



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

Технології розроблення програмного забезпечення

Лабораторна робота №2

Виконала:
Студентка групи ІА-22
Патрик Вікторія

Київ 2024

Діаграма варіантів використання. Сценарії варіантів використання. Діаграми UML. Діаграми класів.

Концептуальна модель системи.

Мета: проаналізувати тему та намалювати схему прецеденту, діаграму класів, розробити основні класи і структуру системи баз даних.

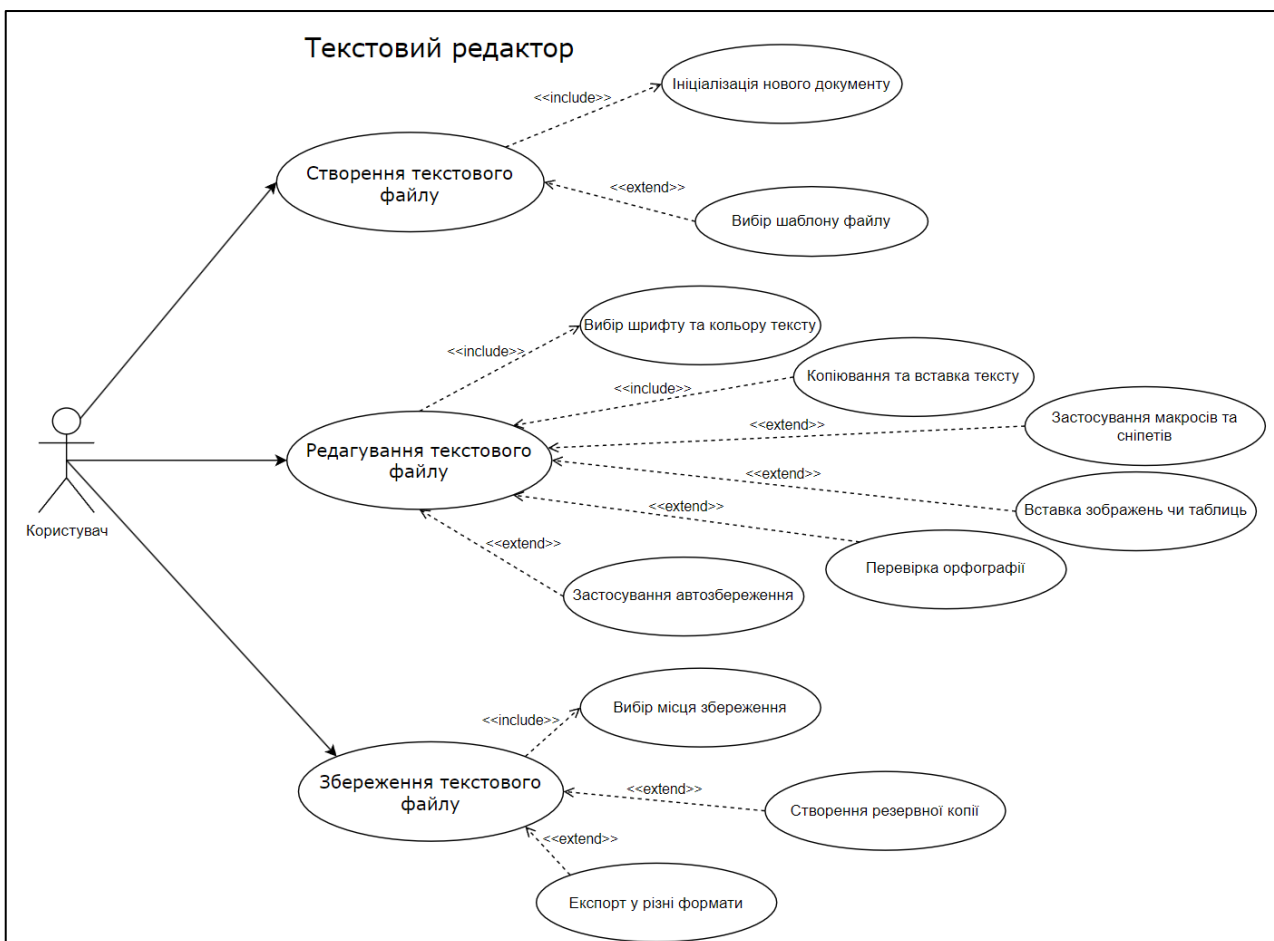
Хід роботи:

Тема:

..3 Текстовий редактор (strategy, command, observer, template method, flyweight, SOA)

Текстовий редактор повинен вміти розпізнавати текстові файли в будь-якій кодуванні, мати розширені функції редагування: макроси, сніппети, підказки, закладки, перехід на рядок / сторінку, підсвічування синтаксису (для однієї мови програмування або розмітки на розсуд студента).

