблиці).

Таблиця 3.1 – Тест-кейс КО0001

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|---------------------------------------------|
| Тест-кейс ID: КО0001 | Пріоритет: 1 | Створено: 04.06.2021, І.Корольчук |
| Назва: Перевірка відображення переліку фільмів на обговоренні Вхідні дані: Перелік фільмів з бази даних | | |
| Кроки | Очікуваний результат | |
| <ol style="list-style-type: none"> 1. Запустити веб-сервіс 2. Переглянути головну сторінку сайту 3. Порівняти фактичний результат з очікуваним | Відображення фільмів, які є на обговоренні з бази даних | |
| Результат виконання тест-кейсу: пройдено успішно | | |

Другий тестовий випадок (таблиця 3.2) перевіряє вивід детальної інформації про обраний кінофільм. Після обрання кінофільму користувачем та натискання кнопки «Детальніше» під фільмом, має відкритись сторінка (рисунок 3.8) з детальним описом кінофільму (опис, рік прем'єри, жанр, країна, враження які залишали користувачі про фільм).

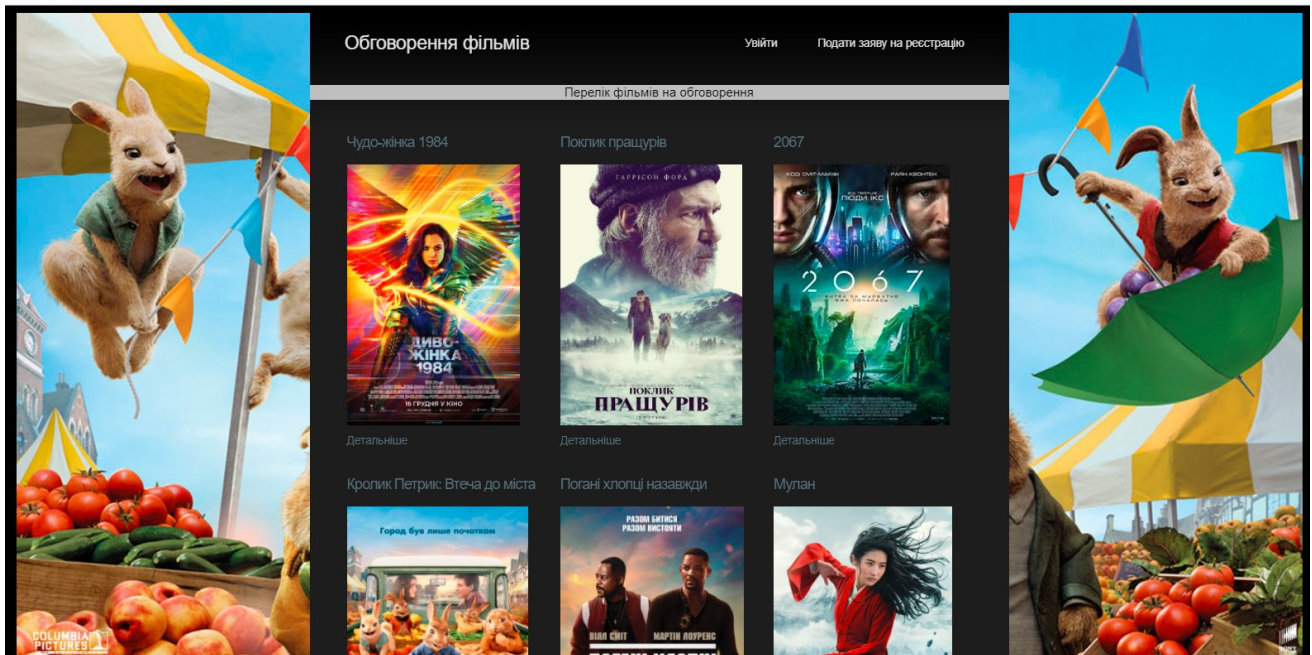


Рисунок 3.7 – Перелік фільмів на обговоренні

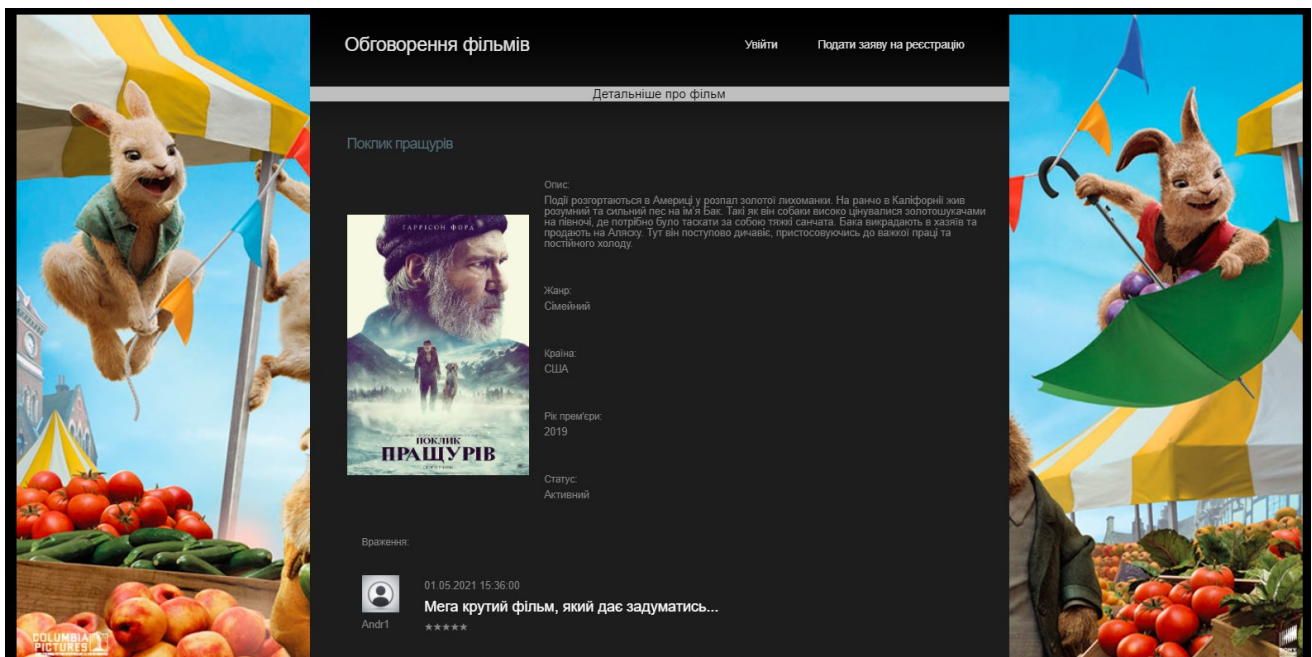


Рисунок 3.8 – Сторінка детальніше про фільм

Третім тестовим випадком (таблиця 3.3) перевіряється функція додавання фільму зареєстрованим користувачем на веб-сервіс. Користувач має авторизуватись, далі перейти на форму додавання кінофільму (рисунок 3.9), натиснувши кнопку «Додати кінофільм», заповнити форму додавання та натиснути кнопку «Додати фільм». Після виконаних маніпуляцій користувач

перейде на головну сторінку де буде додано вже новий щойно введений фільм (рисунок 3.10).

