**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»**

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

Виконав:

студент 1 курсу, групи ІА-31

Губар Богдан Олександрович \_\_\_\_\_\_\_\_\_\_

(підпис)

Керівник:

асистент кафедри ІСТ

Тюляков Дмитро Ігорович \_\_\_\_\_\_\_\_\_\_

(підпис)

Засвідчую, що у цій курсовій роботі 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)

[ДОДАТОК Б Назва додатку Б (за наявності) 37](#_Toc158383277)

[ДОДАТОК В Назва додатку В (за наявності) 38](#_Toc158383278)

# ВСТУП

## На сьогоднішній день зростає тенденція в онлайн-сервісах, таких як: інтернет-магазини, сайти аптек, онлайн-банкінги, бронювання готелів та подорожей в Інтернеті, тощо. Це пов’язано насамперед через їх зручність та відсутність необхідності фізичної наявності в магазині.

## Актуальність створення веб-сайту для замовлення квитків полягає в зростаючому попиті на онлайн-сервіси в різних галузях, зокрема у галузі культурних подій та розважальної індустрії. З підвищенням інтересу до кіно, театральних вистав, музичних концертів та інших подій, користувачі шукають зручний і надійний спосіб замовлення квитків без необхідності особистої присутності в касі або на місці події.

## Метою роботи є створення системи, яка дозволяє користувачам зручно та швидко замовляти квитки на різноманітні події через інтернет. Для цього система повинна мати такі можливості та функціональність:

- Зручний інтерфейс для перегляду різних подій, вистав та концертів.

- Можливість обрати місце та кількість квитків для замовлення.

- Нагадування та інформаційна підтримка щодо подій, на які були придбані квитки.

- Система повинна бути надійною та безпечною з точки зору обробки особистих даних користувачів та фінансових операцій.

## Такий веб-сайт забезпечить користувачам зручний спосіб отримання квитків на різні події, сприяючи розвитку онлайн-культурної індустрії та задоволенню потреб сучасного споживача.

# 1 ВИМОГИ ДО СИСТЕМИ

## 1.1 Функціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

* Незареєстрований користувач(далі - гість) має право переглядати інформацію про сеанси
* Гість має право замовлювати квитки, встановлявати ціну та кількість квитків для конкретного сеансу, а також місце та дату;
* Адміністратор має право редагувати інформацію про квитки та сеанси а також видаляти та додавати сеанси;
* Адміністратор може бачити деталі замовлення квитків користувачами;

## 1.2 Нефункціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

– система повинна мати архітектуру MVC;

– система повинна мати веб-інтерфейс;

– інтерфейс користувача має бути зручним та інтуїтивно-зрозумілим;

– система повинна бути крос-платформною.

# 2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

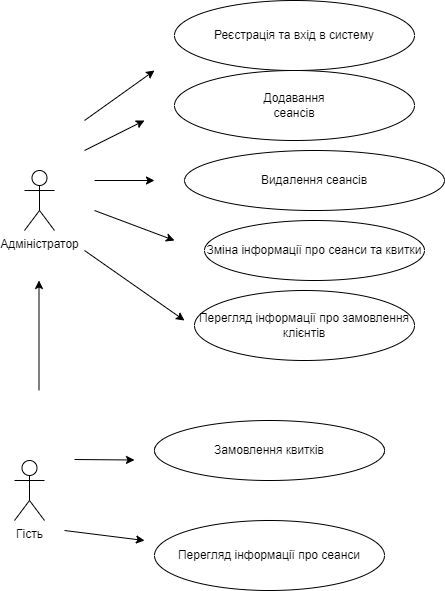
## 2.1 Діаграма прецедентів

Діаграма прецедентів системи представлена на рис. 2.1.

