



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 розроблення програмного забезпечення»*

**Тема: «ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ
ВАРІАНТІВ ВИКОРИСТАННЯ.
ДІАГРАМИ UML. ДІАГРАМИ
КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ
СИСТЕМИ»**

Виконав:

Студент групи ІА-23

Хохол М.В.

Перевірив:

Мягкий М.Ю.

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторної.
3. Намалюйте діаграму класів для реалізованої частини системи.
4. Виберіть 3 прецеденти і напишіть на їх основі прецеденти.
5. Розробити основні класи і структуру системи баз даних.
6. Класи даних повинні реалізувати шаблон Репозиторію для взаємодії з базою даних.
7. Підготувати звіт про хід виконання лабораторних робіт. Звіт, що подається повинен містити: діаграму прецедентів, діаграму класів системи, вихідні коди класів системи, а також зображення структури бази даних.

Зміст

<u>Аналіз теми та шаблонів проектування</u>	<u>3</u>
<u>Створення діаграми прецедентів</u>	<u>6</u>
<u>Опис 3 прецедентів</u>	<u>7</u>
<u>Створення діаграми класів для частини майбутньої системи</u>	<u>10</u>
<u>Представлення схеми таблиць бази даних</u>	<u>13</u>
<u>Висновок</u>	<u>14</u>

Хід роботи:

Крок 1. Аналіз теми та шаблонів проектування.

Варіант 7

Редактор зображень (state, prototype, memento, facade, composite, client-server) Редактор зображень має такі функціональні можливості: відкриття/збереження зображень у найпопулярніших форматах (5 на вибір студента), застосування ефектів, наприклад поворот, розтягування, стиснення, кадрування зображення, можливість створення колажів шляхом «нашарування» зображень.

Я обрав відкриття/збереження зображень в форматах ("jpg", "png", "tiff", "bmp", "gif")

Та обрав такі ефекти як:

- 1) оператор Собеля або виявленням країв методом Собеля. Це один із методів обробки зображень, який підкреслює межі об'єктів на фото, виділяючи контрастні переходи між різними кольорами або відтінками. З точки зору користувача, цей ефект виглядає як обведення контурів білими або світлими лініями на темному тлі, що створює враження схематичного, майже рентгенівського, вигляду зображення.
- 2) Блур (розмиття). Реалізований за допомогою гауссового розмиття.
- 3) Створення колажів (реалізація цілих композицій з різноманітним розташуванням картинок, декором)
- 4) Чорно-білий чи монохромний ефект. Цей ефект видаляє всі кольори, залишаючи тільки відтінки сірого. У результаті кожен кольоровий піксель замінюється його яскравим еквівалентом у градаціях сірого.
- 5) Збільшення яскравості зображення на вказаний відсоток.
- 6) Пікселізація зі встановленням розміру пікселю

Перед початком роботи над лабораторною роботою, варто проаналізувати надані шаблони проектування та визначитись з варіантами їх використання.

Шаблон	Суть	Варіант застосування
State	Шаблон дозволяє об'єкту змінювати свою поведінку в залежності від свого стану, забезпечуючи динамічну зміну класів.	В залежності від вибраного ефекту редактор переходить у відповідний стан для роботи з цими змінами
Prototype	Дозволяє створювати нові об'єкти копіюванням існуючих. Використовується, коли створення об'єкта є складним або ресурсозатратним.	Створення нових проектів на основі старих
Memento	Дозволяє робити знімки внутрішнього стану об'єктів, а потім відновлювати їх	Кожного разу, коли користувач застосовує ефект, програма може зберігати поточний стан зображення до якого можна буде повернутись у майбутньому
Facade	Використовується для спрощення доступу до складної системи, надаючи єдиний інтерфейс для взаємодії з нею	Забезпечує спрощений інтерфейс для взаємодії з усіма ефектами

Composite	Призначений для роботи з ієрархією об'єктів, де індивідуальні об'єкти та групи об'єктів обробляються однаково	Можливість застосовувати ефекти до всіх зображень одночасно
Client-server	Архітектурний патерн, який розділяє систему на клієнтську частину (фронтенд), що надсилає запити, та серверну частину (бекенд), що обробляє ці запити, виконує потрібну логіку та надсилає відповіді клієнту	Взаємодія між інтерфейсом користувача та сервером для збереження, обробки та отримання даних зображень і проектів.
Репозиторій	Абстрагування логіки доступу до даних і ізолювати її від решти коду, що дозволяє зробити систему гнучкішою та легшою для підтримки.	Організація доступу до даних.

