**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

|  |  |
| --- | --- |
|  |  |

**Курсова робота**

**з дисципліни «Програмування. Частина 2. Структури даних та алгоритми»**

**на тему: «Веб-сайт замовлення квитків у кінотеатр»**

Виконав:

студент 1 курсу, групи ІА-33

Дядюшка Ростислав Русланович \_\_\_\_\_\_\_\_\_\_

(підпис)

Керівник:

асистент кафедри ІСТ

Мягкий Михайло Юрійович \_\_\_\_\_\_\_\_\_\_

(підпис)

Засвідчую, що у цій дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_\_\_\_\_\_

(підпис)

Київ – 2024 року

ЗМІСТ

[ВСТУП 3](#_Toc168380637)

[1 ВИМОГИ ДО СИСТЕМИ 5](#_Toc168380638)

[1.1 Функціональні вимоги до системи 5](#_Toc168380639)

[1.2 Нефункціональні вимоги до системи 5](#_Toc168380640)

[2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ 7](#_Toc168380641)

[2.1 Діаграма прецедентів 7](#_Toc168380642)

[2.2 Опис сценаріїв використання системи 8](#_Toc168380643)

[3 АРХІТЕКТУРА СИСТЕМИ 12](#_Toc168380644)

[ВИСНОВКИ 14](#_Toc168380645)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 16](#_Toc168380646)

[ДОДАТОК А Лістінг програми 16](#_Toc168380647)

# ВСТУП

У сучасному світі, де технології стають невід'ємною частиною нашого повсякденного життя, індустрія розваг також адаптується до цифрової ери. Кінотеатри, як один із найпопулярніших видів дозвілля, не є винятком. Традиційний процес придбання квитків у касі кінотеатру часто пов'язаний з незручностями: довгі черги, обмежений час роботи кас, відсутність детальної інформації про фільми та сеанси. Ці фактори можуть негативно впливати на досвід відвідування кінотеатру та, як наслідок, на його прибутковість.

Актуальність створення веб-сайту для замовлення квитків у кінотеатр обумовлена кількома важливими факторами:

– зростання популярності онлайн-сервісів: сучасні споживачі все частіше віддають перевагу онлайн-послугам через їхню зручність та доступність 24/7.

– економія часу: можливість зарезервувати квитки онлайн дозволяє уникнути черг та заощадити час відвідувачів.

– підвищення якості обслуговування: онлайн-система надає більше інформації про фільми, сеанси та місця, що допомагає клієнтам зробити більш усвідомлений вибір.

– конкурентна перевага: кінотеатри, які пропонують онлайн-сервіси, мають перевагу над тими, що не адаптувалися до цифрових тенденцій.

Метою створення веб-сайту замовлення квитків у кінотеатр є модернізація та оптимізація процесу продажу квитків, що призведе до підвищення задоволеності клієнтів та збільшення прибутку кінотеатру.

Для досягнення цієї мети, система вирішуватиме наступні задачі:

– надання зручного та інтуїтивно зрозумілого інтерфейсу для перегляду розкладу сеансів, вибору місць та бронюванню квитків онлайн.

– забезпечення детальної інформації про фільми, включаючи назви, обкладинки, вікові обмеження, опис та жанри.

– впровадження функцій бронювання та резервування місць.

– забезпечення адаптивного дизайну для комфортної роботи на різних пристроях (ПК, планшети, смартфони).

# 1 ВИМОГИ ДО СИСТЕМИ

## 1.1 Функціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

– незареєстрований користувач повинен мати можливість переглядати інформацію про фільми, включаючи назву, жанр, вікове обмеження, тривалість, короткий опис;

– незареєстрований користувач повинен мати можливість створити профіль користувача;

– незареєстрований користувач повинен мати можливість переглядати загальну інформацію про кінотеатр;

– зареєстрований користувач повинен мати усі можливості, що є у незареєстрованого користувача;

– зареєстрований користувач повинен мати можливість переглядати розклад сеансів з інформацією про дату, час;

– зареєстрований користувач повинен мати можливість переглядати схему залу та наявність вільних місць для кожного сеансу;

– зареєстрований користувач повинен мати можливість вибрати місця та забронювати їх;

– адміністратор системи повинен мати усі можливості, що є у незареєстрованого користувача;

– адміністратор системи повинен мати можливість додавати, редагувати та видаляти інформацію про фільми та сеанси.

## 1.2 Нефункціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

– система повинна мати відкриту архітектуру;

– система повинна мати веб-інтерфейс;

– інтерфейс користувача має бути інтуїтивно зрозумілим, з мінімальною кількістю натискань для завершення певного процесу;

– система повинна бути крос-платформною;

– час відповіді на запит не повинен перевищувати 2 секунди при нормальному навантаженні;

– архітектура системи повинна дозволяти легко додавати нові функції без повної переробки.