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

Адміністратору доступна інформація про замовлення, можливість додавати, видаляти сеанси та редагувати їх, також він може переглядати замовлення клієнтів. Гість – незареєстрований користувач, може лише замовити квиток, та переглянути інформацію про сеанси, які є на сайті, на які можна замовити квиток. 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 «Реєстрація та вхід»

|  |  |
| --- | --- |
| Назва | Реєстрація та вхід |
| ID | 1 |
| Опис | Адміністратор може зареєструватись в сервісі, і надалі керувати процесами на сайті, додавати та змінювати інформацію на сайті |
| Актори | Адміністратор |
| Вигоди компанії | Сайт буде мати постійні оновлення та актуальну інформацію про сеанси, завдяки роботі адміністраторів |
| Частота користування | Часто |
| Тригери | Користувач переходить на сторінку реєстрації або входу |
| Передумови | Правильні дані, активний обліковий запис(для входу) |
| Постумови | Невдалий або успішний вхід або реєстраціія |
| Основний розвиток | Користувач вводить запит у пошукову строку, натискає на кнопку пошуку чи Enter |
| Альтернативні розвитки | - |
| Виняткові ситуації | Заблокований обліковий запис |

В таблиці 2.2 представлений сценарій використання «Додавання сеансів»

Таблиця 2.2 – Сценарій використання «Додавання сеансів»

|  |  |
| --- | --- |
| Назва | Додавання сеансів |
| ID | 2 |
| Опис | Адміністратор додає новий сеанс на сторінку перегляду сеансів |
| Актори | Адміністратор |
| Вигоди компанії | - |
| Частота користування | Часто |
| Тригери | Адміністратор натискає кнопку «Додати новий сеанс» |
| Передумови | Введені необхідні дані(назва, опис сеансу) |
| Постумови | Додається новий сеанс |
| Основний розвиток | Адміністратор натиснув «Додати сеанс»  На сайті з’явився новий сеанс, на який можна замовити квитки |
| Альтернативні розвитки | – |
| Виняткові ситуації | Після натискання кнопки сеанс не додається або неправильно додається через неправильно вказані дані при додаванні |

В таблиці 2.3 представлений сценарій використання «Видалення сеансів»

Таблиця 2.3 – Сценарій використання «Видалення сеансів»

|  |  |
| --- | --- |
| Назва | Видалення сеансів |
| ID | 3 |
| Опис | Адміністратор видаляє обраний сеанс |
| Актори | Адміністратор |
| Вигоди компанії | Сайт не буде містити неактуальні сеанси, тому користувачі ресурсу завжди матимуть актуальну інформацію щодо нових фільмів |
| Частота користування | Часто |
| Тригери | Адміністратор натискає кнопку «Видалити сеанс» |
| Передумови | Було перейдено на сторінку перегляду інформації про сеанс |
| Постумови | Сеанс видаляється з сторінки |
| Основний розвиток | Сеанс зникає з головної сторінки |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.4 представлений сценарій використання «Зміна інформації про сеанси»

Таблиця 2.4 – Сценарій використання «Зміна інформації про сеанси»

|  |  |
| --- | --- |
| Назва | Зміна інформації про сеанси |
| ID | 4 |
| Опис | Адміністратор змінює інформацію про сеанс |
| Актори | Адміністратор |
| Вигоди компанії | Завдяки цьому користувачі сервісу в динаміці бачитимуть зміни і завжди розумітимуть точно інформацію про кожен сеанс на даний момент |
| Частота користування | Невідомо, як часто ця опція буде використовуватися на практиці |
| Тригери | Натиснута кнопка «Змінити інформацію» на сторінці перегляду сеансу |
| Передумови | Адміністратор перейшов на сторінку сеансу |
| Постумови | Змінюється відповідно інформація про конкретний сеанс |
| Основний розвиток | На сайті змінюється певна інформація щодо того сеансу, який редагував адміністратор |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.5 представлений сценарій використання «Перегляд інформації про замовлення клієнтів»

Таблиця 2.5 – Сценарій використання «Перегляд інформації про замовлення клієнтів»

|  |  |
| --- | --- |
| Назва | Перегляд інформації про замовлення клієнтів |
| ID | 5 |
| Опис | Адміністратор переглядає замовлення клієнтів |
| Актори | Адміністратор |
| Вигоди компанії | – |
| Частота користування | Напряму залежить від потоку клієнтів |
| Тригери | Адміністратор перейшов на вкладку «Останні замовлення» |
| Передумови | Було створене замовлення гостями |
| Постумови | Адміністратор бачить інформацію про останні замовлення на сайті та їх деталі |
| Основний розвиток | В вкладці перегляд замовлень видно останні замовлення |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