Крок 2. Створення діаграми прецедентів.

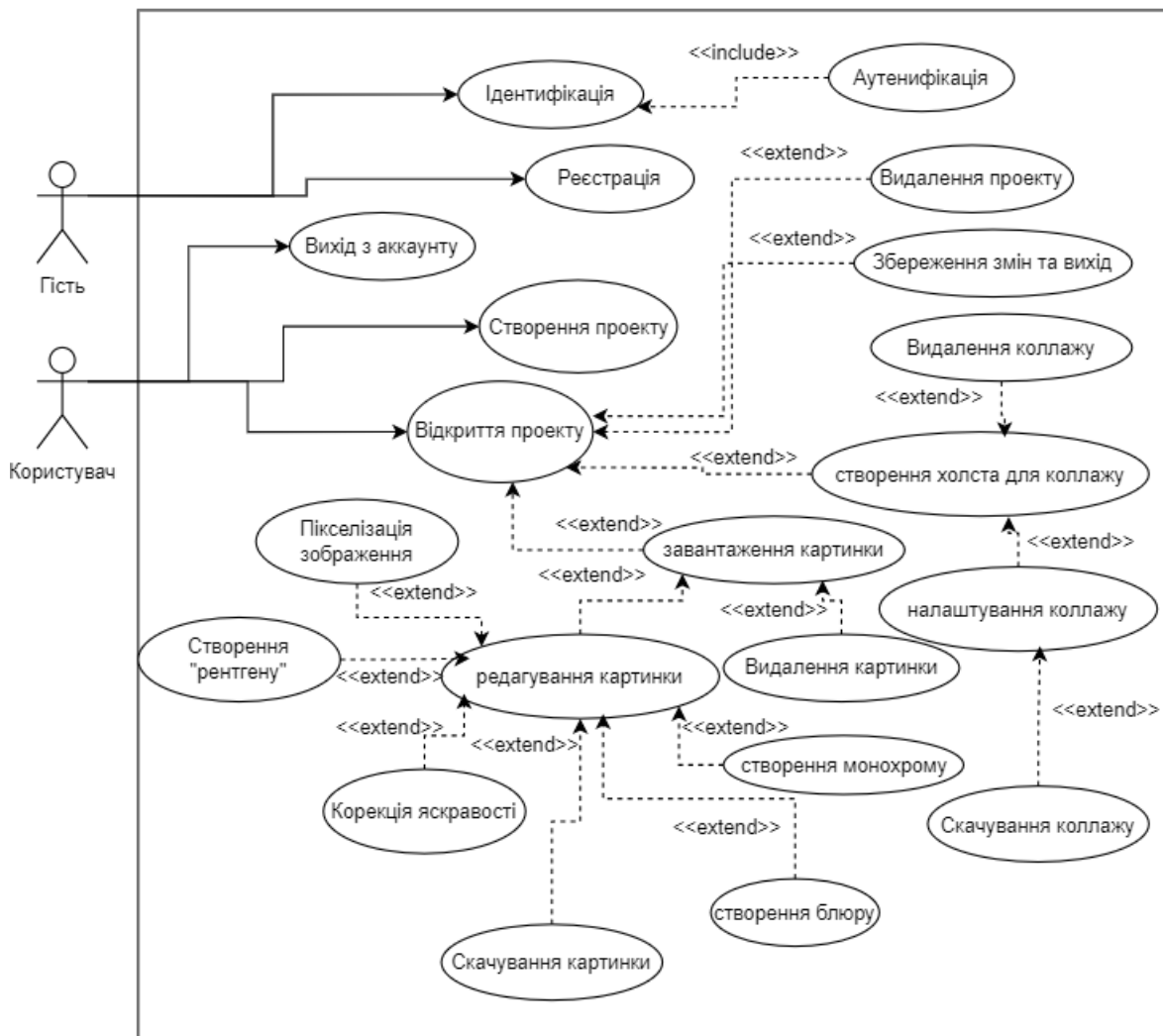


Рис. 1 - Діаграма прецедентів

Крок 3. Опис 3 прецедентів.

