

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 розробки програмного забезпечення - 3”

Тема: «Використання. Сценарії варіантів використання. Діаграми uml.
Діаграми класів. Концептуальна модель системи»

Виконала:

Студентка групи ІА-01:

Дронь О. В.

Перевірів:

вик. Колеснік В. М.

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

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.
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.

Виконання:

Тема: JSON Tool

Метою роботи є створення системи, яка дозволяє виконувати такі дії:

Display JSON schema with syntax highlight. Validate JSON schema and display errors. Create user friendly table\list box\other for read and update JSON schema properties metadata (description, example, data type, format, etc.). Auto save\restore when edit, maybe history. Can check JSON value by schema (Put schema and JSON = valid\invalid, display errors). Export schema as markdown table. JSON to "flat" view.

Для цього система повинна мати певний перелік властивостей та вирішувати такі задачі:

- Система повинна мати можливість створення файлу формату JSON
- Система повинна мати можливість читання файлів формату JSON
- Система повинна мати можливість редагування файлу формату JSON
- Система повинна мати можливість підсвічування синтаксису
- Система повинна мати можливість валідувати файл та виділяти помилки у файлі

- Система повинна мати можливість переглянути метадані та редагувати їх
- Система повинна мати можливість зберігати та відновлювати зміни у файлі
- Система повинна мати можливість перевіряти відповідність файлу до його схеми та висвітлювати помилки
- Система повинна мати можливість експортувати файл як markdown table
- Система повинна мати можливість перетворювати файл у «плоский» вигляд

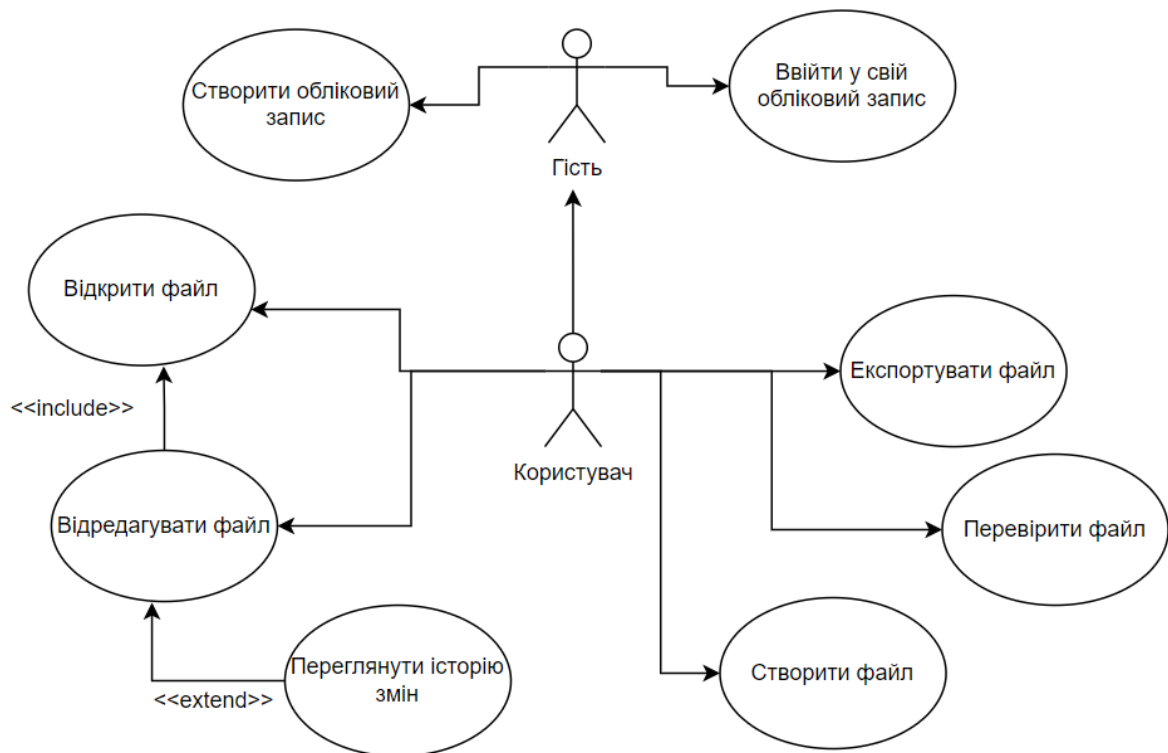
Система має відповідати наступним функціональним вимогам:

- Неаутефікований користувач має можливість лише зареєструватися або увійти до свого облікового запису
- Аутефікований користувач повинен мати можливість відкрити свої файли та створити нові
- Аутефікований користувач повинен мати можливість редагувати свої файли
- Аутефікований користувач повинен мати можливість зберігати свої файли