В таблиці 2.6 представлений сценарій використання «Замовлення квитків»

Таблиця 2.6 – Сценарій використання «Замовлення квитків»

|  |  |
| --- | --- |
| Назва | Замовлення квитків |
| ID | 6 |
| Опис | Гість (рідко – адміністратор) може замовити квитки на будь який актуальний сеанс |
| Актори | Гість, Адміністратор |
| Вигоди компанії | Завдяки замовленням, кошти йтимуть в бюджет компанії |
| Частота користування | Залежить від потоку клієнтів але зазвичай часто  Адміністратор – рідко |
| Тригери | Було натиснуто кнопку «Замовити квиток» |
| Передумови | Користувачі знаходяться на сторінці сеансу |
| Постумови | Було створено нове замовлення |
| Основний розвиток | Було створено нове замовлення на конкретний сеанс, місце і час |
| Альтернативні розвитки | Через відсутність вільних місць користувач може не потрапити на бажаний час місце і навіть сеанс |
| Виняткові ситуації | – |

В таблиці 2.7 представлений сценарій використання «Перегляд інформації про сеанси»

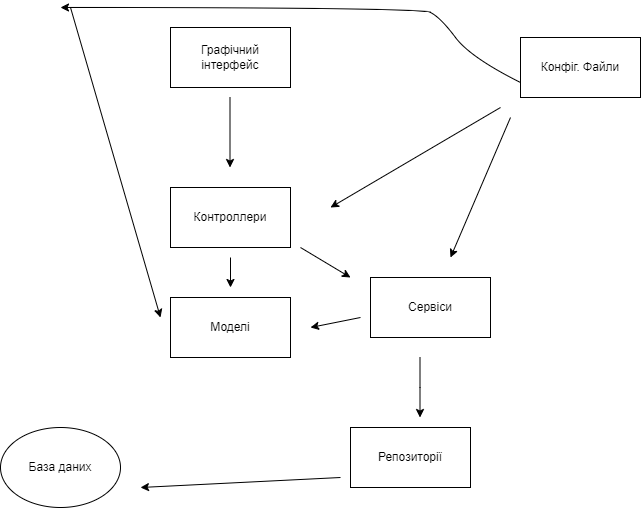
Таблиця 2.7 – Сценарій використання «Перегляд інформації про сеанси»

|  |  |
| --- | --- |
| Назва | Перегляд інформації про сеанси |
| ID | 7 |
| Опис | Користувачі можуть переглянути сторінку з сеансами та конкретно інформацію про кожен сеанс |
| Актори | Адміністратор, Гість |
| Вигоди компанії | На сторінці буде активність клієнтів, і це може призвести до покупок квитків |
| Частота користування | Дуже часто |
| Тригери | Було перейдено на сайт ресурсу або на сторінку перегляду інформації про конкретний сеанс |
| Передумови | Було перейдено на веб-ресурс |
| Постумови | Користувачі можуть переглянути потрібну їм інформацію про сеанси |
| Основний розвиток | Користувачі переходять на сторінку сайту, бачать перелік сеансів, або якщо переходять на сторінку конкретного сеансу, то бачать інформацію про сеанс |
| Альтернативні розвитки | – |
| Виняткові ситуації | Через відсутність будь яких сеансів або технічних збоїв користувачі не бачать ніяких сеансів та інформації |

# 3 АРХІТЕКТУРА СИСТЕМИ

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 Додатку Б

До серверної частини належать наступні елементи:

– контролер;

– модель та вигляд;

– сервіс;

– репозиторій;

– конфігураційні файли для авторизації та безпеки.

**Графічний інтерфейс:**

Користувач взаємодіє з графічним інтерфейсом, вводячи дані або взаємодіючи з елементами інтерфейсу.

**Контролери:**

Контролери отримують дані від графічного інтерфейсу та визначають, які дії потрібно виконати на основі цих даних.

Також вони створюють об'єкти та дані для відображення на графічному інтерфейсі.

**Сервіси:**

Контролери передають отримані дані в сервіси для виконання бізнес-логіки та обробки даних.

