**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](#_Toc158383261)

[1 ВИМОГИ ДО СИСТЕМИ 4](#_Toc158383262)

[1.1 Функціональні вимоги до системи 4](#_Toc158383263)

[1.2 Нефункціональні вимоги до системи 4](#_Toc158383264)

[2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ 5](#_Toc158383265)

[2.1 Діаграма прецедентів 6](#_Toc158383266)

[2.2 Опис сценаріїв використання системи 7](#_Toc158383267)

[3 АРХІТЕКТУРА СИСТЕМИ 24](#_Toc158383268)

[4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ 26](#_Toc158383269)

[4.1 Загальна структура проекту 27](#_Toc158383270)

[4.2 Компоненти рівня доступу до даних 27](#_Toc158383271)

[4.3 Компоненти рівня бізнес-логіки 29](#_Toc158383272)

[4.4 Компоненти рівня інтерфейсу користувача 30](#_Toc158383273)

[ВИСНОВКИ 31](#_Toc158383274)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 33](#_Toc158383275)

[ДОДАТОК А Лістінг програми 36](#_Toc158383276)

# ВСТУП

Об’єм: 1-2 сторінки.

У вступі потрібно обґрунтувати актуальність системи, що створюється, сформулювати мету створення системи, описати задачі, які будуть нею вирішуватись.

Приклад:

З розвитком інформаційних технологій створення і поширення мультимедійної інформації стало просте як ніколи. Разом з цим з’являється потреба обробляти ці дані таким чином, щоб був можливий зручний пошук по цим даним. Якщо розглядати окремо лише музичні файли, то проблема їх індексації полягає у змінних у часі характеристик та доволі нечіткого поняття їх схожості, на відміну від, наприклад, текстових документів. Крім того, жоден існуючий рекомендаційний сервіс не може повністю задовольнити потреби користувачів. Так, наприклад, Last.fm та Spotify не використовують інформацію, що міститься безпосередньо в аудіо, сервіс Pandora, хоча і покладається на розмітку музичних файлів експертами, не може повноцінно поповнюватися в автоматичному режимі, що випливає у найменшій серед своїх конкурентів бібліотеці, а Shazam має можливість шукати лише повне співпадіння композиції. Звідси у користувачів випливає потреба у сервісі, що може вирішити ці проблеми.

Метою роботи є створення системи, що дозволяє знаходити схожі музичні композиції за контентом. Для цього система повинна мати певний перелік властивостей та вирішувати такі задачі:

* система повинна виробляти рекомендації базуючись лише на інформації, що безпосередньо міститься в аудіо;
* система повинна мати можливість обробляти дані в автоматичному режимі, в тому числі й ті дані, що раніше не оброблялися системою, а тому вона не має залежати від роботи експертів чи будь–якої іншої розмітки користувачами;
* аудіо інформація повинна бути подана у стислому вигляді для можливості подальшого зручного індексування та кластеризації.

# 1 ВИМОГИ ДО СИСТЕМИ

Об’єм: 1-2 сторінки

Перелічуються вимоги до системи, які цікавлять умовного «замовника» даної системи. Складаються з функціональних та нефункціональних вимог. Приклад:

## 1.1 Функціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

– незареєстрований користувач повинен мати можливість переглядати інформацію про фільми;

– незареєстрований користувач повинен мати можливість здійснювати пошук фільмів по словам у назві та/або у описі фільму;

– незареєстрований користувач повинен мати можливість сортувати список фільмів за абеткою, за кількістю коментарів, за кількістю «вподобайок», та за часом, коли інформація про фільм буда додана у систему;

– незареєстрований користувач повинен мати можливість переглядати коментарі інших користувачів;

– зареєстрований користувач повинен мати усі можливості, що є у незареєстрованого користувача, а також він повинен мати можливість залишати коментарі, ставити та знімати (свої) «вподобайки».

## 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 зручним та інтуїтивно-зрозумілим;

– система повинна бути крос-платформною.

# 2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

Об’єм: 4-6 сторінок

Наводиться діаграма прецедентів (use-case diagram) та опис сценаріїв використання системи.

Малювати діаграму можна в будь-якому графічному редакторі. Безкоштовно це можна зробити у веб-редакторі http://draw.io

Про те, як малювати use-case діаграми можна почитати та подивитися тут:

<https://www.uml-diagrams.org/use-case-diagrams.html>

<https://www.uml-diagrams.org/use-case-actor.html>

<https://www.uml-diagrams.org/use-case.html>

<https://www.uml-diagrams.org/use-case-actor-association.html>

<https://www.uml-diagrams.org/use-case-extend.html>

<https://www.uml-diagrams.org/use-case-include.html>

<https://www.uml-diagrams.org/use-case-diagrams-examples.html>

Опис сценаріїв використання рекомендується робити у вигляді однотипних таблиць по принципу «один сценарій – одна таблиця».

