

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

## Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ  
ОРГАНІЗАЦІЇ КАМЕРНИХ КОНЦЕРТІВ  
DESIGN OF A SOFTWARE SYSTEM FOR  
CHAMBER CONCERT ORGANIZATION

Виконав(ла): студент(ка) 4 курсу, групи КНТ-213сп  
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

РЕЗНИК В.Д.

(ПРИЗВИЩЕ та ініціали)

Керівник ЛЬОВКІН В.М.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРИЗВИЩЕ та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет «Запорізька політехніка»**  
 (повне найменування закладу вищої освіти)

Факультет КНТ  
 Кафедра програмних засобів  
 Ступінь вищої освіти бакалавр  
 Спеціальність 122 Комп'ютерні науки  
(код і найменування)  
 Освітня програма (спеціалізація) Комп'ютерні науки  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
Сергій СУББОТІН  
 “ ” 2026 року

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

РЕЗНИКА Владислава Денисовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної системи для організації камерних концертів. Design of a Software System for Chamber Concert Organization

керівник проєкту (роботи) к.т.н., доцент, ЛЬОВКІН Валерій Миколайович,  
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 07 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 10 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програмної системи. 3. Розробка програмної системи. 4. Експлуатація та тестування програмної системи.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | ПРИЗВИЩЕ, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-4 Основна частина | ЛЬОВКІН В.М., доцент                      |                |                           |
| Нормоконтроль       | БСЛОВА А.В., асистент                     |                |                           |

7. Дата видачі завдання « 20 » квітня 2026 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                             | Строк виконання етапів проєкту ( роботи ) | Примітка     |
|-------|----------------------------------------------------------------------|-------------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                          | 1 тиждень                                 | Завдання, ТЗ |
| 2     | Аналіз предметної області.                                           | 1 тиждень                                 | Розділ 1     |
| 3     | Вибір мови програмування та інших технологій розробки.               | 2 тиждень                                 | Розділ 2     |
| 4     | Розробка архітектури програми.                                       | 2 тиждень                                 | Розділ 2     |
| 5     | Розробка програми.                                                   | 3-4 тижні                                 | Розділ 3, 4  |
| 6     | Тестування та експериментальне дослідження програмного забезпечення. | 5 тиждень                                 | Розділ 4     |
| 7     | Оформлення пояснювальної записки та документів до неї.               | 6 тиждень                                 | Додатки      |
| 8     | Нормоконтроль та рецензування.                                       | 7 тиждень                                 |              |
| 9     | Захист роботи.                                                       | 8 тиждень                                 |              |

Студент(ка)

\_\_\_\_\_  
( підпис )      Владислав РЕЗНИК  
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_  
( підпис )      Валерій ЛЬОВКІН  
(Ім'я ПРИЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 96 с., 3 табл., 26 рис., 3 дод., 10 джерел.

МУЗИЧНИЙ КОЛЕКТИВ, КАМЕРНИЙ КОНЦЕРТ, ПРИМІЩЕННЯ,  
ПРОГРАМНА СИСТЕМА, ФРЕЙМВОРК.

Об'єкт роботи – процес побудови програмних систем для організації камерних концертів. Предмет роботи – програмні системи для організації камерних концертів. Мета роботи – збільшення кількості варіантів організації камерних концертів за рахунок надання доступу власниками до невеликих приміщень за допомогою програмної системи.

Матеріали, методи та технічні засоби: мова Python та фреймворк Django як структуруючі одиниці створення програмної системи, система MySQL для роботи з даними.

Результати. У результаті створена програмна система, яка розроблена за результатами аналізу предметної області, виділення основних можливостей, визначених відмінностей з формулюванням концепції розробки, реалізована через розробку підсистем у вигляді моделей, представлень, шаблонів сторінок.

Висновки. Програмна система призначена для організації камерних концертів шляхом зв'язування між собою людей, які є власниками камерних приміщень та готові проводити в себе такі концерти, музикантів, які готові проводити камерні концерти, та відвідувачів, які готові відвідати такі концерти. Наявність такого майданчика повинна дозволити комунікувати всім сторонам між собою та сприяти за рахунок надання додаткової інформації від кожної сторони зручності роботи кожної зі сторін.

Галузь використання. Організація проведення камерних концертів, тобто музичних концертів у невеликих приміщеннях, в частині зв'язування між собою власників приміщень, музикантів та відвідувачів.

## ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 96 pages, 3 tables, 26 figures, 3 appendixes, 10 sources.

MUSIC GROUP, CHAMBER CONCERT, VENUE, SOFTWARE SYSTEM, FRAMEWORK.

The object of the work is the process of building software systems for organizing chamber concerts. The subject of the work is software systems for organizing chamber concerts. The purpose of the work is to increase the number of options for organizing chamber concerts by providing owners with access to small spaces using a software system.

Materials, methods and technical means: Python language and Django framework as structuring units for creating a software system, MySQL system for working with data.

Results. As a result, a software system was created, which was developed based on the results of the analysis of the subject area, highlighting the main capabilities, identified differences with the formulation of the development concept, implemented through the development of subsystems in the form of models, representations, page templates.

Conclusions. The software system is designed to organize chamber concerts by connecting people who are owners of chamber spaces and are ready to hold such concerts in their spaces, musicians who are ready to hold chamber concerts, and visitors who are ready to attend such concerts. The presence of such a platform should allow all parties to communicate with each other and facilitate the convenience of each party's work by providing additional information from each party.

Field of use. Organization of chamber concerts in terms of connecting owners of rooms, musicians and visitors.

## ЗМІСТ

|                                                                                                               |    |
|---------------------------------------------------------------------------------------------------------------|----|
|                                                                                                               | С. |
| Перелік скорочень та умовних познач.....                                                                      | 8  |
| Вступ.....                                                                                                    | 9  |
| 1 Аналіз предметної області.....                                                                              | 10 |
| 1.1 Огляд програмних рішень для організації камерних концертів .....                                          | 10 |
| 1.2 Порівняльний аналіз програмних рішень для організації камерних<br>концертів.....                          | 13 |
| 1.3 Висновки за розділом 1 .....                                                                              | 15 |
| 2 Проектування програмної системи .....                                                                       | 16 |
| 2.1 Вибір засобів проектування і розробки .....                                                               | 16 |
| 2.2 Моделювання роботи з програмною системою організації камерних<br>концертів.....                           | 17 |
| 2.3 Визначення структури бази даних програмної системи для організації<br>камерних концертів .....            | 25 |
| 2.4 Висновки за розділом 2 .....                                                                              | 35 |
| 3 Розробка програмної системи.....                                                                            | 36 |
| 3.1 Порядок організації роботи з користувачем в програмній системі для<br>організації камерних концертів..... | 36 |
| 3.2 Рішення з реалізації програмної системи для організації камерних<br>концертів.....                        | 37 |
| 3.3 Висновки за розділом 3 .....                                                                              | 51 |
| 4 Експлуатація та тестування програмної системи .....                                                         | 52 |
| 4.1 Призначення програмної системи .....                                                                      | 52 |
| 4.2 Умови виконання програмної системи .....                                                                  | 52 |
| 4.3 Експлуатація програмної системи для організації камерних концертів.....                                   | 52 |
| 4.4 Тестування програмної системи організації камерних концертів .....                                        | 55 |
| 4.5 Висновки за розділом 4 .....                                                                              | 58 |
| Висновки .....                                                                                                | 59 |

|                                   |    |
|-----------------------------------|----|
| Перелік джерел посилання .....    | 61 |
| Додаток А Технічне завдання ..... | 62 |
| Додаток Б Текст програми .....    | 68 |
| Додаток В Слайди презентації..... | 89 |

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**

БД – база даних;

ПС – програмна система.

## ВСТУП

Камерні концерти розраховані на невелику аудиторію, а тому проводитись можуть в різних невеликих приміщеннях. Це може бути і невеликий концерт на вулиці. Відповідно варіантів приміщень, в яких можуть проводитись такі концерти, є доволі багато. Якщо масштабний концерт дуже обмежений в потенційних майданчиках, то камерний концерт може проводитись і в місцях, які зазвичай не використовуються для проведення концертів. Наприклад, це може бути певний музей або подібний майданчик, який готовий прийняти в себе концерт. Музиканти, які проводять такий концерт, можуть мати різний рівень, тож загалом варіантів для організації концертів може бути багато. Окрім того є і відвідувачі, які зацікавлені саме в таких камерних концертах та готові відвідувати їх за участі музикантів різного рівня. Відповідно необхідна програмна система, яка б зв'язала всіх цих учасників між собою.

Мета роботи, покладаючись на описану актуальність, визначена як збільшення кількості варіантів організації камерних концертів за рахунок надання доступу власниками до невеликих приміщень за допомогою програмної системи (ПС).

Завдання роботи:

- аналіз проблеми розробки ПС для організації камерних концертів;
- проєктування ПС для організації камерних концертів;
- розробка ПС для організації камерних концертів.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Огляд програмних рішень для організації камерних концертів

DICE (рис. 1.1) є платформою, яка пов'язує між собою артистів, представників місць проведення концертів та промоутерів [1]. Це дещо відрізняє дане програмне рішення від предметної області, яка саме досліджується в роботі. По-перше, в даній роботі не приділяється окрема увага промоутерам. Оскільки музичний колектив може не мати промоутера, але якщо він його має, то представник у програмі музичного колективу і є таким промоутером. Тобто це не скасовує такої можливості, однак це не є направленістю застосунку, що розробляється. Камерні концерти є невеликими, музиканти можуть бути тільки на початку своєї кар'єри, тому промоутери не є обов'язковою складовою процесів. По-друге, DICE не враховує особливостей камерних концертів та приміщень для таких концертів, які взагалі типово можуть використовуватися для інших процесів.

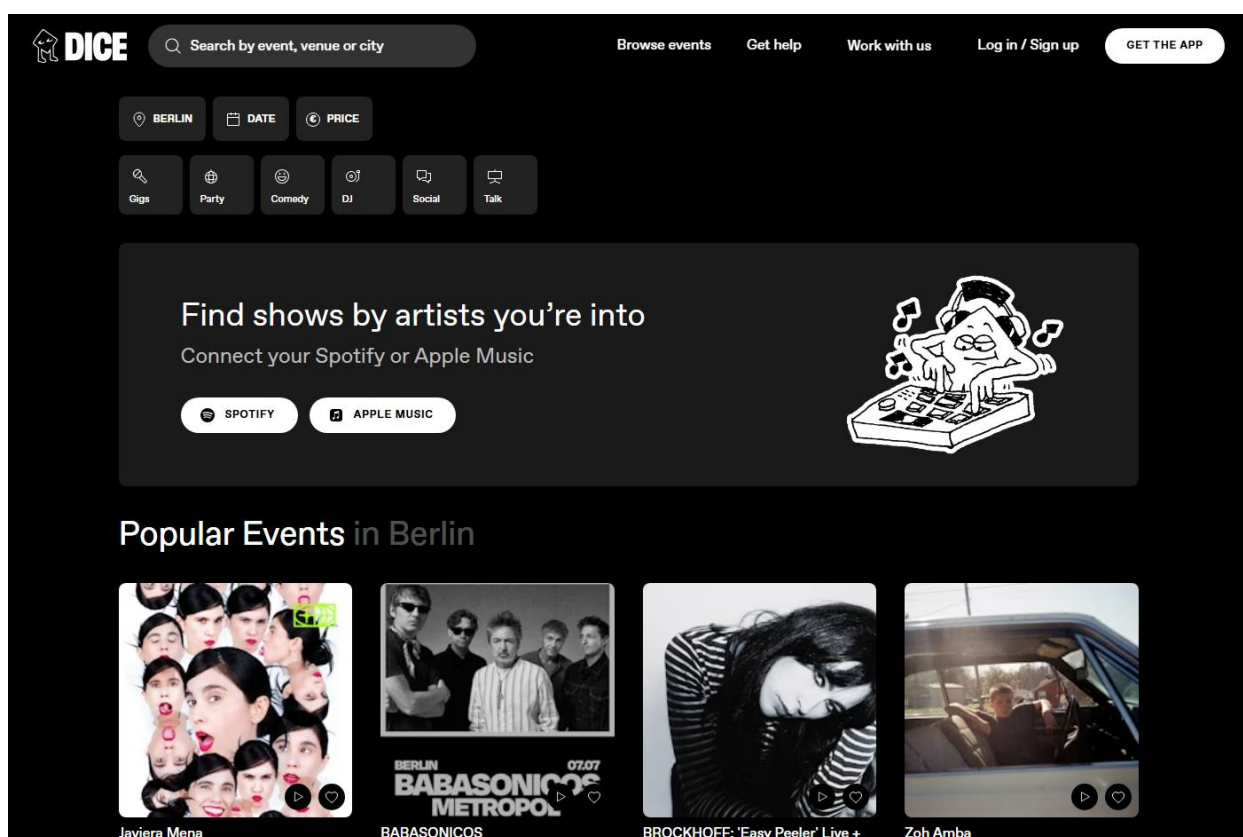


Рисунок 1.1 – Програма DICE

З іншого боку DICE дозволяє шукати концерти, місця проведення концертів, виконавців. Програма дозволяє не тільки бронювати місця, але і оплачувати їх вартість, змінювати квитки на події з друзями. Можливо підписатись на місця проведення музичних концертів, на музичні колективи, в результаті чого можна переглядати заплановані концерти за ними.

Fever (рис. 1.2) надає можливості пошуку не тільки камерних концертів, але і не тільки музичних концертів [2]. Застосунок об'єднує в собі можливості пошуку серед різних подій, включаючи нічні події, різного роду культурні події, події, пов'язані з їжею, кінотеатральні події, спортивні події. Тому специфіку саме камерних концертів застосунок не враховує. Фактично це застосунок, який дозволяє шукати різного роду події, бронювати місця на цих подіях. Користувачі можуть залишати відгуки про події. При цьому місця проведення подій не можуть мати відгуків, на них не можна підписатися, щоб потім бачити перелік запланованих у них подій.

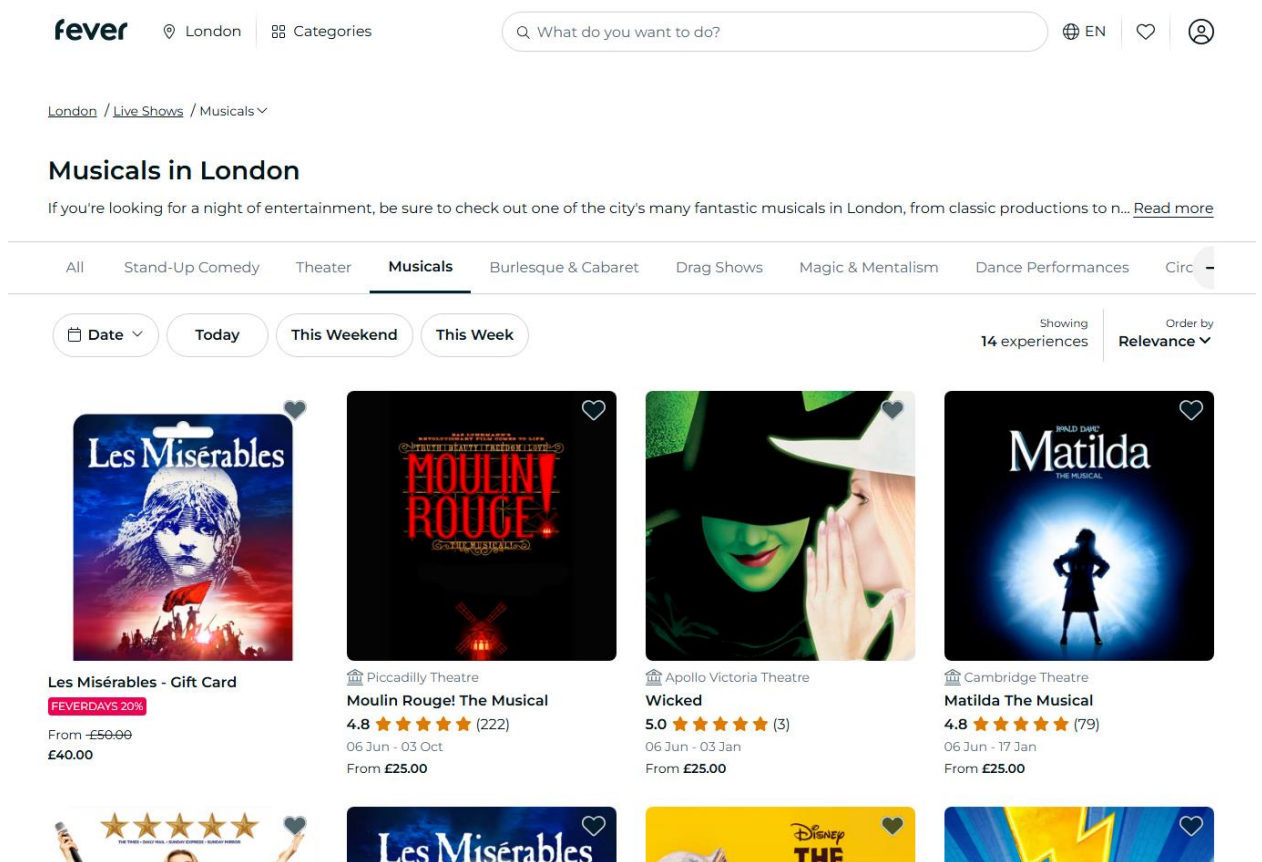


Рисунок 1.2 – Програма Fever

Програма Songkick (рис. 1.3) на відміну від попередньої зконцентрована саме на музичних подіях, при чому велику частку цих подій займають саме камерні концерти [3]. Надається можливість придбати квитки на концерти, при тому залишити відгуки про концерти, місця проведення подій не можна. Місця проведення концертів мають свої окремі сторінки, можна шукати за ними. При цьому так чи інакше підписатися на місце проведення концертів, щоб потім бачити концерти, які за цим місцем заплановані, не можна. Відповідно фактично функціональність направлена на інформування про музичні концерти, в тому числі камерні, та придбання квитків на такі концерти.