Таблиця 3.2 – Тест-кейс КО0002

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|---------------------------------------------|
| Тест-кейс ID: КО0002 | Пріоритет: 1 | Створено: 04.06.2021, І.Корольчук |
| Назва: Перевірка відображення детальної інформації про фільм Вхідні дані: Кінофільм = «Поклик пращурів» | | |
| Кроки | Очікуваний результат | |
| <ol style="list-style-type: none"> 1. Запустити веб-сервіс 2. Обрати кінофільм «Поклик пращурів» 3. Натиснути кнопку «Детальніше» 4. Порівняти фактичний результат з очікуваним | Відображення детальної інформації про обраний фільм | |
| Результат виконання тест-кейсу: пройдено успішно | | |

Таблиця 3.3 – Тест-кейс КО0003

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|---------------------------------------------|
| Тест-кейс ID: КО0003 | Пріоритет: 2 | Створено: 04.06.2021, І.Корольчук |
| Назва: Перевірка додавання нового кінофільму Вхідні дані: Назва = «Форсаж 9: нестримна сага», постер = «film7.jpg», жанр = «бойовик», країна = «США», рік прем'єри = «2021», логін = «log1», пароль = «123» | | |
| Кроки | Очікуваний результат | |
| <ol style="list-style-type: none"> 1. Запустити веб-сервіс 2. Авторизуватись ввівши логін = «log1» та пароль = «123» 3. Натиснути кнопку «Ввійти» 4. Натиснути кнопку «Додати кінофільм» 5. Ввести дані про фільм в відповідні поля назва = «Форсаж 9: нестримна сага», постер = «film7.jpg», жанр = «бойовик», країна = «США», рік прем'єри = «2021» 6. Натиснути кнопку «Додати фільм» 7. Порівняти фактичний результат з очікуваним | Відображення доданого фільму на головній сторінці | |
| Результат виконання тест-кейсу: пройдено успішно | | |

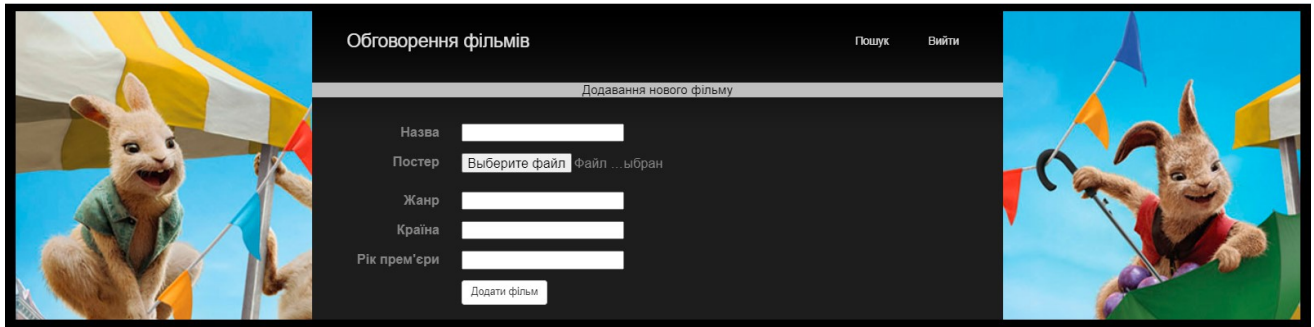


Рисунок 3.9 – Форма додавання фільму

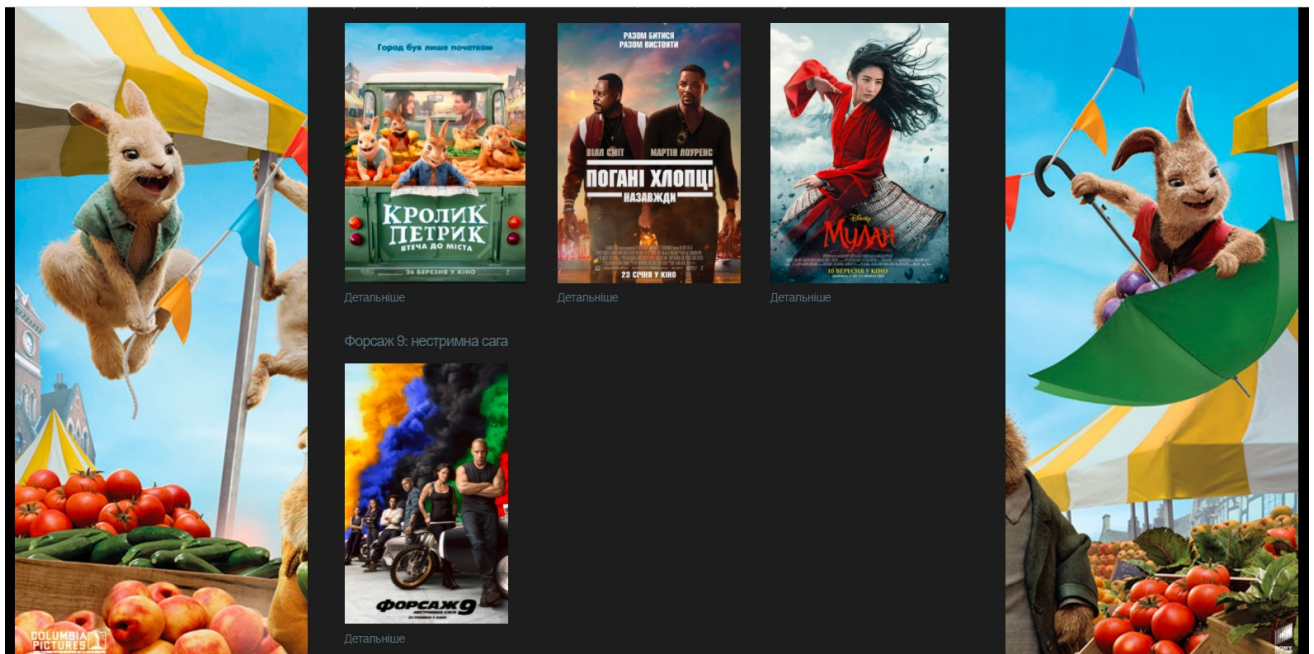


Рисунок 3.10 – Головна сторінка з доданим фільмом

Отже, в результаті тестування дійшли висновку що функції системи працюють коректно, а саме функції: перегляд переліку кінофільмів на обговоренні, пошук кінофільму за назвою, роком прем'єри, жанром чи країною, перегляд докладного опису обраного кінофільму та вражень інших користувачів, долучення враження користувача про обраний кінофільм, оцінка обраного кінофільму, оцінка обраного враження іншого користувача, додавання нового кінофільму до веб-сервісу, перегляд переліку користувачів веб-сервісу та їх статистики.

3.4 Інструкція користувача

Перейшовши на сайт ще незареєстрованому користувачу відкривається початкова сторінка (рисунок 3.11).

