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

з дисципліни «Технології розроблення програмного забезпечення»

Тема: «Основи проектування»

Виконав:
студент групи ІА-з31
Курбатов Кіріл Андрійович

Перевірив:
Амонс Олександр Анатолійович

Київ 2025

Мета: Обрати зручну систему побудови UML-діаграм та навчитися будувати
діаграми варіантів використання для системи що проєктується, розробляти
сценарії варіантів використання та будувати діаграми класів предметної області.

Завдання

• Ознайомитись з короткими теоретичними відомостями.
• Проаналізувати тему та спроєктувати діаграму варіантів використання відповідно до обраної теми лабораторного циклу.
• Спроєктувати діаграму класів предметної області.
• Вибрати 3 варіанти використання та написати за ними сценарії використання.
• На основі спроєктованої діаграми класів предметної області розробити основні класи та структуру бази даних системи. Класи даних повинні реалізувати шаблон Repository для взаємодії з базою даних.
• Нарисувати діаграму класів для реалізованої частини системи.
• Підготувати звіт щодо виконання лабораторної роботи. Поданий звіт повинен
містити: діаграму варіантів використання відповідно, діаграму класів системи,
вихідні коди класів системи, а також зображення структури бази даних.

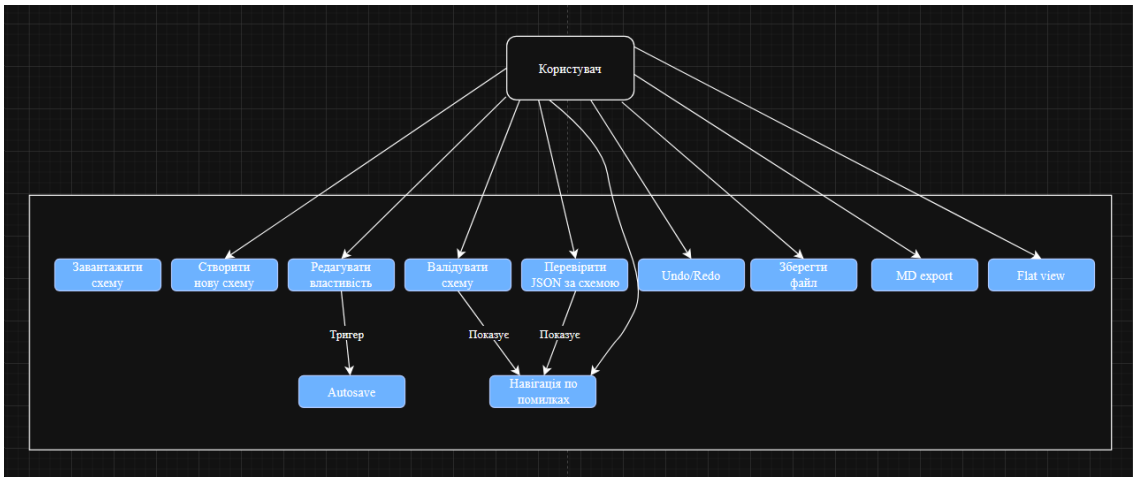

Рис. 1 — Use Case діаграма системи

Сценарії варіантів використання

Завантаження JSON Schema

Параметри: Назва — Завантаження JSON Schema. Актор — Користувач. Мета — Відкрити файл JSON Schema для редагування та валідації.

Передумови: Застосунок запущено. Файл JSON Schema існує на диску.

Основний сценарій:

Користувач натискає File → Open або використовує комбінацію клавіш Ctrl+O.

Система відкриває діалогове вікно вибору файлу з активованим фільтром для файлів типу *.json.

Користувач обирає потрібний файл JSON Schema.

Система завантажує вміст вибраного файлу.

Система відображає завантажений JSON-код у вбудованому редакторі з підсвічуванням синтаксису.

Система аналізує (парсить) схему та автоматично заповнює список властивостей (Properties) на основі її вмісту.

