



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 №3

з дисципліни «Технології розроблення програмного забезпечення»

Тема: «Основи проектування розгортання»

Виконав:

студент групи IA-331

Курбатов Кіріл Андрійович

Перевірив:

Амонс Олександр Анатолійович

Мета: Навчитися проєктувати діаграмами розгортання та компонентів для системи що проєктується, а також розробляти діаграмами взаємодії, а саме діаграмами послідовностей, на основі сценаріїв зроблених в попередній лабораторній роботі.

Завдання

- Ознайомитись з короткими теоретичними відомостями.
- Проаналізувати діаграми створені в попередній лабораторній роботі а також тему системи та спроєктувати діаграму розгортання використання відповідно до обраної теми лабораторного циклу.
- Розробити діаграму компонентів для проєктованої системи.
- Розробити діаграму розгортання для проєктованої системи.
- Розробити як мінімум дві діаграми послідовностей для сценаріїв прописаних в попередній лабораторній роботі.
- На основі спроєктованих діаграм розгортання та компонентів доопрацювати програмну частину системи. Реалізація системи, додатково до попередньої реалізації, повинна містити як мінімум дві візуальні форми. В системі вже повинен бути повністю реалізована архітектура (повний цикл роботи з даними від вводу на формі до збереження їх в БД і подальшій виборці з БД та відображенням на UI).
- Підготувати звіт щодо виконання лабораторної роботи. Поданий звіт повинен містити: діаграму розгортання з описом, діаграму компонентів системи з описом, діаграми послідовностей, а також вихідний код системи, який було додано в цій лабораторній роботі.

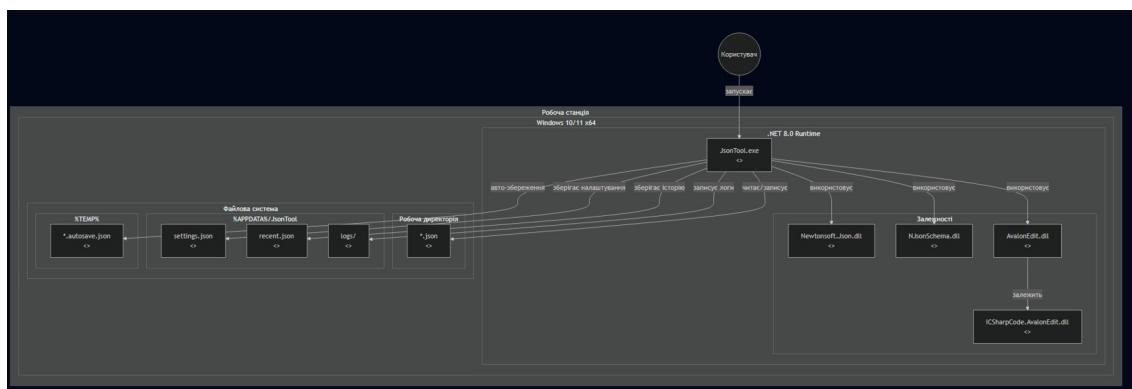


Рис. 1 — Діаграма розгортання системи JSON Schema Tool

Опис кроків:

1. Ініціація: Користувач натискає гарячу клавішу Ctrl+O або обирає пункт меню File → Open в головному вікні (MainWindow).
2. Виклик команди: Головне вікно через механізм data binding викликає команду відкриття файлу (OpenFileCommand).
3. Діалог вибору: Команда ініціює відкриття стандартного діалогового вікна вибору файлу (OpenFileDialog). У діалозі

активовано фільтр для відображення лише файлів з розширенням *.json.

4. Вибір файлу: Користувач знаходить та обирає потрібний файл JSON Schema в діалоговому вікні та підтверджує вибір.
5. Завантаження: Логіка завантаження делегується сервісу роботи зі схемами (JsonSchemaService). Цей сервіс звертається до файлової системи для читання вмісту обраного файлу.
6. Парсинг: Отримавши сирий JSON, сервіс парсить (розбирає) його, перетворюючи текст у внутрішню структуру даних, та витягає з неї список властивостей (properties) схеми.
7. Валідація: Сервіс виконує базову валідацію структури завантаженої схеми, перевіряючи її на відповідність стандарту JSON Schema.
8. Оновлення UI: Результат роботи сервісу (об'єкт схеми) передається до ViewModel (моделі подання). ViewModel оновлює свої властивості, такі як CurrentSchema та Properties. Ці зміни автоматично, через механізм прив'язки даних (data binding), викликають оновлення користувачького інтерфейсу (View): редактор JSON, список властивостей тощо.
9. Сповіщення: Для координації різних частин програми використовується шаблон Спостерігач (Observer). Суб'єкт документа (DocumentSubject) сповіщає всіх зареєстрованих спостерігачів (наприклад, модуль автозбереження, панель результатів) про те, що в роботі тепер знаходиться нова схема.

