



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №2

із дисципліни «Технології розробки програмного забезпечення»

**Тема: «Діаграма варіантів використання. Сценарії варіантів використання.
Діаграма UML. Діаграми класів. Концептуальна модель системи»**

Виконав:
Студент групи ІА-24
Зелінський І.О.

Перевірив:
Мякий М. Ю.

Зміст

Тема.....	3
Мета.....	3
Обрана тема	3
Опис трьох прецедентів.....	3
1. Адміністратор керує піснями (видаляє/додає).....	3
2. Користувач переглядає історію прослуховувань	4
3. Користувач створює свій плейлист.....	4
Діаграма прецедентів.....	5
Діаграма класів.....	6
Опис моделей.....	7
Зв'язки між моделями.....	8
Структура БД	9
Висновок.....	9

Тема: Діаграма варіантів використання. Сценарії варіантів використання. Діаграма UML. Діаграми класів. Концептуальна модель системи

Мета: Вивчення основних принципів побудови та проектування програмних систем з використанням схем прецедентів, діаграм класів та структури бази даних.

Хід роботи

Обрана тема: Online radio station (iterator, adapter, factory method, facade, visitor, client-server)

Додаток повинен служити сервером для радіостанції з можливістю мовлення на радіостанцію (64, 92, 128, 196, 224 kb/s) в потоковому режимі; вести облік підключених користувачів і статистику відвідувань і прослуховувань; налаштувати папки з піснями і можливість вести списки програвання або playlists (не відтворювати всі пісні).

Опис трьох прецедентів:

1. Адміністратор керує піснями (видаляє/додає)

Передумови:

Адміністратор увійшов у систему з правами адміністратора.

У системі вже є наявний набір пісень, з якими можна взаємодіяти (для видалення).

Післяумови:

Якщо пісню було додано, вона зберігається в репозиторії пісень і стає доступною для використання (наприклад, у плейлистах).

Якщо пісню було видалено, вона більше не доступна для користувачів і плейлистів.

Хід подій:

Адміністратор відкриває розділ керування піснями.

Вибирає опцію додавання пісні або видалення наявної пісні.

У випадку додавання, вводить деталі нової пісні і зберігає її.

У випадку видалення, вибирає пісню зі списку і видаляє її.

Система підтверджує успішне додавання або видалення пісні.

Винятки:

Пісня не може бути додана: Якщо дані пісні неповні або некоректні (наприклад, відсутній виконавець).

2. Користувач переглядає історію прослуховувань

Передумови:

Користувач увійшов у систему.

Післяумови:

Користувач бачить список пісень, які він прослуховував.

Хід подій:

Користувач заходить на свою сторінку або в розділ історії прослуховувань.

Система відображає користувачу список його прослуханих пісень.

Користувач може переглянути деталі кожного запису історії та послухати пісню/додати у плейлист.

Винятки:

Історія недоступна: Якщо історія прослуховувань відсутня (наприклад, користувач не прослуховував жодної пісні), система може відобразити відповідне повідомлення.

3. Користувач створює свій плейлист

Передумови:

Користувач увійшов у систему.

У системі доступні пісні, які можна додати до плейлисту.

Післяумови:

Плейлист створено та збережено з унікальним ID, користувач може його редагувати або переглядати.

Користувач може слухати свій плейлист, а також додавати або видаляти пісні з плейлисту.

Хід подій:

Користувач відкриває розділ зі створенням нового плейлисту.

Вводить назву плейлисту, вибирає статус (публічний, приватний, закритий) та додає початкові пісні зі списку доступних.