# 2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

## 2.1 Діаграма прецедентів

Діаграма прецедентів системи представлена на рис. 2.1.

Акторами є користувачі системи: незареєстрований (гість) та зареєстрований (користувач).

Зареєстрованому користувачу доступна уся функціональність, що і незареєстрованому, а також . Детально усі сценарії використання описані у наступному підрозділі.

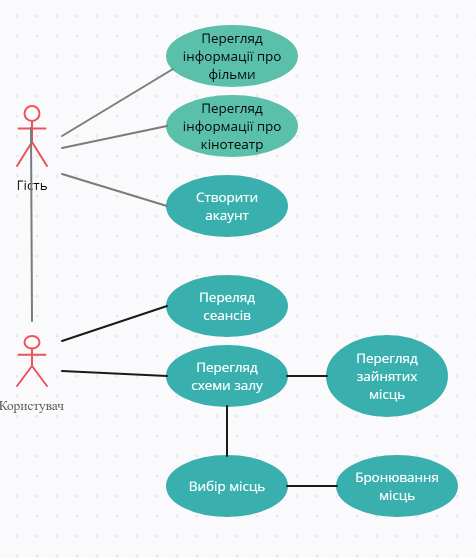


Рисунок 2.1 – Діаграма прецедентів

## 2.2 Опис сценаріїв використання системи

Детальні описи сценаріїв використання наведено у таблицях 2.1 – 2.5.

Таблиця 2.1 – Сценарій використання «Перегляд наявних фільмів»

|  |  |
| --- | --- |
| Назва | Перегляд наявних фільмів |
| ID | 1 |
| Опис | Користувач переглядає список всіх фільмів, що наразі демонструються в кінотеатрі |
| Актори | Незареєстрований користувач, зареєстрований користувач |
| Вигоди компанії | Надання повного списку фільмів допомагає користувачам знайти цікаві варіанти, збільшуючи ймовірність покупки |
| Частота користування | Дуже часто |
| Тригери | Користувач заходить натискає «» в меню |
| Передумови | Доступ до інтернету та відкритий сайт |
| Постумови | Користувач бачить список всіх поточних фільмів |
| Основний розвиток | 1. Користувач відкриває сайт 2. Натискає «Фільми» 3. Переглядає список фільмів з постерами та назвами |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

Таблиця 2.2 – Сценарій використання «Перегляд сеансів та вибір місць»

|  |  |
| --- | --- |
| Назва | Перегляд сеансів та вибір місць |
| ID | 2 |
| Опис | Користувач вибирає фільм, переглядає доступні сеанси та обирає місця |
| Актори | Зареєстрований користувач |
| Вигоди компанії | Зручний вибір сеансу та місць підвищує шанси на бронювання |
| Частота користування | Часто |
| Тригери | Користувач натискає на кнопку поруч з фільмом |
| Передумови | Користувач на сторінці списку фільмів |
| Постумови | Користувач вибрав сеанс і місця |
| Основний розвиток | Користувач переглядає інформацію про фільм  Натискає на кнопку поруч з фільмом  Обирає дату та сеанс  На схемі залу вибирає вільні місця |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

Таблиця 2.3 – Сценарій використання «Бронювання обраних місць»

|  |  |
| --- | --- |
| Назва | Бронювання обраних місць |
| ID | 3 |
| Опис | Користувач бронює вибрані місця на сеанс |
| Актори | Зареєстрований користувач |
| Вигоди компанії | Бронювання гарантує збереження місць, підвищуючи довіру користувача |
| Частота користування | Часто |
| Тригери | Користувач натискає «Підтвердити бронювання» після вибору місць |
| Передумови | Користувач авторизований і вибрав місця |
| Постумови | Місця заброньовані на певний час та дату сеансу |
| Основний розвиток | Користувач підтверджує вибір місць  Натискає «Підтвердити бронювання»  Отримує повідомлення про успішне бронювання |
| Альтернативні розвитки | – |
| Виняткові ситуації | Якщо місце стає зайнятим, система повідомляє про це  Якщо користувач обрав менше 1 або більше 15 місць, система повідомляє про це |