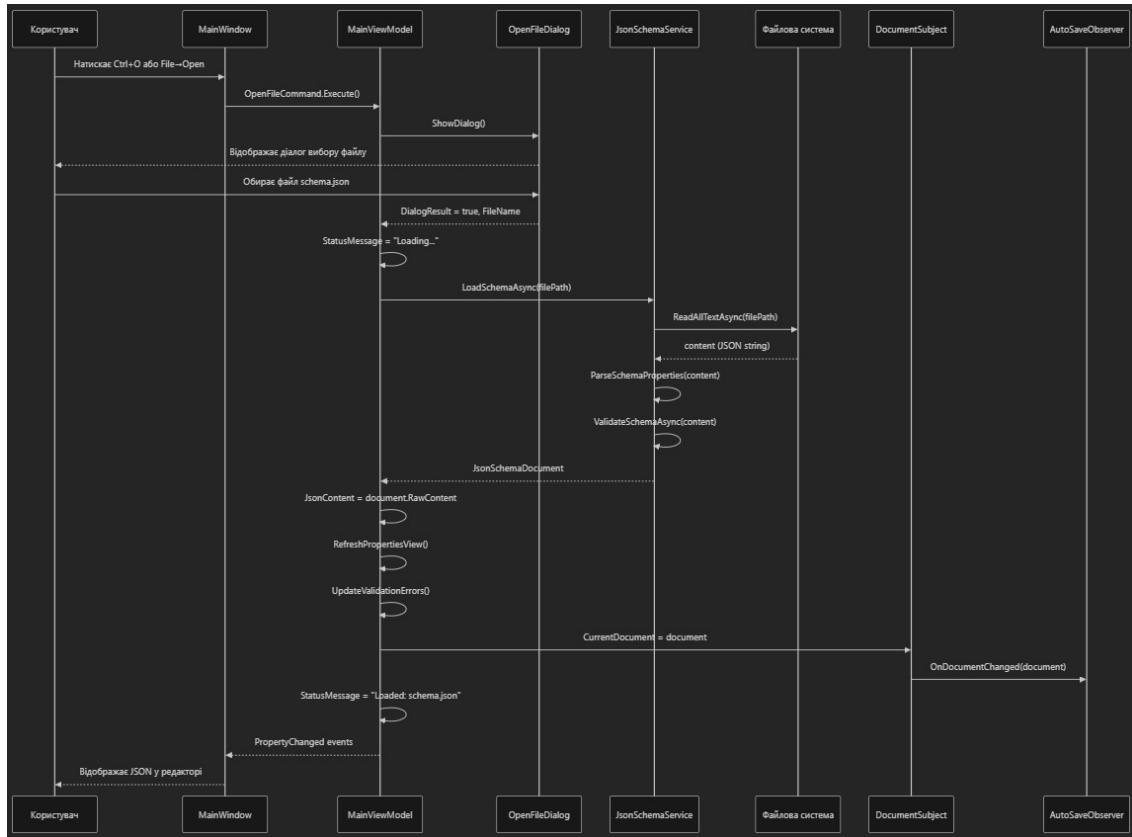


Рис. 2 — Діаграма послідовності завантаження JSON Schema

1. Вибір властивості: Користувач обирає конкретну властивість в панелі списку властивостей (Properties Panel).
2. Відображення панелі: Подія вибору призводить до того, що ViewModel встановлює значення властивості SelectedProperty. Це, в свою чергу, призводить до відображення панелі редагування властивостей (Edit Property Panel), поля якої заповнюються поточними значеннями обраної властивості (тип, формат, опис, приклади).
3. Редагування: Користувач змінює значення в одному або кількох полях панелі редагування.
4. Застосування: Для підтвердження змін користувач натискає кнопку "Apply".
5. Оновлення схеми: Команда редагування звертається до JsonSchemaService з проханням оновити внутрішню структуру JSON-схеми новими метаданими обраної властивості.
6. Сповіщення: Після успішного оновлення DocumentSubject сповіщає всіх спостерігачів про зміну в документі.
7. Debounce-затримка: Один з спостерігачів, AutoSaveObserver, отримавши сповіщення, запускає таймер з затримкою (наприклад, 3 секунди). Якщо протягом цих 3 секунд надійде нове сповіщення

про зміну, таймер перезапуститься.