Приклади:

Не лякайтесь великої кількості таблиць – це просто для прикладу, щоб було приблизно зрозуміло, що там писати.

У вас в роботі буде значно менше сценаріїв використання (приблизно 3-6), і відповідно значно менше таких таблиць.

## 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 можливість писати коментарі та ставити/прибирати «вподобайки». Детально усі сценарії використання описані у наступному підрозділі.

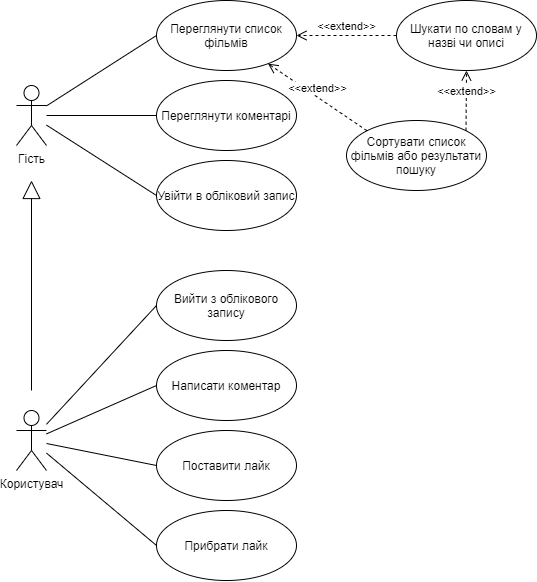


Рисунок 2.1 – Діаграма прецедентів

## 2.2 Опис сценаріїв використання системи

Детальні описи сценаріїв використання наведено у таблицях 2.1 – 2.17.

Таблиця 2.1 – Сценарій використання «Пошук по виконавцю, альбому, назві композиції»

|  |  |
| --- | --- |
| Назва | Пошук по виконавцю, альбому, назві композиції |
| ID | 1 |
| Опис | Користувач, використовуючи поле для пошуку, шукає аудіо, що відповідає критерію пошуку по виконавцю, альбому або назві |
| Актори | Користувач, Преміумний користувач |
| Вигоди компанії | Якщо користувачі не можуть шукати вже відому їм музику, то вони навряд чи продовжать користуватися сервісом |
| Частота користування | Постійно |
| Тригери | Користувач вводить пошуковий запит у полі для пошуку |
| Передумови | Пошукове поле доступне у будь–якому вікні |
| Постумови | Користувач потрапляє на вікно з результатами пошуку |
| Основний розвиток | Користувач вводить запит у пошукову строку, натискає на кнопку пошуку чи Enter |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.2 представлений сценарій використання «Прослуховування обраних композицій»

Таблиця 2.2 – Сценарій використання «Прослуховування обраних композицій»

|  |  |
| --- | --- |
| Назва | Прослуховування обраних композицій |
| ID | 2 |
| Опис | Користувач обирає трек, який хоче прослухати і слухає |
| Актори | Користувач, Преміумний користувач |
| Вигоди компанії | Стримінговий сервіс без можливості прослуховування не буде користуватися попитом |
| Частота користування | Постійно |
| Тригери | Користувач натискає на кнопку “Грати” біля треку |
| Передумови | У вікні є елементи треків, які можна програти |
| Постумови | Музика грає |
| Основний розвиток | Користувач натискає на кнопку “Грати”  Аудіо завантажується з серверу  Музика грає |
| Альтернативні розвитки | Користувач натискає на кнопку “Грати”  Аудіо завантажується з кешу пристрою  Музика грає |
| Виняткові ситуації | – |

В таблиці 2.3 представлений сценарій використання «Перегляд головної сторінки із плейлистами та рекомендаціями»

Таблиця 2.3 – Сценарій використання «Перегляд головної сторінки із плейлистами та рекомендаціями»

|  |  |
| --- | --- |
| Назва | Перегляд головної сторінки із плейлистами та рекомендаціями |
| ID | 3 |
| Опис | Користувач переглядає головну сторінку із плейлистами та рекомендаціями |
| Актори | Користувач, Преміумний користувач |
| Вигоди компанії | Головна сторінка важлива для зацікавлення нових користувачів, та актуальна інформація буде утримувати старих |
| Частота користування | Постійно |
| Тригери | Користувач переходить на сайт/запускає додаток |
| Передумови | Немає |
| Постумови | Користувач потрапляє на головну сторінку |
| Основний розвиток | Користувач запускає додаток/переходить на сайт |
| Альтернативні розвитки | – |
| Виняткові ситуації | Сервіс недоступний у країні користувача, тоді показується повідомлення про помилку |