Сервіси виконують різні операції з даними в залежності від вимог.

**Моделі:**

Моделі в свою чергу використовуються для збереження та обробки інформації у сервісах та репозиторіях.

**Репозиторії:**

Репозиторії взаємодіють з моделями для збереження та отримання даних з бази даних. Є своєрідним мостом між БД та системою

Сервіси використовують репозиторії для роботи з даними, включаючи створення, оновлення, видалення та отримання даних.

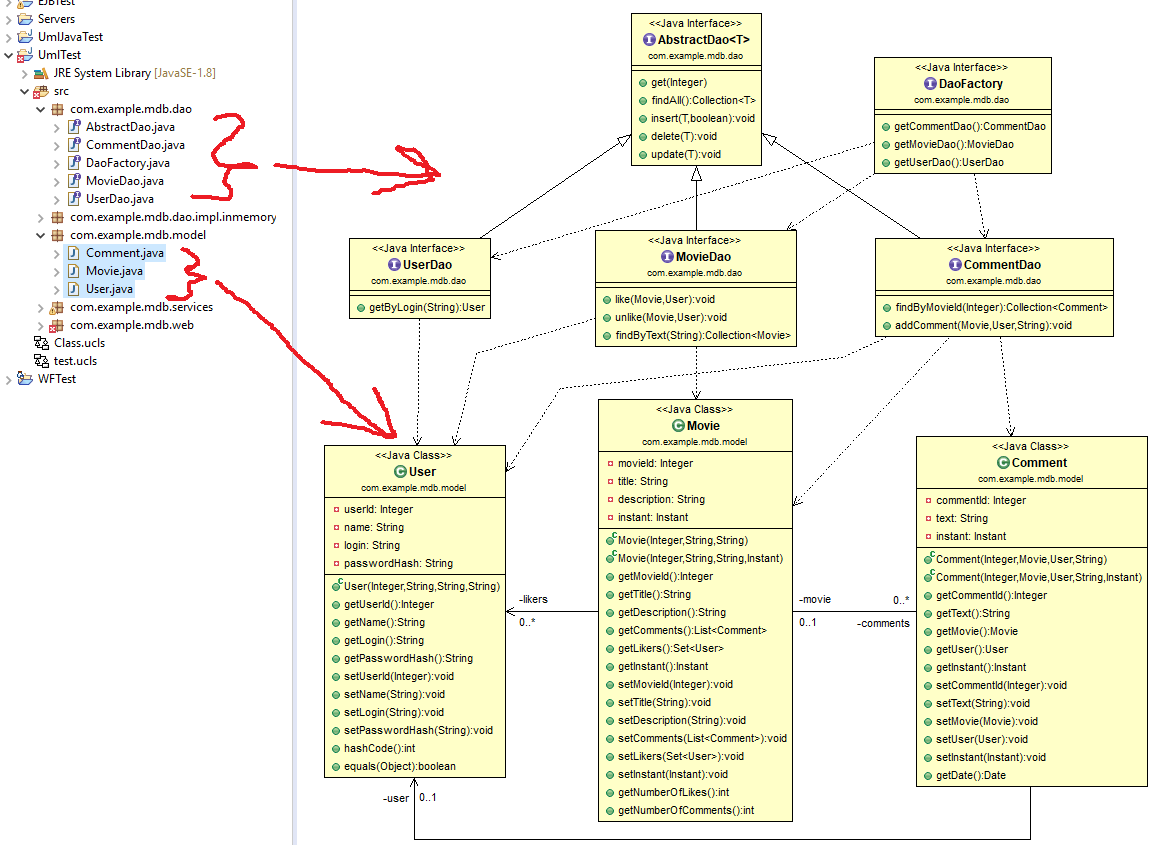
**Конфігураційні файли:**

Конфігураційні файли визначають налаштування та параметри сервісу, наприклад, права для відвідування певних сторінок, можливість редагування тощо.