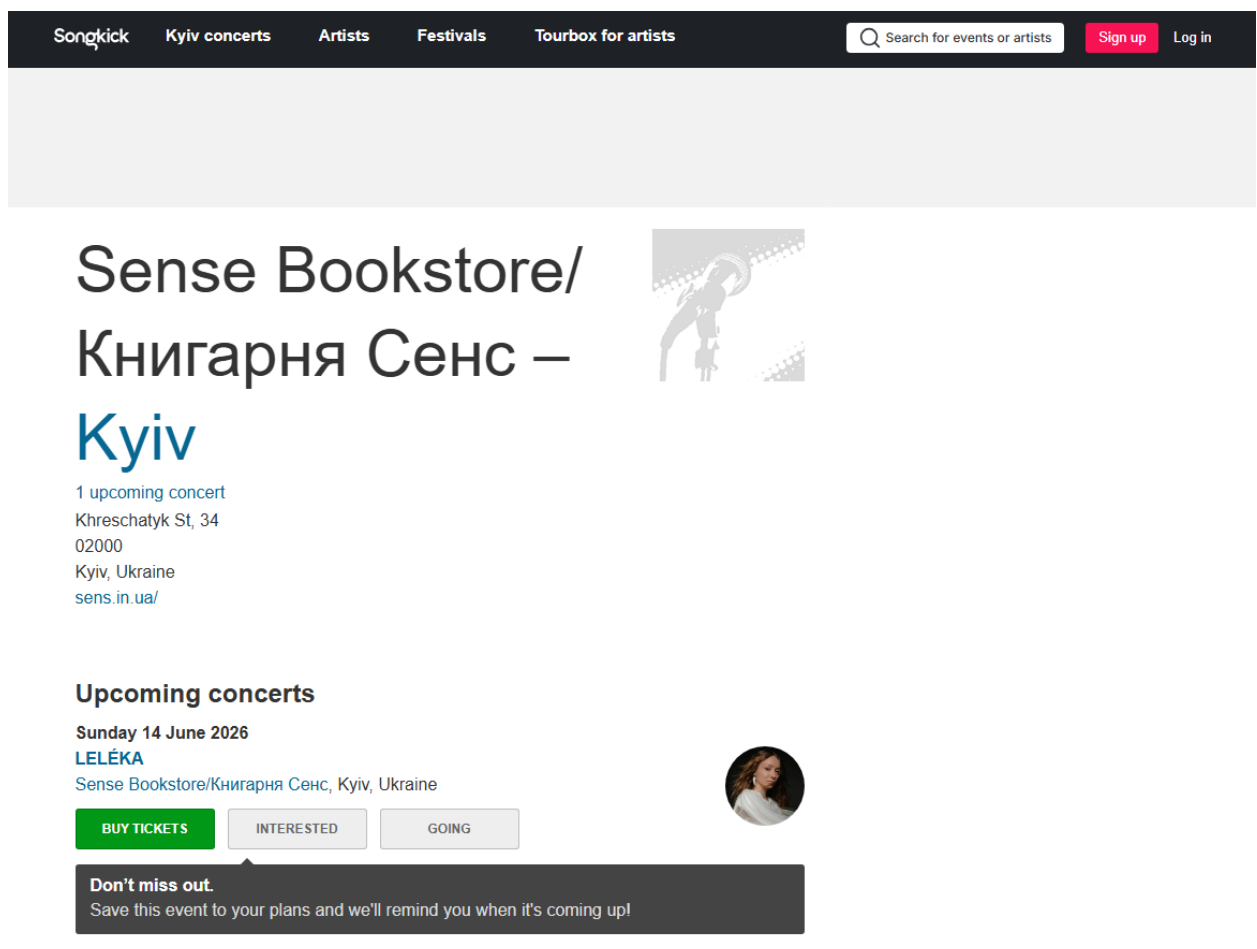


Рисунок 1.3 – Програма Songkick

## **1.2 Порівняльний аналіз програмних рішень для організації камерних концертів**

Порівняльний аналіз програмних рішень для організації камерних концертів виконано за програмами DICE, Fever, Songkick та власною розробкою, яку заплановано в роботі (табл. 1.1).

Особливостями розробки ПС для організації камерних концертів є те, що вона дозволяє адресувати запити на проведення концертів саме у програмі між власниками приміщень для проведення камерних концертів та представниками музичних колективів, окрім того так само як і в DICE представники приміщень для концертів (у даному випадку камерних) можуть додаватись до програми і таким чином розширювати кількість пропозицій місць для проведення концертів. У двох інших аналогів набір місць проведення концертів стандартний.

Слідкування за концертами у розробці передбачає можливість зберегти місця проведення, колективи для того, щоб потім за цим переліком дізнатись про нові заплановані у них концерти. З аналогів тільки в DICE є подібна можливість. Не всі аналоги мають можливість пошуку за місцями проведення концертів. Зокрема Fever такої можливості не має.

Окрім того відгуки у розробці можливі про місця проведення, концерти та колективи. У аналогів всього цього набору немає, вони обмежені фактично тільки одним варіантом. Відповідно тоді досвід використання нових приміщень для проведення камерних концертів, що у випадку пропонування нових місць, є важливим, не враховується.

Тож в підсумку це дозволяє визначити, що розробка ПС для організації камерних концертів має свої особливості, які виділяють її порівняно з аналогами, при чому застосовуючи їх основні можливості для роботи в сфері камерних концертів.

Таблиця 1.1 – Порівняння програмних рішень для організації камерних концертів

| Критерій                                    | DICE                                                 | Fever             | Songkick                      | Розробка                                               |
|---------------------------------------------|------------------------------------------------------|-------------------|-------------------------------|--------------------------------------------------------|
| Коло місць проведення концертів             | Може розширюватись                                   | Стандартне        | Стандартне, включаючи камерні | Може розширюватись за рахунок нестандартних (камерних) |
| Слідкування за концертами                   | Так, в тому числі за місцями проведення, колективами | Тільки пошук      | Тільки пошук                  | Так, в тому числі за місцями проведення, колективами   |
| Пошук за місцями проведення концертів       | Так                                                  | Ні                | Так                           | Так                                                    |
| Адресування запитів на проведення концертів | Ні                                                   | Ні                | Ні                            | Так                                                    |
| Відгуки                                     | Ні                                                   | Так, про концерти | Так, про концерти колективу   | Так, про місця проведення, концерти та колективи       |

### **1.3 Висновки за розділом 1**

Перший розділ роботи представляє основні результати, отримані під час аналізу проблеми розробки ПС для організації камерних концертів. Серед аналогів було проаналізовано застосунки DICE, Fever, Songkick. Виконувана розробка була визначена стосовно відмінностей, присутніх у її можливостях, які головним чином стосуються адресування запитів на проведення камерних концертів у самій програмі, що дозволяє в підсумку створити майданчик для комунікації між власниками камерних приміщень, представниками музичних колективів та відвідувачами камерних концертів.

## 2 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Вибір засобів проєктування і розробки

Розробку ПС для організації камерних концертів вирішено забезпечити через підтримку в основі мовою програмування Python. Подальший вибір базувався на цьому і включав порівняння двох Python-фреймворків, включаючи Django [4]-[5] і Pyramid [6]-[7] (табл. 2.1). Як і вибір мови Python, так і вибір фреймворку базувались в першу чергу на скороченні часу розробки. Це було критичним фактором прийняття рішення тому, що розробка згідно з календарним планом була дуже обмеженою в часі, враховуючи обсяг функціональних можливостей, які треба було розробити.

Таблиця 2.1 – Порівняння фреймворків для створення ПС для організації камерних концертів

| Критерій                        | Django                                                         | Pyramid                                                                       |
|---------------------------------|----------------------------------------------------------------|-------------------------------------------------------------------------------|
| Витрати на розробку             | Менші за часом                                                 | Більші за часом                                                               |
| Спосіб взаємодії з базою даних  | На основі вбудованого об'єктно-реляційного відображення        | На основі об'єктно-реляційного відображення, яке потрібно обрати і встановити |
| Структура ПС                    | Регламентована за підсистемами представлень, моделей, шаблонів | Змінювана                                                                     |
| Засоби авторизації користувачів | Наявні вбудовані                                               | Підлягають налаштуванню                                                       |

З одного боку враховуючи, що безпосередньо розробка на основі Django зазвичай триває менше ніж на основі Pyramid, а також те, що Django

має готовий налаштований спосіб організації взаємодії з базою даних (БД), авторизації користувачів, визначену структуру ПС, саме Django обраний фреймворком розробки.

## **2.2 Моделювання роботи з програмною системою організації камерних концертів**

Сценарії роботи користувача до авторизації з ПС організації камерних концертів включають у основному потоці:

- авторизація;
- пошук приміщень для проведення камерних концертів;
- пошук колективів, які проводять камерні концерти;
- пошук камерних концертів;
- перегляд даних камерного концерту;
- перегляд даних приміщення для проведення камерного концерту;
- перегляд даних музичного колективу для камерних концертів;
- перегляд відгуків за камерним концертом;
- перегляд відгуків за музичним колективом;
- перегляд запланованих камерних концертів колективу;
- перегляд проведених камерних концертів колективу.
- перегляд відгуків за приміщенням для проведення камерного концерту (рис. 2.1).

Альтернативний сценарій користувач до авторизації має тільки один – реєстрація, який активується тоді, коли користувач не має акаунту.

Сценарії роботи відвідувача з ПС організації камерних концертів включають у основному потоці додатково до сценаріїв користувача до авторизації:

- перегляд заброньованих камерних концертів;
- бронювання місця на камерному концерті;
- збереження камерного концерту;

- перегляд збережених камерних концертів;
- збереження колективу для камерних концертів;
- перегляд збережених колективів для камерних концертів;
- створення відгуку про проведений камерний концерт.

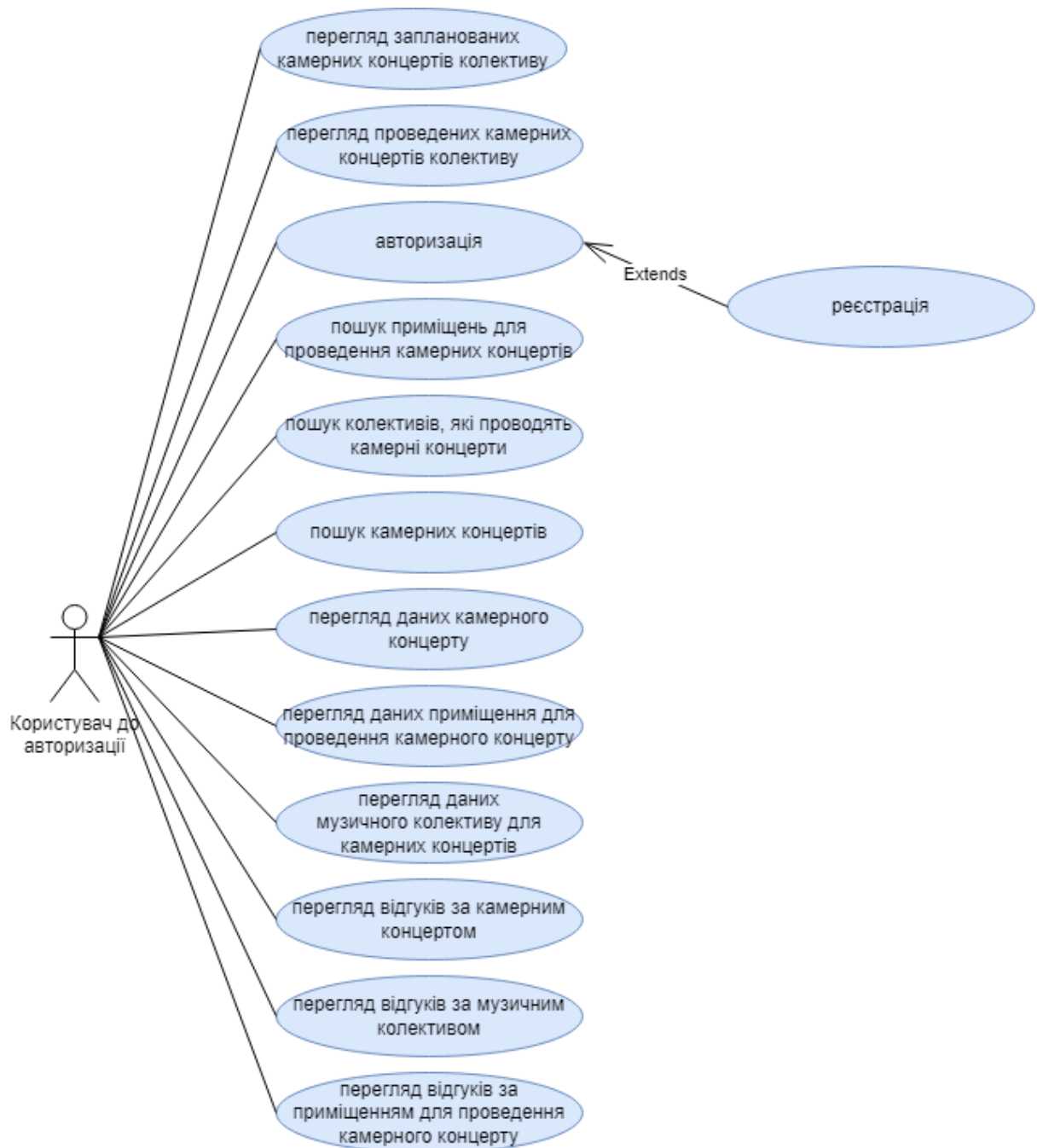


Рисунок 2.1 – Діаграма прецедентів для користувача ПС організації камерних концертів до авторизації



Також сценарії роботи відвідувача з ПС організації камерних концертів включають у основному потоці:

- збереження приміщення для проведення камерних концертів;
- перегляд збережених приміщень для проведення камерних концертів;
- перегляд запланованих камерних концертів за збереженими приміщеннями;
- перегляд запланованих камерних концертів за збереженими колективами.

Альтернативні сценарії для роботи відвідувача з ПС організації камерних концертів:

- скасування бронювання місця на камерному концерті для бронювання місця на камерному концерті, коли бронювання місця на камерному концерті виконане цим відвідувачем, камерний концерт ще не розпочався, а відвідувач не зможе відвідати концерт;
- скасування збереження камерного концерту для збереження камерного концерту, коли збереження цього камерного концерту цим відвідувачем виконане, але більше не є актуальним;
- скасування збереження колективу для камерних концертів для збереження колективу для камерних концертів, коли збереження цього колективу для камерних концертів цим відвідувачем виконане, але більше не є актуальним;
- скасування збереження приміщення для проведення камерних концертів для збереження приміщення для проведення камерних концертів, коли збереження цього приміщення для проведення камерних концертів цим відвідувачем виконане, але більше не є актуальним.

Сценарії роботи власника приміщення для проведення камерних концертів з ПС організації камерних концертів включають у основному потоці додатково до сценаріїв користувача до авторизації:

- збереження колективу для камерних концертів;
- перегляд збережених колективів для камерних концертів;

- визначення власного приміщення для проведення камерних концертів;
- перегляд власних приміщень для проведення камерних концертів;
- пропонування власного приміщення для проведення камерного концерту;
- підтвердження запиту на проведення камерного концерту в приміщенні;
- перегляд власних запитів на проведення камерного концерту в приміщенні;
- перегляд адресованих запитів на проведення камерних концертів за приміщенням;
- перегляд адресованих запитів на проведення камерних концертів за приміщеннями власника;
- перегляд запланованих камерних концертів за збереженими колективами.

Альтернативні сценарії роботи власника приміщення для проведення камерних концертів з ПС організації камерних концертів:

- скасування збереження колективу для камерних концертів для збереження колективу для камерних концертів, коли збереження цього колективу для камерних концертів цим власником приміщення для проведення камерних концертів виконане, але більше не є актуальним;
- скасування запиту на проведення камерного концерту в приміщенні для підтвердження запиту на проведення камерного концерту в приміщенні, коли такий запит на проведення камерного концерту в приміщенні не може бути виконаний цим власником приміщення для проведення камерних концертів;
- зняття власного приміщення для проведення камерних концертів з розгляду для визначення власного приміщення для проведення камерних концертів, коли це власне приміщення для проведення камерних концертів представлено в ПС, але більше не надається для проведення камерних

концертів власником;

– редагування даних власного приміщення для проведення камерних концертів для визначення власного приміщення для проведення камерних концертів, коли це власне приміщення для проведення камерних концертів визначене в ПС, але зазначені дані не є повністю актуальними (рис. 2.3).

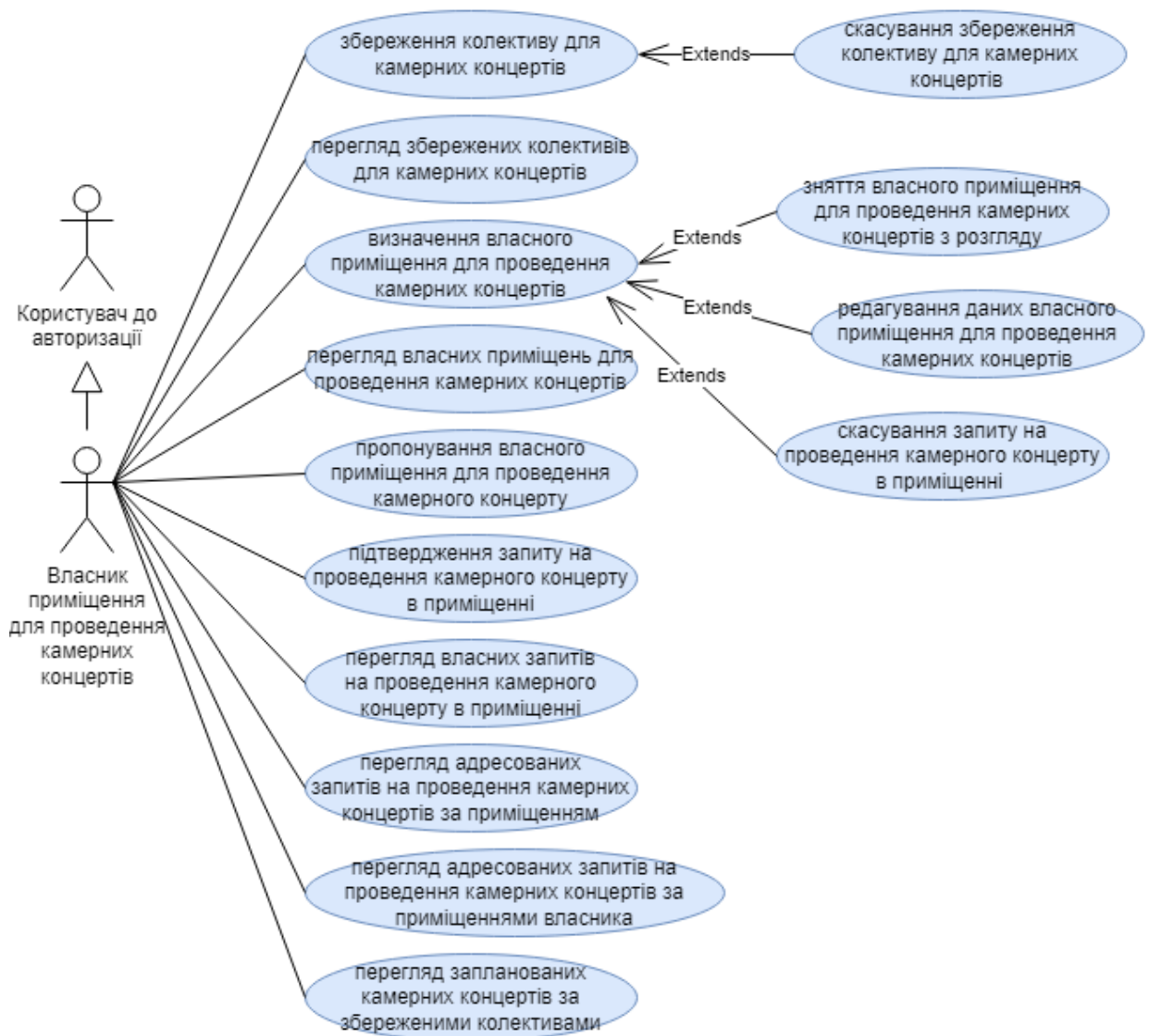


Рисунок 2.3 – Діаграма прецедентів для власника приміщення для проведення камерних концертів у ПС організації камерних концертів

Сценарії роботи представника музичного колективу з ПС організації камерних концертів включають у основному потоці додатково до сценаріїв користувача до авторизації:

- перегляд переліку відвідувачів камерного концерту;
- запит на проведення камерного концерту в приміщенні;
- підтвердження запиту на проведення камерного концерту в приміщенні;
- перегляд власних запитів на проведення камерного концерту в приміщенні;
- перегляд адресованих запитів на проведення камерних концертів;
- визначення власного колективу для камерних концертів;
- перегляд власних запланованих камерних концертів;
- перегляд власних проведених камерних концертів;
- перегляд власних камерних концертів на розгляді;
- перегляд запланованих камерних концертів за збереженими приміщеннями;
- перегляд збережених приміщень для проведення камерних концертів;
- збереження приміщення для проведення камерних концертів;
- перегляд запланованих камерних концертів за збереженими приміщеннями.