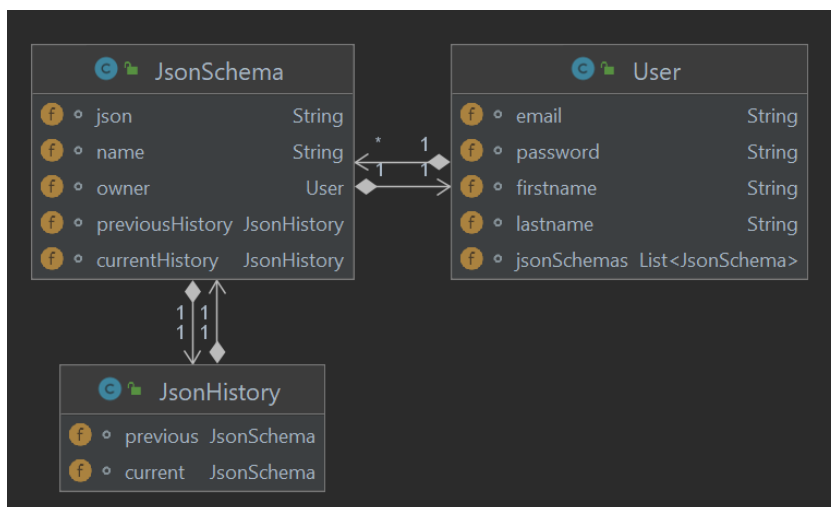
Система має відповідати наступним нефункціональним вимогам:

- система повинна мати відкриту архітектуру;
- інтерфейс користувача має бути зручним та інтуїтивно-зрозумілим;
- система повинна бути крос-платформною.

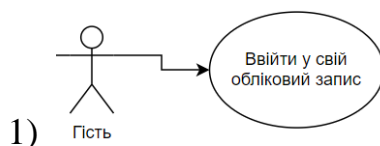
Діаграма прецендентів:



Діаграма класів:



Преценденти:



Передумови. Відсутні.

Постумови. У разі успішного виконання, гість входить в систему. У протилежному випадку стан системи не змінюється.

Взаємодіючі сторони. Гість, сервіс.

Короткий опис. Цей варіант використання описує вхід гостя до системи JSON tool.

Основний потік подій:

Цей варіант використання починає виконуватися, коли гість хоче увійти до системи JSON tool.

1. Система запитує пошту користувача та пароль.
2. Гість вводить пошту та пароль.
3. Система перевіряє поштову адресу та пароль, після чого відкривається доступ до системи. Якщо пошта та/або пароль неправильні, Виняток №1.

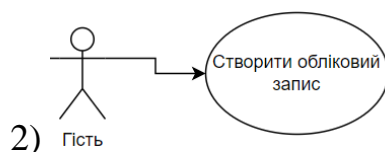
Винятки

Виняток №1

Неправильна пошта/пароль. Якщо під час виконання Основного потоку виявиться, що гість ввів неправильну пошту та/або пароль, система виводить повідомлення про помилку. Гість може повернутися до початку Основного потоку або відмовитись від входу в систему, при цьому виконання варіанту використання завершується.

Примітки

Відсутні.



Передумови. Відсутні.

Постумови. У разі успішного виконання, створюється новий користувач та гість входить в систему. У протилежному випадку стан системи не змінюється.

Взаємодіючі сторони. Гість, сервіс.

Короткий опис. Цей варіант використання описує вхід гостя до системи вперше та створення облікового запису.

Основний потік подій:

Цей варіант використання починає виконуватися, коли гість хоче вперше увійти у систему.

1. Система запитує пошту користувача та пароль або пропонує зареєструватись.
2. Гість обирає зареєструватись та вводить необхідні поля.
3. Система перевіряє введені поля на валідність, після чого, в разі вдалої реєстрації, відкривається доступ до системи. Якщо заповнені гостем поля не валідні, Виняток №1. Якщо введена пошта неунікальна, Виняток №2.

Винятки

Виняток №1

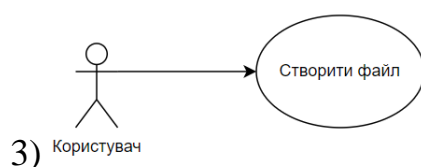
Невалідні поля. Якщо під час виконання Основного потоку виявиться, що гість ввів некоректне значення, система виводить повідомлення про помилку. Гість може повернутися до початку Основного потоку або відмовитись від входу в систему, при цьому виконання варіанта використання завершується.

