



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
Технології розроблення програмного забезпечення
ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ
ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ
КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ
Office communicator

Виконав
студент групи ІА – 22:
Ільніцький П А

Перевірив:

Київ 2024

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

Мета: Проаналізувати тему, намалювати схему прецеденту, діаграму класів, розробити основні класи і структуру бази

Хід роботи

Схема прецеденту, що відповідає обраній темі, зображена на рисунку 1.

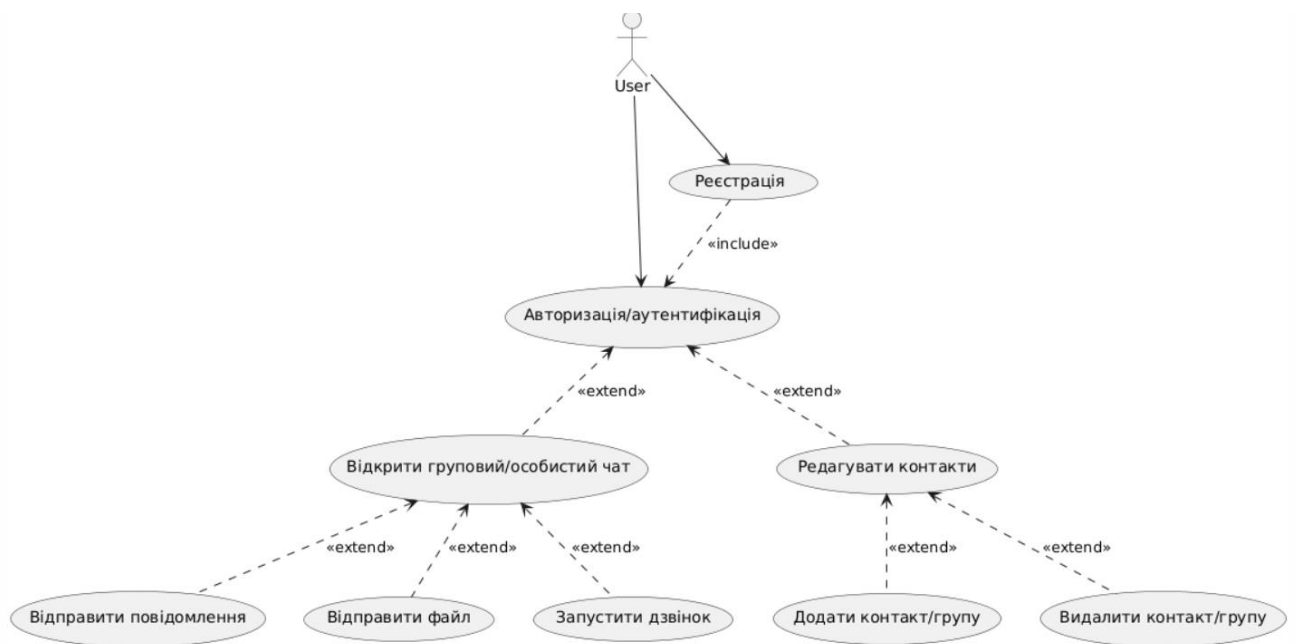


Рисунок 1 – Схема прецеденту

Діаграма прецедентів (Use Case Diagram) — це тип діаграми в UML (Unified Modeling Language), яка описує взаємодію акторів (користувачів або зовнішніх систем) із системою через її функціональність. Вона показує, що саме система робить з точки зору користувачів, не заглиблюючись у технічні деталі реалізації.

Основні елементи діаграми прецедентів:

1. Актори (Actors): Зовнішні сутності, що взаємодіють із системою. Актор може бути користувачем, пристроєм або іншою системою.

2. Прецеденти (Use Cases): Дії або функції, які система надає акторам. Вони представляють собою завдання, які виконує система на вимогу актора.

3. Система (System): Представлена у вигляді прямокутника, який охоплює всі прецеденти. Актори завжди знаходяться поза межами системи.

Відносини в діаграмі прецедентів:

1. Association (Асоціація): Прямий зв'язок між актором і прецедентом, що показує, що актор ініціює або бере участь у виконанні прецеденту.

2. Include (Включення): Використовується, коли один прецедент завжди виконує інший. Це означає, що прецедент включає в себе частину функціональності іншого прецеденту.

3. Extend (Розширення): Використовується, коли один прецедент опціонально додає або змінює поведінку іншого. Ця дія виконується за певних умов.

Призначення діаграми: Діаграма прецедентів допомагає виявити вимоги до системи, визначити її функціональність і зрозуміти, як вона буде використовуватися користувачами або взаємодіяти з іншими системами.