Альтернативні сценарії роботи представника музичного колективу з ПС організації камерних концертів:

- скасування запиту на проведення камерного концерту в приміщенні для підтвердження запиту на проведення камерного концерту в приміщенні, коли такий запит на проведення камерного концерту в приміщенні отримано представником музичного колективу, але він не погоджується на це;
- скасування збереження приміщення для проведення камерних концертів для збереження приміщення для проведення камерних концертів, коли збереження цього приміщення для проведення камерних концертів цим представником музичного колективу виконане, але більше не є актуальним;
- редагування даних власного колективу для камерних концертів для визначення власного колективу для камерних концертів, коли цей власний колектив для камерних концертів визначений в ПС, але зазначені дані не є

повністю актуальними (рис. 2.4).

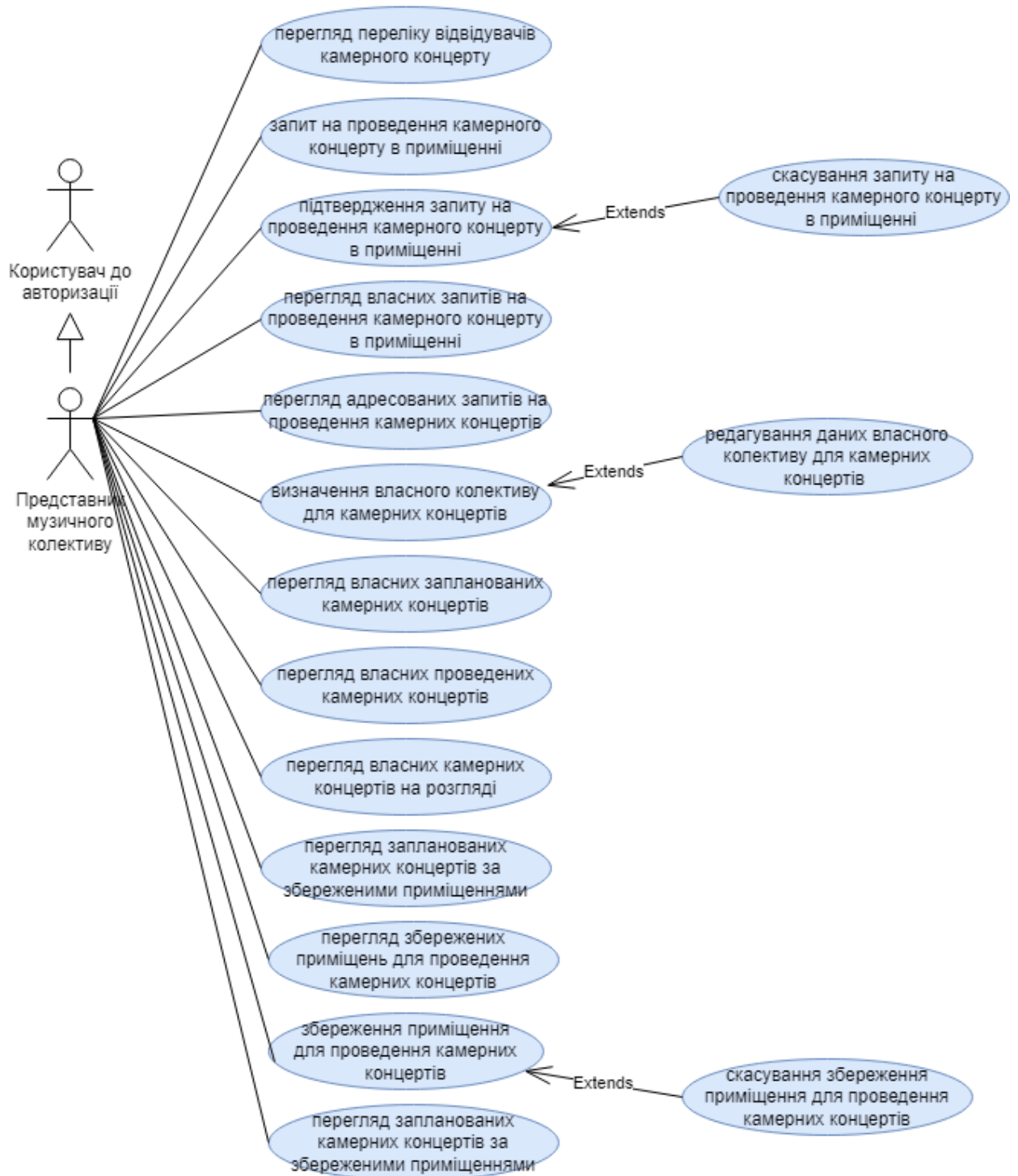


Рисунок 2.4 – Діаграма прецедентів для представника музичного колективу в ПС організації камерних концертів

### **2.3 Визначення структури бази даних програмної системи для організації камерних концертів**

БД реалізована на основі системи MySQL зі створенням описаної нижче структури.

Таблиця `Chamberspacacity` зберігає дані міст, в яких можуть проводитись камерні концерти, зі структурою стовпців:

- `id` – ідентифікатор міста, в якому можуть проводитись камерні концерти;
- `chamberspacacityname` – назва міста, в якому можуть проводитись камерні концерти.

Таблиця `Chamberspacetype` зберігає дані типів приміщень для проведення камерного концерту зі структурою стовпців:

- `id` – ідентифікатор типу приміщення для проведення камерного концерту;
- `chamberspacetypename` – назва типу приміщення для проведення камерного концерту.

Таблиця `Chamberspace` зберігає дані приміщень для проведення камерних концертів зі структурою стовпців:

- `id` – ідентифікатор приміщення для проведення камерного концерту;
- `chamberspaceownerid` – власник, який у програмі представляє це приміщення для проведення камерного концерту: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspaceowner`;
- `chamberspacetitle` – назва приміщення для проведення камерного концерту;
- `chamberspacedescription` – опис приміщення для проведення камерного концерту;
- `chamberspacetypeid` – тип приміщення для проведення камерного концерту, до якого належить це приміщення: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspacetype`;

– `chamberspacecityid` – місто, в якому знаходиться це приміщення для проведення камерного концерту: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspacecity`;

– `chamberspaceaddress` – адреса, за якою знаходиться це приміщення для проведення камерного концерту;

– `chamberspacereviewmark` – усереднена оцінка, виставлена у відгуках цьому приміщенню для проведення камерного концерту;

– `chambersspaceplacenum` – кількість місць, доступна у цьому приміщенні для проведення камерного концерту;

– `chambersspacephone` – номер телефону для з'єднання з приміщенням для проведення камерного концерту.

Таблиця `Chambersspacephoto` зберігає дані фотографій приміщення для проведення камерного концерту зі структурою стовпців:

– `id` – ідентифікатор фотографії приміщення для проведення камерного концерту;

– `chamberspaceid` – приміщення для проведення камерного концерту, показане на фотографії: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspace`;

– `chambersspacephotodata` – сама фотографія приміщення для проведення камерного концерту;

– `chambersspacephotodescription` – опис фотографії приміщення для проведення камерного концерту.

Таблиця `Chamberspacereview` зберігає дані відгуків за приміщенням для проведення камерного концерту зі структурою стовпців:

– `id` – ідентифікатор відгуку за приміщенням для проведення камерного концерту;

– `chamberspaceid` – приміщення для проведення камерного концерту, за яким створено відгук: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspace`;

– `chambervisitorid` – відвідувач, який створив цей відгук за приміщенням для проведення камерного концерту: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `User`;

– `chamberspacereviewtexted` – текст відгуку за приміщенням для проведення камерного концерту;

– `chamberspacereviewdt` – дата і час створення відгуку за приміщенням для проведення камерного концерту;

– `chamberspacereviewmark` – оцінка відгуку за приміщенням для проведення камерного концерту.

Таблиця `Chamberspacereviewphoto` зберігає дані фотографії, доданої до відгуку за приміщенням для проведення камерного концерту, зі структурою стовпців:

– `id` – ідентифікатор фотографії, доданої до відгуку за приміщенням для проведення камерного концерту;

– `chamberspacereviewid` – відгук за приміщенням для проведення камерного концерту, до якого додано цю фотографію: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspacereview`;

– `chamberspacereviewphotodata` – дані фотографії, доданої до відгуку за приміщенням для проведення камерного концерту.

Таблиця `Chamberspacemusiciansave` зберігає дані збережених представниками музичних колективів приміщень для проведення камерних концертів зі структурою стовпців:

– `id` – ідентифікатор збереження представником музичного колективу приміщення для проведення камерних концертів;

– `chamberspaceid` – приміщення для проведення камерних концертів, яке збережено цим представником музичного колективу: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspace`;

– `chambermusicianid` – представник музичного колективу, який зберіг це приміщення для проведення камерних концертів: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chambermusician`.

Таблиця `Chamberspacevisitorsave` зберігає дані збережених відвідувачами приміщень для проведення камерних концертів зі структурою стовпців:

- `id` – ідентифікатор збереження відвідувачем приміщення для проведення камерних концертів;
- `chamberspaceid` – приміщення для проведення камерних концертів, яке збережено відвідувачем: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspace`;
- `chambervisitorid` – відвідувач, який зберіг це приміщення для проведення камерних концертів: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `User`.

Таблиця `Chamberspaceconcertinquiry` зберігає дані запитів на проведення камерних концертів в приміщеннях зі структурою стовпців:

- `id` – ідентифікатор запиту на проведення камерного концерту в приміщенні;
- `chamberspaceid` – приміщення для проведення камерного концерту, проведення в якому запитано музичним колективом або власник якого запитів у музичного колективу можливість проведення концерту в нього: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspace`;
- `chamberconcertid` – камерний концерт, проведення якого запитано в цьому приміщенні для проведення камерних концертів: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberconcert`;
- `chamberinquirystate` – стан запиту на проведення камерного концерту в приміщенні;
- `chamberinquirydt` – дата і час створення запиту на проведення камерного концерту в приміщенні;
- `chamberinquirytexted` – текст примітки до запиту на проведення камерного концерту в приміщенні від власника приміщення для проведення камерних концертів;

– `chamberinquirybandtexted` – текст примітки до запиту на проведення камерного концерту в приміщенні від представника музичного колективу;

– `chamberinquiryinitializer` – сторона, яка ініціювала запит на проведення камерного концерту в приміщенні: власник приміщення або представник музичного колективу.

Таблиця `Chamberspaceowner` зберігає дані користувачів, які є власниками приміщень для проведення камерних концертів, зі структурою стовпців:

– `id` – ідентифікатор користувача, який є власником приміщення для проведення камерного концерту;

– `chamberuseraccountid` – акаунт у програмі, який має цей власник приміщення для проведення камерних концертів: встановлено зовнішнім ключем зі зв'язком один-до-одного до таблиці `User`;

– `chamberspaceownerphone` – номер телефону користувача, який є власником приміщення для проведення камерного концерту.

Таблиця `Chambermusicband` зберігає дані музичних колективів, які проводять камерні концерти, зі структурою стовпців:

– `id` – ідентифікатор музичного колективу, який проводить камерні концерти;

– `chambermusicbandtitle` – назва музичного колективу, який проводить камерні концерти;

– `chambermusicbanddescription` – опис музичного колективу, який проводить камерні концерти;

– `chambermusicstyle` – основний стиль, в якому виконує музику цей музичний колектив, який проводить камерні концерти;

– `chambermusicbandurl` – вебадреса сайту музичного колективу, який проводить камерні концерти;

– `chambermusicbandurladditional` – додаткова вебадреса сайту музичного колективу, який проводить камерні концерти;

– `chambermusicbandreviewmark` – усереднена оцінка, виставлена у відгуках музичному колективу.

Таблиця `Chambermusicbandphoto` зберігає дані фотографій музичних колективів зі структурою стовпців:

– `id` – ідентифікатор фотографії музичного колективу, який проводить камерні концерти;

– `chambermusicbandid` – музичний колектив, який проводить камерні концерти: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chambermusicband`;

– `chambermusicbandphotodata` – сама фотографія музичного колективу.

Таблиця `Chambermusicbandvideo` зберігає дані відео виступів музичних колективів, які проводять камерні концерти, зі структурою стовпців:

– `id` – ідентифікатор відео виступу музичного колективу, який проводить камерні концерти;

– `chambermusicbandid` – музичний колектив, який проводить камерні концерти та виступ якого показаний на відео: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chambermusicband`;

– `chambermusicbandvideodata` – саме посилання на відео виступу музичного колективу, який проводить камерні концерти.

Таблиця `Chambermusicbandreview` зберігає дані відгуків про музичні колективи, які проводять камерні концерти, зі структурою стовпців:

– `id` – ідентифікатор відгуку про музичний колектив, який проводить камерні концерти;

– `chambermusicbandid` – музичний колектив, який проводить камерні концерти та відгук на який створено: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chambermusicband`;

– `chambervisitorid` – відвідувач, який створив цей відгук про музичний колектив, який проводить камерні концерти: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `User`;

– `chambermusicbandreviewtexted` – текст відгуку про музичний колектив, який проводить камерні концерти;

– `chambermusicbandreviewdt` – дата і час створення відгуку про музичний колектив, який проводить камерні концерти;

– `chambermusicbandreviewmark` – оцінка відгуку про музичний колектив, який проводить камерні концерти.

Таблиця `Chambermusicbandreviewphoto` зберігає дані фотографій, прикріплених до відгуків про музичні колективи, які проводять камерні концерти, зі структурою стовпців:

– `id` – ідентифікатор фотографії, прикріпленої до відгуку про музичний колектив, який проводить камерні концерти;

– `chambermusicbandreviewid` – відгук про музичний колектив, який проводить камерні концерти, до якого прикріплено фотографію: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chambermusicband`;

– `chambermusicbandreviewphotodata` – сама фотографія, прикріплена до відгуку про музичний колектив, який проводить камерні концерти.

Таблиця `Chambermusicbandownersave` зберігає дані збережених власниками приміщень для проведення камерних концертів музичних колективів зі структурою стовпців:

– `id` – ідентифікатор збереження власником приміщення для проведення камерного концерту музичного колективу;

– `chambermusicbandid` – музичний колектив, який збережено власником приміщення для проведення камерного концерту: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chambermusicband`;

– `chamberspaceownerid` – власник приміщення для проведення камерного концерту, який зберіг цей музичний колектив: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspaceowner`.

Таблиця `Chambermusicbandvisitorsave` зберігає дані збережених відвідувачами музичних колективів зі структурою стовпців:

- id – ідентифікатор збереження відвідувачем музичного колективу;
- chambermusicbandid – музичний колектив, який збережено цим відвідувачем: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці Chambermusicband;

- chambervisitorid – відвідувач, який зберіг музичний колектив: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці User.

Таблиця Chambermusician зберігає дані представників музичних колективів зі структурою стовпців:

- id – ідентифікатор представника музичного колективу;
- chamberuseraccountid – акаунт у програмі, який має цей представник музичного колективу: встановлено зовнішнім ключем зі зв'язком один-до-одного до таблиці User;

- chambermusicianphone – номер телефону представника музичного колективу;

- chambermusicbandid – музичний колектив, який представляє цей користувач: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці Chambermusicband.

Таблиця Chamberconcert зберігає дані камерних концертів зі структурою стовпців:

- id – ідентифікатор камерного концерту;
- chambermusicbandid – музичний колектив, який виконує цей камерний концерт: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці Chambermusicband;

- chamberconcertstartdt – дата і час початку камерного концерту;
- chamberconcertenddt – дата і час закінчення камерного концерту;
- chamberconcertreviewmark – усереднена оцінка цього камерного концерту від відвідувачів;
- chamberconcertleftplaces – кількість місць, яка залишилась на цьому камерному концерті для бронювання;

– `chambermusicstyle` – основний музичний стиль, якому відповідає цей камерний концерт;

– `chamberconcerturl` – вебадреса сайту, де представлено камерний концерт;

– `chamberconcertdescription` – опис текстом камерного концерту.

Таблиця `Chamberconcertplace` зберігає дані бронювань місць на камерному концерті зі структурою стовпців:

– `id` – ідентифікатор бронювання місця на камерному концерті;

– `chamberconcertid` – камерний концерт, бронювання місця на якому відбулось: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberconcert`;

– `chambervisitorid` – відвідувач, який забронював це місце на цьому камерному концерті: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `User`;

– `chamberconcertplacestatus` – стан бронювання місця на камерному концерті.

Таблиця `Chamberconcertreview` зберігає дані відгуків про камерні концерти зі структурою стовпців:

– `id` – ідентифікатор відгуку про камерний концерт;

– `chamberconcertid` – камерний концерт, відгук про який залишив відвідувач: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberconcert`;

– `chambervisitorid` – відвідувач, який залишив відгук про камерний концерт: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `User`;

– `chamberconcertreviewtexted` – текст відгуку про камерний концерт;

– `chamberconcertreviewdt` – дата і час створення відгуку про камерний концерт;

– `chamberconcertreviewmark` – оцінка відгуку про камерний концерт

(рис. 2.5).



Рисунок 2.5 – Схема БД ПС для організації камерних концертів

Таблиця БД Chamberconcertreviewphoto зберігає дані фотографій, прикріплених до відгуку про камерні концерти, зі структурою стовпців:

- id – ідентифікатор фотографії, прикріпленої до відгуку про камерний концерт;
- chamberconcertreviewid – відгук про камерний концерт, до якого прикріплено цю фотографію: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці Chamberconcertreview;

– chamberconcertreviewphotodata – сама фотографія, прикріплена до відгуку про камерний концерт.