В таблиці 2.4 представлений сценарій використання «Пошук композицій по наданому користувачем аудіо із мікрофону або файлу»

Таблиця 2.4 – Сценарій використання «Пошук композицій по наданому користувачем аудіо із мікрофону або файлу»

|  |  |
| --- | --- |
| Назва | Пошук композицій по наданому користувачем аудіо із мікрофону або файлу |
| ID | 4 |
| Опис | Користувач може знайти схожі треки, надавши запис з мікрофону або файл як запит |
| Актори | Користувач, Преміумний користувач |
| Вигоди компанії | Така стратегія сприяє кращому пошуку нової музики, ніж класичні моделі, тому повинна стати головною цікавинкою компанії |
| Частота користування | Невідомо, як часто ця опція буде використовуватися на практиці |
| Тригери | Користувач натискає на відповідні кнопки у пошуковій строці |
| Передумови | Користувач знаходиться на будь–якій сторінці із пошуковим запитом |
| Постумови | Користувач потрапляє на сторінку із пошуковими видачами |
| Основний розвиток | Користувач натискає на відповідні кнопки у пошуковій строці. Обирає файл. Отримує пошукову видачу |
| Альтернативні розвитки | Користувач натискає на відповідні кнопки у пошуковій строці. Записує з мікрофону  Отримує пошукову видачу |
| Виняткові ситуації | – |

В таблиці 2.5 представлений сценарій використання «Формування персоналізованих плейлистів на основі результатів видачі»

Таблиця 2.5 – Сценарій використання «Формування персоналізованих плейлистів на основі результатів видачі»

|  |  |
| --- | --- |
| Назва | Формування персоналізованих плейлистів на основі результатів видачі |
| ID | 5 |
| Опис | Формуються персоналізовані плейлисти на основі результатів видачі |
| Актори | Користувач, Преміумний користувач |
| Вигоди компанії | Легке створення якісних нових плейлистів потрібно утримати існуючих користувачів |
| Частота користування | Користувачи зазвичай досить часто використовують плейлисти |
| Тригери | Користувач здійснив пошук |
| Передумови | Користувач знаходиться у вікні із результатами пошуку |
| Постумови | Користувач зберігає новий плейлист |
| Основний розвиток | У вікні із результатами пошуку користувач натискає на спеціальну кнопку |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.6 представлений сценарій використання «Додавання композицій в плейлисти/обране»

Таблиця 2.6 – Сценарій використання «Додавання композицій в плейлисти/обране»

|  |  |
| --- | --- |
| Назва | Додавання композицій в плейлисти/обране |
| ID | 6 |
| Опис | Користувач додає композиції в плейлисти/обране |
| Актори | Користувач, Преміумний користувач |
| Вигоди компанії | Плейлисти потрібні для утримання існуючих користувачі, як необхідний функціонал |
| Частота користування | Часто |
| Тригери | Користувач натискає на відповідну кнопку |
| Передумови | Користувач знаходиться на сторінці із треками, обирає якийсь із них |
| Постумови | Трек додано до відповідного плейлисту |
| Основний розвиток | Користувач натискає на відповідну кнопку біля треку для додавання його до обраного |
| Альтернативні розвитки | Користувач натискає на відповідну кнопку біля треку для додавання. Відкривається віконце із існуючими плейлистами або варіантом створення ного, до якого додасться трек |
| Виняткові ситуації | Спроба створити плейлист з тією ж назвою |

В таблиці 2.7 представлений сценарій використання «Сплата платної підписки»

Таблиця 2.7 – Сценарій використання «Сплата платної підписки»

|  |  |
| --- | --- |
| Назва | Сплата платної підписки |
| ID | 7 |
| Опис | Користувач може сплатити підписку, що надає більше можливостей |
| Актори | Користувач, Преміумний користувач |
| Вигоди компанії | Платні підписки основний дохід компанії |
| Частота користування | Приблизно третина користувачів сплачує підписку на Spotify, проте вони генерують найбільший прибуток |
| Тригери | Користувач вибирає відповідну опцію в особистому кабінеті |
| Передумови | Користувач знаходиться в особистому кабінеті |
| Постумови | Користувач стає преміумним користувачем, або подовжує свою підписку, якщо він уже преміумний |
| Основний розвиток | Користувач вибирає відповідну опцію в особистому кабінеті, проходить процедуру оплати |
| Альтернативні розвитки | – |
| Виняткові ситуації | Користувач відміняє сплату, відмова оплати банком, виводиться повідомлення про невдалу транзакцію, поповнення не відбувається |

В таблиці 2.8 представлений сценарій використання «Налаштування якості звуку»