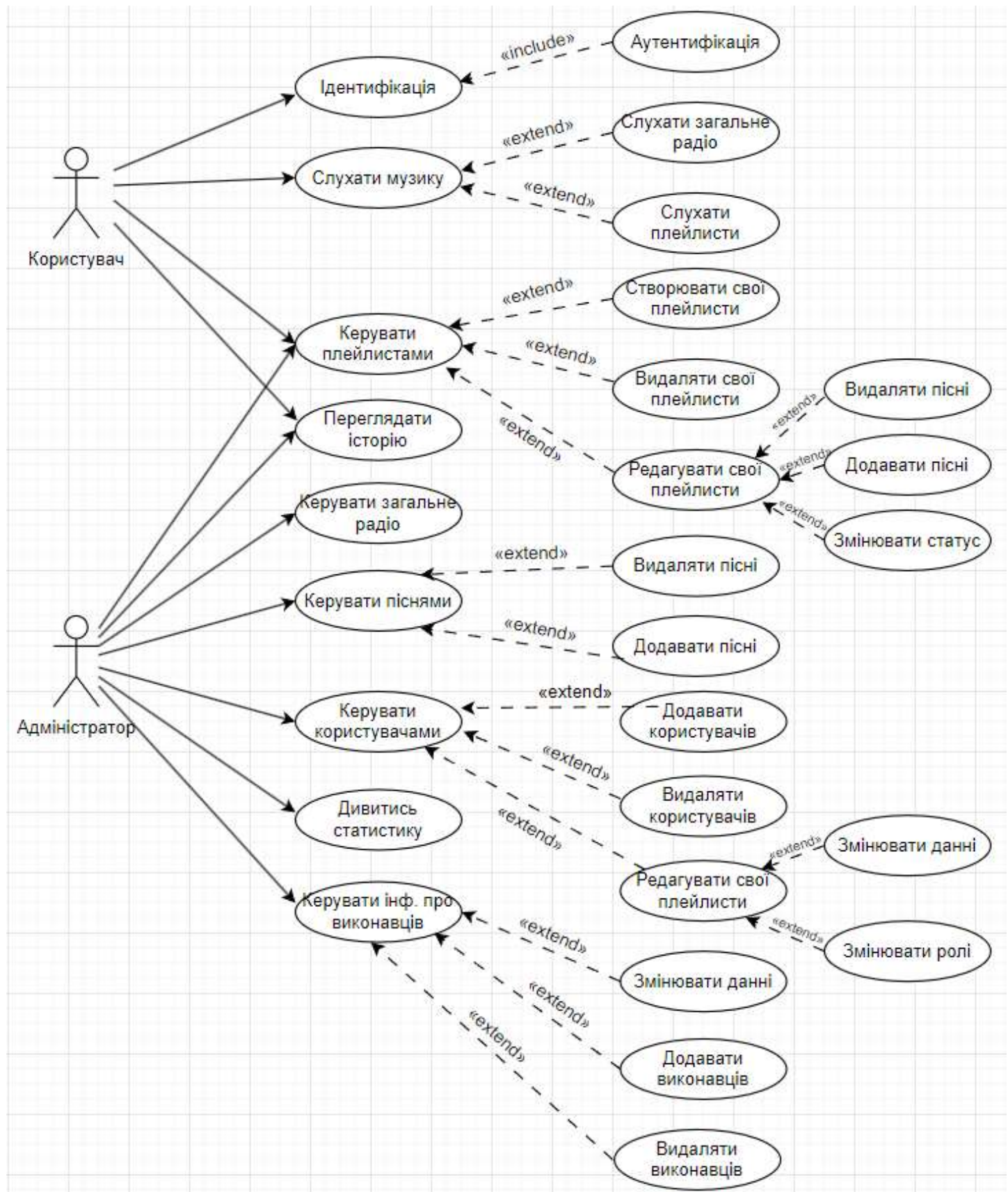
Система зберігає плейлист, прив'язуючи його до ID користувача.

Користувач може повернутися до редагування плейлисту пізніше (додавати чи видаляти пісні).

Винятки:

Неможливо створити плейлист: Якщо користувач не вказав назву плейлисту, система може повідомити про помилку і вимагати введення необхідних даних.

Діаграма прецедентів:



Діаграма класів



Опис моделей:

1. User

Поля:

- id: Унікальний ідентифікатор користувача.
- login: Логін користувача для входу.
- password: Пароль користувача.
- ipAddress: IP-адреса користувача.
- role: Роль користувача (адмін, звичайний).
- statisticIds: Список ID статистик користувача.
- playlistIds: Список ID плейлистів користувача.

Методи:

- Геттери/сеттери;
- addPlaylistId(): Додати ID плейлиста.

2. Playlist

Поля:

- id: Унікальний ідентифікатор плейлиста.
- name: Назва плейлиста.
- songIds: Список ID пісень у плейлисті.
- ownerId: ID користувача-власника.
- status: Статус плейлиста (публічний, приватний тощо).

Методи:

- Геттери/сеттери;
- addSongId(): Додати ID пісні.

3. Song

Поля:

- id: Унікальний ідентифікатор пісні.
- title: Назва пісні.
- artistIds: Список ID виконавців пісні.

Методи:

- Геттери/сеттери;
- addArtistId(): Додати ID виконавця.

4. Artist (Виконавець)

Поля:

- id: Унікальний ідентифікатор виконавця.
- name: Ім'я виконавця.

Методи:

- Геттери/сеттери.

5. Statistic (Статистика)

Поля:

- id: Унікальний ідентифікатор статистики.
- visitTime: Час відвідування.
- listeningDuration: Тривалість прослуховування.
- songIds: Список ID пісень, що прослуховувалися.

Методи:

- Геттери/сеттери;
- addSongId(): Додати ID пісні.

6. Role (Роль)

Поля:

- roleName: Назва ролі (адмін, звичайний користувач).

Методи:

- getRoleName()/setRoleName(): Отримати/встановити назву ролі.

Зв'язки між моделями:

User → Role (Один до багатьох)

Один користувач має одну роль, яка визначає його права (наприклад, адмін чи звичайний користувач), але може бути багато представників одної ролі.