Таблиця Chamberconcertreviewvideo зберігає дані відео, прикріплених до відгуку про камерні концерти, зі структурою стовпців:

– id – ідентифікатор відео, прикріпленого до відгуку про камерний концерт;

– chamberconcertreviewid – відгук про камерний концерт, до якого прикріплено це відео: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці Chamberconcertreview;

– chamberconcertreviewvideodata – саме відео, прикріплене до відгуку про камерний концерт.

Таблиця БД Chamberconcertvisitorsave зберігає дані збережень камерних концертів відвідувачами зі структурою стовпців:

– id – ідентифікатор збереження камерного концерту відвідувачем;

– chamberconcertid – камерний концерт, який збережено відвідувачем: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці Chamberconcert;

– chambervisitorid – відвідувач, який зберіг цей камерний концерт: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці User.

## **2.4 Висновки за розділом 2**

Другий розділ роботи представляє основні результати, отримані під час проєктування ПС для організації камерних концертів, включаючи виконане моделювання сценаріїв роботи користувача до авторизації, відвідувача камерних концертів, представника музичного колективу, власника приміщень для проведення камерних концертів, визначення технологій розробки зі стеком, що включає мову Python, систему керування БД MySQL, фреймворк Django, виконане проєктування БД.

## 3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

### 3.1 Порядок організації роботи з користувачем в програмній системі для організації камерних концертів

Порядок організації роботи з користувачем в ПС для організації камерних концертів влаштовано за допомогою розподілу користувачів на групи з ідентифікацією їх в одній з цих груп. Базовою групою є користувач до авторизації, який отримує доступ до роботи з засобами пошуку, а далі може обрати або завершити роботу, або авторизуватись. Якщо користувач авторизувався, то подальша логіка та послідовність визначаються тим, в якій саме ролі перебуває користувач (рис. 3.1).

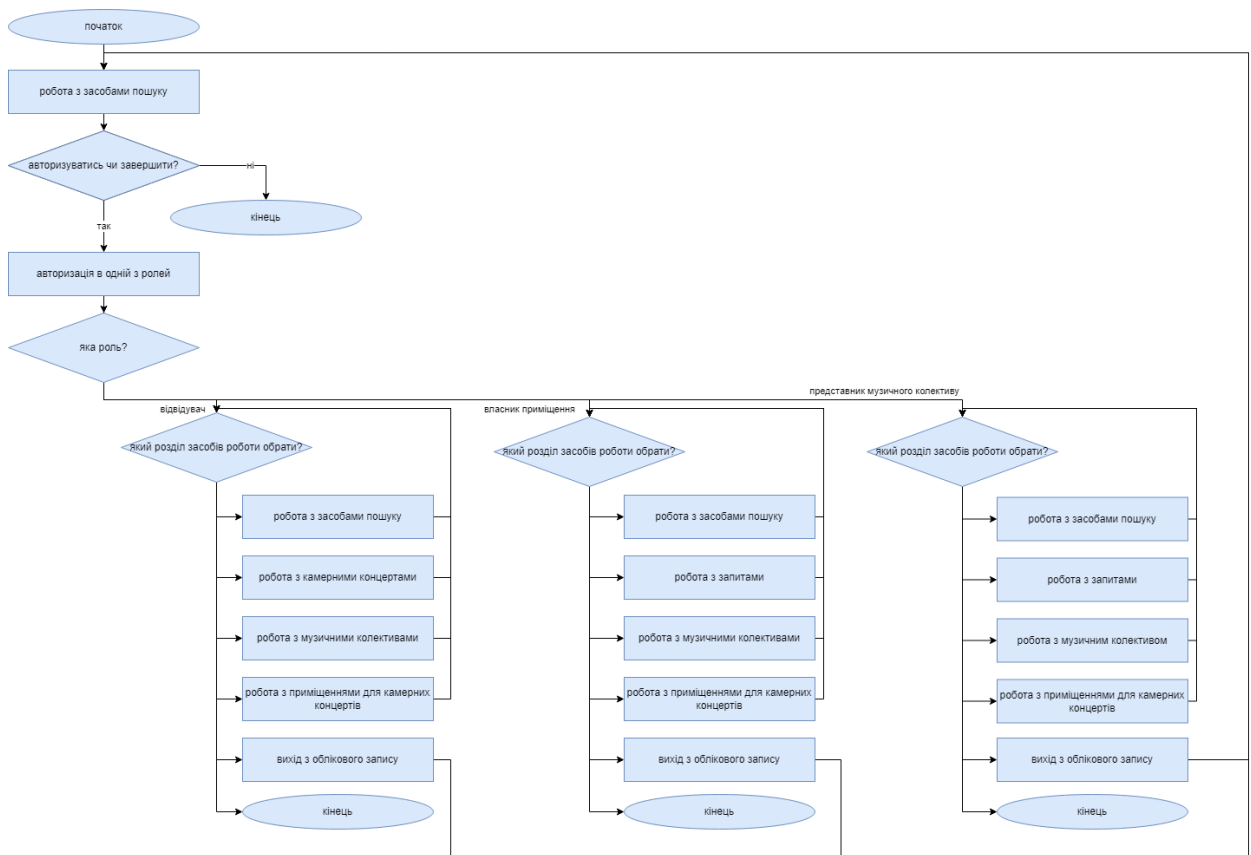


Рисунок 3.1 – Функціональна схема ПС для організації камерних концертів

### 3.2 Рішення з реалізації програмної системи для організації камерних концертів

ПС для організації камерних концертів реалізована у всіх своїх складових, з яких основними є моделі, представлення та шаблони вебсторінок, що відповідає структурі Django-застосунку.

Клас `Chamberspacecity` реалізує модель міст, в яких можуть проводитись камерні концерти. Поле `chamberspacecityname` у цьому класі реалізує назву міста, в якому можуть проводитись камерні концерти, як об'єкт класу `CharField`.

Клас `Chamberspacetype` реалізує модель типів приміщень для проведення камерного концерту. Поле `chamberspacetypename` у цьому класі реалізує назву типу приміщення для проведення камерного концерту як об'єкт класу `CharField`.

Клас `Chamberspace` реалізує модель приміщень для проведення камерних концертів.

Поле `chamberspaceowner` у цьому класі реалізує власника, який у програмі представляє це приміщення для проведення камерного концерту: встановлено зовнішнім ключем зі зв'язком один-до-багатьох до таблиці `Chamberspaceowner`.

Поле `chamberspacetitle` у цьому класі реалізує назву приміщення для проведення камерного концерту як об'єкт класу `CharField`.

Поле `chamberspacedescription` у цьому класі реалізує опис приміщення для проведення камерного концерту як об'єкт класу `TextField`.

Поле `chamberspacetype` у цьому класі реалізує тип приміщення для проведення камерного концерту, до якого належить це приміщення, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberspacetype`.

Поле `chamberspacecity` у цьому класі реалізує місто, в якому знаходиться це приміщення для проведення камерного концерту, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberspacecity`.

Поле `chamberspaceaddress` у цьому класі реалізує адресу, за якою знаходиться це приміщення для проведення камерного концерту, як об'єкт класу `CharField`.

Поле `chamberspacereviewmark` у цьому класі реалізує усереднену оцінку, виставлену у відгуках цьому приміщенню для проведення камерного концерту, як об'єкт класу `DecimalField`.

Поле `chambersspaceplacenumber` у цьому класі реалізує кількість місць, доступну у цьому приміщенні для проведення камерного концерту, як об'єкт класу `PositiveIntegerField`.

Поле `chambersspacephone` у цьому класі реалізує номер телефону для з'єднання з приміщенням для проведення камерного концерту як об'єкт класу `CharField`.

Поле `musiciansaves` у цьому класі реалізує збереження представниками музичних колективів цього приміщення для проведення камерних концертів як об'єкт класу `ManyToManyField` зі зв'язком з моделлю `Chambermusician`.

Поле `visitorsaves` у цьому класі реалізує збереження відвідувачами цього приміщення для проведення камерних концертів як об'єкт класу `ManyToManyField` зі зв'язком з моделлю `User`.

Клас `Chambersspacephoto` реалізує модель фотографій приміщення для проведення камерного концерту.

Поле `chambersspace` у цьому класі реалізує приміщення для проведення камерного концерту, показане на фотографії, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberspace`.

Поле `chambersspacephotodata` у цьому класі реалізує саму фотографію приміщення для проведення камерного концерту як об'єкт класу `ImageField`.

Поле `chambersspacephotodescription` у цьому класі реалізує опис фотографії приміщення для проведення камерного концерту як об'єкт класу `TextField`.

Клас `Chamberspacereview` реалізує модель відгуків за приміщенням для проведення камерного концерту.

Поле `chamberspace` у цьому класі реалізує приміщення для проведення камерного концерту, за яким створено відгук, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberspace`.

Поле `chambervisitor` у цьому класі реалізує відвідувача, який створив цей відгук за приміщенням для проведення камерного концерту, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `User`.

Поле `chamberspacereviewtexted` у цьому класі реалізує текст відгуку за приміщенням для проведення камерного концерту як об'єкт класу `CharField`.

Поле `chamberspacereviewdt` у цьому класі реалізує дату і час створення відгуку за приміщенням для проведення камерного концерту як об'єкт класу `DateTimeField`.

Поле `chamberspacereviewmark` у цьому класі реалізує оцінку відгуку за приміщенням для проведення камерного концерту як об'єкт класу `PositiveSmallIntegerField`.

Клас `Chamberspacereviewphoto` реалізує модель фотографії, доданої до відгуку за приміщенням для проведення камерного концерту.

Поле `chamberspacereview` у цьому класі реалізує відгук за приміщенням для проведення камерного концерту, до якого додано цю фотографію, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberspacereview`.

Поле `chamberspacereviewphotodata` у цьому класі реалізує дані фотографії, доданої до відгуку за приміщенням для проведення камерного концерту, як об'єкт класу `ImageField`.

Клас `Chamberspaceconcertinquiry` реалізує модель запитів на проведення камерних концертів в приміщеннях.

Поле `chamberspace` у цьому класі реалізує приміщення для проведення камерного концерту, проведення в якому запитано музичним колективом або власник якого запитів у музичного колективу можливість проведення концерту в нього, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberspace`.

Поле `chamberconcert` у цьому класі реалізує камерний концерт, проведення якого запитано в цьому приміщенні для проведення камерних концертів, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberconcert`.

Поле `chamberinquirystate` у цьому класі реалізує стан запиту на проведення камерного концерту в приміщенні як об'єкт класу `CharField`.

Поле `chamberinquirydt` у цьому класі реалізує дату і час створення запиту на проведення камерного концерту в приміщенні як об'єкт класу `DateTimeField`.

Поле `chamberinquiryspacetexted` у цьому класі реалізує текст примітки до запиту на проведення камерного концерту в приміщенні від власника приміщення для проведення камерних концертів як об'єкт класу `CharField`.

Поле `chamberinquirybandtexted` у цьому класі реалізує текст примітки до запиту на проведення камерного концерту в приміщенні від представника музичного колективу, як об'єкт класу `CharField`.

Поле `chamberinquiryinitializer` у цьому класі реалізує сторону, яка ініціювала запит на проведення камерного концерту в приміщенні (власник приміщення або представник музичного колективу) як об'єкт класу `BooleanField`.

Клас `Chamberspaceowner` реалізує модель користувачів, які є власниками приміщень для проведення камерних концертів.

Поле `chamberuseraccount` у цьому класі реалізує акаунт у програмі, який має цей власник приміщення для проведення камерних концертів, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `User`.

Поле `chamberspaceownerphone` у цьому класі реалізує номер телефону користувача, який є власником приміщення для проведення камерного концерту, як об'єкт класу `CharField`.

Клас `Chambermusicband` реалізує модель музичних колективів, які проводять камерні концерти.

Поле `chambermusicbandtitle` у цьому класі реалізує назву музичного колективу, який проводить камерні концерти, як об'єкт класу `CharField`.

Поле `chambermusicbanddescription` у цьому класі реалізує опис музичного колективу, який проводить камерні концерти, як об'єкт класу `TextField`.

Поле `chambermusicstyle` у цьому класі реалізує основний стиль, в якому виконує музику цей музичний колектив, який проводить камерні концерти, як об'єкт класу `CharField`.

Поле `chambermusicbandurl` у цьому класі реалізує вебадресу сайту музичного колективу, який проводить камерні концерти, як об'єкт класу `URLField`.

Поле `chambermusicbandurladditional` у цьому класі реалізує додаткову вебадресу сайту музичного колективу, який проводить камерні концерти, як об'єкт класу `URLField`.

Поле `chambermusicbandreviewmark` у цьому класі реалізує усереднену оцінку, виставлену у відгуках музичному колективу, як об'єкт класу `DecimalField`.

Поле `chamberspaceownersaves` у цьому класі реалізує збереження власниками приміщень для проведення камерних концертів цього музичного колективу як об'єкт класу `ManyToManyField` зі зв'язком з моделлю `Chambermusician`.

Поле `visitorsaves` у цьому класі реалізує збереження відвідувачами цього музичного колективу як об'єкт класу `ManyToManyField` зі зв'язком з моделлю `User`.

Клас `Chambermusicbandphoto` реалізує модель фотографій музичних колективів.

Поле `chambermusicband` у цьому класі реалізує музичний колектив, який проводить камерні концерти, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chambermusicband`.

Поле `chambermusicbandphotodata` у цьому класі реалізує саму фотографію музичного колективу як об'єкт класу `ImageField`.

Клас `Chambermusicbandvideo` реалізує модель відео виступів музичних колективів, які проводять камерні концерти.

Поле `chambermusicband` у цьому класі реалізує музичний колектив, який проводить камерні концерти та виступ якого показаний на відео, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chambermusicband`.

Поле `chambermusicbandvideodata` у цьому класі реалізує саме посилання на відео виступу музичного колективу, який проводить камерні концерти, як об'єкт класу `URLField`.

Клас `Chambermusicbandreview` реалізує модель відгуків про музичні колективи, які проводять камерні концерти.

Поле `chambermusicband` у цьому класі реалізує музичний колектив, який проводить камерні концерти та відгук на який створено, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chambermusicband`.

Поле `chambervisitor` у цьому класі реалізує відвідувача, який створив цей відгук про музичний колектив, який проводить камерні концерти, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `User`.

Поле `chambermusicbandreviewtexted` у цьому класі реалізує текст відгуку про музичний колектив, який проводить камерні концерти як об'єкт класу `TextField`.

Поле `chambermusicbandreviewdt` у цьому класі реалізує дату і час створення відгуку про музичний колектив, який проводить камерні концерти, як об'єкт класу `DateTimeField`.

Поле `chambermusicbandreviewmark` у цьому класі реалізує оцінку відгуку про музичний колектив, який проводить камерні концерти, як об'єкт класу `PositiveSmallIntegerField`.

Клас `Chambermusicbandreviewphoto` реалізує модель фотографій, прикріплених до відгуків про музичні колективи, які проводять камерні концерти.

Поле `chambermusicbandreview` у цьому класі реалізує відгук про музичний колектив, який проводить камерні концерти, до якого прикріплено

фотографію, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chambermusicband`.

Поле `chambermusicbandreviewphotodata` у цьому класі реалізує саму фотографію, прикріплену до відгуку про музичний колектив, який проводить камерні концерти, як об'єкт класу `ImageField`.

Клас `Chambermusician` реалізує модель представників музичних колективів.

Поле `chamberuseraccount` у цьому класі реалізує акаунт у програмі, який має цей представник музичного колективу, як об'єкт класу `OneToOneField` зі зв'язком з моделлю `User`.

Поле `chambermusicianphone` у цьому класі реалізує номер телефону представника музичного колективу як об'єкт класу `CharField`.

Поле `chambermusicband` у цьому класі реалізує музичний колектив, який представляє цей користувач, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chambermusicband`.

Клас `Chamberconcert` реалізує модель камерних концертів.

Поле `chambermusicband` у цьому класі реалізує музичний колектив, який виконує цей камерний концерт, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chambermusicband`.

Поле `chamberconcertstartdt` у цьому класі реалізує дату і час початку камерного концерту як об'єкт класу `DateTimeField`.

Поле `chamberconcertenddt` у цьому класі реалізує дату і час закінчення камерного концерту як об'єкт класу `DateTimeField`.

Поле `chamberconcertreviewmark` у цьому класі реалізує усереднену оцінку цього камерного концерту від відвідувачів як об'єкт класу `DecimalField`.

Поле `chamberconcertleftplaces` у цьому класі реалізує кількість місць, яка залишилась на цьому камерному концерті для бронювання, як об'єкт класу `PositiveSmallIntegerField`.

Поле `chambermusicstyle` у цьому класі реалізує основний музичний стиль, якому відповідає цей камерний концерт, як об'єкт класу `CharField`.

Поле `chamberconcerturl` у цьому класі реалізує вебадресу сайту, де представлено камерний концерт, як об'єкт класу `URLField`.

Поле `chamberconcertdescription` у цьому класі реалізує опис текстом камерного концерту як об'єкт класу `TextField`.

Поле `visitorsaves` у цьому класі реалізує збереження відвідувачами цього камерного концерту як об'єкт класу `ManyToManyField` зі зв'язком з моделлю `User`.

Клас `Chamberconcertplace` реалізує модель бронювань місць на камерному концерті.

Поле `chamberconcert` у цьому класі реалізує камерний концерт, бронювання місця на якому відбулось, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberconcert`.

Поле `chambervisitor` у цьому класі реалізує відвідувача, який забронював це місце на цьому камерному концерті, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `User`.

Поле `chamberconcertplacestatus` у цьому класі реалізує стан бронювання місця на камерному концерті як об'єкт класу `CharField`.

Клас `Chamberconcertreview` реалізує модель відгуків про камерні концерти.

Поле `chamberconcert` у цьому класі реалізує камерний концерт, відгук про який залишив відвідувач, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberconcert`.

Поле `chambervisitor` у цьому класі реалізує відвідувача, який залишив відгук про камерний концерт, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `User`.

Поле `chamberconcertreviewtexted` у цьому класі реалізує текст відгуку про камерний концерт як об'єкт класу `TextField`.

Поле `chamberconcertreviewdt` у цьому класі реалізує дату і час створення відгуку про камерний концерт як об'єкт класу `DateTimeField`.

Поле `chamberconcertreviewmark` у цьому класі реалізує оцінку відгуку про камерний концерт як об'єкт класу `PositiveSmallIntegerField`.

Клас `Chamberconcertreviewphoto` реалізує модель фотографій, прикріплених до відгуку про камерні концерти.

Поле `chamberconcertreview` у цьому класі реалізує відгук про камерний концерт, до якого прикріплено цю фотографію, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberconcertreview`.

Поле `chamberconcertreviewphotodata` у цьому класі реалізує саму фотографію, прикріплену до відгуку про камерний концерт, як об'єкт класу `ImageField`.

Клас `Chamberconcertreviewvideo` реалізує модель відео, прикріплених до відгуку про камерні концерти.