Таблиця 2.8 – Сценарій використання «Налаштування якості звуку»

|  |  |
| --- | --- |
| Назва | Налаштування якості звуку |
| ID | 8 |
| Опис | Преміумний користувач може вибрати більш високу якість звуку |
| Актори | Преміумний користувач |
| Вигоди компанії | Висока якість звучання необхідна для залучення деяких клієнтів |
| Частота користування | Близько третини користувачів |
| Тригери | Користувач натискає меню поряд із строкою програвання |
| Передумови | Користувач що–небудь прослуховує |
| Постумови | Змінюється якість передаваємого звуку |
| Основний розвиток | Користувач натискає меню поряд із строкою програвання, вибирає бажану якість із списку |
| Альтернативні розвитки | – |
| Виняткові ситуації | Інтернет користувача занадто повільний для обраної якості, виводиться попереджувальне повідомлення |

В таблиці 2.9 представлений сценарій використання «Підстройка аудіо еквалайзером»

Таблиця 2.9 – Сценарій використання «Підстройка аудіо еквалайзером»

|  |  |
| --- | --- |
| Назва | Підстройка аудіо еквалайзером |
| ID | 9 |
| Опис | Користувач налаштовує АЧХ звуку за допомогою еквалайзера |
| Актори | Преміумний користувач |
| Вигоди компанії | Висока якість звучання необхідна для залучення деяких клієнтів |
| Частота користування | Близько третини користувачів |
| Тригери | Користувач натискає меню поряд із строкою програвання |
| Передумови | Користувач знаходиться на сторінці із треками, обирає якийсь із них |
| Постумови | Змінюються АЧХ передаваємого звуку |
| Основний розвиток | Користувач натискає меню поряд із строкою програвання, відкривається меню із слайдерами налаштування, користувач може їх змінювати |
| Альтернативні розвитки | – |
| Виняткові ситуації |  |

В таблиці 2.10 представлений сценарій використання «Кешування аудіо на пристрої користувача»

Таблиця 2.10 – Сценарій використання «Кешування аудіо на пристрої користувача»

|  |  |
| --- | --- |
| Назва | Кешування аудіо на пристрої користувача |
| ID | 10 |
| Опис | Користувач кешує обрані треки |
| Актори | Преміумний користувач |
| Вигоди компанії | Можливість прослуховування офлайн повинна мотивувати користувачів сплачувати підписку |
| Частота користування | Третина користувачів |
| Тригери | Користувач натискає на відповідну кнопку |
| Передумови | Користувач знаходиться на сторінці із треками, обирає якийсь із них |
| Постумови | Трек кешується на пристрої користувача |
| Основний розвиток | Користувач натискає на відповідну кнопку біля треку для додавання до кешу |
| Альтернативні розвитки | – |
| Виняткові ситуації | Не вистачає вільного місця на пристрої, виводиться повідомлення про помилку |

В таблиці 2.11 представлений сценарій використання «Розмістити банер»

Таблиця 2.11 – Сценарій використання «Розмістити банер»

|  |  |
| --- | --- |
| Назва | Розмістити банер |
| ID | 11 |
| Опис | Рекламодавець може придбати місця для банеру, щоб він інколи відображався клієнтам |
| Актори | Рекламодавець |
| Вигоди компанії | Таким чином можна монетизувати користувачів без платної підписки |
| Частота користування | Часто |
| Тригери | Рекламодавець із особистого кабінету вибирає банер, який хоче розмістити і сплачує розміщення |
| Передумови | Рекламодавець знаходиться у власному кабінеті |
| Постумови | Банер буде випадково показано користувачам |
| Основний розвиток | Рекламодавець із особистого кабінету вибирає банер, який хоче розмістити і сплачує розміщення |
| Альтернативні розвитки | – |
| Виняткові ситуації | Відбувається помилка під час сплати, показується повідомлення про помилку |

В таблиці 2.12 представлений сценарій використання «Розмістити аудіорекламу»

Таблиця 2.12 – Сценарій використання «Розмістити аудіорекламу»

|  |  |
| --- | --- |
| Назва | Розмістити аудіорекламу |
| ID | 12 |
| Опис | Рекламодавець може придбати місця для аудіореклами, щоб вони інколи програвалися клієнтам |
| Актори | Рекламодавець |
| Вигоди компанії | Таким чином можна монетизувати користувачів без платної підписки |
| Частота користування | Часто |
| Тригери | Рекламодавець із особистого кабінету вибирає аудіо, який хоче розмістити і сплачує розміщення |
| Передумови | Рекламодавець знаходиться у власному кабінеті |
| Постумови | Аудіорекламу буде випадково програно користувачам |
| Основний розвиток | Рекламодавець із особистого кабінету вибирає аудіо, яке хоче розмістити і сплачує розміщення |
| Альтернативні розвитки | – |
| Виняткові ситуації | Відбувається помилка під час сплати, показується повідомлення про помилку. Аудіо занадто довге |