Таблиця 1.1 – Сценарій використання «Авторизація»

Назва	Авторизація
ID	1
Опис	Гість виконує процес входу в свій обліковий запис
Актори	Гість
Вигоди компанії	Забезпечення безпеки облікових даних, персоналізований доступ до функцій та можливість відстежування активності користувача
Частота користування	Постійно
Тригери	Користувач бажає отримати доступ до свого облікового запису
Передумови	Користувач має дійсний обліковий запис та правильні облікові дані (логін та пароль)
Постумови	Користувач отримує доступ до свого облікового запису та може використовувати функціонал системи, який доступний для його рівня доступу
Основний розвиток	Користувач відкриває сторінку входу на сайті або додатку. Користувач вводить свій логін та пароль. Система перевіряє правильність облікових даних. Якщо дані введені правильно, система надає доступ до облікового запису користувача.
Альтернативні розвитку	Якщо користувач вводить неправильні облікові дані, система повідомляє про помилку та просить ввести правильні дані
Виняткові ситуації	Система недоступна через технічні проблеми

Таблиця 1.2 – Сценарій використання «Вихід з облікового запису»

Назва	Вихід з облікового запису
ID	2
Опис	Користувач виконує процес виходу зі свого облікового запису
Актори	Користувач
Вигоди компанії	Забезпечення безпеки облікових даних та закриття доступу до функцій облікового запису
Частота користування	Часто, проте залежить від користувача. Сесія автоматично закінчується після 30хв
Тригери	Користувач бажає завершити сеанс роботи та вийти зі свого облікового запису
Передумови	Користувач має активний обліковий запис та перебуває у системі
Постумови	Користувач завершує сеанс роботи та виходить з облікового запису, не маючи доступу до функцій та конфіденційної інформації
Основний розвиток	<p>Користувач знаходиться в системі та бажає вийти з облікового запису. Користувач натискає на кнопку або виконує дію, щоб вийти з облікового запису.</p> <p>Система підтверджує намір користувача і завершує сеанс його облікового запису. Перекидує на сторінку авторизації</p>
Альтернативні розвитку	-
Виняткові ситуації	-

Таблиця 1.3 – Сценарій використання «Блур зображення»

Назва	Блур зображення
ID	3
Опис	Користувач виконує процес побітової зміни зображення
Актори	Користувач
Вигоди компанії	-
Частота користування	Постійно
Тригери	Користувач бажає отримати заблюрене зображення
Передумови	Користувач має дійсний обліковий запис та правильні облікові дані (логін та пароль), а також створений та відкритий проект
Постумови	Користувач отримує оброблене зображення з подальшою можливістю скачати його
Основний розвиток	Користувач відкриває проект. Користувач завантажує зображення. Клієнтська складова системи відображає зображення (без відправлення на серверну частину) Користувач натискає на кнопку “Заблюрити зображення” Зображення передається на сервер, оброблюється та картинка повертається на клієнтську частину
Альтернативні розвитку	-
Виняткові ситуації	Картинка вже була заблюрена, тому при повторному запиті система не буде робити це

Крок 4. Створення діаграми класів для частини майбутньої системи.

Опис класів:

User - клас який представляє користувача, містить:

- id: Унікальний ідентифікатор користувача в системі.
- username: Ім'я користувача для ідентифікації.
- email: Електронна адреса користувача (повинна бути унікальною).
- password: Пароль користувача для авторизації.

Project – клас який представляє проект, містить:

- id: Унікальний ідентифікатор проекту.
- name: Назва проекту, яку задає користувач.
- collages: Список колажів, пов'язаних з проектом.
- images: Список зображень, що належать проекту.