# 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

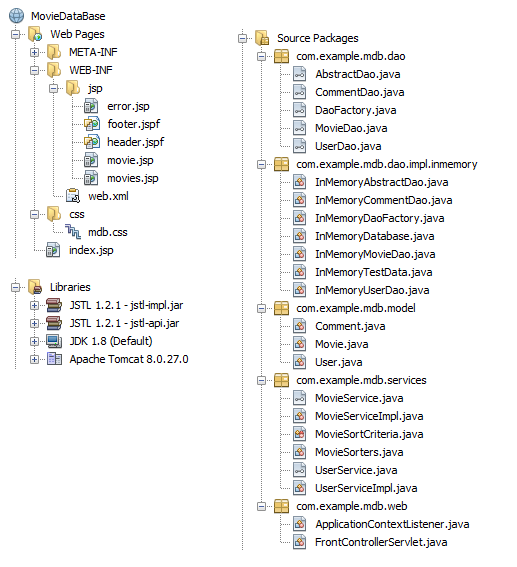


Рисунок 4.1 – Загальна структура проекту

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

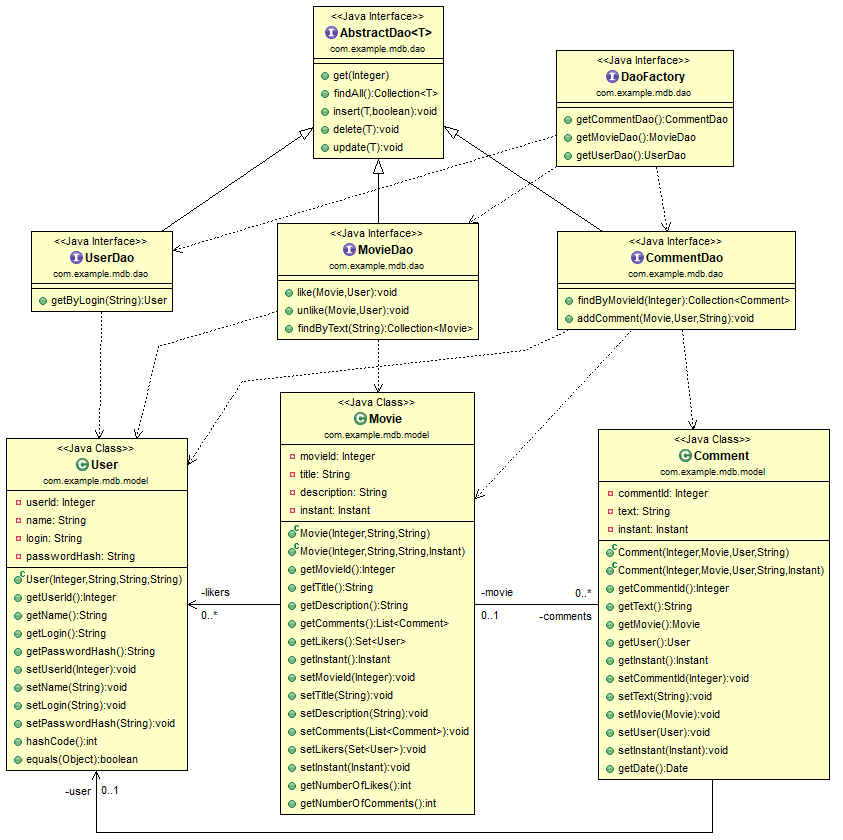


Рисунок 4.2 – Основні сутності та інтерфейси рівня доступу до даних

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

Обсяг: 1-2 сторінки.