Таблиця 2.4 – Сценарій використання «Реєстрація на сайті»

|  |  |
| --- | --- |
| Назва | Реєстрація на сайті |
| ID | 4 |
| Опис | Новий користувач створює обліковий запис |
| Актори | Незареєстрований користувач |
| Вигоди компанії | В майбутньому реєстрація дозволить відстежувати поведінку користувачів і надсилати персоналізовані пропозиції |
| Частота користування | Одноразово для кожного користувача |
| Тригери | Користувач натискає кнопку «Зареєструватися» |
| Передумови | Користувач не має облікового запису |
| Постумови | Створено новий обліковий запис |
| Основний розвиток | Користувач натискає "Реєстрація" на головній сторінці.  Заповнює форму (ім'я, email, пароль, телефон, нікнейм, прізвище).  Натискає кнопку «Зареєструватися» |
| Альтернативні розвитки | Якщо користувач має акаунт, пропонується увійти в акаунт |
| Виняткові ситуації | Якщо введено некоректний email, телефон, система повідомляє про це  Якщо введено вже існуючий email або логін, система повідомляє про це  Якщо введено закороткий або задовгий пароль, логін, ім’я або прізвище, система повідомляє про це |

Таблиця 2.5 – Сценарій використання «»

|  |  |
| --- | --- |
| Назва | Перегляд інформації про кінотеатр |
| ID | 5 |
| Опис | Користувач дізнається більше про кінотеатр та його контакти |
| Актори | Незареєстрований користувач, зареєстрований користувач |
| Вигоди компанії | Надання інформації підвищує довіру та допомагає користувачам знайти кінотеатр |
| Частота користування | Рідко |
| Тригери | Користувач натискає "Про нас" або "Контакти" |
| Передумови | – |
| Постумови | Користувач отримує необхідну інформацію |
| Основний розвиток | Користувач натискає відповідний пункт меню.  Читає інформацію про історію, технології кінотеатру.  Переглядає адресу, телефон, email кінотеатру. |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

# 3 АРХІТЕКТУРА СИСТЕМИ

У проекті використовується архітектура на основі паттерну MVC (Model-View-Controller). Цей підхід дозволяє чітко розділити логіку презентації, бізнес-логіку та роботу з даними, що робить систему більш модульною, легкою для розуміння та підтримки.

Проект реалізований на мові Java з використанням фреймворку Spring Boot, який надає багато корисних інструментів та конвенцій для швидкої розробки веб-додатків.

Загальна архітектура системи наведена на рис. 3.1.

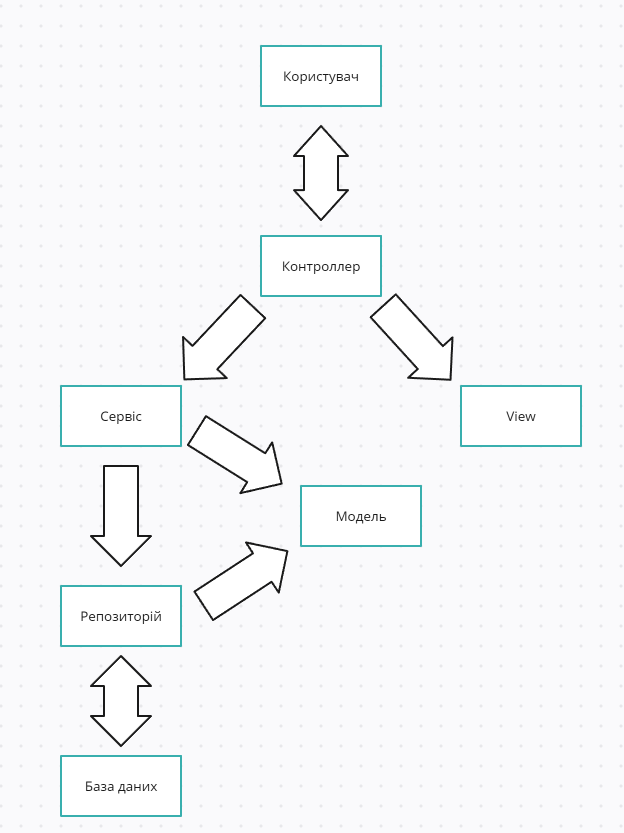


Рисунок 3.1 – Загальна архітектура системи

Система складається з наступних елементів:

– клієнтська частина (View);

– контролер (Controller);

– сервісний шар (Service);

– репозиторний шар (Repository);

– модель (Model);

– база даних.

Клієнтська частина відповідає за презентацію даних користувачеві. У проекті використовуються HTML-файли з шаблонізатором Thymeleaf, який дозволяє динамічно заповнювати сторінки даними з бекенду. CSS використовується для стилізації.

Контролери обробляють HTTP-запити від клієнтської частини. Вони отримують дані з форм, параметрів URL, викликають відповідні сервіси для обробки даних і повертають відповідні View.