Ось три сценарії використання для офісного комунікатора, які враховують основні функції, як у Skype: відео/аудіо зв'язок, обмін повідомленнями, додавання контактів і груп.

Оберемо 3 прецеденти і напишемо для них сценарії використання

Сценарій використання 1: Відправити текстове повідомлення

Передумови: Користувач авторизований та має доступ до чатів.

Післяумови: Повідомлення успішно доставлене отримувачу.

Актори: Користувач.

Опис: Цей сценарій описує процес відправки текстового повідомлення в особистий або груповий чат.

Основний хід подій:

1. Користувач обирає опцію "Відкрити чат" і вибирає контакт або групу.
2. Система відкриває вікно чату з обраним контактом або групою.
3. Користувач вводить текст повідомлення у вікно для написання повідомлень.
4. Користувач натискає "Відправити".
5. Система відправляє повідомлення обраному контакту чи групі.
6. Отримувач бачить повідомлення в своєму чаті в реальному часі.

Винятки:

- Якщо користувач втрачає зв'язок з мережею, система показує повідомлення про неможливість відправки і повторює спробу автоматично при відновленні зв'язку.

Примітки: Немає

Сценарій використання 2: Запустити відео дзвінок

Передумови: Користувач авторизований і має доступ до контактів.

Післяумови: Відео дзвінок завершено.

Актори: Користувач, Отримувач дзвінка.

Опис: Цей сценарій описує процес запуску відео дзвінка для обраного контакту.

Основний хід подій:

1. Користувач обирає контакт у списку контактів.
2. Користувач натискає на опцію "Запустити відео дзвінок".
3. Система відправляє запит на дзвінок отримувачу.
4. Отримувач отримує повідомлення про вхідний дзвінок.
5. Якщо отримувач приймає дзвінок, система встановлює відео з'єднання.
6. Користувач і отримувач можуть бачити та чути один одного під час дзвінка.
7. Після завершення дзвінка користувач або отримувач натискає "Завершити дзвінок".

Винятки:

- Якщо отримувач не відповідає, система показує користувачу повідомлення про неможливість встановити з'єднання.
- Якщо мережа переривається під час дзвінка, система спробує автоматично відновити з'єднання.

Примітки: Немає

Сценарій використання 3: Додати контакт у список

Передумови: Користувач авторизований у системі.

Післяумови: Новий контакт доданий у список контактів користувача.

Актори: Користувач.

Опис: Цей сценарій описує процес додавання нового контакту до списку контактів.

Основний хід подій:

1. Користувач відкриває меню "Контакти".
2. Користувач обирає опцію "Додати контакт".
3. Система показує поле для введення контактних даних (ім'я, ID, email тощо).
4. Користувач вводить інформацію про новий контакт і натискає "Зберегти".
5. Система перевіряє наявність контакту в мережі.
6. Якщо контакт знайдено, система додає його до списку контактів користувача.
7. Контакт відображається у списку, доступному для чату або дзвінків.

Винятки:

- Якщо контакт з такими даними не знайдено, система повідомляє користувача про помилку.
- Якщо контакт уже є в списку, система повідомляє про це.