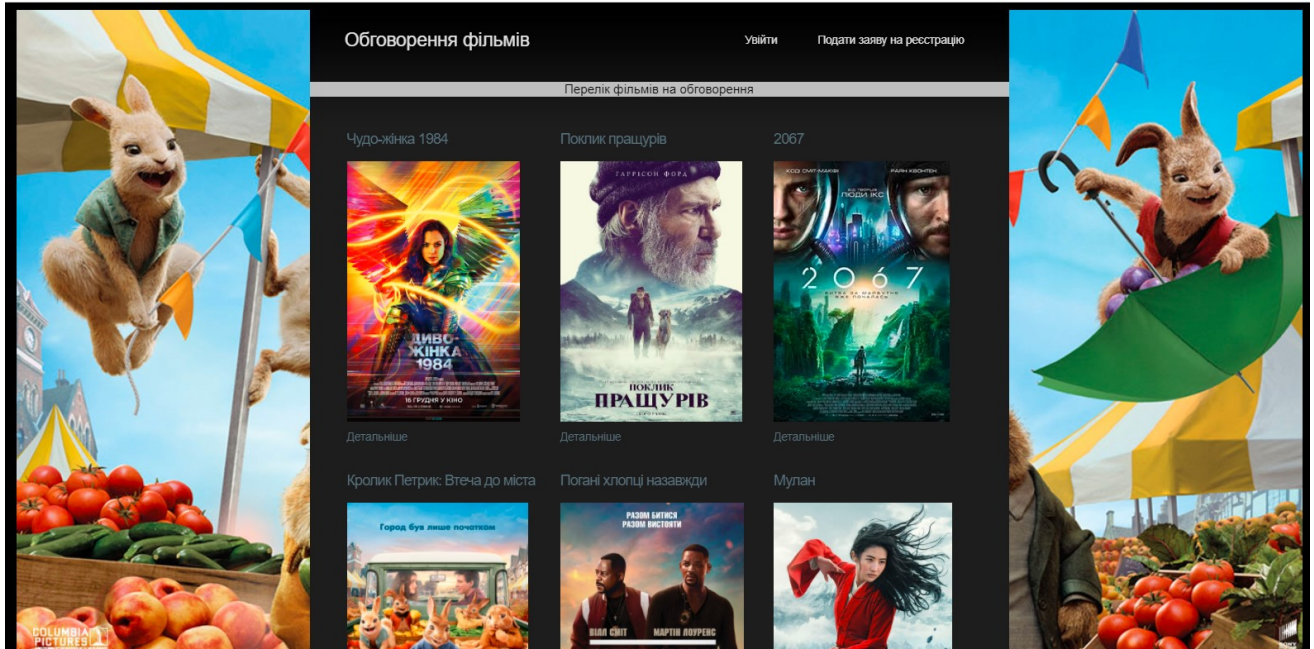


Рисунок 3.11 – Стартує сторінка веб-сайту

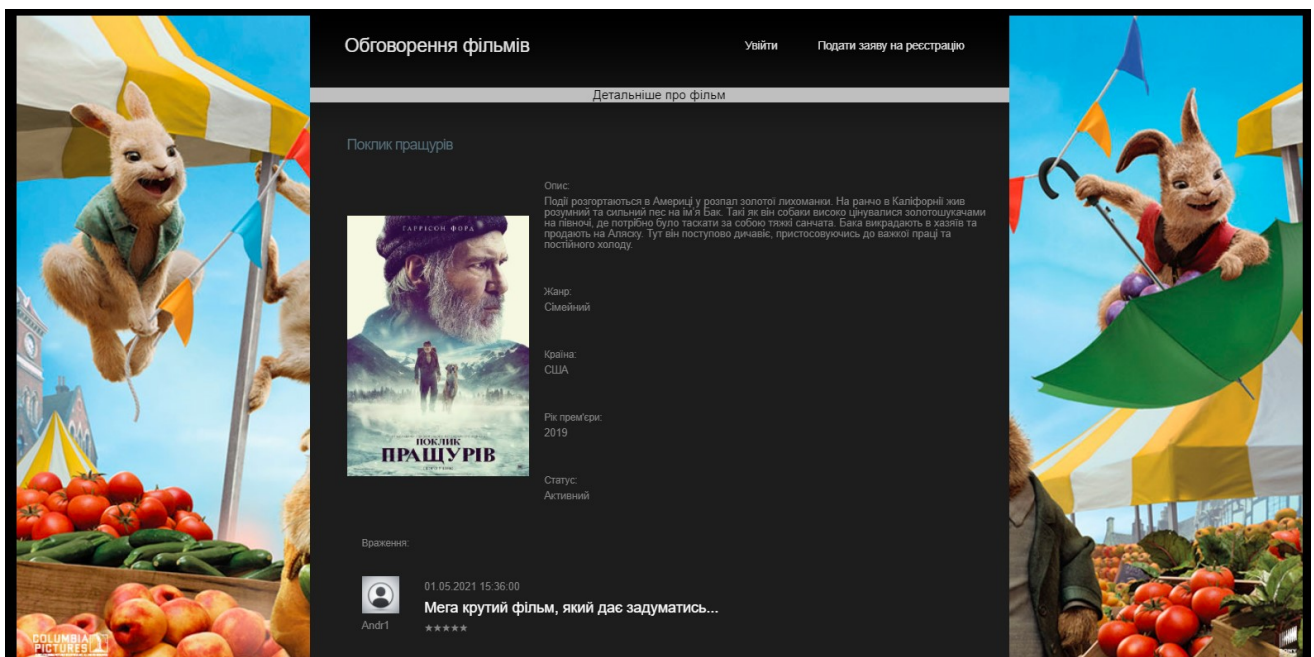


Рисунок 3.12 – Сторінка детальніше про фільм

Користувач може переглянути фільми, які зараз є на обговоренні на сайті, також, натиснувши на одну з кнопок «Детальніше», переглянути детальну інформацію про обраний фільм. Наприклад, було обрано фільм «Поклик пращурів», результат детального опису зображено на рисунку 3.12.

Незарєєстрований користувач може подати заяву на реєстрацію, яка потім надсилається адміністратору. Щоб надіслати заяву користувачу необхідно натиснути відповідну кнопку, що розташована у верхньому правому куті, та заповнити власними даними поля, які необхідні для реєстрації (рисунок 3.13).

Рисунок 3.13 – Форма заяви на реєстрацію у веб-сервісі

Зареєстрований користувач може виконати авторизацію, натиснувши кнопку «Увійти», яка знаходиться на головній сторінці сайту у верхньому правому куті. Для входу необхідно ввести логін та пароль користувача, якщо такий користувач є в базі даних, то йому відкриється початкова сторінка, проте вже з іншим інтерфейсом (рисунок 3.14).

В зарєєстрованого користувача з'являються нові можливості, такі як: можливість додавати кінофільм, можливість переглядати користувачів сайту та їх статистику, можливість додавати враження та виконувати пошук по назві фільму, жанру, року прем'єри та країні.

Щоб додати кінофільм користувачу необхідно натиснути на кнопку «Додати кінофільм», яка знаходиться на головній сторінці, та ввести всі необхідні дані в форму, що зображена на рисунку 3.15.

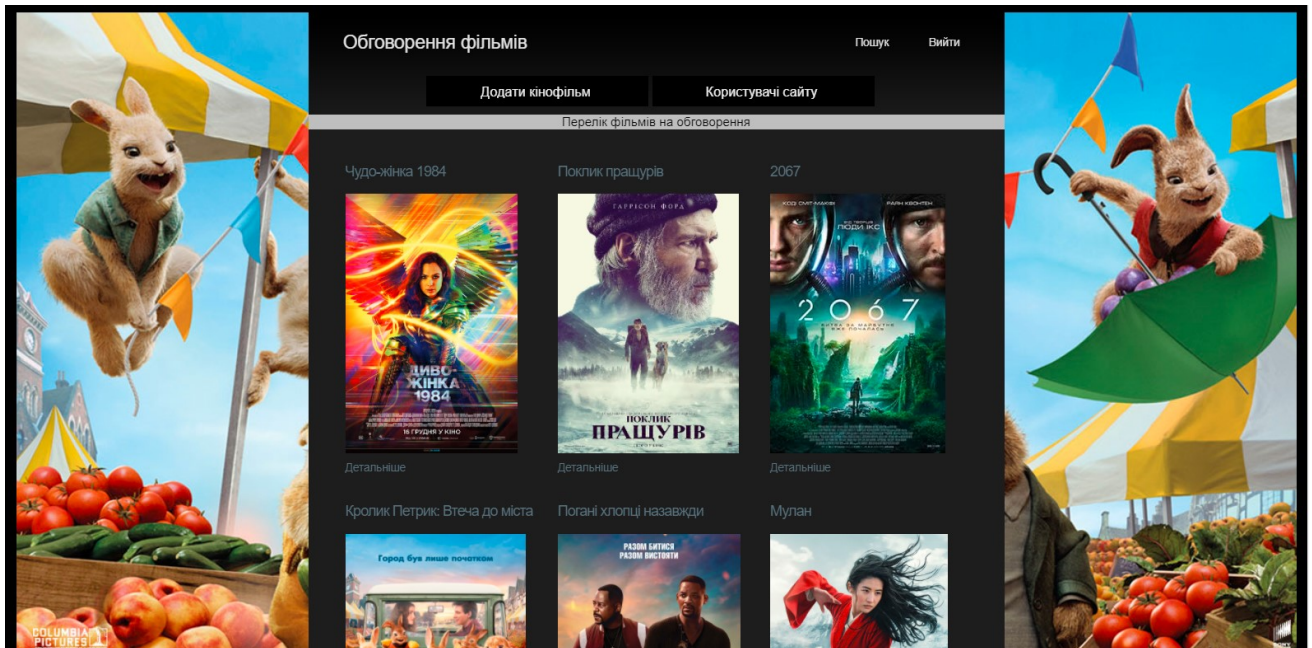


Рисунок 3.14 – Стартова сторінка зареєстрованого користувача



Рисунок 3.15 – Форма додавання фільму

Для перегляду інформації про інших зареєстрованих користувачів, авторизованому користувачу необхідно натиснути на кнопку «Користувачі веб-сервісу», яка знаходиться на головній сторінці сайту. Перелік користувачів з їх даними зображено на рисунку 3.16.