Image – клас який представляє завантажене зображення, містить:

- id: Унікальний ідентифікатор зображення.
- name: Назва зображення (корисно для ідентифікації).
- type: Тип файлу зображення (формат, наприклад, "jpg", "png").
- picByte: Дані зображення, збережені у вигляді масиву байтів.
- project: Проект, до якого прив'язане це зображення.

Collage – клас який представляє колаж, містить:

- id: Унікальний ідентифікатор колажу.
- backgroundColor: Колір фону для колажу.
- gridGap: Відстань між клітинками в сітці колажу.
- gridX: Кількість колонок у сітці колажу.
- gridY: Кількість рядків у сітці колажу.
- width: Ширина колажу.
- height: Висота колажу.

- padding: Відступ всередині колажу.
- left: Позиція колажу по осі X (зліва).
- top: Позиція колажу по осі Y (зверху).
- project: Проект, до якого належить цей колаж.
- mergedCells: Список об'єднаних клітинок у колажі (для збереження комбінацій клітинок).

MergedCells – клас який представляє об'єднані клітинки колажу, містить:

- id: Унікальний ідентифікатор об'єднаної клітинки.
- startRow: Початковий рядок, з якого починається об'єднання.
- endRow: Кінцевий рядок об'єднання.
- startCol: Початкова колонка об'єднання.
- endCol: Кінцева колонка об'єднання.
- collage: Колаж, до якого належить ця об'єднана клітинка.

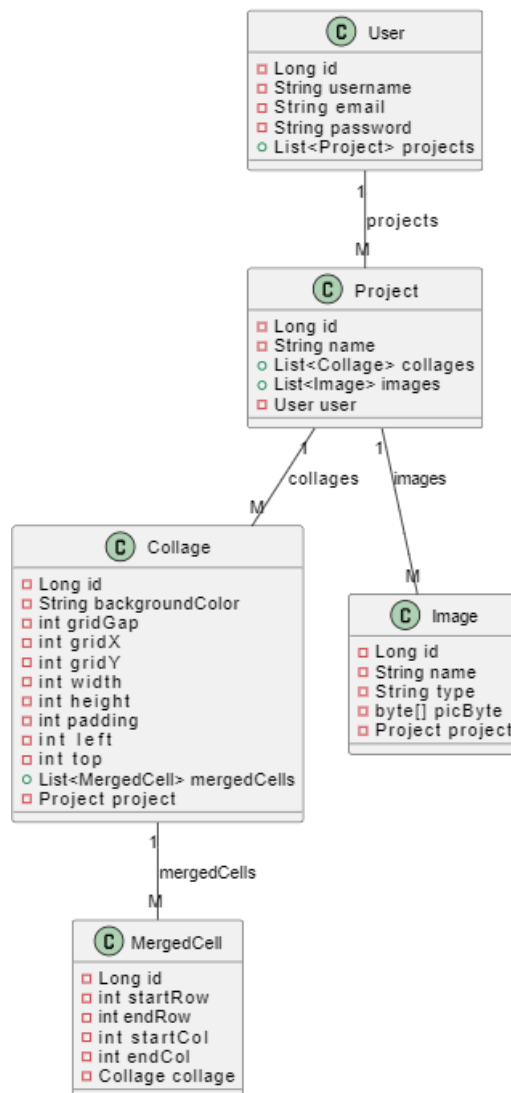


Рис. 2 - Діаграма класів

Крок 5. Представлення схеми таблиць бази даних.

Опис зв'язків між таблицями.

User і Project:

Тип зв'язку: Один-до-багатьох (One-to-Many)

Опис: Один *User* може мати кілька *Project*, але кожен *Project* належить тільки одному *User*.

Реалізація: У таблиці *Project* є зовнішній ключ `user_id`, який посилається на `id` у таблиці *User*.

Project і Collage:

Тип зв'язку: Один-до-багатьох (One-to-Many)

Опис: Один *Project* може містити кілька *Collage*, але кожен *Collage* належить тільки одному *Project*.

Реалізація: У таблиці *Collage* є зовнішній ключ `project_id`, який посилається на `id` у таблиці *Project*.