Система виконує попередню валідацію схеми та відображає її результат у статус-барі.

Система запам'ятовує шлях до відкритого файлу для можливості швидкого збереження змін у майбутньому.

Альтернативний сценарій А (некоректний JSON):

На етапі завантаження (крок 4) система виявляє синтаксичну помилку у файлі.

Система все одно відображає вміст файлу в редакторі.

Система показує деталі помилки в спеціальній панелі "Validation Results".

Список властивостей (Properties) залишається порожнім, оскільки схема не може бути коректно проаналізована.

Альтернативний сценарій В (файл не знайдено):

На етапі відкриття (крок 4) система не може отримати доступ або знайти вказаний файл.

Система показує користувачу повідомлення про помилку з відповідним описом.

Стан застосунку (відкритий документ, список властивостей тощо) залишається незмінним.

Пост-умови:

Вміст файлу відображається в редакторі.

Список властивостей заповнено (якщо схема синтаксично валідна та коректно розпарсена).

Шлях до відкритого файлу збережено в пам'яті застосунку.

Редагування властивості схеми

Параметри: Назва — Редагування властивості схеми. Актор — Користувач. Мета — Змінити метадані властивості (type, format, description, examples).

Передумови: Завантажено валідну JSON Schema. Список Properties містить хоча б одну властивість.

Основний сценарій:

Користувач обирає конкретну властивість у списку Properties.

Система відображає панель "Edit Property", заповнену поточними значеннями обраної властивості.

Користувач вносить зміни в одне або кілька полей (наприклад, Type, Format, Description, Examples).

Користувач підтверджує зміни, натискаючи кнопку "Apply".

Система створює об'єкт команди EditPropertyCommand, який інкапсулює старе та нове значення властивості.

Система виконує цю команду через центральний виконавець команд (CommandInvoker).

Система автоматично оновлює JSON-код у текстовому редакторі відповідно до змін.

Система сповіщає всі підписані модулі (за допомогою шаблону Observer Pattern) про оновлення схеми.

Модуль AutoSaveObserver, отримавши сповіщення, запускає таймер з debounce-логікою для відкладеного автоматичного збереження.

Альтернативний сценарій (відміна змін):

На кроці 4 користувач вирішує скасувати редагування і натискає кнопку

"Cancel".

Система відновлює початкові значення у всіх полях панелі редагування.

Жодні зміни до схеми не застосовуються.

Пост-умови:

JSON Schema оновлено відповідно до внесених змін.

Виконана операція збережена в історії команд, що дозволяє її скасування (Undo).

Документ позначено як такий, що має незбережені зміни.

Перевірка JSON документа за схемою

Параметри: Назва — Перевірка JSON документа за схемою. Актор — Користувач. Мета — Перевірити відповідність JSON-даних поточній схемі.

Передумови: Завантажено валідну JSON Schema.

Основний сценарій:

Користувач натискає кнопку "Check Data" в інтерфейсі.

Система відкриває діалогове вікно вибору JSON-файлу, що містить дані для перевірки.

Користувач обирає потрібний файл з даними.

Система завантажує та читає вміст вибраного файлу даних.

Система створює контекст валідації (ValidationContext), використовуючи стратегію JsonBySchemaValidationStrategy.

Система передає завантажені дані та поточну активну схему в двигун валідації відповідно до обраної стратегії.

Система відображає детальний результат перевірки (успіх або список помилок) у панелі "Validation Results".

У разі наявності помилок, система для кожної з них показує шлях (JSON Path) до невалідного елементу в даних.

Альтернативний сценарій (помилка читання файлу):

На етапі завантаження (крок 4) система не може прочитати або обробити

файл даних (наприклад, через пошкодження або відсутність прав доступу).

Система показує користувачу повідомлення про помилку читання файлу. Процес валідації даних не ініціюється та не виконується.