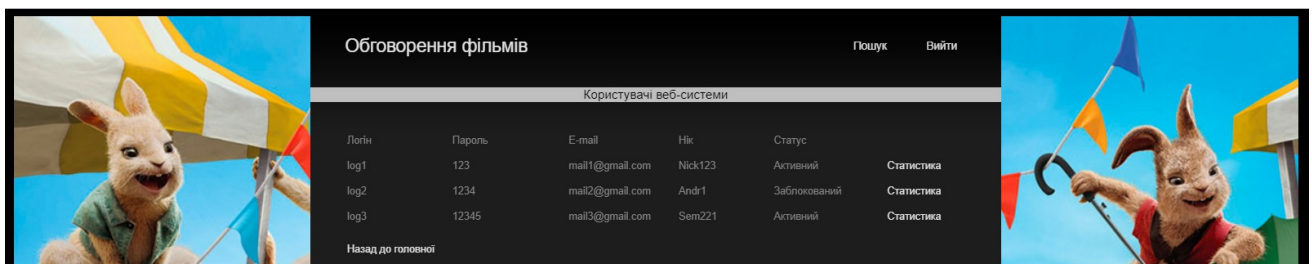


Рисунок 3.16 – Користувачі веб-сервісу

Таким чином, розроблена веб-система реалізує наступні функції: перегляд переліку кінофільмів на обговоренні, пошук кінофільму за назвою, роком прем'єри, жанром чи країною, перегляд докладного опису обраного кінофільму та вражень інших користувачів, долучення враження користувача про обраний кінофільм, оцінка обраного кінофільму, оцінка обраного враження іншого користувача, додавання нового кінофільму до веб-сервісу, перегляд переліку користувачів веб-сервісу та їх статистики.

3.5 Вимоги до розгортання інформаційної системи

Вимоги до браузера:

- підтримка CSS 3;
- підтримка HTML 5.

Мінімальні вимоги до провайдера:

- використання серверу IIS;
- підтримка фреймворку ASP.NET MVC 5.

Вимоги до СКБД:

- СКБД MS SQL SERVER 2012.

Мінімальні вимоги до користувача:

- підключення до Інтернету;
- використання пристроїв з великим розширенням екрану (не підтримується мобільна версія);
- не підтримується Internet Explorer та інші застарілі браузери або старі версії сучасних браузерів.

Висновки

В результаті виконання кваліфікаційної роботи бакалавра було розроблено веб-сервіс для обговорення кінофільмів на платформі ASP.NET. Для створення програмного продукту було використано такий набір засобів розробки: мову програмування C#, фреймворк MVC 5, редактор Microsoft Visual Studio 2019 Professional та СКБД SQL Server 2012.

Розроблена система виконує наступні основні функції для незареєстрованого користувача:

- перегляд переліку кінофільмів на обговоренні;
- перегляд докладного опису обраного кінофільму;
- подання заявки на реєстрацію в системі.

Для зареєстрованого користувача система виконує такі основні функції:

- авторизація в системі;
- перегляд переліку кінофільмів на обговоренні;
- пошук кінофільму за назвою, роком прем'єри, жанром чи країною;
- перегляд докладного опису обраного кінофільму та вражень інших користувачів;

- долучення власного враження про обраний кінофільм;
- оцінка обраного кінофільму;
- оцінка обраного враження іншого користувача;
- додавання нового кінофільму до веб-сервісу;
- перегляд переліку користувачів веб-сервісу та їх статистики.

Для адміністратора веб-сервісу система виконує функції:

- авторизація в системі;
- затвердження та відхилення заявок на реєстрацію у веб-сервісі;
- перегляд переліку кінофільмів на обговоренні з можливістю їх блокування;
- перегляд докладного опису обраного кінофільму та вражень інших користувачів з можливістю блокування обраних вражень;

- додавання нового кінофільму до веб-сервісу;
- перегляд переліку користувачів веб-сервісу з можливістю їх блокування.

Згідно проведеного тестування, створений веб-сервіс для обговорення кінофільмів відповідає поставленому завданню і виконує усі функції, які необхідно було реалізувати. ІС може бути впроваджена в кінотеатри, сервіси обговорення, та ін.

Перелік посилань

1. Коротка історія кінематографа. URL: <http://ispu.ru/book/export/html/7631>.
2. Коротка історія кінофестивалів. URL: <https://mirfest.com/articles/item/314.html>.
3. 10 популярних сервісів для перегляду фільмів: обираємо найкращий. URL: https://yamobi.ru/posts/best_videoservice.html.
4. Кінематограф. URL: <https://ru.wikipedia.org/wiki/Кинематограф>.
5. Історія кінематографа. URL: <http://ispu.ru/node/7631>.
6. Цікаві факти про німе кіно. URL: <https://ok-vmeste.ru/articles/1417-interesnye-fakty-o-nemom-kino.html>.
7. Що таке рейтинг МРАА? URL: <https://tvkinoradio.ru/news/new43-chtotakoe-rejting-mraa>.
8. 250 кращих фільмів по версії IMDb. URL: https://ru.wikipedia.org/wiki/250_лучших_фильмов_по_версии_IMDb.
9. Кращі фільми по ivi.рейтингу. URL: <https://www.ivi.ru/rating>.
10. Кіноафіша. URL: <https://kinoafisha.ua/ratings/>.
11. Куди поїхати на відпочинок кінофанату, щоб побачити місця, де знімалися відомі фільми. URL: <https://novate.ru/blogs/tags/%EA%E8%ED%EE%F4%E0%ED%E0%F2%FB/>.
12. Останній переглянутий фільм v.9. URL: https://forums.playground.ru/movies/poslednij_prosmotrennyj_film_v_9-961665/.
13. Група любителів кіно. URL: <https://www.facebook.com/groups/284071992200398>.
14. Розробка десктопних додатків. URL: <http://mongo-software.ru/desktop>.

15. WEB-додаток чи мобільний додаток. Що обрати? URL: <https://punicapp.com/blog/pages/893/web-prilozhenie-ili-mobilnoe-prilozhenie-cto-vybrat>.

16. Web application. URL: https://en.wikipedia.org/wiki/Web_application.

17. Кращі PHP-фреймворки, які спрощують розробку в 2021 році. URL: <https://wezom.com.ua/blog/luchshie-php-frejmvoriki-kotorye-uproshchayut-razrabotku-v-2019-godu>.

18. Що представляє собою PHP 7? Можливості для програмістів. URL: <https://oracle-patches.com/web/что-представляет-собой-php-7-возможности-для-программиста>.

19. Сайти на Java: плюси та мінуси. URL: <https://landing.ua/blog/sajty-na-java-plyusy-i-minusy.html>.

20. .NET Framework и ASP.NET — платформи для веб-розробки. URL: https://web-creator.ru/articles/dot_net_and_asp.

21. Programming.NET. URL: <http://sharp-generation.narod.ru/ArchitectureNet/Glava1/netfrmwrk.html>

22. Введення в VB.NET. URL: <https://metanit.com/visualbasic/tutorial/1.1.php>.

23. Переваги, недоліки і критика Visual Basic for Application. URL: https://studbooks.net/2220661/informatika/dostoinstva_nedostatki_kritika_visual_basi_c_application.

24. Мова JavaScript та її можливості. URL: <https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti>.

25. C++. URL: <https://uk.wikipedia.org/wiki/C%2B%2B>.

26. C Sharp. URL: https://uk.wikipedia.org/wiki/C_Sharp

27. Особливості ASP.NET MVC. Що нового в MVC 5. URL: <https://metanit.com/sharp/mvc5/1.1.php>.

28. ASP.NET значна технологія для веб творчості. URL: <https://www.interestprograms.ru/source-codes-asp-net-technology>.

29. MySQL проти MSSQL. URL: <https://www.dnsstuff.com/mysql-vs-mssql-performance>

ДОДАТКИ

Додаток А

Програмні коди основних модулів веб-сервісу для обговорення кінофільмів

Лістинг HomeController.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Data.Entity;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Film.Models;
using System.Web.Security;

namespace Film.Controllers
{
    public class HomeController : Controller
    {
        private FilmsEntities db = new FilmsEntities();
        public ActionResult Index()
        {
            return View(db.Фільми.ToList());
        }

        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Фільми films = db.Фільми.Find(id);

            if (films == null)
            {
                return HttpNotFound();
            }

            ViewData["Враження"] = db.Враження.ToList();
            return View(films);
        }

        public ActionResult Register()
        {
            return View();
        }

        public ActionResult Login()
        {
            return View();
        }
    }
}
```

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Login(UserLoginRegister.UserLogin model)
{
    if (ModelState.IsValid)
    {
        // поиск пользователя в бд
        User user = null;
        using (UserContext db = new UserContext())
        {
            user = db.Users.FirstOrDefault(u => u.Login == model.Login && u.Password ==
model.Password);
        }
        if (user != null)
        {
            FormsAuthentication.SetAuthCookie(model.Login, true);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            ModelState.AddModelError("", "Користувача з таким логіном та паролем немає");
        }
    }

    return View(model);
}

