



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
Download manager

Виконала

студентка групи ІА–22:

Степанюк-Б. Анастасія

Перевірів:

Мягкий Михайло Юрійович

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

..26 Download manager (iterator, command, observer, template method, composite, p2p)

Інструмент для скачування файлів з інтернету по протоколах http або https з можливістю продовження завантаження в зупиненому місці, розподілу швидкостей активним завантаженням, ведення статистики завантажень, інтеграції в основні браузери (firefox, opera, internet explorer, chrome).

Побудуємо схему прецедентів. Схема зображена на рисунку 1.

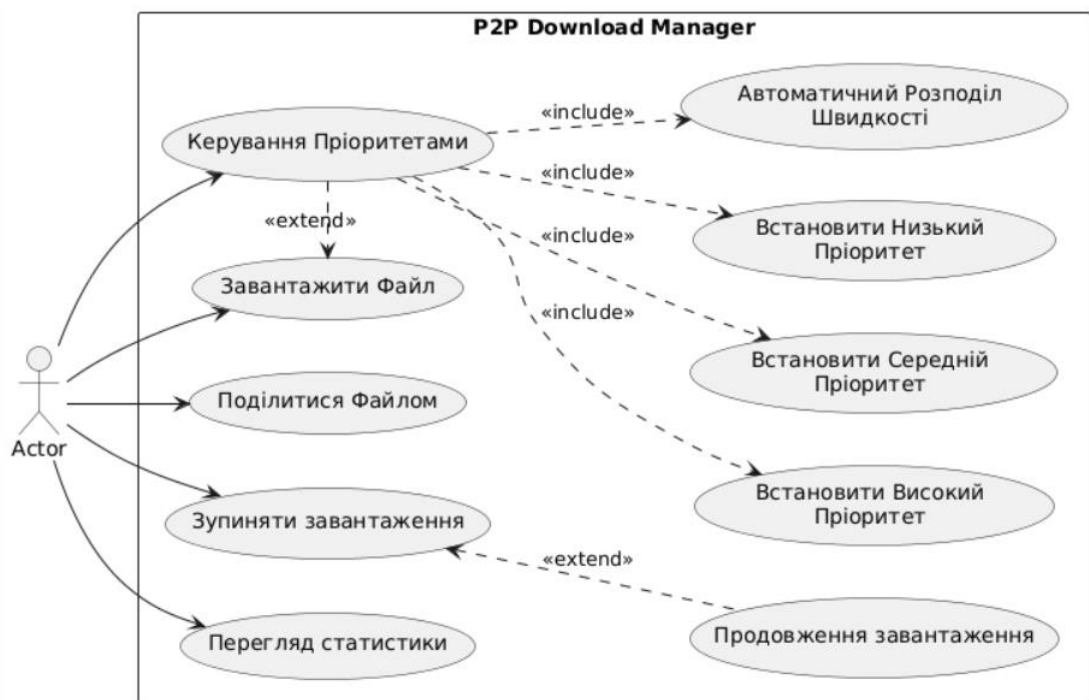


Рисунок 1. – Схема прецеденту

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.

Оберемо 3 прецеденти і напишемо для них сценарії використання

Сценарій використання 1: Завантажити Файл через P2P

Передумови: Користувач підключений до P2P-мережі.

Післяумови: Користувач успішно завантажив файл.

Актори: Користувач.

Опис: Цей сценарій описує процес завантаження файлу через P2P-мережу.

Основний хід подій:

1. Користувач обирає опцію "Завантажити файл".
2. Система відображає список доступних файлів для завантаження в мережі.
3. Користувач обирає файл для завантаження.
4. Система починає завантаження файлу, отримуючи частини файлу від інших учасників мережі.
5. Система показує користувачу прогрес завантаження та швидкість.
6. Після завершення завантаження файл зберігається на комп'ютері користувача.

Винятки: Якщо файл недоступний, система повідомляє користувача про помилку.

Примітки: P2P-протокол дозволяє завантаження частин файлу одночасно з кількох джерел, що покращує швидкість.

Сценарій використання 2: Керування Пріоритетами Завантажень

Передумови: Користувач активував кілька завантажень одночасно.

Післяумови: Пріоритет завантаження файлів встановлений.

Актори: Користувач.

Опис: Цей сценарій описує процес встановлення пріоритетів для активних завантажень файлів.

Основний хід подій:

1. Користувач обирає опцію "Керування Пріоритетами".
2. Система відображає список активних завантажень.
3. Користувач вибирає файл і обирає пріоритет (високий, середній, низький).
4. Система застосовує обраний пріоритет до обраного завантаження, змінюючи швидкість або послідовність завантажень.

5. Користувач може активувати "Автоматичний розподіл швидкості", і система оптимізує пріоритети між завантаженнями автоматично.

Винятки: Якщо користувач не має активних завантажень, система повідомляє про це.

Примітки: Пріоритети дозволяють ефективно керувати ресурсами мережі та збалансовувати завантаження файлів.

Сценарій використання 3: Поділитися Файлом через P2P

Передумови: Користувач має файли, які він готовий зробити доступними в мережі.

Післяумови: Файл доступний для інших користувачів мережі.

Актори: Користувач.

Опис: Цей сценарій описує процес публікації файлу в P2P-мережі для доступу іншим користувачам.