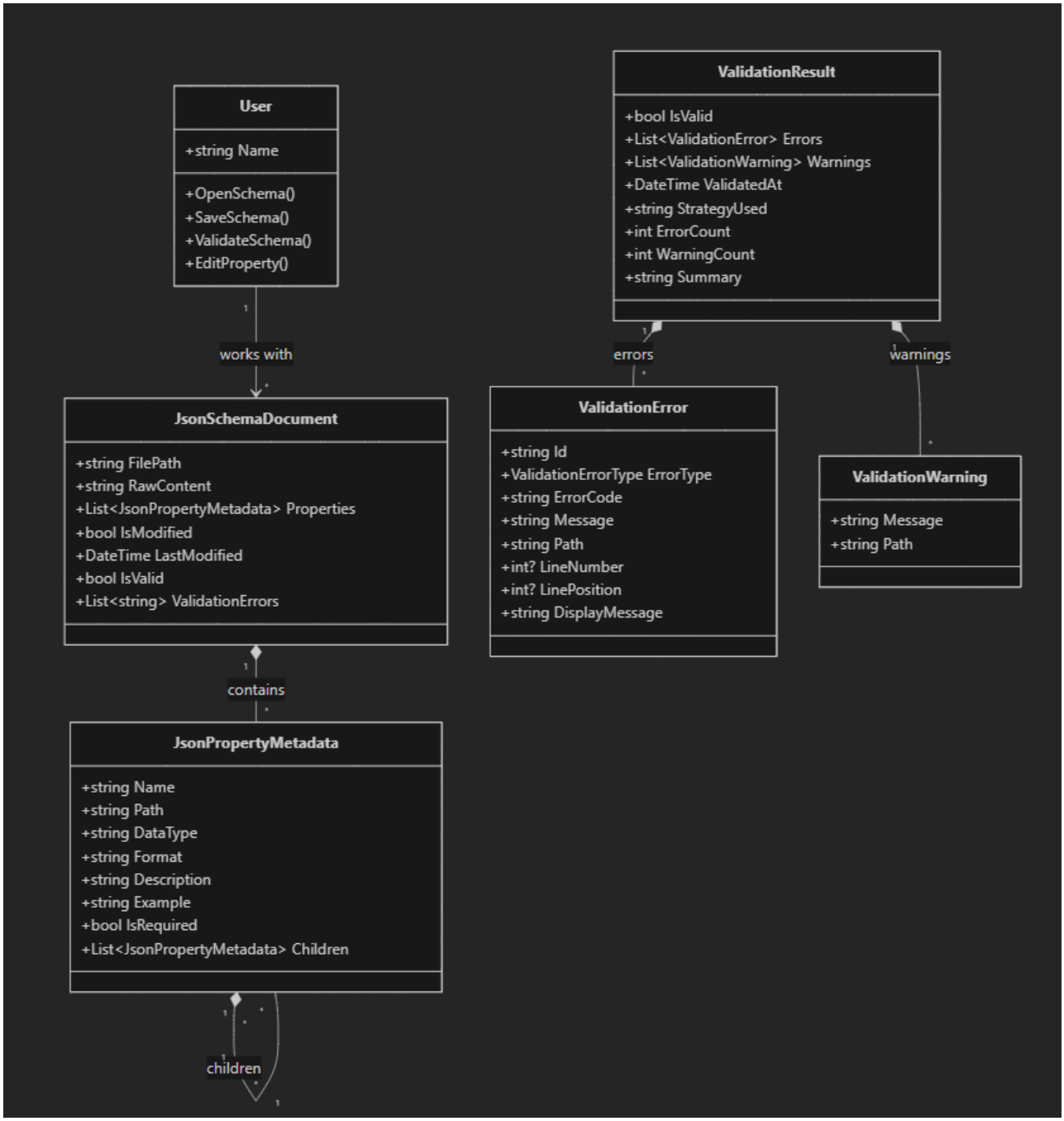

Рис. 2 — Діаграма класів предметної області