```

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Register(UserLoginRegister.UserRegister model)
{
    if (ModelState.IsValid)
    {
        User user = null;
        using (UserContext db = new UserContext())
        {
            user = db.Users.FirstOrDefault(u => u.Login == model.Login);
        }
        if (user == null)
        {
            using (UserContext db = new UserContext())
            {
                db.Users.Add(new User { Login = model.Login, Password = model.Password,
Nick = model.Nick });
                db.SaveChanges();

                user = db.Users.Where(u => u.Login == model.Login && u.Password ==
model.Password).FirstOrDefault();
            }
            if (user != null)
            {
                FormsAuthentication.SetAuthCookie(model.Login, true);
                return RedirectToAction("Index", "Home");
            }
        }
        else
        {
            ModelState.AddModelError("", "Користувача з таким логіном вже існує");
        }
    }
}

```

```

        }
    }

    return View(model);
}
public ActionResult Logoff()
{
    FormsAuthentication.SignOut();
    return RedirectToAction("Index", "Home");
}
}
}

```

Лістинг Index.cshtml:

```

@model IEnumerable<Film.Models.Фільми>

@{

}

<head>
    <title>Home - Home Page | Cinema - Free Website Template from Templates.com</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="description" content="Place your description here" />
    <meta name="keywords" content="put, your, keyword, here" />
    <meta name="author" content="Templates.com - website templates provider" />
    <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />
    <script src="js/jquery-1.4.2.min.js" type="text/javascript"></script>
    <script src="js/cufon-yui.js" type="text/javascript"></script>
    <script src="js/cufon-replace.js" type="text/javascript"></script>
    <script src="js/Gill_Sans_400.font.js" type="text/javascript"></script>
    <script src="js/script.js" type="text/javascript"></script>
</head>

<body id="page1">
    <div class="tail-top">
        <div class="tail-bottom">
            <div id="main">

                <div id="header">
                    <div class="row-1">
                        <div class="fleft">@Html.ActionLink("Обговорення фільмів", "Index",
"Home")<span></span></div>

                            <ul>
                                <if (User.Identity.IsAuthenticated)
                                {
                                    <li> @Html.ActionLink("logout...", "Logoff")</li>
                                }
                                else
                                {
                                    @*<li>@Html.ActionLink("Пошук", "Index", "Home") </li>*@
                                    <li> <a href="/User/Search/Поклик пращурів/1">
                                        Пошук
                                    </a></li>
                                    <li>@Html.ActionLink("Вийти", "Index", "Home") </li>
                                }
                            </ul>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

    }

    </ul>
  </div>
  <div class="row-2">
    <ul>
      <li>@Html.ActionLink("Додати кінофільм", "Create", "User") </li>
      <li>@Html.ActionLink("Користувачі сайту", "Index", "Home") </li>

    </ul>
  </div>
</div>
<!-- CONTENT -->
<div id="content">

  <div class="box">
    <div class="border-right">
      <div class="border-left">

        <h3>Перелік фільмів на обговорення</h3>

      </div>
    </div>
  </div>
  <div class="content">

    <ul class="movies">
      @foreach (var film in Model)
      {

        <li>

          <h4>@film.Назва</h4>
          
          @*<div
class="wrapper"><span><span>@Html.ActionLink("Детальніше", "Details", new { id = film.ІД
})</span></span></div>*@
          <div>
            <a class="det" href="/Home/Details/@film.ІД">
              Детальніше
            </a>
          </div>
        </li>
      }

      <li class="clear">&nbsp;</li>
    </ul>
  </div>
</div>

  </div>
</div>
</div>
<script type="text/javascript">Cufon.now();</script>
</body>

```

Лістинг Create.cshtml:

```

@model Film.Models.Фільми

@{

}

<body id="page1">
  <div class="tail-top">
    <div class="tail-bottom">
      <div id="main">

        <div id="header">
          <div class="row-1">
            <div class="fleft">@Html.ActionLink("Обговорення фільмів", "Index",
"Home")<span></span></div>

            <ul>
              <li>
                <a href="/User/Search/Поклик пращурів/1">
                  Пошук
                </a>
              </li>
              <li>@Html.ActionLink("Вийти", "Index", "Home") </li>
            </ul>
          </div>

        </div>
        <!-- CONTENT -->
        <div id="content">

          <div class="box">
            <div class="border-right">
              <div class="border-left">

                <h3>Додавання нового фільму</h3>

              </div>
            </div>
          </div>
        </div>
        <div class="content">

          @using (Html.BeginForm())
          {
            @Html.AntiForgeryToken()

            <div class="form-horizontal">
              @Html.ValidationSummary(true)

              <div class="form-group">
                <label class="control-label col-md-2">Назва</label>
                <div class="col-md-10">
                  @Html.EditorFor(model => Model.Назва)
                  @Html.ValidationMessageFor(model => Model.Назва)
                </div>
              </div>
            </div>
          }
        </div>
      </div>
    </div>
  </div>
</body>

```

```

</div>

<div class="form-group">
    <label class="control-label col-md-2">Постер</label>
    <div class="col-md-10">
        <input type="file" name="uploadFile" id="uploadFile"><br />
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Жанр</label>
    <div class="col-md-10">
        @Html.EditorFor(model => Model.Жанри.Назва)
        @Html.ValidationMessageFor(model => Model.Жанри.Назва)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Країна</label>
    <div class="col-md-10">
        @Html.EditorFor(model => Model.Країни.Назва)
        @Html.ValidationMessageFor(model => Model.Країни.Назва)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Рік прем'єри</label>
    <div class="col-md-10">
        @Html.EditorFor(model => Model.Рік_премєри)
        @Html.ValidationMessageFor(model => Model.Рік_премєри)
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Додати фільм" class="btn btn-
default" />
    </div>
</div>
</div>
}
</div>
</div>

</div>
</div>
</div>
<script type="text/javascript">Cufon.now();</script>

</body>

```

Лістинг Details.cshtml:

```
@model Film.Models.Фільми
```

```

@{
    ViewBag.Title = "Details";
    var vrazz = ViewData["Враження"] as List<Film.Models.Враження>;
}

<body id="page1">
    <div class="tail-top">
        <div class="tail-bottom">
            <div id="main">

                <div id="header">
                    <div class="row-1">
                        <div class="fleft">@Html.ActionLink("Обговорення фільмів", "Index",
"Home")<span></span></div>

                            <ul>

                                <li>@Html.ActionLink("Увійти", "Index", "Home") </li>
                                <li>@Html.ActionLink("Подати заяву на реєстрацію", "Index", "Home")
</li>

                            </ul>
                        </div>

                    </div>
                    <!-- CONTENT -->
                    <div id="content">

                        <div class="box">
                            <div class="border-right">
                                <div class="border-left">

                                    <h3>Детальніше про фільм</h3>

                                </div>
                            </div>
                        </div>
                        <div class="content">
                            <table>

                                <tr><td><h4> @Model.Назва</h4></td><td></td></tr>

                                <tr>
                                    <th rowspan="5"></th>

                                    <td class="row2_det">Опис: <h5>@Model.Опис</h5></td>
                                </tr>
                                <tr><td class="row2_det"> Жанр: <h5>@Model.Жанри.Назва</h5></td></tr>
                                <tr><td class="row2_det"> Країна:
<h5>@Model.Країни.Назва</h5></td></tr>
                                <tr><td class="row2_det"> Рік прем'єри:
<h5>@Model.Рік_премєри</h5></td></tr>
                                <tr><td class="row2_det"> Статус:
<h5>@Model.Статуси.Назва</h5></td></tr>

                            </table>

```

```

<table>
  <tr><td class="row2_det"> Враження: </td></tr>
  @foreach (var vr in vrazz)
  {
      if (vr.ФК_Фільм == Model.ІД)
      {
          <tr>

          <td class="row2_det">

              
              <h5>@vr.Користувачі.Нік </h5>
          </td>
          <td>

              <h5> @vr.Дата_час</h5>
              <h5 class="table-col">@vr.Текст</h5>
              <h5> @vr.Скрін</h5>

          </td>

          </tr> }
      }

      @*}*@
  </table>

  </div>
</div>
<div class="but1">

    @* @Html.ActionLink("Edit", "Edit", new { id = Model.ІД }) |*@
    @Html.ActionLink("Назад до переліку фільмів", "Index")
</div>

</div>
</div>
</div>
<script type="text/javascript">Cufon.now();</script>

</body>

```

Лістинг Register.cshtml:

```

@model Film.Models.Користувачі

@{
    ViewBag.Title = "Регистрация";
}

<body id="page1">
    <div class="tail-top">
        <div class="tail-bottom">
            <div id="main">

```