На усі джерела мають бути посилання у тексті

1. J. Stephen Downie. Music information retrieval [Електронний ресурс] – Режим доступу до ресурсу: <http://www.music-ir.org/archive/downie_mir_arist37.pdf>
2. Дискретное преобразование Фурье. [Електронний ресурс] – Режим доступу до ресурсу: <http://ru.dsplib.org/content/dft.html>
3. Мел (высота звука) [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Mel_scale>
4. Pulse-code modulation [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Pulse-code_modulation>
5. Теорема Котельникова [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem>
6. Waveform Audio File Format. [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/WAV>
7. TIMIT Acoustic-Phonetic Continuous Speech Corpus [Електронний ресурс] – Режим доступу до ресурсу: <https://catalog.ldc.upenn.edu/LDC93S1>
8. CMU\_ARCTIC speech synthesis databases Електронний ресурс] – Режим доступу до ресурсу: <http://festvox.org/cmu_arctic/>
9. Lee et. al. Unsupervised feature learning for audio classification using convolutional deep-belif networks. [Електронний ресурс]. Режим доступу: <https://papers.nips.cc/paper/3674-unsupervised-feature-learning-for-audio-classification-using-convolutional-deep-belief-networks.pdf>
10. Tao Feng. Deep learning for music genre classification. [Електронний ресурс]. Режим доступу: <https://courses.engr.illinois.edu/ece544na/fa2014/Tao_Feng.pdf>
11. Haggblade et. al. Music genre classification. [Електронний ресурс]. Режим доступу: <http://cs229.stanford.edu/proj2011/HaggbladeHongKao-MusicGenreClassification.pdf>
12. George Tzanetakis et. al. Automatic Musical Genre Classification Of Audio Signals [Електронний ресурс]. Режим доступу: http://ismir2001.ismir.net/pdf/tzanetakis.pdf
13. About Last.fm [Електронний ресурс]. Режим доступу: <https://www.last.fm/about>
14. Last.fm – Wikipedia [Електронний ресурс]. Режим доступу <https://en.wikipedia.org/wiki/Last.fm>
15. Spotify – [Електронний ресурс]. Режим доступу: Wikipedia <https://en.wikipedia.org/wiki/Spotify>
16. М. Тім Джонс. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы [Електронний ресурс]. Режим доступу: <https://www.ibm.com/developerworks/ru/library/os-recommender1/>
17. Clustering
18. Pandora Radio [Електронний ресурс]. Режим доступу: <https://www.pandora.com>
19. How Pandora Radio Works [Електронний ресурс]. Режим доступу: http://computer.howstuffworks.com/internet/basics/pandora2.htm
20. VV Show #54 - Tim Westergren of Pandora [Електронний ресурс]. Режим доступу: <http://www.venturevoice.com/2009/03/vv_show_54_tim_westergren_of_p.html>
21. Mikolov et al. Distributed Representations of Words and Phrases and their Compositionality [Електронний ресурс]. Режим доступу: [http://web2.cs.columbia.edu/~blei/seminar/2016\_discrete\_data/readings/MikolovSutskeverChenCorradoDean2012.pdf](http://web2.cs.columbia.edu/~blei/seminar/2016_discrete_data/readings/MikolovSutskeverChenCorradoDean2013.pdf)
22. Vector Representations of Words [Електронний ресурс]. Режим доступу: <https://www.tensorflow.org/tutorials/word2vec>
23. How does Shazam work [Електронний ресурс]. Режим доступу: <http://coding-geek.com/how-shazam-works/>
24. Li-Chun Wang. An Industrial-Strength Audio Search Algorithm [Електронний ресурс]. Режим доступу: <https://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf>
25. Kyunghyun Cho, Bart van Merrienboer, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation [Електронний ресурс]. Режим доступу: <https://arxiv.org/abs/1406.1078>
26. Yaroslav Yuriiovych Dorogyi, Vasyl Vasylovych Tsurkan, Oleksandr S Khapilin Автоматичне розпізнавання музичних жанрів глибокими згортковими нейронними мережами [Електронний ресурс]. Режим доступу: <http://feltran.kpi.ua/article/view/105159/119678>
27. Aaron van den Oord, Yazhe Li, Oriol Vinyals Representation Learning with Contrastive Predictive Coding [Електронний ресурс]. Режим доступу: <https://arxiv.org/abs/1807.03748>

# 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).

# ДОДАТОК Б Назва додатку Б (за наявності)

Вміст додатку Б (за наявності)

В цьому додатку може бути:

* лістинг фрагментів коду;
* скан статті/тез доповіді на конференції/акта впровадження і т.п.;
* довідкові таблиці/рисунки.

# ДОДАТОК В Назва додатку В (за наявності)

Вміст додатку В (за наявності)

В цьому додатку може бути:

* лістинг фрагментів коду;
* скан статті/тез доповіді на конференції/акта впровадження і т.п.;
* довідкові таблиці/рисунки.