В таблиці 2.13 представлений сценарій використання «Видалити аудіорекламу»

Таблиця 2.13 – Сценарій використання «Видалити аудіорекламу»

|  |  |
| --- | --- |
| Назва | Видалити аудіорекламу |
| ID | 13 |
| Опис | Рекламодавець може видалити аудіорекламу |
| Актори | Рекламодавець |
| Вигоди компанії | Необхідний функціонал рекламодавця |
| Частота користування | Часто |
| Тригери | Рекламодавець із особистого кабінету вибирає аудіо, який хоче видалити |
| Передумови | Рекламодавець знаходиться у власному кабінеті |
| Постумови | Аудіорекламу буде видалено |
| Основний розвиток | Рекламодавець із особистого кабінету вибирає аудіо, який хоче видалити, натискає відповідну кнопку |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.14 представлений сценарій використання «Видалити банер»

Таблиця 2.14 – Сценарій використання

|  |  |
| --- | --- |
| Назва | Видалити банер |
| ID | 14 |
| Опис | Рекламодавець може видалити банер |
| Актори | Рекламодавець |
| Вигоди компанії | Необхідний функціонал рекламодавця |
| Частота користування | Часто |
| Тригери | Рекламодавець із особистого кабінету вибирає банер, який хоче видалити |
| Передумови | Рекламодавець знаходиться у власному кабінеті |
| Постумови | Банер буде видалено |
| Основний розвиток | Рекламодавець із особистого кабінету вибирає банер, який хоче видалити, натискає відповідну кнопку |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.15 представлений сценарій використання «Редагування інформації про виконавця, альбом тощо»

Таблиця 2.15 – Сценарій використання «Редагування інформації про виконавця, альбом тощо»

|  |  |
| --- | --- |
| Назва | Редагування інформації про виконавця, альбом тощо |
| ID | 15 |
| Опис | Контент менеджер редагує інформацію про виконавця, альбом тощо |
| Актори | Контент менеджер |
| Вигоди компанії | Можливість виправляти помилки |
| Частота користування | Часто |
| Тригери | Контент менеджер вибирає, що він бажає редагувати |
| Передумови | Контент менеджер знаходиться на сторінці із треками, виконавцями або альбомами |
| Постумови | Зміна інформації |
| Основний розвиток | Контент менеджер знаходиться на сторінці із треками, виконавцями або альбомами, вибирає, що він бажає редагувати, редагує та зберігає зміни |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.16 представлений сценарій використання «Додавання виконавця, альбому, аудіотреку»

Таблиця 2.16 – Сценарій використання «Додавання виконавця, альбому, аудіотреку»

|  |  |
| --- | --- |
| Назва | Додавання виконавця, альбому, аудіотреку |
| ID | 16 |
| Опис | Контент менеджер додає інформацію про виконавця, альбом тощо |
| Актори | Контент менеджер |
| Вигоди компанії | Можливість додавати нове |
| Частота користування | Часто |
| Тригери | Контент менеджер вибирає, що він бажає додати |
| Передумови | Контент менеджер знаходиться на сторінці власного кабінету |
| Постумови | Зміна інформації |
| Основний розвиток | Контент менеджер знаходиться на сторінці власного кабінету, контент менеджер вибирає, що він бажає додати, додає |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.16 представлений сценарій використання «Видалення записів за вимогами правовласників»

Таблиця 2.17 – Сценарій використання

|  |  |
| --- | --- |
| Назва | Видалення записів за вимогами правовласників |
| ID | 17 |
| Опис | Контент менеджер видаляє треки |
| Актори | Контент менеджер |
| Вигоди компанії | Можливість видаляти старе або незаконне |
| Частота користування | Часто |
| Тригери | Контент менеджер вибирає, що він бажає видалити |
| Передумови | Контент менеджер знаходиться на сторінці із треками, виконавцями або альбомами |
| Постумови | Зміна інформації |
| Основний розвиток | Контент менеджер знаходиться на сторінці із треками, виконавцями або альбомами, вибирає, що він бажає видалити, видаляє та зберігає зміни |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

# 3 АРХІТЕКТУРА СИСТЕМИ

Об’єм: 1-3 сторінки

Описується загальна архітектура системи, з яких компонентів вона складається, як вони пов’язані, які технології використовуються. Приклад як це приблизно може виглядати. Попередження: в рисунок навмисно внесені помилки, щоб студенти його не «копіпастили».