```

<div id="header">
  <div class="row-1">
    <div class="fleft">@Html.ActionLink("Обговорення фільмів", "Index",
"Home")<span></span></div>

    <ul>

      <li>@Html.ActionLink("Увійти", "Index", "Home") </li>

    </ul>
  </div>

</div>
<!-- CONTENT -->
<div id="content">

  <div class="box">
    <div class="border-right">
      <div class="border-left">

        <h3>Заява на реєстрації у веб-сервісі</h3>

      </div>
    </div>
  </div>
</div>
<div class="content">

  @using (Html.BeginForm())
  {
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
      @Html.ValidationSummary(true)

      <div class="form-group">
        @Html.LabelFor(model => Model.Логін, new { @class = "control-
label col-md-2" })

        <div class="col-md-10">
          @Html.EditorFor(model => Model.Логін)
          @Html.ValidationMessageFor(model => Model.Логін)
        </div>
      </div>

      <div class="form-group">
        @Html.LabelFor(model => Model.Пароль, new { @class = "control-
label col-md-2" })

        <div class="col-md-10">
          @Html.EditorFor(model => Model.Пароль)
          @Html.ValidationMessageFor(model => Model.Пароль)
        </div>
      </div>

      <div class="form-group">
        @Html.LabelFor(model => Model.Нік, new { @class = "control-
label col-md-2" })

        <div class="col-md-10">
          @Html.EditorFor(model => Model.Нік)
          @Html.ValidationMessageFor(model => Model.Нік)
        </div>
      </div>
    </div>
  }

```

```

</div>

<div class="form-group">
  @Html.LabelFor(model => Model.Емейл, new { @class = "control-
label col-md-2" })

  <div class="col-md-10">
    @Html.EditorFor(model => Model.Емейл)
    @Html.ValidationMessageFor(model => Model.Емейл)
  </div>
</div>

<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Подати заяву" class="btn btn-
default" />
  </div>
</div>
</div>
}
</div>
</div>

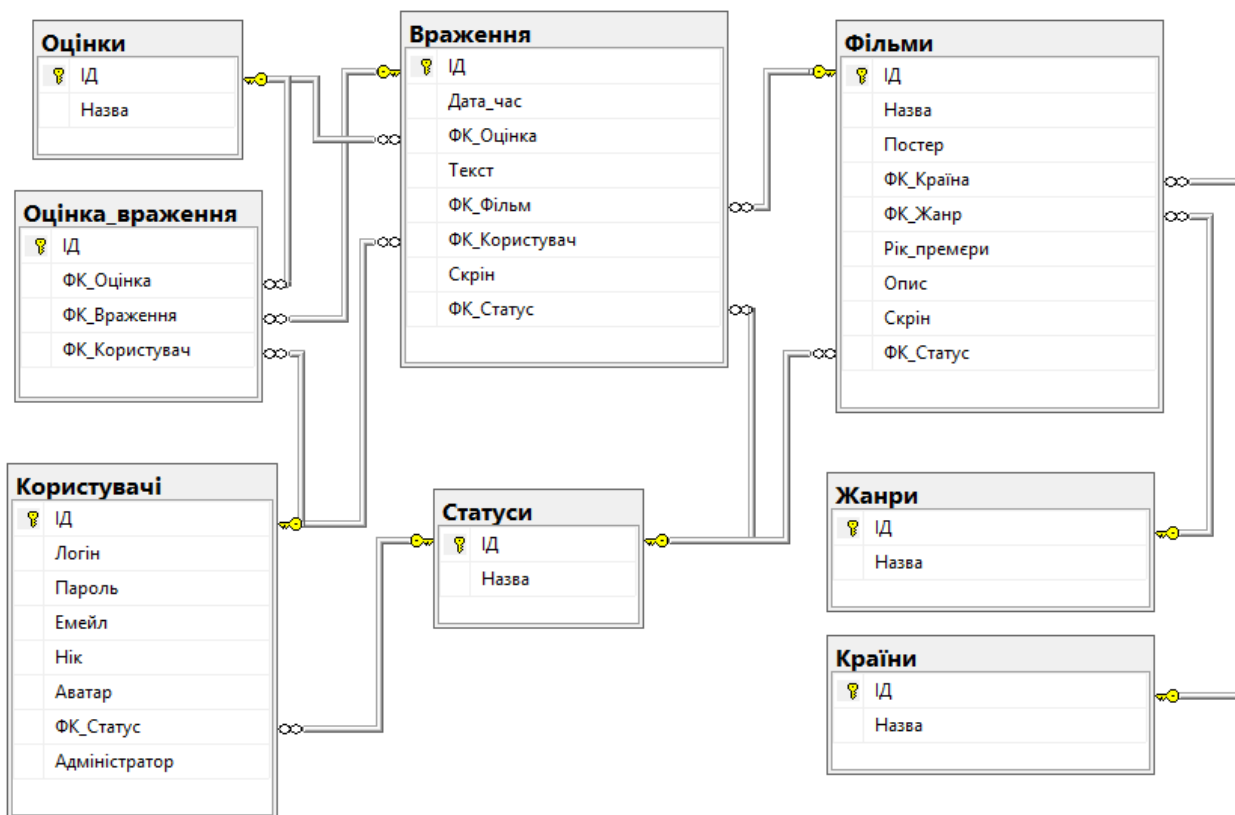
</div>
</div>
</div>
<script type="text/javascript">Cufon.now();</script>