Поле `chamberconcertreview` у цьому класі реалізує відгук про камерний концерт, до якого прикріплено це відео, як об'єкт класу `ForeignKey` зі зв'язком з моделлю `Chamberconcertreview`.

Поле `chamberconcertreviewvideodata` у цьому класі реалізує посилання на відео, прикріплене до відгуку про камерний концерт, як об'єкт класу `URLField`.

Далі описані реалізовані функції представлень [8] у ПС для організації камерних концертів.

Перегляд власних приміщень для проведення камерних концертів реалізовано функцією `own_chamberspaces_view`, а в підсистемі шаблонів вебсторінок [9] було створено шаблон `ownchamberspaces.html`, який організований як і всі наступні шаблони за допомогою шаблону [10].

Перегляд власних запитів на проведення камерного концерту в приміщенні реалізовано функцією `own_chamberspaceconcertinquiries_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `ownchamberinquiries.html`.

Перегляд адресованих запитів на проведення камерних концертів за приміщенням реалізовано функцією

own\_space\_chamberspaceconcertinquiries\_view, а в підсистемі шаблонів вебсторінок було створено шаблон ownspacechamberinquiries.html. Функція має параметр chamberspace\_id, який задає ідентифікатор приміщення для проведення камерних концертів.

Перегляд адресованих запитів на проведення камерних концертів за приміщеннями власника реалізовано функцією own\_spaces\_chamberspaceconcertinquiries\_view, а в підсистемі шаблонів вебсторінок було створено шаблон ownspaceschamberinquiries.html.

Перегляд адресованих запитів на проведення камерних концертів музичному колективу реалізовано функцією own\_band\_chamberspaceconcertinquiries\_view, а в підсистемі шаблонів вебсторінок було створено шаблон ownbandchamberinquiries.html.

Перегляд власних запланованих камерних концертів реалізовано функцією own\_chamberconcerts\_view, а в підсистемі шаблонів вебсторінок було створено шаблон ownchamberconcerts.html.

Перегляд власних проведених камерних концертів реалізовано функцією own\_done\_chamberconcerts\_view, а в підсистемі шаблонів вебсторінок було створено шаблон ownspacechamberdoneconcerts.html.

Перегляд власних камерних концертів на розгляді реалізовано функцією own\_inquired\_chamberconcerts\_view, а в підсистемі шаблонів вебсторінок було створено шаблон ownspacechamberinqconcerts.html.

Перегляд даних музичного колективу для камерних концертів реалізовано функцією chamber\_band\_view, а в підсистемі шаблонів вебсторінок було створено шаблон chamberband.html. Функція має параметр chambermusicband\_id, який задає ідентифікатор музичного колективу для камерних концертів.

Перегляд запланованих камерних концертів колективу реалізовано функцією chamber\_band\_concerts\_view, а в підсистемі шаблонів вебсторінок було створено шаблон chamberbandconcerts.html. Функція має параметр

`chambermusicband_id`, який задає ідентифікатор музичного колективу для камерних концертів.

Перегляд проведених камерних концертів колективу реалізовано функцією `chamber_band_archive_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberarchiveconcerts.html`. Функція має параметр `chambermusicband_id`, який задає ідентифікатор музичного колективу для камерних концертів.

Перегляд відгуків за камерним концертом реалізовано функцією `chamberconcert_reviews_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberconcertreviews.html`. Функція має параметр `chamberconcert_id`, який задає ідентифікатор камерного концерту.

Перегляд відгуків за музичним колективом реалізовано функцією `chamberband_reviews_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberbandreviews.html`. Функція має параметр `chamberband_id`, який задає ідентифікатор музичного колективу для камерних концертів.

Перегляд відгуків за приміщенням для проведення камерного концерту реалізовано функцією, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberspacereviews.html`. Функція має параметр `chamberspace_id`, який задає ідентифікатор приміщення для камерних концертів.

Перегляд заброньованих камерних концертів реалізовано функцією `booked_chamberconcerts_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `bookedchamberconcerts.html`.

Перегляд переліку відвідувачів камерного концерту реалізовано функцією `chamberconcert_users_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberconcertusers.html`. Функція має параметр `chamberconcert_id`, який задає ідентифікатор камерного концерту.

Визначення власного приміщення для проведення камерних концертів реалізовано функцією `chamberspace_add_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberspaceadd.html`.

Зняття власного приміщення для проведення камерних концертів з розгляду реалізовано функцією `chamberspace_cancel_view`, а параметр `chamberspace_id` задає ідентифікатор приміщення для камерних концертів.

Редагування даних власного приміщення для проведення камерних концертів реалізовано функцією `chamberspace_edit_view`, а параметр `chamberspace_id` задає ідентифікатор приміщення для камерних концертів.

Запит на проведення камерного концерту в приміщенні реалізовано функцією `chamberspaceconcert_inquiry_add_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberspaceconcertinquiryadd.html`. Параметр `chamberspace_id` задає ідентифікатор приміщення для камерних концертів.

Підтвердження запиту на проведення камерного концерту в приміщенні реалізовано функцією `chamberspaceconcert_inquiry_prove_view`, а параметр `chamberspaceconcertinquiry_id` задає ідентифікатор запиту на проведення камерного концерту в приміщенні.

Скасування запиту на проведення камерного концерту в приміщенні реалізовано функцією `chamberspaceconcert_inquiry_cancel_view`, а параметр `chamberspaceconcertinquiry_id` задає ідентифікатор запиту на проведення камерного концерту в приміщенні.

Пошук приміщень для проведення камерних концертів реалізовано функцією `chamberspace_search_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberspacesearch.html`. Усі параметри передаються через GET-запит.

Визначення власного колективу для камерних концертів реалізовано функцією `chambermusicband_add_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chambermusicbandadd.html`.

Редагування даних власного колективу для камерних концертів реалізовано функцією `chambermusicband_edit_view`, а параметр `chambermusicband_id` задає ідентифікатор музичного колективу.

Збереження приміщення для проведення камерних концертів реалізовано функцією `chamberspace_save_view`, а параметр `chamberspace_id` задає ідентифікатор приміщення для проведення камерних концертів. Скасування збереження приміщення для проведення камерних концертів реалізовано тим же самим представленням, але при перевірці існування такого збереження встановлюється, що якщо воно вже існує, то реалізується скасування.

Перегляд збережених приміщень для проведення камерних концертів у них реалізовано функцією `chamberspaces_saved_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chambersspacesaved.html`.

Перегляд запланованих камерних концертів за збереженими приміщеннями реалізовано функцією `chamberconcerts_saved_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberconcertsaved.html`.

Пропонування власного приміщення для проведення камерного концерту реалізовано функцією `space_chamberspaceconcert_inquiry_add_view`, а параметри `chamberspace_id` задає ідентифікатор приміщення для проведення камерного концерту, `chamberconcert_id` задає ідентифікатор камерного концерту.

Бронювання місця на камерному концерті реалізовано функцією `chamberconcert_book_place_view`, а параметр `chamberconcert_id` задає ідентифікатор камерного концерту.

Скасування бронювання місця на камерному концерті реалізовано функцією `chamberconcert_cancel_book_place_view`, а параметр `chamberconcertplace_id` задає ідентифікатор бронювання.

Збереження камерного концерту реалізовано функцією `chamberconcert_save_view`, а параметр `chamberconcert_id` задає ідентифікатор камерного концерту. Скасування збереження камерного концерту реалізовано тим же самим представленням, але при перевірці існування такого збереження встановлюється, що якщо воно вже існує, то реалізується скасування.

Збереження колективу для камерних концертів реалізовано функцією `chamberband_save_view`, а параметр `chamberband_id` задає ідентифікатор колективу для камерних концертів. Скасування збереження колективу для камерних концертів реалізовано тим же самим представленням, але при перевірці існування такого збереження встановлюється, що якщо воно вже існує, то реалізується скасування.

Перегляд збережених камерних концертів реалізовано функцією `chamberconcerts_saved_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberconcertssaved.html`.

Перегляд запланованих камерних концертів за збереженими колективами реалізовано функцією `chamberconcerts_saved_bands_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberconcertssavedbands.html`.

Пошук колективів, які проводять камерні концерти реалізовано функцією `chamberband_search_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberbandsearch.html`. Усі параметри передаються через GET-запит.

Пошук камерних концертів реалізовано функцією `chamberconcert_search_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberconcertsearch.html`. Усі параметри передаються через GET-запит.

Перегляд збережених колективів для камерних концертів реалізовано функцією `chamberbands_saved_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberbandssaved.html`.

Створення відгуку про проведений камерний концерт реалізовано функцією `chamberconcert_review_add_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberconcertreviewadd.html`. Параметр функції `chamberconcert_id` задає ідентифікатор камерного концерту.

Створення відгуку про музичний колектив за камерним концертом реалізовано функцією `chamberband_review_add_view`, а в підсистемі шаблонів

вебсторінок було створено шаблон `chamberbandreviewadd.html`. Параметр функції `chamberband_id` задає ідентифікатор музичного колективу, який виконує камерні концерти.

Створення відгуку про приміщення для проведення камерного концерту реалізовано функцією `chamberspace_review_add_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberspacereviewadd.html`. Параметр функції `chamberspace_id` задає ідентифікатор приміщення для проведення камерного концерту.

Перегляд даних камерного концерту реалізовано функцією `chamberconcert_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberconcert.html`. Параметр функції `chamberconcert_id` задає ідентифікатор камерного концерту.

Перегляд даних приміщення для проведення камерного концерту реалізовано функцією `chamberspace_view`, а в підсистемі шаблонів вебсторінок було створено шаблон `chamberspace.html`. Параметр функції `chamberspace_id` задає ідентифікатор приміщення для проведення камерного концерту.

### **3.3 Висновки за розділом 3**

Третій розділ роботи представляє основні результати, отримані під час реалізації ПС для організації камерних концертів, включаючи послідовність роботи з програмою та відповідні реалізовані структурні одиниці ПС, включаючи моделі, шаблони вебсторінок, представлення.

## **4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ**

### **4.1 Призначення програмної системи**

ПС призначена для організації камерних концертів шляхом зв'язування між собою людей, які є власниками камерних приміщень та готові проводити в себе такі концерти, музикантів, які готові проводити камерні концерти, та відвідувачів, які готові відвідати такі концерти. Наявність такого майданчика повинна дозволити комунікувати всім сторонам між собою та сприяти за рахунок надання додаткової інформації від кожної сторони зручності роботи кожної зі сторін.

### **4.2 Умови виконання програмної системи**

ПС для організації камерних концертів потребує для використання встановлення серверної частини, яка має бути забезпечена інтерпретатором мови Python, системою MySQL та технічною стороною з процесором мінімально з 2 ядрами, оперативною пам'яттю в 4 Гб, 100 Гб вільного місця на жорсткому диску.

### **4.3 Експлуатація програмної системи для організації камерних концертів**

Робота користувача з програмою організована за логікою функціональної схеми (рис. 3.1). Відповідно користувач має різний варіант доступу в залежності від ролі, в якій він авторизувався.

Якщо користувач авторизувався як відвідувач, то він може працювати над пошуком камерних концертів, колективів або приміщень, в яких можуть проводитись камерні концерти, що доступно і без авторизації. Також відвідувач може з розділу концертів переглянути камерні концерти, які він безпосередньо зберіг або місця на яких забронював. З розділу колективів

відвідувач може переглянути збережені колективи та камерні концерти, які запланували ці збережені колективи. Для перемикання між режимами є відповідні кнопки (рис. 4.1).

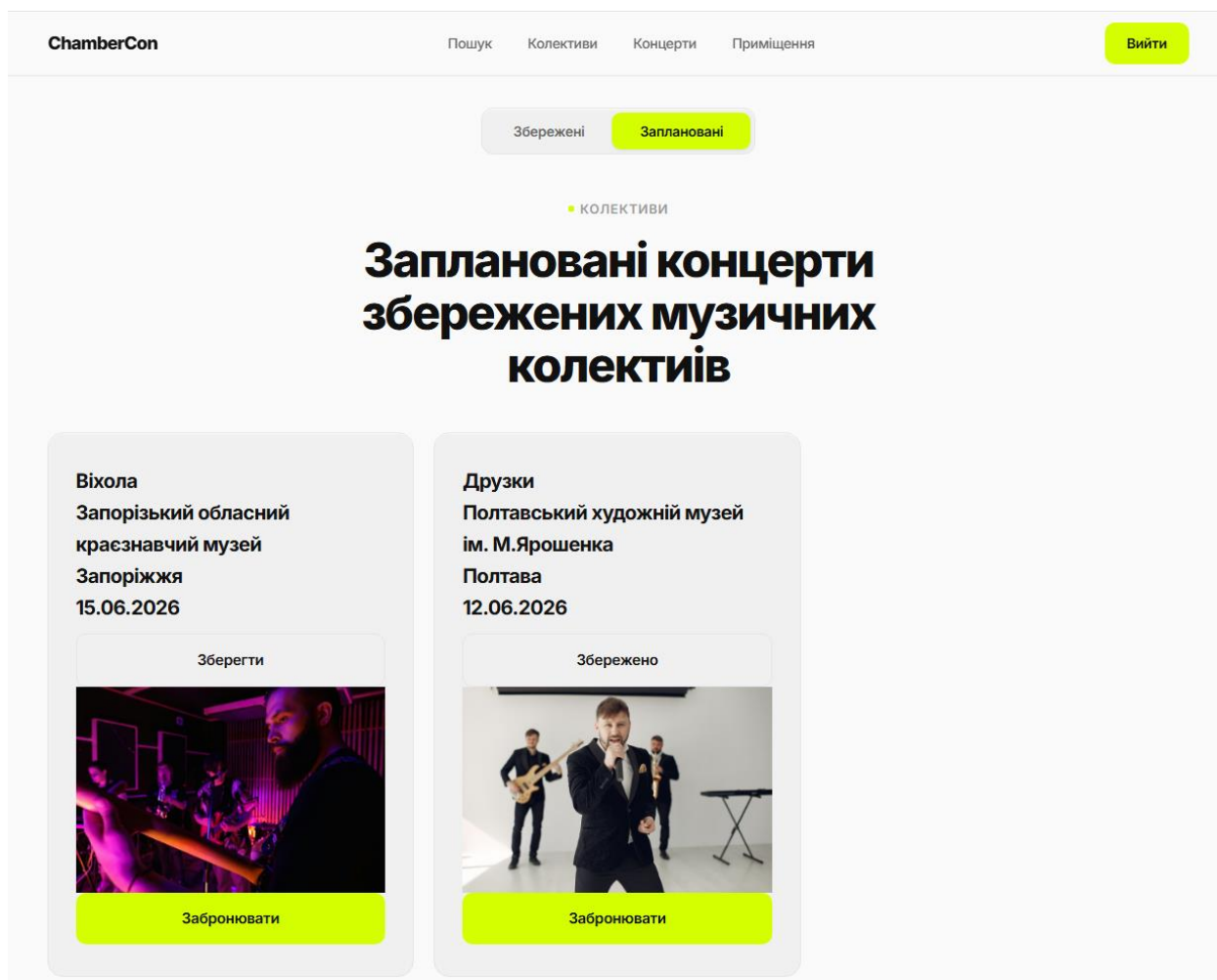


Рисунок 4.1 – Перегляд запланованих камерних концертів за збереженими музичними колективами відвідувачем

З розділу приміщень відвідувач може також вибрати або збережені, або переглянути заплановані концерти за всіма збереженими ним приміщеннями для проведення камерних концертів.

Власники приміщень для проведення камерних концертів можуть використовувати також розділ пошуку, а можуть розділ колективів. З цього розділу він може переглянути збережені колективи та камерні концерти, які запланували ці збережені колективи. З розділу приміщень власник може

додати нове власне приміщення для проведення камерних концертів, може переглянути та керувати власними приміщеннями. З розділу запитів власник приміщення для проведення камерних концертів може обрати або власні запити, або адресовані. З адресованих доступні або всі, або за приміщенням. У останньому випадку тоді далі потрібно обрати конкретне приміщення, а за ним відобразиться перелік запитів (рис. 4.2).

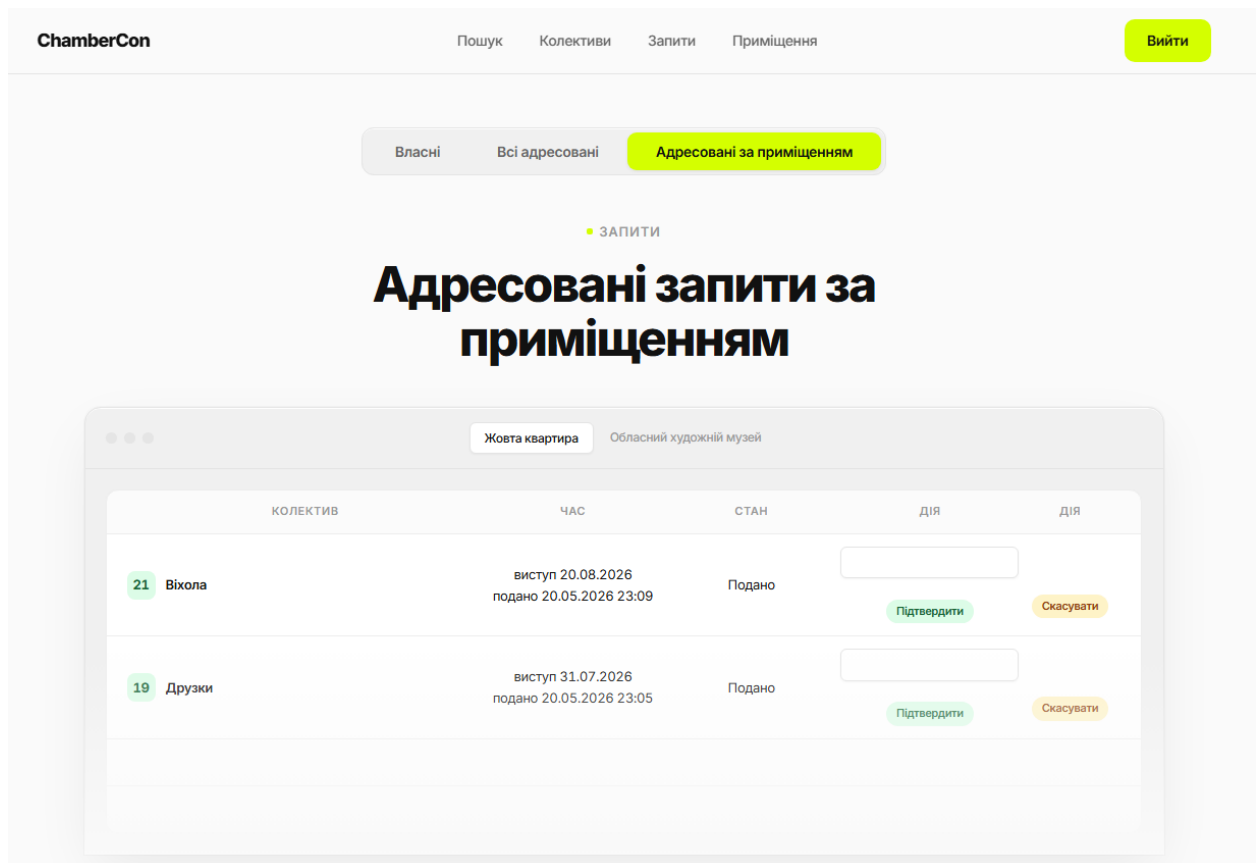


Рисунок 4.2 – Керування запитом на проведення камерних концертів, адресованими за приміщенням власника

Представник музичного колективу може використовувати також розділ пошуку, а може розділ колективу, звідки він може вносити дані за власним колективом. З розділу концертів він може переглянути камерні концерти, які заплановані за його колективом, які вже відбулись або які знаходяться на розгляді за сформованими запитом до власників приміщень для проведення камерних концертів (рис. 4.3).