Загальна архітектура системи наведена на рис. 3.1.

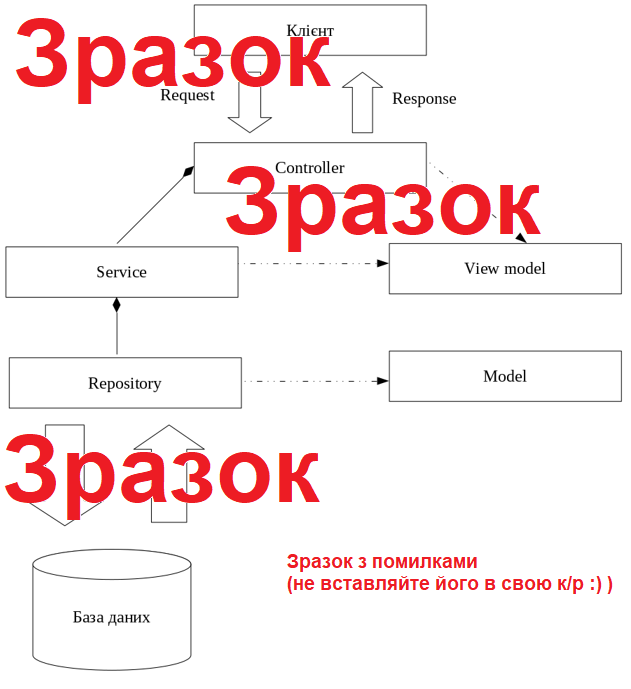


Рисунок 3.1 – Загальна архітектура системи

Система складається з наступних елементів:

– графічний інтерфейс;

– серверна частина;

– база даних.

Графічний інтерфейс необхідний для взаємодії з користувачем. HTTP запит надходить до серверної частини, де оброблюється і повертається відповідь. На серверній частині виконується основна логіка системи. Дані, отриманні з графічного інтерфейсу валідуються, конвертуються. Також, серверна частина формує запит до бази даних та оброблює відповідь і передає її до графічного інтерфейсу. База даних зберігає дані, які були сформовані на серверній частині та повертає їх у разі запиту. Структурна схема зображена в Додатоку Б

До серверної частини належать наступні елементи:

– контролер;

– модель та вигляд;

– сервіс;

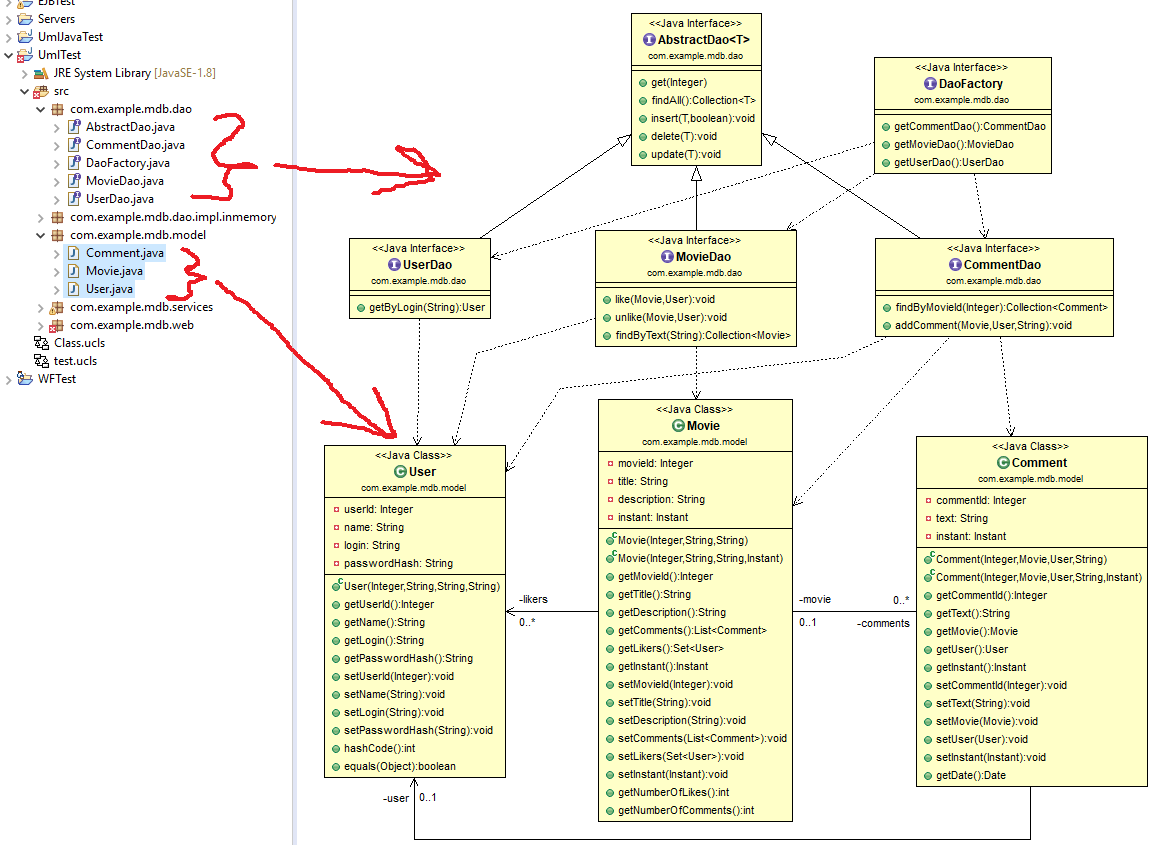
– репозиторій.