</body>

```

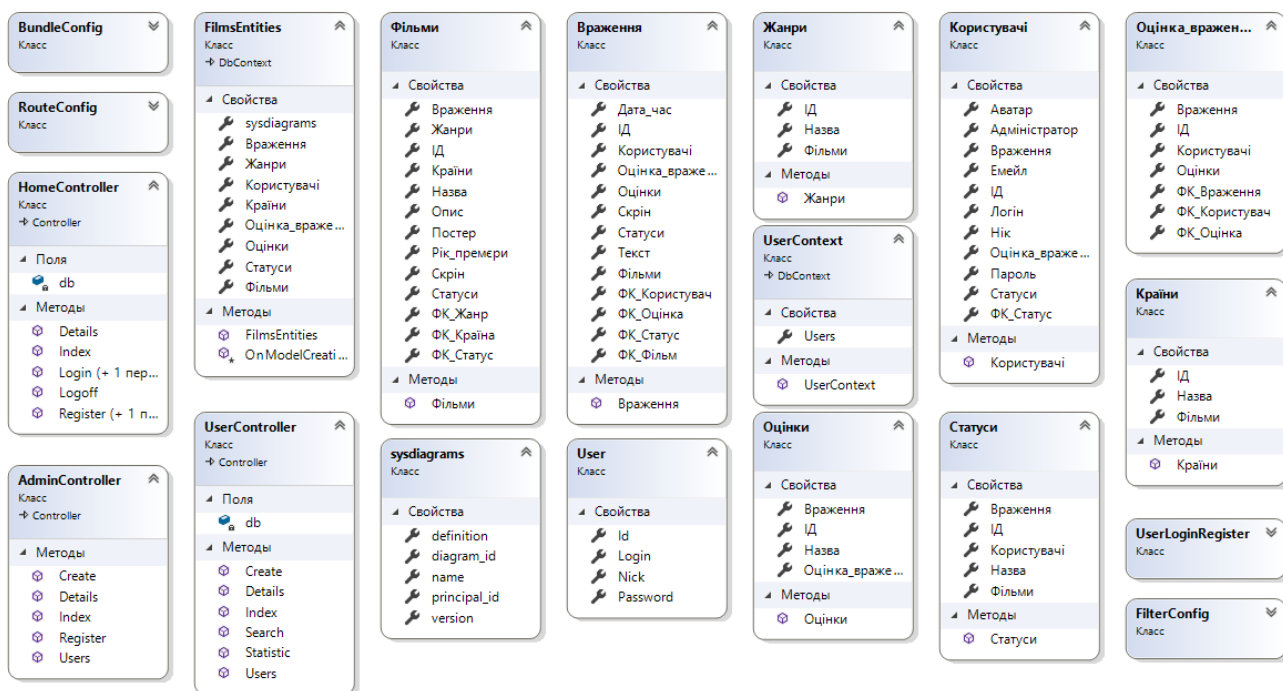
Додаток Б

Структура бази даних веб-сервісу для обговорення кінофільмів



Додаток В

Розгорнута структура класів інформаційної системи



Додаток Г

Презентаційний матеріал

Кваліфікаційна робота бакалавра

Веб-сервіс для обговорення кінофільмів

Виконав: студент 4 курсу групи КН-17-1 Корольчук І.Я.

Керівник: к.ф.-м.н., доцент кафедри КНІТ Міхалевський В.Ц.

Актуальність

Кількість об'єднань кінофанів у соціальних мережах зростає, а застосунків із повноцінним функціоналом і можливістю не тільки перегляду фільмів, а і їх обговорення практично немає. Тому робота у даному напрямку актуальна і потребує подальшого дослідження.



Завдання

Метою кваліфікаційної роботи бакалавра є розробка веб-сервісу для обговорення кінофільмів.

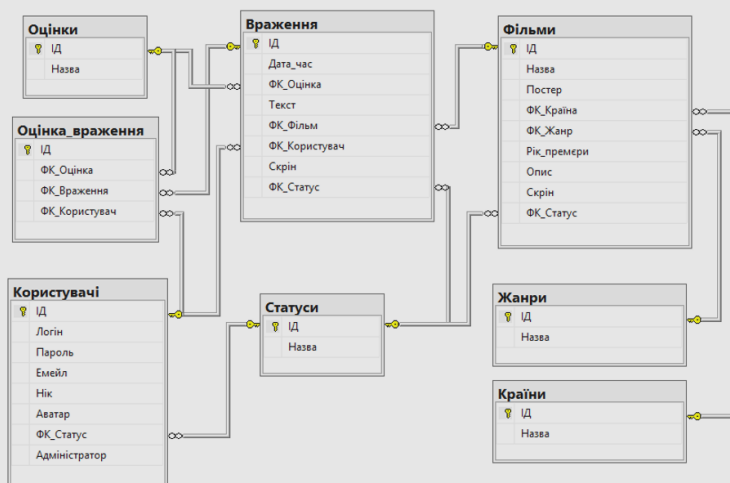
Автоматизувати наступні функції:

1. перегляд переліку кінофільмів на обговоренні
2. пошук кінофільму за назвою, роком прем'єри, жанром чи країною
3. перегляд докладного опису обраного кінофільму та вражень інших користувачів
4. долучення враження користувача про обраний кінофільм
5. оцінка обраного кінофільму
6. оцінка обраного враження іншого користувача
7. додавання нового кінофільму до веб-сервісу
8. перегляд переліку користувачів веб-сервісу та їх статистики.

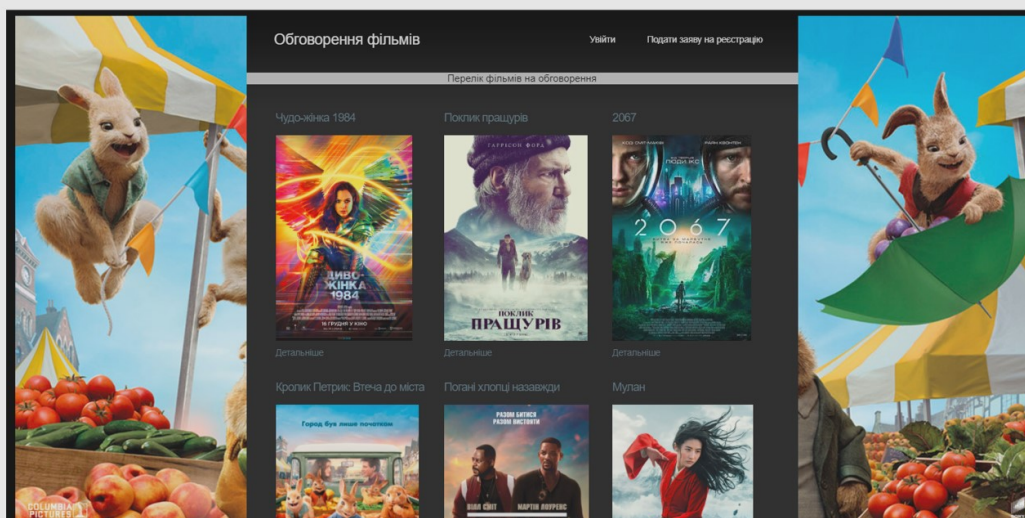
Діаграма класів веб-сервісу



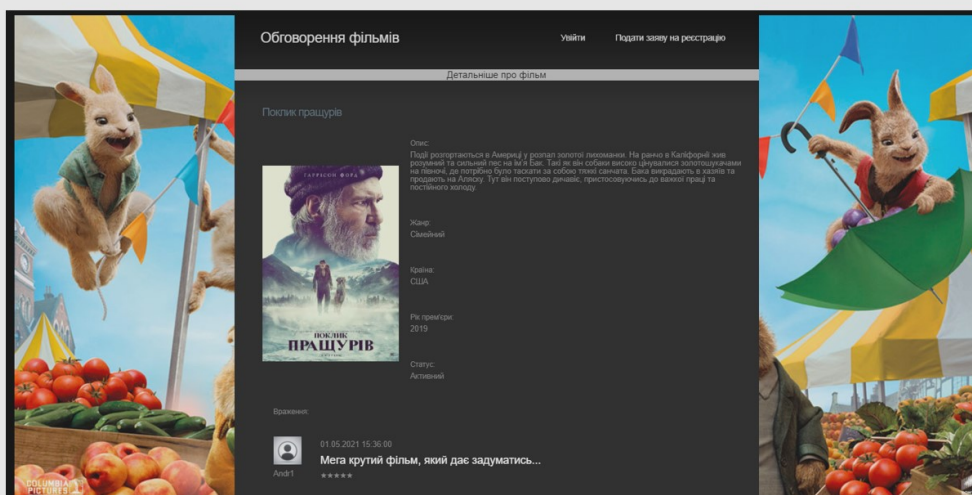
Схема бази даних



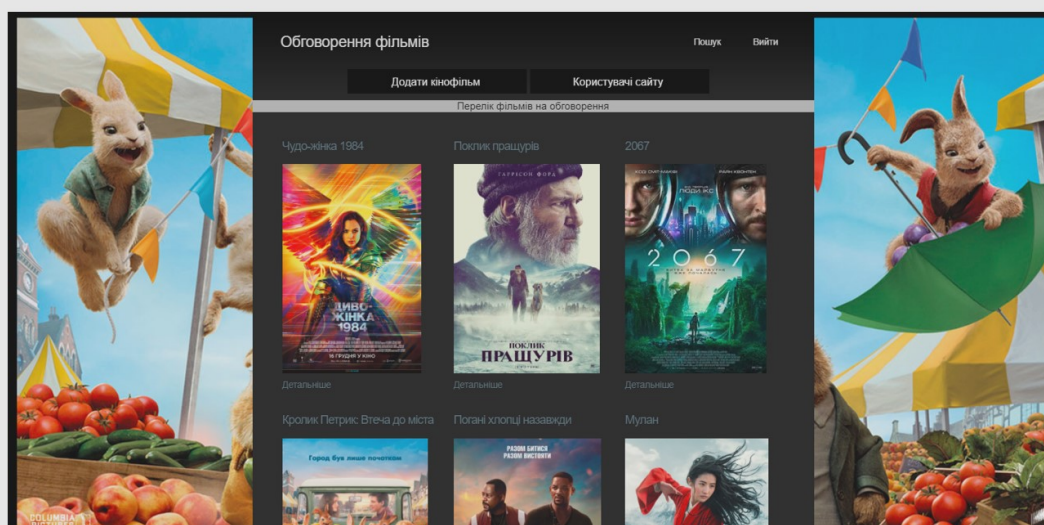
Стартова сторінка



Детальна інформація про фільм



Стартова сторінка зареєстрованого користувача



Презентація

Кваліфікаційна робота бакалавра

Веб-сервіс для обговорення кінофільмів

Виконав: студент 4 курсу групи КН-17-1 Корольчук І.Я.

Керівник: к.ф.-м.н., доцент кафедри КНІТ Міхалевський В.Ц.

Актуальність

Кількість об'єднань кінофанів у соціальних мережах зростає, а застосунків із повноцінним функціоналом і можливістю не тільки перегляду фільмів, а і їх обговорення практично немає. Тому робота у даному напрямку актуальна і потребує подальшого дослідження.



Завдання

Метою кваліфікаційної роботи бакалавра є розробка веб-сервісу для обговорення кінофільмів.

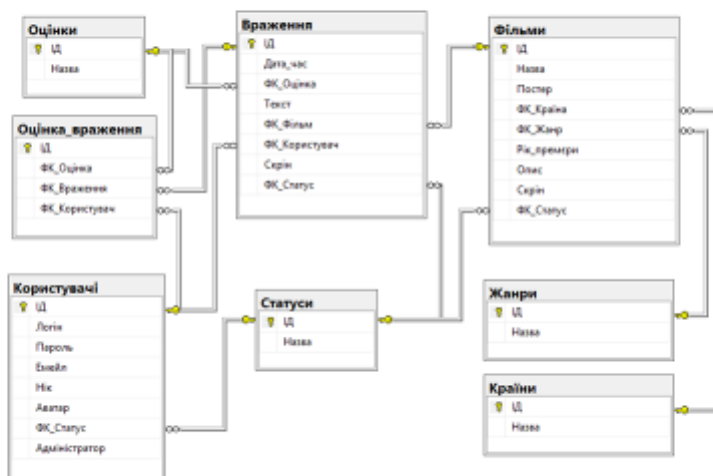
Автоматизувати наступні функції:

1. перегляд переліку кінофільмів на обговоренні
2. пошук кінофільму за назвою, роком прем'єри, жанром чи країною
3. перегляд докладного опису обраного кінофільму та вражень інших користувачів
4. долучення враження користувача про обраний кінофільм
5. оцінка обраного кінофільму
6. оцінка обраного враження іншого користувача
7. додавання нового кінофільму до веб-сервісу
8. перегляд переліку користувачів веб-сервісу та їх статистики.

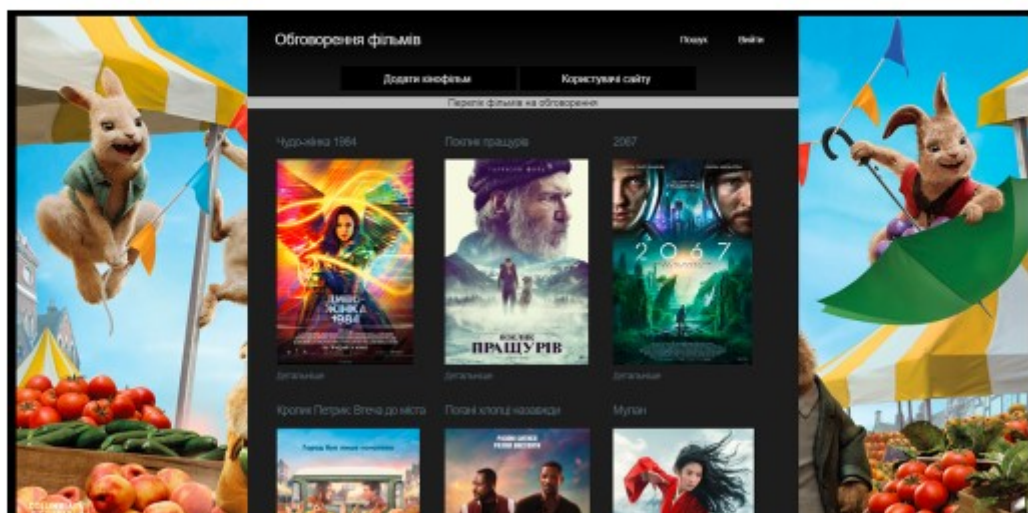
Діаграма класів веб-сервісу



Схема бази даних



Стартова сторінка зареєстрованого користувача



Заява на реєстрацію



Обговорення фільмів Поиск

Заява на реєстрацію у веб-сервісі

Логін

Пароль

Ім'я

Електронна пошта

Форма додавання нового фільму



Обговорення фільмів Поиск

Додавання нового фільму

Назва

Постер

Жанр

Країна

Рівняння

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Корольчук І. Я. на захист дипломного проекту (роботи)
(прізвище, ініціали)
 за спеціальністю 122 - Комп'ютерні науки
 На тему: Веб-сервіс для обговорення кінофільмів

Дипломний проект (робота), рецензія і довідка про перевірку на плагиат додаються.

Декан факультету



САВЕНКО О. С.

(прізвище та ініціали)

ДОВІДКА УСПІШНОСТІ

Корольчук І. Я. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки повністю виконав навчальний план спеціальності з такими розподілом оцінок за:
 національною шкалою: відмінно 0,00 %, добре 6,25 %, задовільно 93,75 %.
 шкалою ЄКТС: А 0,00 %, В 0,00 %, С 3,64 %, D 0,00%, Е 96,36 %.

Методист факультету

[Signature]
(підпис)

(прізвище та ініціали)

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент Корольчук І. Я. отримав роботу по темі національно оцінювальної роботи бакалавра виконав повністю. Тема роботи актуальна, розроблена модель системи веб-сервісу та алгоритми роботи. Інформаційна система виконана сучасним методом КРБ виконав згідно теоретичних вимог. Зауважень, що впливають на роботу програми забезпечення, не виступило.