Основний хід подій:

1. Користувач обирає опцію "Поділитися файлом".
2. Система відкриває вікно для вибору файлу, який користувач бажає зробити доступним.
3. Користувач обирає файл і натискає "Поділитися".
4. Система додає файл до загального каталогу P2P-мережі.
5. Інші користувачі можуть завантажувати цей файл через мережу.

Винятки: Якщо користувач спробує поділитися файлом, доступ до якого обмежений або заборонений, система видасть повідомлення про помилку.

Примітки: P2P-мережа дозволяє децентралізоване розповсюдження файлів, де кожен користувач є рівноправним учасником процесу.

Побудуємо структуру бази даних (рисунок 2)

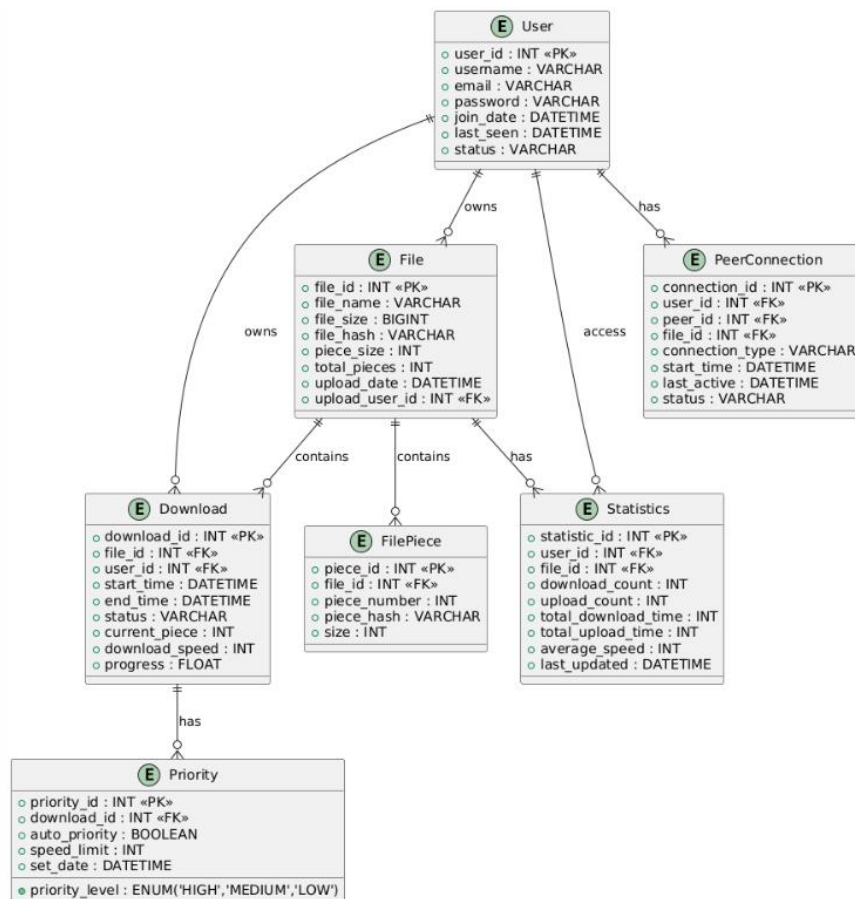


Рисунок 2. Структура бази даних

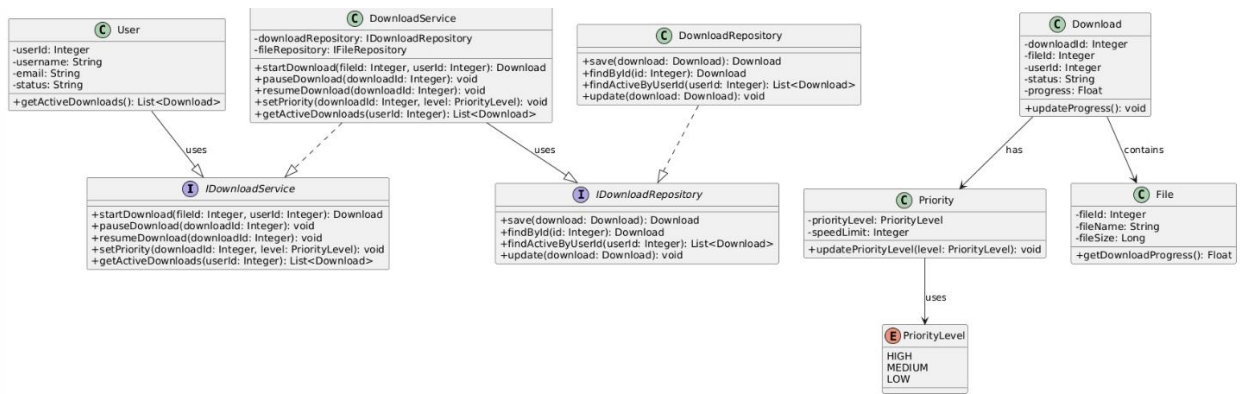
Теорія для діаграми бази даних

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).

Побудуємо діаграму 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.

Висновки: під час виконання лабораторної роботи я проаналізувала тему, намалював схему прецедентів, діаграму класів, розробив основні класи.