На контролер надходять дані з графічного інтерфейсу. З контролеру, дані формуються в сервісі для запиту в репозиторій. З репозиторію дані надсилаються до бази даних і зберігаються. Також в контролері формується вид, тобто об’єкт і його ім’я для відображення на графічному інтерфейсі.

# 4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

Обсяг: 10-20 сторінок, включаючи фрагменти коду, рисунки з UML-діаграмами класів або іншими типами діаграм.

Для економії часу рекомендується не малювати діаграми класів «руками», а генерувати їх на основі написаного коду. Наприклад, це вміє робити компонент plug-in компонент ObjectAid для Eclipse (<https://www.objectaid.com/>). Достатньо просто перетягти необхідні елементи з дерева проекту на полотно діаграми, і усі елементи з’являться автоматично:



Рекомендується розбити цей розділ на підрозділи відповідно до архітектури системи

## 4.1 Загальна структура проекту

Загальна структура проекту представлена на рис.4.1

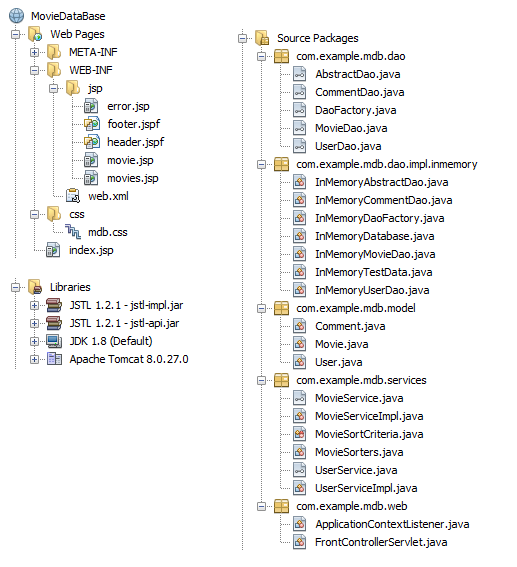


Рисунок 4.1 – Загальна структура проекту

Проект складається з веб-ресурсів, бібліотек, та вихідного коду, який в свою чергу можна поділити на компоненти рівня доступу до даних, компоненти бізнес-логіки та веб-компоненти.