Collage і MergedCell:

Тип зв'язку: Один-до-багатьох (One-to-Many)

Опис: Один *Collage* може містити кілька *MergedCell*, але кожен *MergedCell* належить тільки одному *Collage*.

Реалізація: У таблиці *MergedCell* є зовнішній ключ `collage_id`, який посилається на `id` у таблиці *Collage*.

Project і Image:

Тип зв'язку: Один-до-багатьох (One-to-Many)

Опис: Один *Project* може мати кілька *Image*, але кожен *Image* належить тільки одному *Project*.

Реалізація: У таблиці *Image* є зовнішній ключ `project_id`, який посилається на `id` у таблиці *Project*.

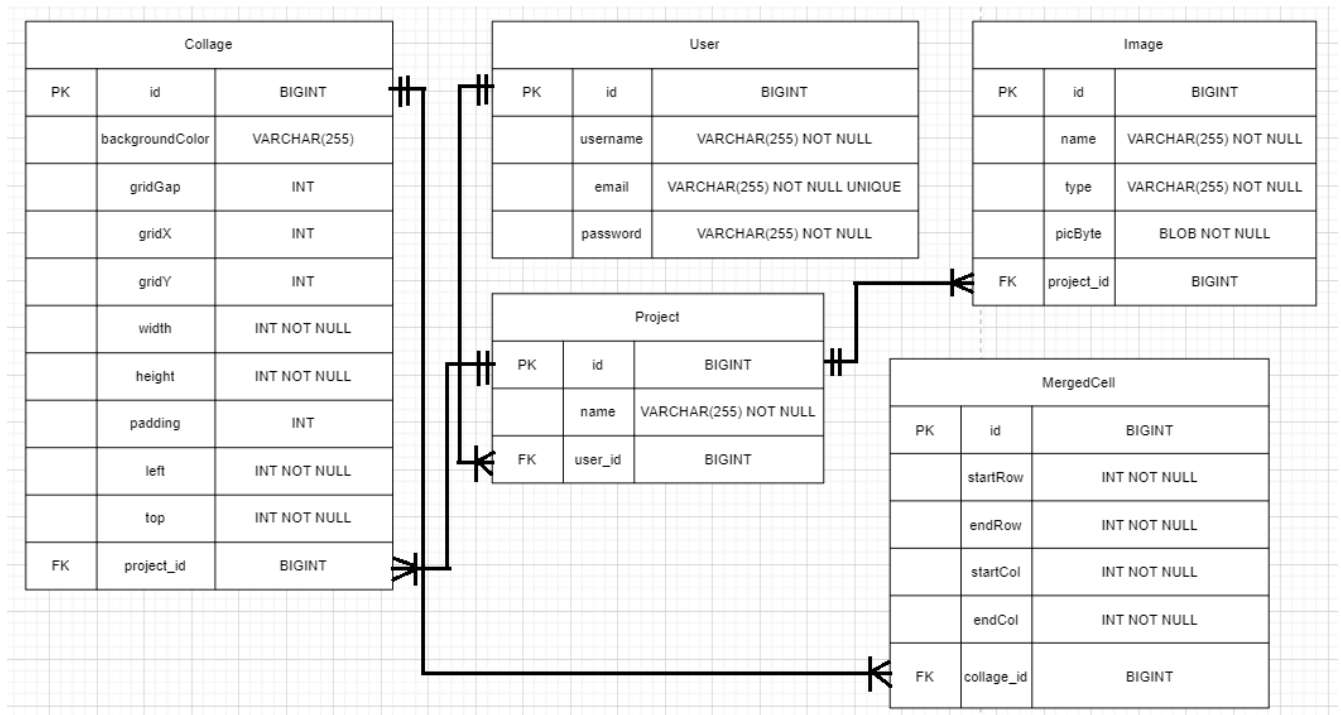


Рис. 3 - Схема таблиць бази даних

Висновок: у даній лабораторній роботі я проаналізував тему та на основі даних створив діаграму прецедентів та діаграму класів, спроектував схему бази даних та описав зв'язки між таблицями.