Схема прецеденту, що відповідає темі «Текстовий редактор»:



Прецеденти та сценарії їх використання:

Прецедент: Створити текстовий файл

Актор: Користувач

Опис: Користувач створює новий текстовий файл для редагування та збереження.

Основний сценарій:

1. Користувач відкриває текстовий редактор.
2. Користувач натискає кнопку "Створити новий файл".
3. Програма ініціалізує новий порожній документ.
4. Користувач вводить текст у новий файл.
5. Користувач може змінювати шрифт та колір тексту.
6. Користувач натискає кнопку "Зберегти" і обирає місце на пристрої для збереження файлу.

Альтернативний сценарій:

- Якщо користувач не обирає місце збереження або залишає поле пустим, система відображає повідомлення про помилку і запитує повторне введення місця збереження.

Прецедент: Редагувати текстовий файл

Актор: Користувач

Опис: Користувач відкриває існуючий текстовий файл для редагування в текстовому редакторі.

Основний сценарій:

1. Користувач відкриває текстовий редактор.
2. Користувач натискає кнопку "Відкрити файл" і вибирає файл з пристрою.
3. Програма завантажує файл і відображає його в редакторі.
4. Користувач редагує текст: додає або видаляє символи, змінює форматування.
5. Користувач може використовувати функції перевірки орфограми, копіювання та вставки тексту.
6. Користувач натискає кнопку "Зберегти зміни".

Альтернативний сценарій:

- Якщо файл, який користувач намагається відкрити, не підтримується, система відображає повідомлення про помилку.
- Якщо у користувача немає доступу до файлу, система повідомляє про це і відмовляє у збереженні змін.

Прецедент: Перевірка орфографії

Актор: Користувач

Опис: Користувач використовує функцію перевірки орфографії для виправлення помилок у тексті.

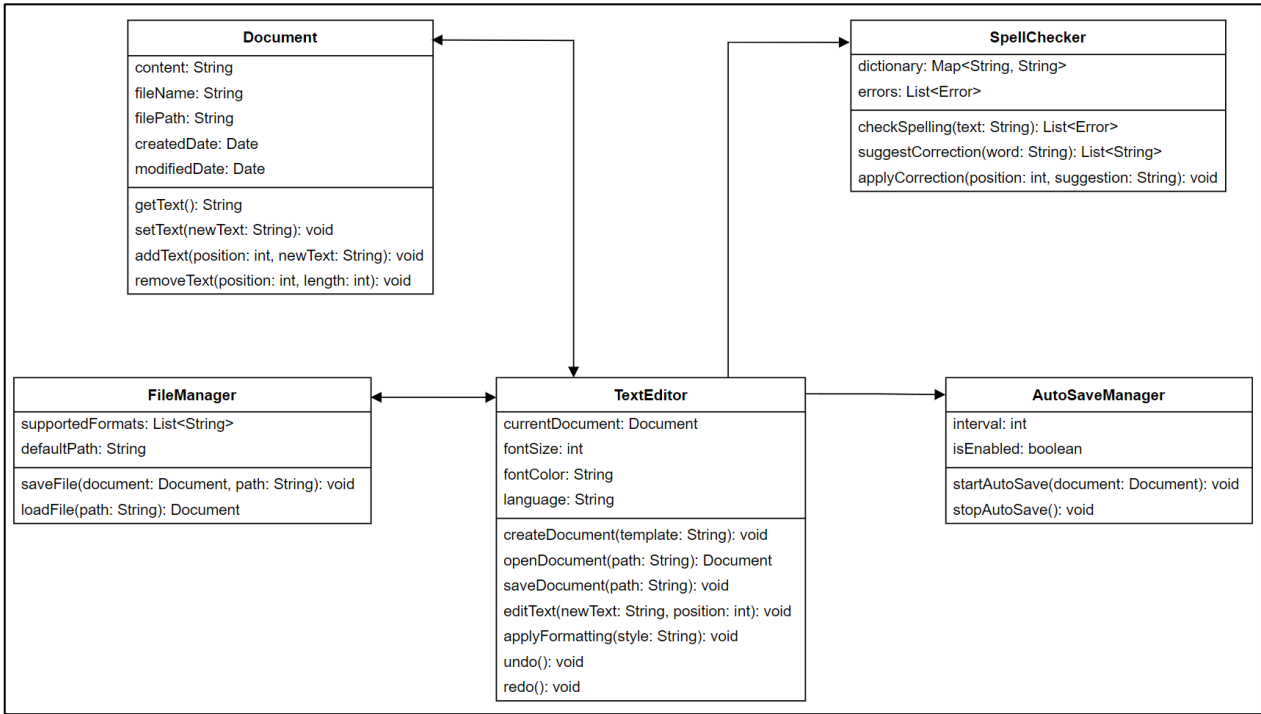
Основний сценарій:

- 1. Користувач відкриває текстовий редактор і завантажує файл або створює новий текстовий документ.
- 2. Користувач натискає кнопку "Перевірка орфографії".
- 3. Програма сканує текст на наявність орфографічних помилок.
- 4. Програма підсвічує помилки та пропонує варіанти виправлення.
- 5. Користувач обирає відповідне виправлення або ігнорує помилку.
- 6. Користувач зберігає виправлений документ.

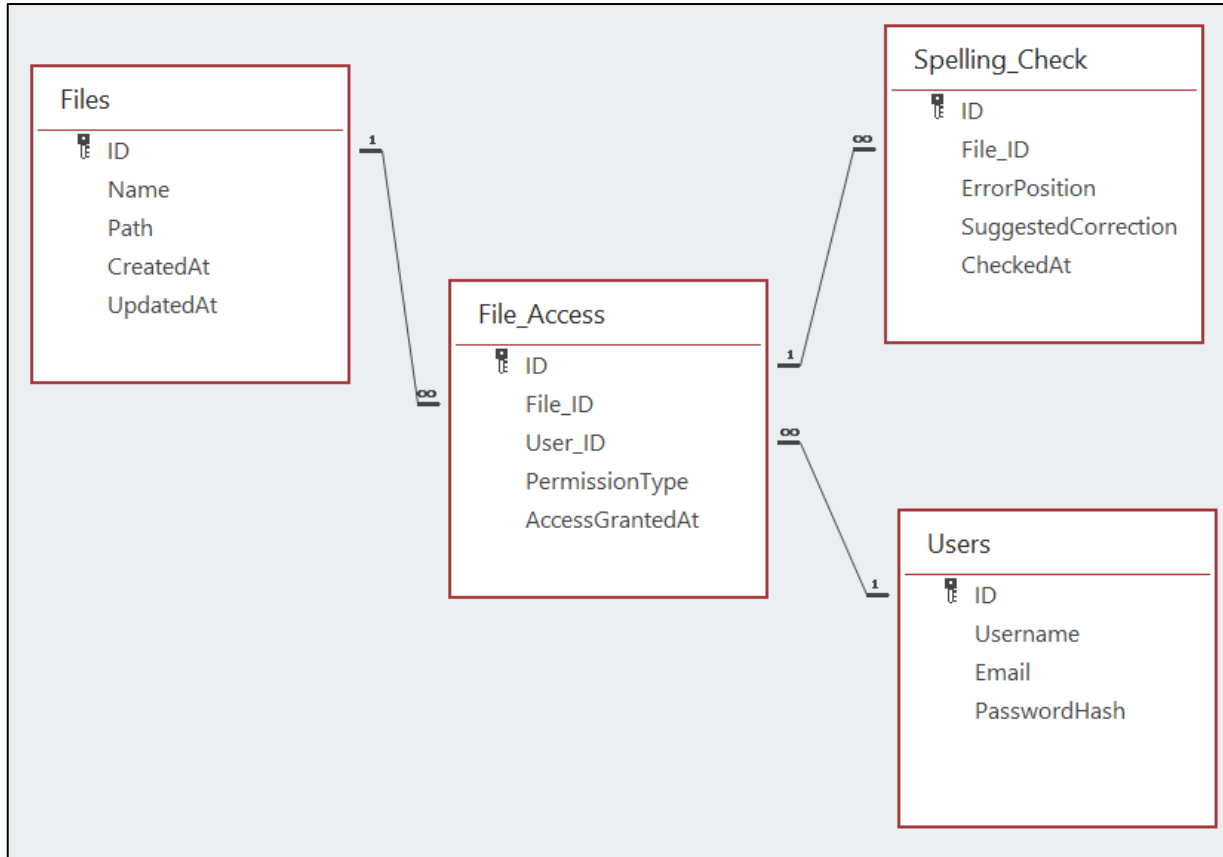
Альтернативний сценарій:

- Якщо у документі немає помилок, система повідомляє про це і завершує процес перевірки.
- Якщо користувач закриває документ, не зберігши виправлення, система запитує підтвердження про вихід без збереження.

Діаграма класів:



Структура бази даних:



Висновок: завдяки цій лабораторній роботі, я проаналізувала тему своєї роботи та намалювала схему прецеденту, діаграму класів, розробила основні класи і структуру системи баз даних.