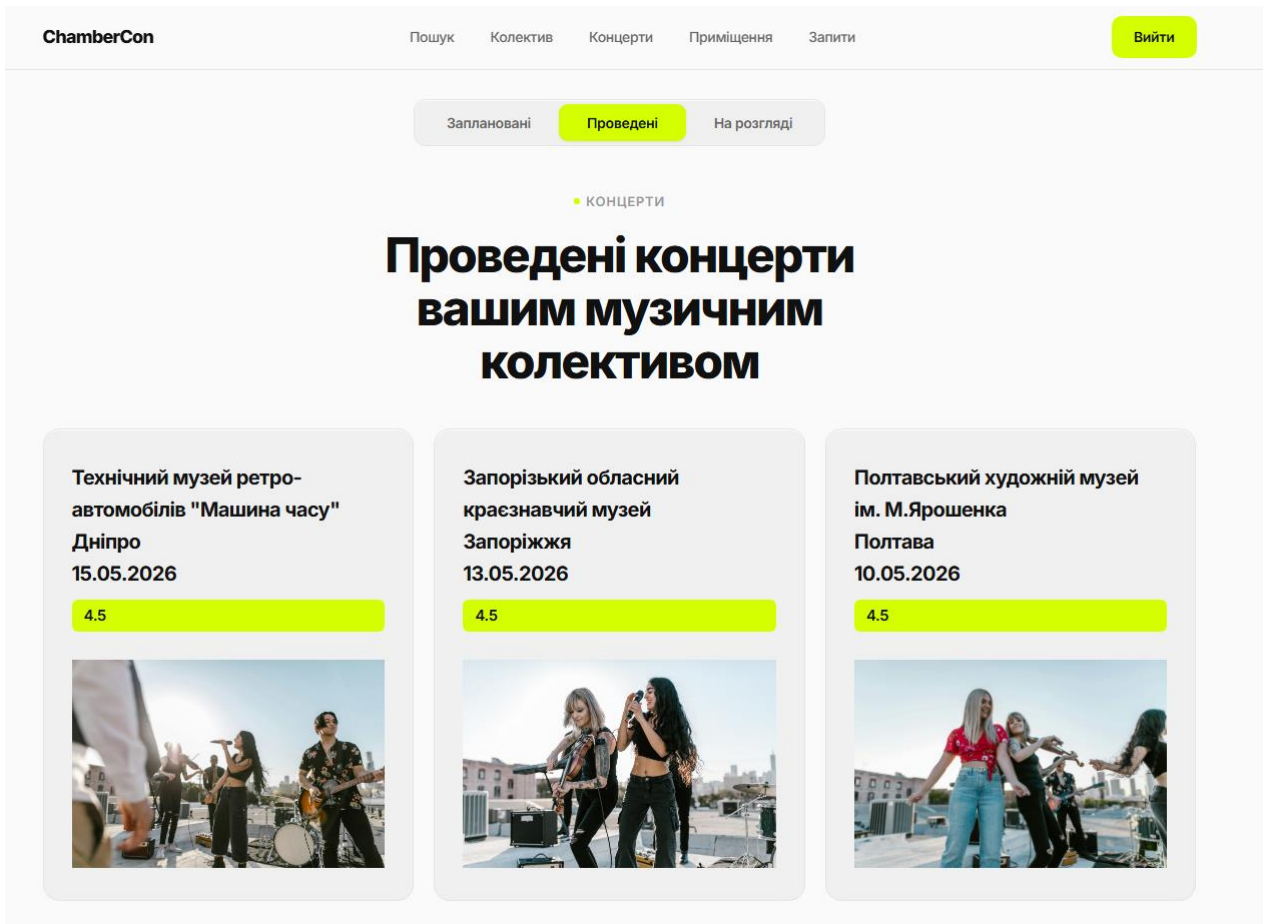


Рисунок 4.3 – Перегляд проведених камерних концертів музичним колективом авторизованого представника

З розділу приміщень представник музичного колективу може або переглянути збережені приміщення, або переглянути заплановані концерти за збереженими приміщеннями. З розділу запитів представник музичного колективу може мати доступ або до власних запитів, або до адресованих його колективу запитів власників приміщень для проведення камерних концертів, в яких вони пропонують провести концерт у них.

#### 4.4 Тестування програмної системи організації камерних концертів

Результати проведеного тестування ПС організації камерних концертів представлені в цьому підрозділі. У табл. 4.1 подано функціональні можливості

за сценаріями з розділу 2 та за кожним сценарієм наведено результат, який позначає успіх у випадку позначки «+» та невдачу у випадку позначки «-».

Таблиця 4.1 – Результати тестування ПС організації камерних концертів

| Функціональна можливість                                                          | Результат |
|-----------------------------------------------------------------------------------|-----------|
| 1                                                                                 | 2         |
| Визначення власного приміщення для проведення камерних концертів                  | +         |
| Зняття власного приміщення для проведення камерних концертів з розгляду           | +         |
| Редагування даних власного приміщення для проведення камерних концертів           | +         |
| Перегляд власних приміщень для проведення камерних концертів                      | +         |
| Запит на проведення камерного концерту в приміщенні                               | +         |
| Підтвердження запиту на проведення камерного концерту в приміщенні                | +         |
| Скасування запиту на проведення камерного концерту в приміщенні                   | +         |
| Перегляд власних запитів на проведення камерного концерту в приміщенні            | +         |
| Перегляд адресованих запитів на проведення камерних концертів за приміщенням      | +         |
| Перегляд переліку відвідувачів камерного концерту                                 | +         |
| Перегляд адресованих запитів на проведення камерних концертів музичному колективу | +         |
| Перегляд запланованих камерних концертів за збереженими колективами               | +         |

Продовження таблиці 4.1

| 1                                                                                      | 2 |
|----------------------------------------------------------------------------------------|---|
| Авторизація                                                                            | + |
| Реєстрація                                                                             | + |
| Пошук приміщень для проведення камерних концертів                                      | + |
| Визначення власного колективу для камерних концертів                                   | + |
| Редагування даних власного колективу для камерних концертів                            | + |
| Перегляд власних запланованих камерних концертів                                       | + |
| Перегляд власних проведених камерних концертів                                         | + |
| Перегляд власних камерних концертів на розгляді                                        | + |
| Збереження приміщення для проведення камерних концертів                                | + |
| Скасування збереження приміщення для проведення камерних концертів                     | + |
| Перегляд збережених приміщень для проведення камерних концертів                        | + |
| Пропонування власного приміщення для проведення камерного концерту                     | + |
| Пошук колективів, які проводять камерні концерти                                       | + |
| Пошук камерних концертів                                                               | + |
| Бронювання місця на камерному концерті                                                 | + |
| Скасування бронювання місця на камерному концерті                                      | + |
| Збереження камерного концерту                                                          | + |
| Перегляд адресованих запитів на проведення камерних концертів за приміщеннями власника | + |
| Перегляд відгуків за приміщенням для проведення камерного концерту                     | + |

Кінець таблиці 4.1

| 1                                                                    | 2 |
|----------------------------------------------------------------------|---|
| Скасування збереження камерного концерту                             | + |
| Перегляд збережених камерних концертів                               | + |
| Збереження колективу для камерних концертів                          | + |
| Скасування збереження колективу для камерних концертів               | + |
| Перегляд збережених колективів для камерних концертів                | + |
| Створення відгуку про проведений камерний концерт                    | + |
| Створення відгуку про музичний колектив за камерним концертом        | + |
| Створення відгуку про приміщення для проведення камерного концерту   | + |
| Перегляд даних камерного концерту                                    | + |
| Перегляд даних приміщення для проведення камерного концерту          | + |
| Перегляд даних музичного колективу для камерних концертів            | + |
| Перегляд відгуків за камерним концертом                              | + |
| Перегляд відгуків за музичним колективом                             | + |
| Перегляд запланованих камерних концертів колективу                   | + |
| Перегляд проведених камерних концертів колективу                     | + |
| Перегляд запланованих камерних концертів за збереженими приміщеннями | + |
| Перегляд заброньованих камерних концертів                            | + |

#### 4.5 Висновки за розділом 4

ПС організації камерних концертів описана за роботою з основними сценаріями, представлено результати тестування ПС. Тестування виявилось успішним, вказуючи на те, що всі функції були реалізовані в повній мірі.

## ВИСНОВКИ

Розробка ПС для організації камерних концертів виконувалась для збільшення кількості варіантів організації камерних концертів за рахунок надання доступу власниками до невеликих приміщень. Мета роботи в підсумку була досягнута, оскільки розроблена програмна система дозволяє власникам невеликих приміщень виставляти їх як варіанти для проведення камерних концертів, знаходити музичні колективи для цих концертів, а представникам музичних колективів відповідно знаходити майданчики для проведення камерних концертів, звертатись до власників цих приміщень. Таким чином кількість варіантів збільшується за рахунок такої можливої комунікації.

Фактично приміщення, в якому може проводитись камерний концерт, може бути не направленим на проведення концертів. Тобто це може бути музей, мистецький простір, торговий майданчик. Відповідно такий майданчик не розглядається як варіант для проведення концерту базово, а за наявності розробленої програмної системи виникає можливість власнику повідомити, що він готовий до такої ініціативи, а представникам музичних колективів шукати серед таких потенційних варіантів і звертатись з пропозиціями. Це і мається на увазі під збільшенням кількості варіантів організації камерних концертів.

ПС для організації камерних концертів розроблена мовою Python з фреймворком та системою MySQL. У результаті роботи створена програмна система, яка розроблена за результатами аналізу предметної області, виділення основних можливостей, визначених відмінностей з формулюванням концепції розробки, реалізована через розробку підсистем у вигляді моделей, представлень, шаблонів сторінок.

ПС призначена для організації камерних концертів шляхом зв'язування між собою людей, які є власниками камерних приміщень та готові проводити в себе такі концерти, музикантів, які готові проводити камерні концерти, та відвідувачів, які готові відвідати такі концерти. Наявність такого майданчика

повинна дозволити комунікувати всім сторонам між собою та сприяти за рахунок надання додаткової інформації від кожної сторони зручності роботи кожної зі сторін.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. DICE. Tickets for your kind of shows. URL: <https://dice.fm/> (date of access: 20.05.2026).
2. Events in London & Things to do. Fever. URL: [https://feverup.com/en/london?\\_gl=1\\*exci1h\\*\\_up\\*MQ..\\*\\_ga\\*ODgwNjA2OTU1LjE3ODA3NTIxNTc.\\*\\_ga\\_D4T4V3RS3\\*czE3ODA3NTIxNTYkbzEkZzAkdDE3ODA3NTIxNTYkajYwJGwwJGgw](https://feverup.com/en/london?_gl=1*exci1h*_up*MQ..*_ga*ODgwNjA2OTU1LjE3ODA3NTIxNTc.*_ga_D4T4V3RS3*czE3ODA3NTIxNTYkbzEkZzAkdDE3ODA3NTIxNTYkajYwJGwwJGgw) (date of access: 20.05.2026).
3. Find the Best Concerts Near You, Tour Dates & Tickets Worldwide. Songkick. URL: <https://www.songkick.com/> (date of access: 20.05.2026).
4. Django at a glance. Django documentation. Django. URL: <https://docs.djangoproject.com/en/6.0/intro/overview/> (date of access: 22.05.2026).
5. Models and databases. Django documentation. Django. URL: <https://docs.djangoproject.com/en/6.0/topics/db/> (date of access: 22.05.2026).
6. Requirements – The Pyramid Web Framework v2.1. URL: [https://docs.pylonsproject.org/projects/pyramid/en/latest/quick\\_tutorial/requirements.html](https://docs.pylonsproject.org/projects/pyramid/en/latest/quick_tutorial/requirements.html) (date of access: 22.05.2026).
7. Templates – The Pyramid Community Cookbook v0.2. URL: [https://docs.pylonsproject.org/projects/pyramid\\_cookbook/en/latest/pylons/templates.html](https://docs.pylonsproject.org/projects/pyramid_cookbook/en/latest/pylons/templates.html) (date of access: 22.05.2026).
8. Writing views. Django documentation. Django. URL: <https://docs.djangoproject.com/en/6.0/topics/http/views/> (date of access: 23.05.2026).
9. Templates. Django documentation. Django. URL: <https://docs.djangoproject.com/en/6.0/topics/templates/> (date of access: 23.05.2026).
10. Free Template 618 The Catalyst. URL: <https://templatemo.com/tm-618-the-catalyst> (date of access: 25.05.2026).

**ДОДАТОК А**  
**Технічне завдання**

## **Вступ**

Камерні концерти розраховані на невелику аудиторію, а тому проводитись можуть в різних невеликих приміщеннях. Це може бути і невеликий концерт на вулиці. Відповідно варіантів приміщень, в яких можуть проводитись такі концерти, є доволі багато. Якщо масштабний концерт дуже обмежений в потенційних майданчиках, то камерний концерт може проводитись і в місцях, які зазвичай не використовуються для проведення концертів. Наприклад, це може бути певний музей або подібний майданчик, який готовий прийняти в себе концерт. Музиканти, які проводять такий концерт, можуть мати різний рівень, тож загалом варіантів для організації концертів може бути багато. Окрім того є і відвідувачі, які зацікавлені саме в таких камерних концертах та готові відвідувати їх за участі музикантів різного рівня. Відповідно необхідна програмна система, яка б зв'язала всіх цих учасників між собою.

### **A.1 Підстави для розробки**

Робота виконана за темою «Створення програмної системи для організації камерних концертів» у відповідності з даними Наказу № 139 від 7.04.2026 року в Національному університеті «Запорізька політехніка».

### **A.2 Призначення розробки**

Програмна система, яка розробляється, призначена для організації камерних концертів шляхом зв'язування між собою людей, які є власниками камерних приміщень та готові проводити в себе такі концерти, музикантів, які готові проводити камерні концерти, та відвідувачів, які готові відвідати такі концерти. Наявність такого майданчика повинна дозволити комунікувати всім

сторонам між собою та сприяти за рахунок надання додаткової інформації від кожної сторони зручності роботи кожної зі сторін.

### **А.3 Основні вимоги до програми**

#### **А.3.1 Вимоги до функціональних характеристик**

Програмна система організації камерних концертів має відповідати функціональним характеристикам, серед яких:

- визначення власного приміщення для проведення камерних концертів;
- зняття власного приміщення для проведення камерних концертів з розгляду;
- редагування даних власного приміщення для проведення камерних концертів;
- перегляд власних приміщень для проведення камерних концертів;
- запит на проведення камерного концерту в приміщенні;
- підтвердження запиту на проведення камерного концерту в приміщенні;
- скасування запиту на проведення камерного концерту в приміщенні;
- перегляд власних запитів на проведення камерного концерту в приміщенні;
- перегляд адресованих запитів на проведення камерних концертів за приміщенням;
- перегляд адресованих запитів на проведення камерних концертів за приміщеннями власника;
- авторизація;
- реєстрація;
- пошук приміщень для проведення камерних концертів;
- визначення власного колективу для камерних концертів;
- редагування даних власного колективу для камерних концертів;

- перегляд власних запланованих камерних концертів;
- перегляд власних проведених камерних концертів;
- перегляд власних камерних концертів на розгляді;
- збереження приміщення для проведення камерних концертів;
- скасування збереження приміщення для проведення камерних концертів;
- перегляд збережених приміщень для проведення камерних концертів у них;
- перегляд запланованих камерних концертів за збереженими приміщеннями;
- пропонування власного приміщення для проведення камерного концерту;
- пошук колективів, які проводять камерні концерти;
- пошук камерних концертів;
- бронювання місця на камерному концерті;
- скасування бронювання місця на камерному концерті;
- збереження камерного концерту;
- скасування збереження камерного концерту;
- перегляд збережених камерних концертів;
- збереження колективу для камерних концертів;
- перегляд запланованих камерних концертів за збереженими колективами;
- скасування збереження колективу для камерних концертів;
- перегляд збережених колективів для камерних концертів;
- створення відгуку про проведений камерний концерт;
- створення відгуку про музичний колектив за камерним концертом;
- створення відгуку про приміщення для проведення камерного концерту;
- перегляд даних камерного концерту;
- перегляд даних приміщення для проведення камерного концерту;

- перегляд даних музичного колективу для камерних концертів;
- перегляд відгуків за камерним концертом;
- перегляд відгуків за музичним колективом;
- перегляд відгуків за приміщенням для проведення камерного концерту;
- перегляд запланованих камерних концертів колективу;
- перегляд проведених камерних концертів колективу;
- перегляд заброньованих камерних концертів;
- перегляд переліку відвідувачів камерного концерту;
- перегляд адресованих запитів на проведення камерних концертів музичному колективу.

### **А.3.2 Умови експлуатації**

Експлуатаційні вимоги до програмної системи організації камерних концертів включають виділення серверу для роботи з програмою шляхом використання існуючого сервера або замовлення хостингу.

У обох випадках сервер повинен забезпечувати роботу з системою керування базами даних, мовою програмування, що застосовуватимуться для роботи з програмою після прийняття цих рішень в даній роботі.

### **А.3.3 Вимоги до складу та параметрів технічних засобів**

Для забезпечення хостингу шляхом оренди сервера або використання власного серверу необхідно враховувати, що мінімальними вимогами, які на сервері виділені виключно під роботу з програмною системою організації камерних концертів передбачають наявність процесора мінімально з 2 ядрами, 4 Гб оперативної пам'яті, 100 Гб сховища на жорсткому диску.

#### **А.4 Порядок контролю та приймання**

Робота над дипломною кваліфікаційною роботою має виконуватися протягом 8-тижневого циклу, під час якого має бути виконано саму роботу, проведено перевірку роботи на нормоконтроль, проведено процедури рецензування та захисту роботи.