## 4.2 Компоненти рівня доступу до даних

Основні сутності та інтерфейси рівня доступу до даних наведені на рис. 4.2

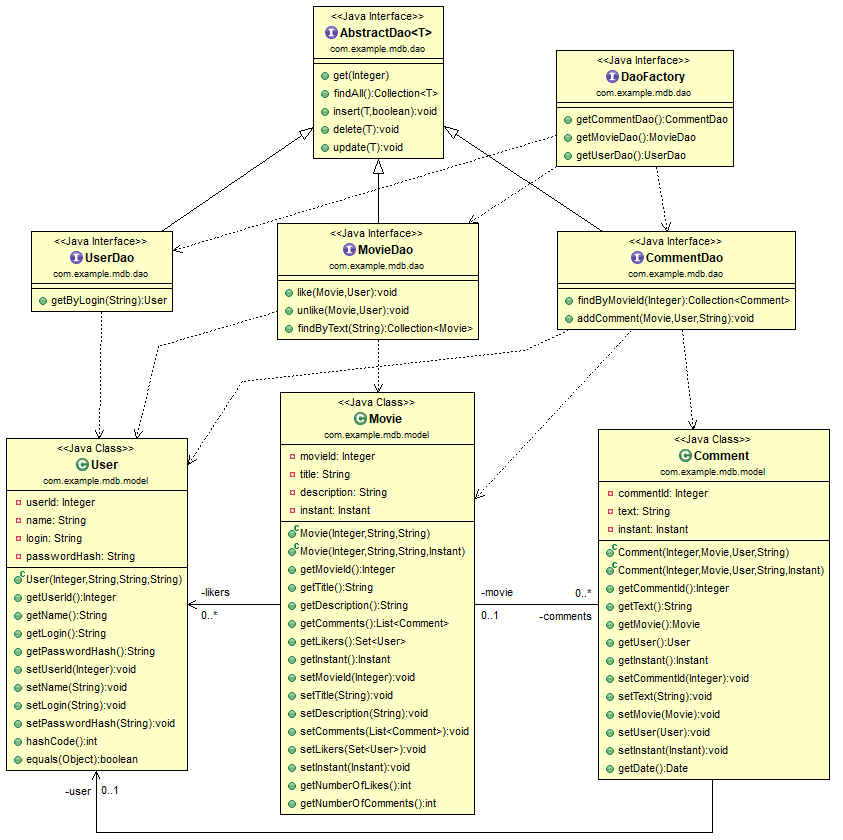


Рисунок 4.2 – Основні сутності та інтерфейси рівня доступу до даних

Далі йде опис сутностей, зв’язків та способів їх використання...

## 4.3 Компоненти рівня бізнес-логіки

## 4.4 Компоненти рівня інтерфейсу користувача

# ВИСНОВКИ

Обсяг: 1-2 сторінки. Коротко підсумовується, що було зроблено в роботі, які цілі були досягнуті, які задачі були вирішені. Наводиться загальна оцінка результату, та робляться пропозиції щодо використання або шляхи подальшого розвитку чи вдосконалення даної системи.

Приклад:

Під час написання роботи, було проаналізовано ринок CRM-систем в Україні, переваги та недоліки існуючих систем. Було вирішено розробити систему, яка дозволить користувачу самостійно керувати своїми телекомунікаційними послугами.

Першим кроком було сформулювати функціональні та нефункціональні вимоги до системи, що визначило очікувану поведінку системи. Наступним, обрано технології на яких буде написана система. Java була обрана в якості мови програмування, середа розробки – IntelliJ IDEA за її можливості та потужність. В якості системи управління базами даних було вирішено використовувати MySQL за легкість її налаштування та підтримки. Для роботи з запитами та налаштуванням з’єднання між системою та базою даних було вирішено використати Hibernate та Spring Data. Величезна перевага фреймворку Spring Data, за яку його було обрано, в можливості не писати запити на SQL, а будувати їх за допомоги назв методів. Для налаштування безпеки системи було вирішено використовувати Spring Security. Цей фреймворк дозволяє власноруч зробити необхідні налаштування, такі як дозволенні ресурси, url-адреси та як саме система визначає ці права доступу. Система повинна мати графічний інтерфейс, тож вирішено було використовувати HTML та Thymeleaf за їх легкість у впровадженні.

Наступним кроком були описані сценарії використання. Система підтримує функціонал перегляду та активації тарифів, сервісів, тарифів домашнього Інтернету. Також була впроваджена бонусна система. Під бонусом мається на увазі – промо-код, активувавши який, користувач може отримати знижку на сторонні продукти. Бонус нараховується у розмірі 10% від вартості купленого продукту на бонусний рахунок користувача. Бонус має термін дії, тож по завершені її – деактивується. Також для симуляції підключення тарифів домашнього Інтернету (проведення кабелів та інше) було описано та впроваджено відповідний сценарій, за яким можна переглянути як змінюється статус по завершенні технічних робіт.

Далі, було розроблено загальну архітектуру системи. Завдяки шаблону проектування MVC, систему було розділено на логічні рівні. Також, окремо було виділені конвертери, помічники та моделі, які роблять систему більш гнучкою та зрозумілою.

Далі, була розроблена ER-модель. Завдяки Hibernate, для встановлення відповідності між базою даних та Java-кодом було створено класи-сутності, що прискорило розробку системи. Для кожної сутності були визначенні необхідні поля та зв’язок з іншими сутностями, якщо в ньому була необхідність.

Отже, сумуючи наведене вище, можна сказати, що основі переваги створеної системи в простоті її налаштування та оновлення функціоналу. Завдяки тому, що система має відкриту архітектуру, її розширення не є проблемою. Також графічний інтерфейс системи є легким в налаштуванні. Система є конкурентоспроможною та має потенціал до розвинення.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

На усі джерела мають бути посилання у тексті

# ДОДАТОК А Лістінг програми

В першому рядочку – посилання на github

Далі – код системи (інтерфейси, класи, шаблони сторінок, конфігураційні файли, файли з тестовими даними і т.д.)

Враховуючи загальний обсяг коду системи, в додатках допускається використовувати шрифт, що відрізняється від стандартного Times New Roman 14 міжрядковий 1.5 (наприклад, розмір – 10, міжрядковий – одинарний).