8. Авто-збереження: Якщо за 3 секунди нових змін не сталося, таймер спрацьовує, і AutoSaveObserver ініціює процес автоматичного збереження поточного стану схеми у файл.

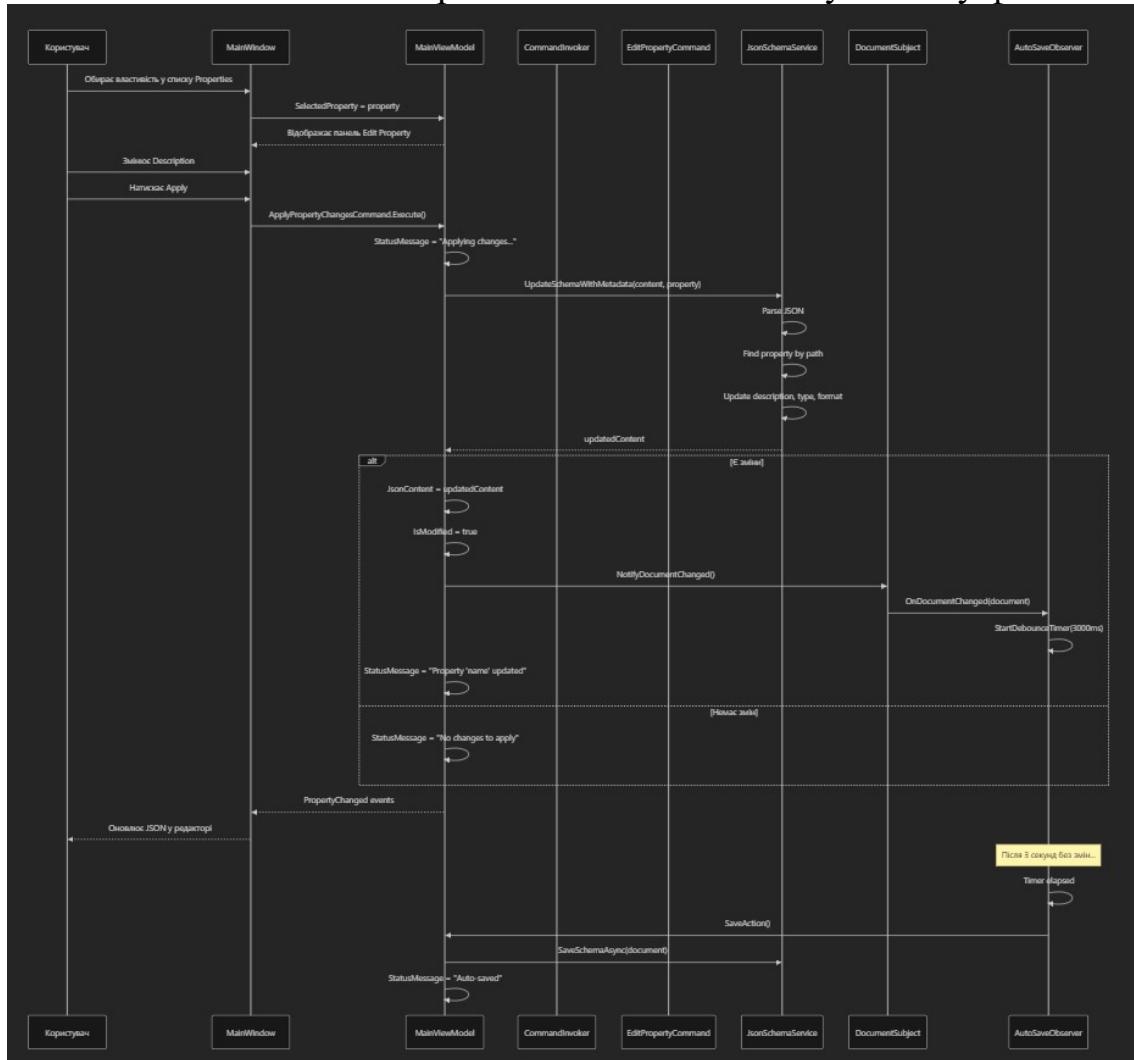


Рис. 3 — Діаграма послідовності редагування властивості схеми