**ДОДАТОК Б**  
**Текст програми**

```

from django.shortcuts import render
from .models import DecorationplantType,
    Chamberspacecity,
    Chamberspacetype,
    Chamberspace,
    Chamberspacephoto,
    Chamberspacereview,
    Chamberspacereviewphoto,
    Chamberspaceconcertinquiry,
    Chamberspaceowner,
    Chambermusicband,
    Chambermusicbandphoto,
    Chambermusicbandvideo,
    Chambermusicbandreview,
    Chambermusicbandreviewphoto,
    Chambermusician,
    Chamberconcert,
    Chamberconcertplace,
    Chamberconcertreview,
    Chamberconcertreviewphoto,
    Chamberconcertreviewvideo

def own_chamberspaces_view(request):
    chamberspace_owner_object =
Chamberspaceowner.objects.select_related("chamberuseraccount").filter(chamberu
seraccount = request.user)
    if not chamberspace_owner_object:
        return render(
            request,

```

```

        "chambercon/ownchamberspaces.html",
        {
            "chamberspaces": [],
            "owner": None
        }
    )
else:
    chamberspace_owner_object = chamberspace_owner_object[0]
    chamber_spaces =
Chamberspace.objects.select_related("chamberspaceowner").select_related("cham
berspacetype").select_related("chamberspacecity").filter(chamberspaceowner =
chamberspace_owner_object)
    if chamber_spaces:
        for chamber_space in chamber_spaces:
            chamber_space.photo =
Chamberspacephoto.objects.filter(chamberspace = chamber_space)[0]
    return render(
        request,
        "chambercon/ownchamberspaces.html",
        {
            "chamberspaces": chamber_spaces,
            "owner": chamberspace_owner_object
        }
    )

def own_chamberspaceconcertinquiries_view(request):
    chamberspace_owner_object =
Chamberspaceowner.objects.select_related("chamberuseraccount").filter(chamberu
seraccount = request.user)
    if chamberspace_owner_object:

```

```

        chamberspace_owner_object = chamberspace_owner_object[0]
        chamberinquiries =
Chamberspaceconcertinquiry.objects.select_related("chamberspace").select_relate
d("chamberspace__chamberspacetype").select_related("chamberspace__chambers
pacecity").select_related("chamberconcert").select_related("chamberconcert__cha
mbermusicband").filter(chamberinquiryinitializer = False,
chamberspace__chamberspaceowner = chamberspace_owner_object).order_by("-
chamberinquirydt")
        if chamberinquiries:
            for chamberinquiry_object in chamberinquiries:
                chamberinquiry_object.bandphoto =
Chambermusicbandphoto.objects.filter(chambermusicband =
chamberinquiry_object.chamberconcert.chambermusicband)[0]
                chambermusic_band_object =
Chambermusicband.objects.prefetch_related("chamberspaceownersaves").filter(id
= chamberinquiry_object.chamberconcert.chambermusicband.id)
                if chamberspace_owner_object in
chambermusic_band_object.chamberspaceownersaves:
                    chamberinquiry_object.saved = True
                else:
                    chamberinquiry_object.saved = False
            return render(
                request,
                "chambercon/ownchamberinquiries.html",
                {
                    "chamberinquiries": chamberinquiries,
                    "user": chamberspace_owner_object
                }
            )
        else:

```

```

        chamber_musician_object =
Chambermusician.objects.select_related("chamberuseraccount").select_related("ch
ambermusicband").filter(chamberuseraccount = request.user)
        if not chamber_musician_object:
            return render(
                request,
                "chambercon/ownchamberinquiries.html",
                {
                    "chamberinquiries": [],
                    "user": None
                }
            )
        else:
            chamber_musician_object = chamber_musician_object[0]
            chamberinquiries =
Chamberspaceconcertinquiry.objects.select_related("chamberspace").select_relate
d("chamberspace__chamberspacetype").select_related("chamberspace__chambers
pacecity").select_related("chamberconcert").select_related("chamberconcert__cha
mbermusicband").filter(chamberinquiryinitializer = True,
chamberconcert__chambermusicband =
chamber_musician_object.chambermusicband).order_by("-chamberinquirydt")
            if chamberinquiries:
                for chamberinquiry_object in chamberinquiries:
                    chamberinquiry_object.spacephoto =
Chamberspacephoto.objects.filter(chamberspace =
chamberinquiry_object.chamberspace)[0]
                    chamber_space_object =
Chamberspace.objects.prefetch_related("musiciansaves").filter(id =
chamberinquiry_object.chamberspace.id)

```

```

        if chamber_musician_object in
chamber_space_object.musiciansaves:
            chamber_musician_object.saved = True
        else:
            chamber_musician_object.saved = False
    return render(
        request,
        "chambercon/ownchamberinquiries.html",
        {
            "chamberinquiries": chamberinquiries,
            "user": chamber_musician_object
        }
    )

def own_band_chamberspaceconcertinquiries_view(request):
    chamber_musician_object =
Chambermusician.objects.select_related("chamberuseraccount").select_related("ch
ambermusicband").filter(chamberuseraccount = request.user)
    if not chamber_musician_object:
        return render(
            request,
            "chambercon/ownbandchamberinquiries.html",
            {
                "chamberinquiries": [],
                "user": None
            }
        )
    else:
        chamber_musician_object = chamber_musician_object[0]

```

```

        chamberinquiries =
Chamberspaceconcertinquiry.objects.select_related("chamberspace").select_relate
d("chamberspace__chamberspacetype").select_related("chamberspace__chambers
pacecity").select_related("chamberconcert").select_related("chamberconcert__cha
mbermusicband").filter(chamberinquiryinitializer = False,
chamberconcert__chambermusicband =
chamber_musician_object.chambermusicband).order_by("-chamberinquirydt")
        if chamberinquiries:
            for chamberinquiry_object in chamberinquiries:
                chamberinquiry_object.spacephoto =
Chamberspacephoto.objects.filter(chamberspace =
chamberinquiry_object.chamberspace)[0]
                chamber_space_object =
Chamberspace.objects.prefetch_related("musiciansaves").filter(id =
chamberinquiry_object.chamberspace.id)
                if chamber_musician_object in
chamber_space_object.musiciansaves:
                    chamber_musician_object.saved = True
                else:
                    chamber_musician_object.saved = False
            return render(
                request,
                "chambercon/ownspacechamberinquiries.html",
                {
                    "chamberinquiries": chamberinquiries,
                    "user": chamber_musician_object
                }
            )

def own_spaces_chamberspaceconcertinquiries_view(request):

```

```

        chamberspace_owner_object =
Chamberspaceowner.objects.select_related("chamberuseraccount").filter(chamberu
seraccount = request.user)
        if chamberspace_owner_object:
            chamberspace_owner_object = chamberspace_owner_object[0]
            chamberinquiries =
Chamberspaceconcertinquiry.objects.select_related("chamberspace").select_relate
d("chamberspace__chamberspacetype").select_related("chamberspace__chambers
pacecity").select_related("chamberconcert").select_related("chamberconcert__cha
mbermusicband").filter(chamberinquiryinitializer = True,
chamberspace__chamberspaceowner = chamberspace_owner_object).order_by("-
chamberinquirydt")
            if chamberinquiries:
                for chamberinquiry_object in chamberinquiries:
                    chamberinquiry_object.bandphoto =
Chambermusicbandphoto.objects.filter(chambermusicband =
chamberinquiry_object.chamberconcert.chambermusicband)[0]
                    chamber_band_object =
Chambermusicband.objects.prefetch_related("chamberspaceownersaves").filter(id
= chamberinquiry_object.chamberconcert.chambermusicband.id)
                    if chamberspace_owner_object in
chamber_band_object.chamberspaceownersaves:
                        chamber_band_object.saved = True
                    else:
                        chamber_band_object.saved = False
        return render(
            request,
            "chambercon/ownspaceschamberinquiries.html",
            {
                "chamberinquiries": chamberinquiries,

```

```

        "user": chamberspace_owner_object
    }
)
else:
    return render(
        request,
        "chambercon/ownspaceschamberinquiries.html",
        {
            "chamberinquiries": [],
            "user": None
        }
    )

def own_space_chamberspaceconcertinquiries_view(request,
chamberspace_id):
    chamberspace_owner_object =
Chamberspaceowner.objects.select_related("chamberuseraccount").filter(chamberu
seraccount = request.user)
    if chamberspace_owner_object:
        chamberspace_owner_object = chamberspace_owner_object[0]
        chamber_spaces =
Chamberspace.objects.select_related("chamberspaceowner").select_related("cham
berspacetype").select_related("chamberspacecity").filter(chamberspaceowner =
chamberspace_owner_object)
        chamberspace_owner_object.chamberspaces = chamber_spaces
        chamberinquiries =
Chamberspaceconcertinquiry.objects.select_related("chamberspace").select_relate
d("chamberspace__chamberspacetype").select_related("chamberspace__chambers
pacecity").select_related("chamberconcert").select_related("chamberconcert__cha
mbermusicband").filter(chamberinquiryinitializer = True,

```

```

chamberspace__chamberspaceowner = chamberspace_owner_object,
chamberspace__id = chamberspace_id).order_by("-chamberinquirydt")
    if chamberinquiries:
        for chamberinquiry_object in chamberinquiries:
            chamberinquiry_object.bandphoto =
Chambermusicbandphoto.objects.filter(chambermusicband =
chamberinquiry_object.chamberconcert.chambermusicband)[0]
            chamber_band_object =
Chambermusicband.objects.prefetch_related("chamberspaceownersaves").filter(id
= chamberinquiry_object.chamberconcert.chambermusicband.id)
            if chamberspace_owner_object in
chamber_band_object.chamberspaceownersaves:
                chamber_band_object.saved = True
            else:
                chamber_band_object.saved = False
            chamberspace_object =
Chamberspace.objects.select_related("chamberspaceowner").select_related("cham
berspacetype").select_related("chamberspacecity").filter(id = chamberspace_id)
            if chamberspace_object:
                chamberspace_object.photo =
Chamberspacephoto.objects.filter(chamberspace = chamberspace_object)[0]
        return render(
            request,
            "chambercon/ownspacechamberinquiries.html",
            {
                "chamberinquiries": chamberinquiries,
                "user": chamberspace_owner_object,
                "chamberspace": chamberspace_object
            }
        )

```

```

else:
    return render(
        request,
        "chambercon/ownspacechamberinquiries.html",
        {
            "chamberinquiries": [],
            "user": None,
            "chamberspace": None
        }
    )

```

```

def own_chamberconcerts_view(request):
    chamber_musician_object =
Chambermusician.objects.select_related("chamberuseraccount").select_related("ch
ambermusicband").filter(chamberuseraccount = request.user)
    if not chamber_musician_object:
        return render(
            request,
            "chambercon/ownchamberconcerts.html",
            {
                "chamberconcerts": [],
                "user": None
            }
        )
    else:
        chamber_musician_object = chamber_musician_object[0]
        chamberconcerts =
Chamberspaceconcertinquiry.objects.select_related("chamberconcert").select_relat
ed("chamberconcert__chambermusicband").select_related("chamberconcert__cha
mberspace").select_related("chamberconcert__chamberspace__chamberspacecity"

```

```

).filter(chamberconcert__chambermusicband =
chamber_musician_object.chambermusicband, chamberinquirystate =
"A").order_by("-chamberconcert__chamberconcertstart")
        return render(
            request,
            "chambercon/ownchamberconcerts.html",
            {
                "chamberconcerts": chamberconcerts,
                "user": chamber_musician_object
            }
        )

def own_done_chamberconcerts_view(request):
    chamber_musician_object =
Chambermusician.objects.select_related("chamberuseraccount").select_related("ch
ambermusicband").filter(chamberuseraccount = request.user)
    if not chamber_musician_object:
        return render(
            request,
            "chambercon/ownspacechamberdoneconcerts.html",
            {
                "chamberconcerts": [],
                "user": None
            }
        )
    else:
        chamber_musician_object = chamber_musician_object[0]
        chamberconcerts =
Chamberspaceconcertinquiry.objects.select_related("chamberconcert").select_relat
ed("chamberconcert__chambermusicband").select_related("chamberconcert__cha

```

```

mberspace").select_related("chamberconcert__chamberspace__chamberspacecity"
).filter(chamberconcert__chambermusicband =
chamber_musician_object.chambermusicband, chamberinquirystate =
"F").order_by("-chamberconcert__chamberconcertstart")

    return render(
        request,
        "chambercon/ownspacechamberdoneconcerts.html",
        {
            "chamberconcerts": chamberconcerts,
            "user": chamber_musician_object
        }
    )

def own_inquired_chamberconcerts_view(request):
    chamber_musician_object =
    Chambermusician.objects.select_related("chamberuseraccount").select_related("ch
ambermusicband").filter(chamberuseraccount = request.user)

    if not chamber_musician_object:
        return render(
            request,
            "chambercon/ownspacechamberinqconcerts.html",
            {
                "chamberconcerts": [],
                "user": None
            }
        )
    else:
        chamber_musician_object = chamber_musician_object[0]
        chamberconcerts =
        Chamberspaceconcertinquiry.objects.select_related("chamberconcert").select_relat

```

```

ed("chamberconcert__chambermusicband").select_related("chamberconcert__cha
mberspace").select_related("chamberconcert__chamberspace__chamberspacecity"
).filter(chamberconcert__chambermusicband =
chamber_musician_object.chambermusicband, chamberinquirystate =
"C").order_by("-chamberconcert__chamberconcertstart")

    return render(
        request,
        "chambercon/ownspacechamberinqconcerts.html",
        {
            "chamberconcerts": chamberconcerts,
            "user": chamber_musician_object
        }
    )

```

```

def chamber_band_view(request, chambermusicband_id):
    chamber_musicband_object =
Chambermusicband.objects.prefetch_related("chamberspaceownersaves").prefetch
_related("visitorsaves").filter(id = chambermusicband_id)
    if request.user.is_authenticated:
        if request.user in chamber_musicband_object.visitorsaves:
            chamber_musicband_object.saved = True
        else:
            chamberspace_owner =
Chamberspaceowner.objects.filter(chamberuseraccount = request.user)
            if chamberspace_owner and chamberspace_owner in
chamber_musicband_object.chamberspaceownersaves:
                chamber_musicband_object.saved = True
            else:
                chamber_musicband_object.saved = False
    if not chamber_musicband_object:

```

```

        return render(
            request,
            "chambercon/chamberband.html",
            {
                "chamberconcerts": [],
                "chamberarchiveconcerts": [],
                "chamberband": None
            }
        )
    else:
        chamber_musicband_object = chamber_musicband_object[0]
        chamberconcerts =
Chamberspaceconcertinquiry.objects.select_related("chamberconcert").select_relat
ed("chamberconcert__chambermusicband").select_related("chamberconcert__cha
mberspace").select_related("chamberconcert__chamberspace__chamberspacecity"
).filter(chamberconcert__chambermusicband =
chamber_musician_object.chambermusicband, chamberinquirystate =
"A").order_by("-chamberconcert__chamberconcertstart")[:5]
        chamberarchiveconcerts =
Chamberspaceconcertinquiry.objects.select_related("chamberconcert").select_relat
ed("chamberconcert__chambermusicband").select_related("chamberconcert__cha
mberspace").select_related("chamberconcert__chamberspace__chamberspacecity"
).filter(chamberconcert__chambermusicband =
chamber_musician_object.chambermusicband, chamberinquirystate =
"F").order_by("-chamberconcert__chamberconcertstart")[:5]
        return render(
            request,
            "chambercon/chamberband.html",
            {
                "chamberconcerts": chamberconcerts,

```

```

        "chamberarchiveconcerts": chamberarchiveconcerts,
        "chamberband": chamber_musicband_object
    }
)

def chamber_band_concerts_view(request, chambermusicband_id):
    chamber_musicband_object = Chambermusicband.objects.filter(id =
chambermusicband_id)
    if request.user.is_authenticated:
        if request.user in chamber_musicband_object.visitorsaves:
            chamber_musicband_object.saved = True
        else:
            chamberspace_owner =
Chamberspaceowner.objects.filter(chamberuseraccount = request.user)
            if chamberspace_owner and chamberspace_owner in
chamber_musicband_object.chamberspaceownersaves:
                chamber_musicband_object.saved = True
            else:
                chamber_musicband_object.saved = False
    if not chamber_musicband_object:
        return render(
            request,
            "chambercon/chamberbandconcerts.html",
            {
                "chamberconcerts": [],
                "chamberband": None
            }
        )
    else:
        chamber_musicband_object = chamber_musicband_object[0]

```

```

        chamberconcerts =
Chamberspaceconcertinquiry.objects.select_related("chamberconcert").select_relat
ed("chamberconcert__chambermusicband").select_related("chamberconcert__cha
mberspace").select_related("chamberconcert__chamberspace__chamberspacecity"
).filter(chamberconcert__chambermusicband =
chamber_musician_object.chambermusicband, chamberinquirystate =
"A").order_by("-chamberconcert__chamberconcertstart")

        return render(
            request,
            "chambercon/chamberbandconcerts.html",
            {
                "chamberconcerts": chamberconcerts,
                "chamberband": chamber_musicband_object
            }
        )

```

```

def chamber_band_archive_view(request, chambermusicband_id):
    chamber_musicband_object = Chambermusicband.objects.filter(id =
chambermusicband_id)

    if request.user.is_authenticated:
        if request.user in chamber_musicband_object.visitorsaves:
            chamber_musicband_object.saved = True
        else:
            chamberspace_owner =
Chamberspaceowner.objects.filter(chamberuseraccount = request.user)
            if chamberspace_owner and chamberspace_owner in
chamber_musicband_object.chamberspaceownersaves:
                chamber_musicband_object.saved = True
            else:
                chamber_musicband_object.saved = False

```

```
if not chamber_musicband_object:
```

```
    return render(
        request,
        "chambercon/chamberarchiveconcerts.html",
        {
            "chamberconcerts": [],
            "chamberarchiveconcerts": [],
            "chamberband": None
        }
    )
```

```
else:
```

```
    chamber_musicband_object = chamber_musicband_object[0]
    chamberarchiveconcerts =
```

```
    Chamberspaceconcertinquiry.objects.select_related("chamberconcert").select_relat
    ed("chamberconcert__chambermusicband").select_related("chamberconcert__cha
    mberspace").select_related("chamberconcert__chamberspace__chamberspacecity"
    ).filter(chamberconcert__chambermusicband =
    chamber_musician_object.chambermusicband, chamberinquirystate =
    "F").order_by("-chamberconcert__chamberconcertstart")
```

```
    return render(
        request,
        "chambercon/chamberarchiveconcerts.html",
        {
            "chamberarchiveconcerts": chamberarchiveconcerts,
            "chamberband": chamber_musicband_object
        }
    )
```

```
def chamberconcert_reviews_view(request, chamberconcert_id):
```

```

    chamber_concert_object =
Chamberconcert.objects.prefetch_related("visitorsaves").select_related("chamberm
usicband").filter(id = chamberconcert_id)
    if request.user.is_authenticated:
        if request.user in chamber_concert_object.visitorsaves:
            chamber_concert_object.saved = True
        else:
            chamber_concert_object.saved = False
    if not chamber_concert_object:
        return render(
            request,
            "chambercon/chamberconcertreviews.html",
            {
                "chamberconcert": None,
                "chamberconcertreviews": []
            }
        )
    else:
        chamber_concert_object = chamber_concert_object[0]
        chamber_concert_reviews =
Chamberconcertreview.objects.select_related("chambervisitor").filter(chambercon
cert = chamber_concert_object).order_by("-chamberconcertreviewdt")
        if chamber_concert_reviews:
            for chamber_concert_review in chamber_concert_reviews:
                chamber_concert_review.photos =
Chamberconcertreviewphoto.objects.filter(chamberconcertreview =
chamber_concert_review)
                chamber_concert_review.videos =
Chamberconcertreviewvideo.objects.filter(chamberconcertreview =
chamber_concert_review)