User → Playlist (Один до багатьох)

Опис: Один користувач може створити кілька плейлистів, але кожен плейлист належить лише одному користувачу.

User → Statistic (Один до багатьох)

Опис: Один користувач має багато статистичних записів, що зберігають інформацію про його прослуховування.

Playlist → Song (Багато до багатьох)

Опис: Один плейлист може містити багато пісень, а одна пісня може входити до різних плейлистів.

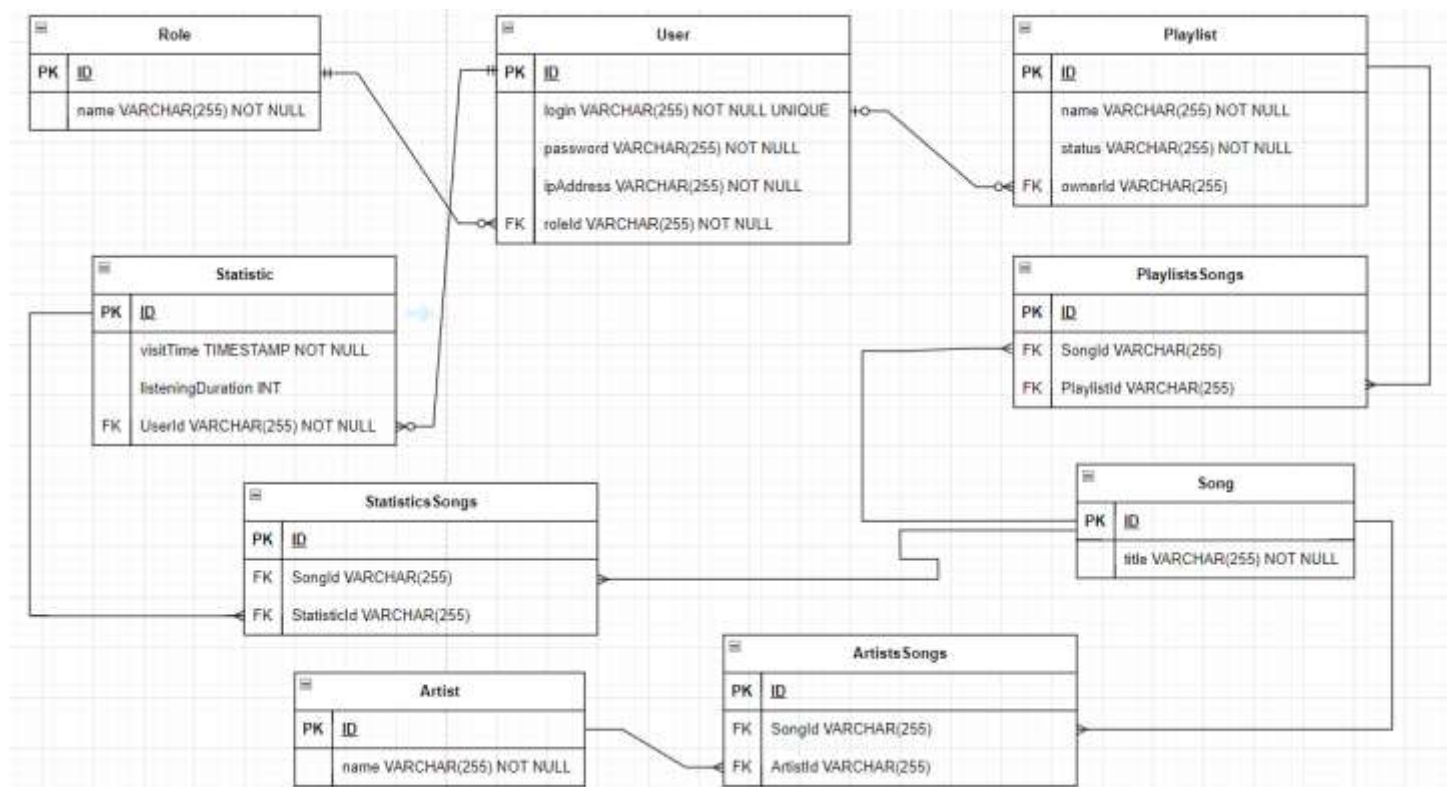
Song → Artist (Багато до багатьох)

Опис: Одна пісня може мати кількох виконавців, так само як один виконавець може брати участь у багатьох піснях.

Statistic → Song (Багато до багатьох)

Опис: Один статистичний запис може містити багато прослуханих пісень, і кожна пісня може бути присутньою у багатьох статистиках.

Структура БД:



Висновок: на цій лаб. роботі я навчився створювати та детально продумувати діаграму прецедентів, також створювати діаграму класів та реалізовувати класи. Після набутих знань я використав їх для створення ПЗ зі своєї теми.