Оцінка дипломного проекту (роботи) добре

Керівник дипломного проекту (роботи)

[Signature]
(підпис)

Михайленко В. С.
(прізвище та ініціали)

" 8 " 06 2021 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Корольчук І. Я. допускається до захисту цього

Завідувач кафедри

КНІТ

(назва)

[Signature]
(підпис, прізвище, ініціали)

Бурмак О. В.

" 9 " 06 2021 р.

11.06.2021

result_4332678570409700215.html

Fri Jun 11 12:25:24 EEST 2021, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 3.0%

 Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 15%**

| | | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------|----------------------------------------|---------|
| ID: 93347 Название: Веб-сервіс для обговорення кінофільмів Добавлено в БД: 2021-06-11 Авторы: І.Я. Корольчук Руководители: В.Ц. Міхалевський Консультанты: Оponentы: | Документ | | Суммарное совпадение по Базе Данных | |
| | Символы | Лексемы | Символы | Лексемы |
| | 57295 | 524 | 3431 (6%) | 42 (8%) |



Ім'я користувача:
Кафедра КН

ID перевірки:
1008271454

Дата перевірки:
11.06.2021 13:22:18 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
11.06.2021 13:28:09 EEST

ID користувача:
100005671

Назва документа: 2021_КРБ_Корольчук 20210610 6 фінал Lite

Кількість сторінок: 56 Кількість слів: 8606 Кількість символів: 63423 Розмір файлу: 13.71 MB ID файлу: 1008341

18.9% Схожість

Найбільша схожість: 7.65% з Інтернет-джерелом (https://studopedia.com.ua/1_47066_aksesuari.html)

12.2% Джерела з Інтернету

114

Сторінка 58

8.65% Джерела з Бібліотеки

61

Сторінка 59

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

1

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Веб-сервіс для обговорення кінофільмів

Автор: Корольчук І.Я., студент групи КН-17-1

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.фіз.-мат.н., доцент кафедри КНІТ Міхалевський В.Ц.

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | - |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | - |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укріплення запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | - |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 18,9% і адресується до першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____

Гарант ОП _____

Завідувач кафедри КНІТ _____

В. Ц. Міхалевський

О. В. Мазурець

О. В. Бармак

Хмельницький національний університет
Кафедра комп'ютерних наук та інформаційних технологій

РЕЦЕНЗІЯ
на кваліфікаційну роботу бакалавра

студента групи КН-17-1 Корольчука Івана Яковича
за темою «Веб-сервіс для обговорення кінофільмів».

Актуальність і значення теми: веб-сервіс для обговорення кінофільмів забезпечить автоматизацію організації та проведення обговорення, збору відгуків та вражень про фільми.

1. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки: веб-сервіс несе у собі актуальність перегляду та обговорення кінофільмів, моделі та алгоритми оптимізації роботи системи використані доцільно, тому такий перегляд дозволяє швидко отримати доступ до потрібної інформації.

2. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність: розроблена система призначена для любителів кінофільмів та адміністратора системи по збору відгуків та вражень про кінофільми. Автоматизовано функції незареєстрованого користувача, зареєстрованого користувача та адміністратора веб-сервісу. Напрямками практичного використання розробленої інформаційної системи визначено автоматизацію процесу перегляду та додавання вражень про той чи інший кінофільм. Дана інформаційна система буде корисною для кінотеатрів та інших місць прокату кінофільмів.

3. Загальний висновок та оцінка: вимоги поставленої задачі виконані в повному обсязі. Розроблений веб-сервіс працює вірно, рекомендовано для провадження.

Робота заслуговує на оцінку « задовільно »

Рецензент к.т.н., доц. Медведчук І.В. Жу