```

```

return render(
    request,
    "chambercon/chamberconcertreviews.html",
    {
        "chamberconcert": chamber_concert_object,
        "chamberconcertreviews": chamber_concert_reviews
    }
)

```

```

def chamberband_reviews_view(request, chamberband_id):
    chamber_band_object =
Chambermusicband.objects.prefetch_related("chamberspaceownersaves").prefetch
_related("visitorsaves").filter(id = chamberband_id)
    if request.user.is_authenticated:
        if request.user in chamber_band_object.visitorsaves:
            chamber_band_object.saved = True
        else:
            chamberspace_owner =
Chamberspaceowner.objects.filter(chamberuseraccount = request.user)
            if chamberspace_owner and chamberspace_owner in
chamber_musicband_object.chamberspaceownersaves:
                chamber_band_object.saved = True
            else:
                chamber_band_object.saved = False
    if not chamber_band_object:
        return render(
            request,
            "chambercon/chamberbandreviews.html",
            {
                "chamberband": None,

```

```

        "chamberbandreviews": []
    }
)
else:
    chamber_band_object = chamber_band_object[0]
    chamber_band_reviews =
Chambermusicbandreview.objects.select_related("chambervisitor").filter(chamber
musicband = chamber_band_object).order_by("-chambermusicbandreviewdt")
    if chamber_band_reviews:
        for chamber_band_review in chamber_band_reviews:
            chamber_band_review.photos =
Chambermusicbandreviewphoto.objects.filter(chambermusicbandreview =
chamber_band_review)
    return render(
        request,
        "chambercon/chamberbandreviews.html",
        {
            "chamberband": chamber_band_object,
            "chamberbandreviews": chamber_band_reviews
        }
    )

```

**ДОДАТОК В**  
**Слайди презентації**

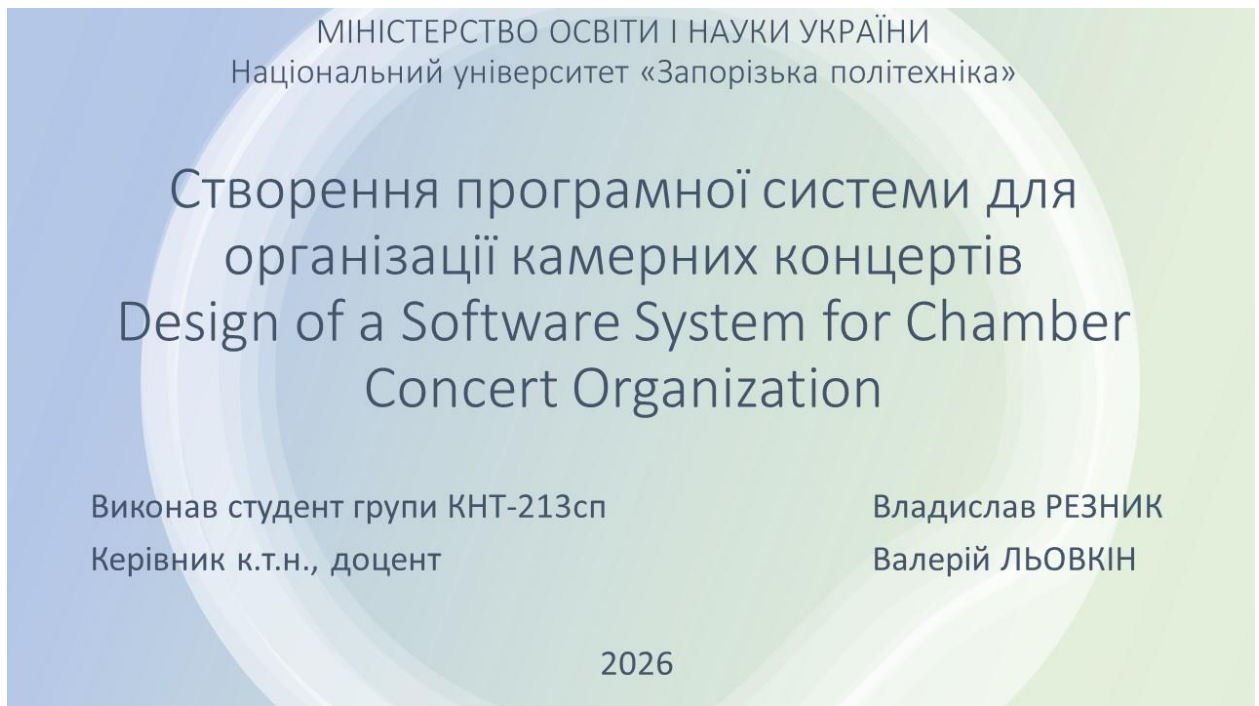


Рисунок В.1 – Початковий слайд

- Об'єкт роботи – процес побудови програмних систем для організації камерних концертів.
- Предмет роботи – програмні системи для організації камерних концертів.
- Мета роботи – збільшення кількості варіантів організації камерних концертів за рахунок надання доступу власниками до невеликих приміщень за допомогою програмної системи.

Рисунок В.2 – Об'єкт, предмет, мета роботи

# Завдання роботи

- аналіз проблеми розробки програмної системи для організації камерних концертів;
- проектування програмної системи для організації камерних концертів;
- розробка програмної системи для організації камерних концертів.

3

Рисунок В.3 – Завдання роботи

# Порівняння програмних рішень для організації камерних концертів

| Критерій                                    | DICE                                                 | Fever             | Songkick                       | Розробка                                               |
|---------------------------------------------|------------------------------------------------------|-------------------|--------------------------------|--------------------------------------------------------|
| Коло місць проведення концертів             | Може розширюватись                                   | Стан-дартне       | Стан-дартне, включаючи камерні | Може розширюватись за рахунок нестандартних (камерних) |
| Слідкування за концертами                   | Так, в тому числі за місцями проведення, колективами | Тільки пошук      | Тільки пошук                   | Так, в тому числі за місцями проведення, колективами   |
| Пошук за місцями проведення концертів       | Так                                                  | Ні                | Так                            | Так                                                    |
| Адресування запитів на проведення концертів | Ні                                                   | Ні                | Ні                             | Так                                                    |
| Відгуки                                     | Ні                                                   | Так, про концерти | Так, про концерти колективу    | Так, про місця проведення, концерти та колективи       |

4

Рисунок В.4 – Порівняння програмних рішень для організації камерних концертів

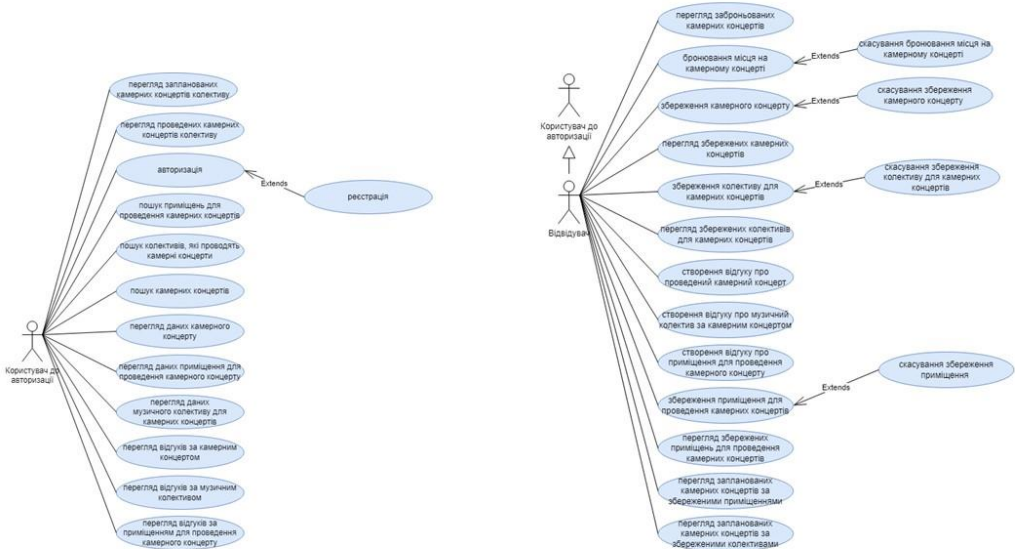
Порівняння фреймворків для створення програмної системи для організації камерних концертів

| Критерій                        | Django                                                         | Pyramid                                                                       |
|---------------------------------|----------------------------------------------------------------|-------------------------------------------------------------------------------|
| Витрати на розробку             | Менші за часом                                                 | Більші за часом                                                               |
| Спосіб взаємодії з базою даних  | На основі вбудованого об'єктно-реляційного відображення        | На основі об'єктно-реляційного відображення, яке потрібно обрати і встановити |
| Структура ПС                    | Регламентована за підсистемами представлень, моделей, шаблонів | Змінювана                                                                     |
| Засоби авторизації користувачів | Наявні вбудовані                                               | Підлягають налаштуванню                                                       |

5

Рисунок В.5 – Порівняння фреймворків для створення програмної системи для організації камерних концертів

Діаграми прецедентів для користувача до авторизації та відвідувача у програмній системі організації камерних концертів

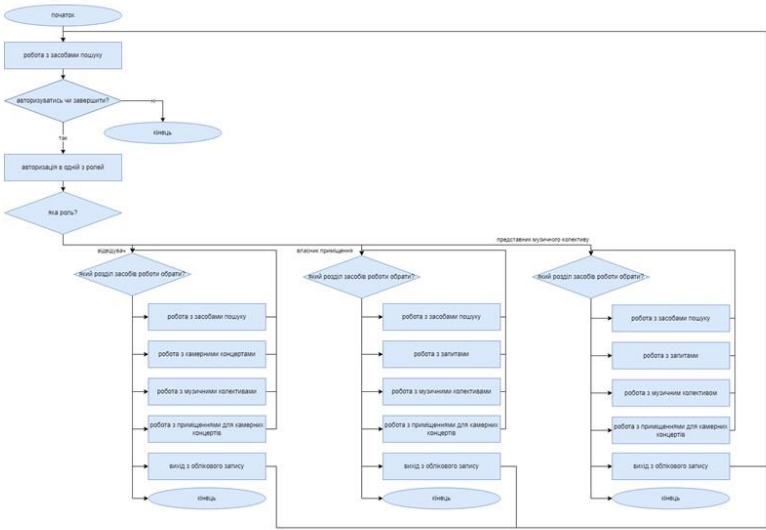


6

Рисунок В.6 – Діаграми прецедентів для користувача до авторизації та відвідувача у програмній системі організації камерних концертів

Рисунок В.8 – Схема бази даних програмної системи для організації камерних концертів

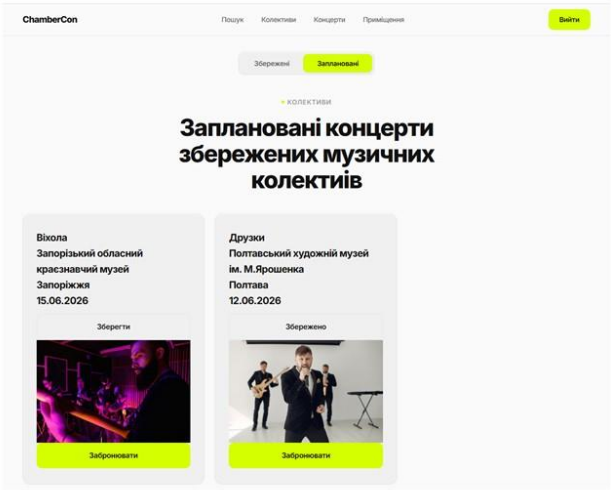
Функціональна схема програмної системи для організації камерних концертів



9

Рисунок В.9 – Функціональна схема програмної системи для організації камерних концертів

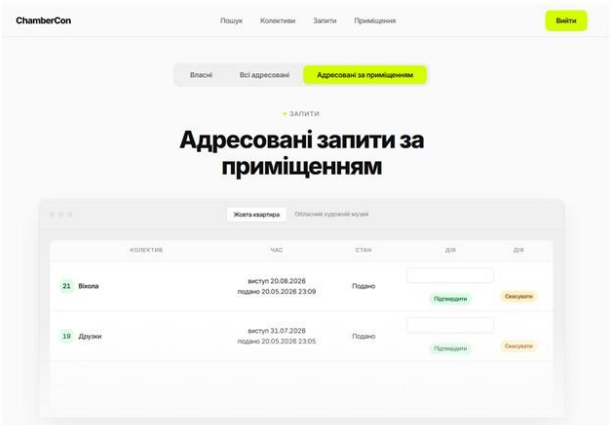
Перегляд запланованих камерних концертів за збереженими музичними колективами відвідувачем у програмній системі



10

Рисунок В.10 – Перегляд запланованих камерних концертів за збереженими музичними колективами відвідувачем у програмній системі

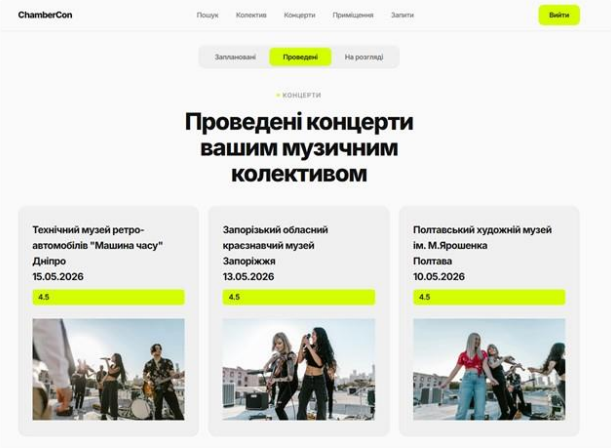
Керування запитами на проведення камерних концертів, адресованими за приміщенням власника у програмній системі



11

Рисунок В.11 – Керування запитами на проведення камерних концертів, адресованими за приміщенням власника у програмній системі

Перегляд проведених камерних концертів музичним колективом авторизованого представника у програмній системі



12

Рисунок В.12 – Перегляд проведених камерних концертів музичним колективом авторизованого представника у програмній системі

## Результати тестування програмної системи організації камерних концертів

| Функціональна можливість                                                               | Результат | Функціональна можливість                                             | Результат | Функціональна можливість                                                          | Результат |
|----------------------------------------------------------------------------------------|-----------|----------------------------------------------------------------------|-----------|-----------------------------------------------------------------------------------|-----------|
| Авторизація                                                                            | +         | Скасування збереження камерного концерту                             | +         | Визначення власного приміщення для проведення камерних концертів                  | +         |
| Реєстрація                                                                             | +         | Перегляд збережених камерних концертів                               | +         | Зняття власного приміщення для проведення камерних концертів з розгляду           | +         |
| Пошук приміщень для проведення камерних концертів                                      | +         | Збереження колективу для камерних концертів                          | +         | Редагування даних власного приміщення для проведення камерних концертів           | +         |
| Визначення власного колективу для камерних концертів                                   | +         | Скасування збереження колективу для камерних концертів               | +         | Перегляд власних приміщень для проведення камерних концертів                      | +         |
| Редагування даних власного колективу для камерних концертів                            | +         | Перегляд збережених колективів для камерних концертів                | +         | Запит на проведення камерного концерту в приміщенні                               | +         |
| Перегляд власних запланованих камерних концертів                                       | +         | Створення відгуку про проведений камерний концерт                    | +         | Підтвердження запиту на проведення камерного концерту в приміщенні                | +         |
| Перегляд власних проведених камерних концертів                                         | +         | Створення відгуку про музичний колектив за камерним концертом        | +         | Скасування запиту на проведення камерного концерту в приміщенні                   | +         |
| Перегляд власних камерних концертів на розгляді                                        | +         | Створення відгуку про приміщення для проведення камерного концерту   | +         | Перегляд власних запитів на проведення камерного концерту в приміщенні            | +         |
| Збереження приміщення для проведення камерних концертів                                | +         | Перегляд даних камерного концерту                                    | +         | Перегляд адресованих запитів на проведення камерних концертів за приміщенням      | +         |
| Скасування збереження приміщення для проведення камерних концертів                     | +         | Перегляд даних музичного колективу для камерних концертів            | +         | Перегляд адресованих запитів на проведення камерних концертів за приміщенням      | +         |
| Перегляд збережених приміщень для проведення камерних концертів                        | +         | Перегляд відгуків за камерним концертом                              | +         | Перегляд переліку відвідувачів камерного концерту                                 | +         |
| Пропонування власного приміщення для проведення камерного концерту                     | +         | Перегляд відгуків за музичним колективом                             | +         | Перегляд адресованих запитів на проведення камерних концертів музичному колективу | +         |
| Пошук колективів, які проводять камерні концерти                                       | +         | Перегляд запланованих камерних концертів колективу                   | +         | Перегляд запланованих камерних концертів за збереженими колективами               | +         |
| Пошук камерних концертів                                                               | +         | Перегляд проведених камерних концертів колективу                     | +         |                                                                                   |           |
| Бронювання місця на камерному концерті                                                 | +         | Перегляд запланованих камерних концертів за збереженими приміщеннями | +         |                                                                                   |           |
| Скасування бронювання місця на камерному концерті                                      | +         | Перегляд заброньованих камерних концертів                            | +         |                                                                                   |           |
| Збереження камерного концерту                                                          | +         |                                                                      |           |                                                                                   |           |
| Перегляд адресованих запитів на проведення камерних концертів за приміщеннями власника | +         |                                                                      |           |                                                                                   |           |
| Перегляд відгуків за приміщенням для проведення камерного концерту                     | +         |                                                                      |           |                                                                                   |           |

13

Рисунок В.13 – Результати тестування програмної системи організації камерних концертів

## Висновки

- Розробка програмної системи для організації камерних концертів виконувалась для збільшення кількості варіантів організації камерних концертів за рахунок надання доступу власникам до невеликих приміщень. Мета роботи в підсумку була досягнута, оскільки розроблена програмна система дозволяє власникам невеликих приміщень виставляти їх як варіанти для проведення камерних концертів, знаходити музичні колективи для цих концертів, а представникам музичних колективів відповідно знаходити майданчики для проведення камерних концертів, звертатись до власників цих приміщень. Таким чином кількість варіантів збільшується за рахунок такої можливої комунікації.

14

Рисунок В.14 – Висновки