Примітки: Немає

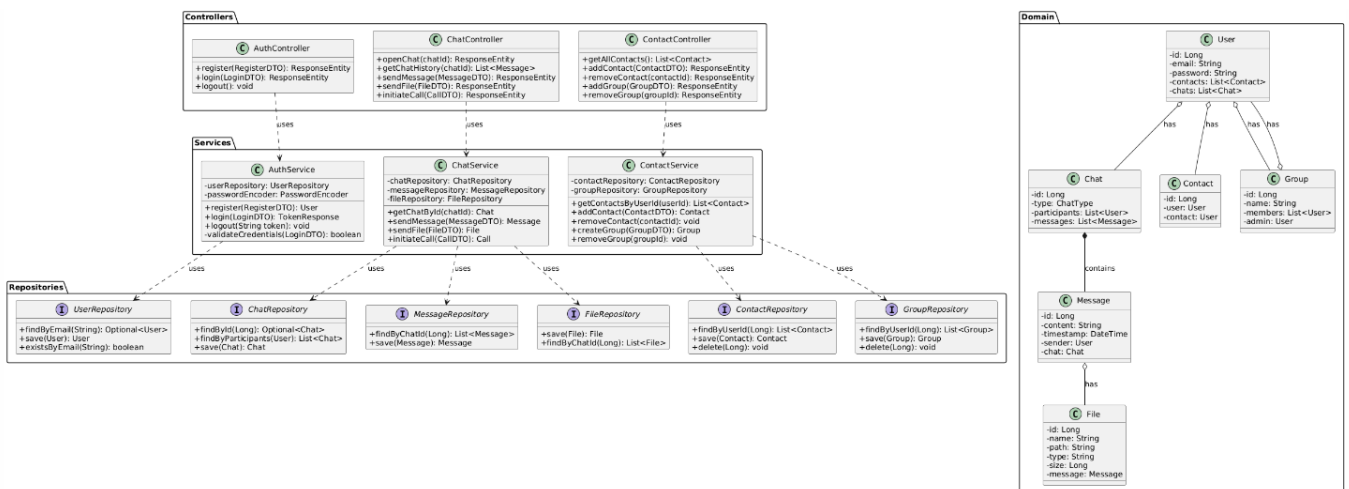


Рисунок 2 – Діаграма класів

Діаграма класів Діаграма класів — це один із типів діаграм, використовуваних в об'єктно орієнтованому програмуванні для візуалізації структури системи. Вона показує класи, їх атрибути, методи та відношення між класами.

Основні елементи:

- Класи: Основні елементи діаграми, які представляють об'єкти в системі. Клас містить атрибути (дані) та методи (функції).
- Атрибути: Властивості класу, які визначають його стан.
- Методи: Функції або дії, які може виконувати клас.
- Відношення: Зв'язки між класами, такі як асоціація, агрегація, композита та спадкування, що описують, як класи взаємодіють між собою.

Важливість: Діаграми класів допомагають зрозуміти структуру системи, її компоненти та їх взаємозв'язки. Вони є важливими для проектування програмного забезпечення, оскільки дозволяють командам розробників обговорювати та планувати реалізацію архітектури системи.

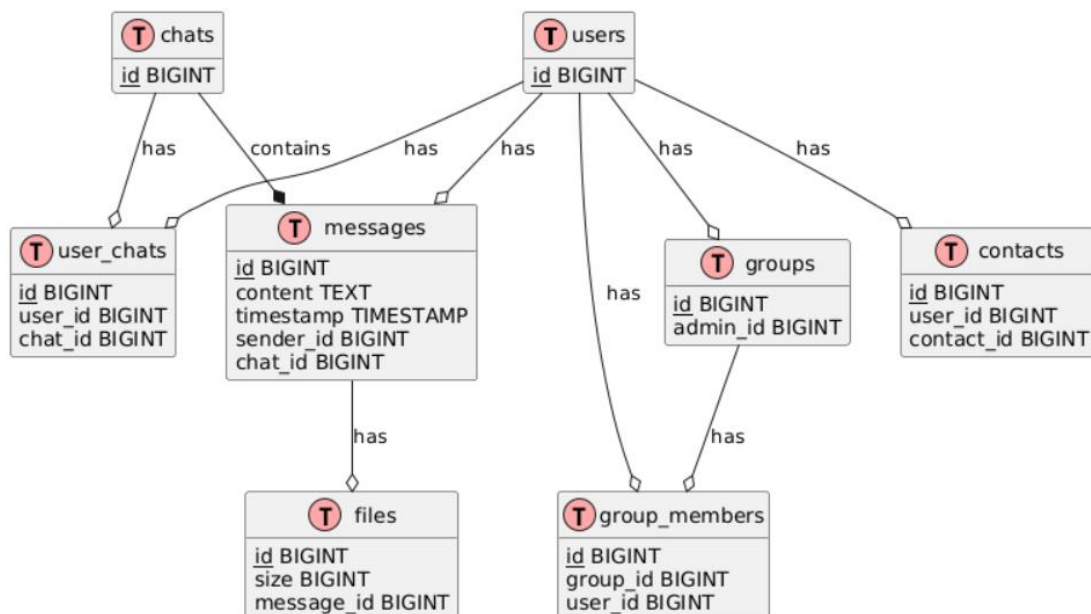


Рисунок 3. Структура бази даних

Діаграма бази даних (ER-діаграма або діаграма сутностей-зв'язків) є графічним поданням структури бази даних, яка відображає сутності (таблиці), їх атрибути (поля) та зв'язки між ними. Це допомагає розробникам та аналітикам системи зрозуміти, як дані організовані, як вони взаємодіють і які відносини між різними частинами даних.

Основні елементи діаграми бази даних:

1. Сутності (Entities): Це таблиці бази даних, що представляють об'єкти або поняття в системі (наприклад, Користувач, Файл, Завантаження).
2. Атрибути (Attributes): Це колонки в таблицях, які містять дані про сутності (наприклад, ім'я користувача, розмір файлу).
3. Зв'язки (Relationships): Це відносини між сутностями, які визначають, як дані взаємодіють одна з одною (наприклад, один Користувач може мати багато Завантажень).

Висновки: я проаналізував тему, намалював схему прецеденту, діаграму класів, розробив основні класи.