Сервіси містять бізнес-логіку додатку. Тут відбувається валідація даних, прийняття рішень (наприклад, чи доступне місце для бронювання), трансформація даних.

Репозиторії відповідають за взаємодію з базою даних. У проекті використовується Spring Data JPA, який значно спрощує роботу з базами даних, автоматично генеруючи SQL-запити на основі імен методів. Hibernate виступає як ORM (Object-Relational Mapping) для маппінгу Java-об'єктів на таблиці бази даних.

Класи моделей представляють сутності бізнес-логіки. Вони містять поля, які відповідають колонкам у таблицях бази даних, геттери, сеттери та бізнес-методи. Анотації JPA (наприклад, @Entity, @Table, @Column) використовуються для конфігурації маппінгу на базу даних.

База даних для зберігання всіх даних додатку. Вона містить таблиці для фільмів, сеансів, місць, бронювань, користувачів тощо. H2 повністю сумісна з JDBC, що робить її чудовим вибором для використання з JPA та Hibernate. Spring Boot має вбудовану підтримку H2, що дозволяє налаштувати базу даних з мінімальною конфігурацією.

# ВИСНОВКИ

У ході виконання роботи було проаналізовано сучасний стан індустрії кінотеатрів та тенденції в напрямку цифровізації послуг. Було виявлено, що традиційний процес придбання квитків у касі кінотеатру має низку недоліків: довгі черги, обмежений час роботи кас, недостатність інформації про фільми. Ці фактори можуть негативно впливати на досвід відвідувачів та, як наслідок, на прибутковість кінотеатрів.

Тому було вирішено розробити систему, яка дозволить користувачам зручно та швидко бронювати квитки онлайн, отримувати детальну інформацію про фільми та сеанси.

Першим кроком було сформулювати функціональні та нефункціональні вимоги до системи. Це допомогло чітко визначити, що система повинна дозволяти переглядати наявні фільми та сеанси, вибирати місця, бронювати квитки, реєструватися та авторизуватися на сайті. Також було визначено, що система має бути швидкою, безпечною та зручною у використанні.

Наступним етапом став вибір технологій. Java була обрана як основна мова програмування завдяки її популярності у веб-розробці та широкій екосистемі. Spring Boot був вибраний як фреймворк через його здатність швидко створювати автономні, готові до виробництва Spring-додатки. Паттерн MVC, який є частиною Spring MVC, дозволив чітко розділити компоненти системи, зробивши її більш модульною та зрозумілою.

Для роботи з даними було обрано H2 - вбудовану базу даних, яка ідеально підходить для розробки та тестування. Це дозволило швидко почати роботу без необхідності налаштовувати зовнішню базу даних. Hibernate та Spring Data JPA були вибрані для взаємодії з базою даних, що значно спростило написання запитів, дозволивши зосередитися на бізнес-логіці.

Для побудови веб-інтерфейсу було використано HTML та Thymeleaf. Thymeleaf, як шаблонізатор, дозволяє легко інтегрувати динамічні дані з Java-коду в HTML-сторінки, що прискорює розробку та робить її більш гнучкою.

Після вибору технологій були описані основні сценарії використання системи: перегляд фільмів, вибір сеансів та місць, бронювання квитків, реєстрація та авторизація, перегляд інформації про кінотеатр. Це допомогло краще зрозуміти, як користувачі будуть взаємодіяти з системою.

Далі, було розроблено загальну архітектуру системи на основі паттерну MVC. Це дозволило розділити систему на логічні компоненти: Model для даних, Repository для роботи з базою даних, Service для бізнес-логіки, Controller для обробки запитів користувачів, та View для відображення інтерфейсу. Така структура робить систему більш гнучкою, полегшує її тестування та майбутнє розширення.

Отже, підсумовуючи, можна сказати, що основними перевагами створеної системи є:

– зручність для користувачів, що підвищує їхнє задоволення та лояльність;

– модульна архітектура, яка полегшує розширення та підтримку системи;

– використання сучасних технологій, що забезпечує ефективність та безпеку;

– легкість у налаштуванні та розгортанні завдяки використанню Spring Boot та H2.

У майбутньому система може бути вдосконалена наступними шляхами:

– додавання системи рекомендацій фільмів на основі історії переглядів та оцінок;

– інтеграція з платіжними системами купівлі квитків;

– впровадження мобільного додатку для ще більшої зручності;

– інтеграція з системами лояльності кінотеатру для автоматичного нарахування та використання бонусів.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

# ДОДАТОК А Лістінг програми

<https://github.com/SilverHinriEaT/Cinemax>