Виняток №2

Неунікальне ім'я нового користувача. Якщо під час реєстрації, гість ввів пошту, яка вже є в системі, то система виводить повідомлення про помилку. Гість може увійти в свій вже існуючий обліковий запис або відмовитись від входу в систему, при цьому виконання варіанта використання завершується.

Примітки

Відсутні.



Передумови. Користувач увійшов у систему.

Постумови. У разі успішного виконання, створюється новий файл. У протилежному випадку стан системи не змінюється.

Взаємодіючі сторони. Гість, сервіс.

Короткий опис. Цей варіант використання описує створення користувачем нового файлу.

Основний потік подій:

Цей варіант використання починає виконуватися, коли користувач хоче

створити новий файл.

1. Система запитує назву нового файлу.
2. Користувач вносить назву нового файлу.
3. Система перевіряє введені дані на унікальність, після чого, в разі вдалої перевірки, створюється новий файл. Якщо заповнені гостем поля неунікальні, Виняток №1. Якщо введена назва файлу неунікальна

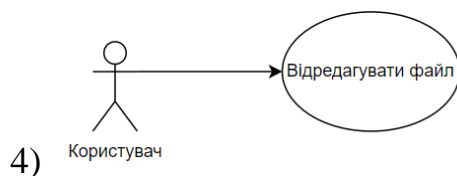
Винятки

Виняток №1

Неунікальне поле. Якщо під час виконання Основного потоку виявиться, що гість ввів неунікальне значення, система виводить повідомлення про помилку. Гість може повернутися до початку Основного потоку або відмовитись від створення нового файлу, при цьому виконання варіанта використання завершується.

Примітки

Відсутні.



Передумови. Користувач увійшов у систему. Користувач відкрив файл.

Постумови. У разі успішного виконання, зміни файлу зберігаються. У протилежному випадку стан системи не змінюється.

Взаємодіючі сторони. Гість, сервіс.

Короткий опис. Цей варіант використання описує зберігання змін у файлі, відредагованого користувачем.

Основний потік подій:

Цей варіант використання починає виконуватися, коли користувач хоче відредагувати файл та зберегти його зміни.

1. Система запитує змінений текст.
2. Користувач вносить зміни.

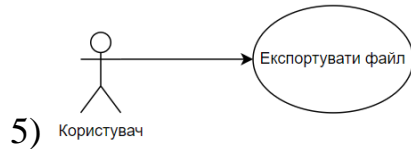
3. Система зберігає зміни та вносить попередній варіант в історію змін.

Винятки

Відсутні.

Примітки

Відсутні.



Передумови. Користувач увійшов у систему.

Постумови. У разі успішного виконання, файл експортується у певному форматі/вигляді. У протилежному випадку стан системи не змінюється.

Взаємодіючі сторони. Гість, сервіс.

Короткий опис. Цей варіант використання описує експортування файлу користувача.

Основний потік подій:

Цей варіант використання починає виконуватися, коли користувач хоче експортувати свій файл.

1. Система запитує який файл потрібно експортувати та вигляд експортування.
2. Користувач вносить дані.
3. Система дає можливість експортувати цей файл.

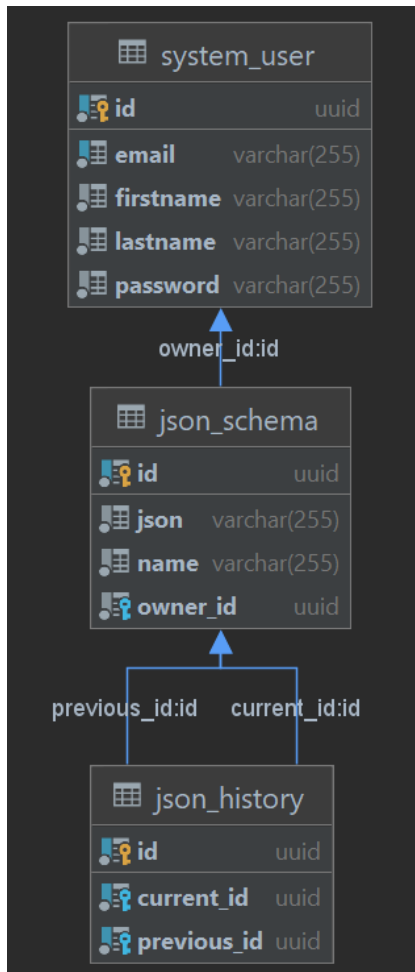
Винятки

Відсутні.

Примітки

Відсутні.

Модель БД:



Висновок: Під час виконання лабораторної роботи було проаналізовано обрану тему JSON tool, відповідно побудовано діаграму варіантів використання та сценарії варіантів використання, діаграму класів. Та спроектовано базу даних.