Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни “Об’єктно-орієнтоване програмування”

ХМАРНЕ СХОВИЩЕ ДАНИХ

Керівник, проф. Бондарєв В. М.

Студент гр. ПЗПІ-23-4 Цисельський А. О.

Комісія:

проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарєв В. М.

доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кравець Н. С.

cт. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю. Ю.

Харків 2024

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра *програмної інженерії*

Рівень вищої освіти *перший (бакалаврський)*

Дисципліна *Об’єктно-орієнтоване програмування*

Спеціальність *121 Інженерія програмного забезпечення*

Освітня програма *Програмна інженерія*

Курс 1 Група ПЗПІ-23-4 Семестр 2

***ЗАВДАННЯ***

***на курсовий проект студента***

Цисельського Андрія Олександровича

1 Тема проекту:

Хмарне сховище даних

2 Термін здачі студентом закінченого проекту: ***“08” - червня - 2024 р.***

3 Вихідні дані до проекту:

*База користувачів: ідентифікаційний номер, прізвище та ім’я , пошта, пароль, об’єм дозволеної пам’яті для зберігання файлів, заповнений об’єм, файли які користувач зберігає. База файлів: ідентифікаційний номер, назва, тип файлу, розмір файлу, дата завантаження файлу на диск, дата видалення, ідентифікаційний номер користувача який володіє файлом, стан файлу: чи є цей файл улюбленим або видаленим.*

Зміст розрахунково-пояснювальної записки:

*Вступ, опис вимог, проектування програми, інструкція користувача, висновки*

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження | 13.02.2024 - 15.03.2024 р. |
| 2 | Формулювання вимог до програми | 16.03.2024 - 20.03.2024 р. |
| 3 | Розробка підсистеми зберігання та пошуку даних | 21.03.2024 -10.04.2024 р |
| 4 | Розробка функцій | 11.04.2024 –25.04.2024 р. |
| 5 | Розробка функцій зберігання та завантаження даних | 26.04.2024 –11.05.2024 р. |
| 6 | Тестування і доопрацювання розробленої програмної системи | 11.05.2024 –15.05.2024 р. |
| 7 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 16.05.2024 – 06.06.2024 р. |
| 8 | Захист | 03.06.2024–08.06.2024 р. |

Студент

Керівник Бондарєв Володимир Михайлович

«21» лютого 2024 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 50 с., 58 рис., 5 джерел.

Ключові слова КОРИСТУВАЧ, ФАЙЛ, ЗВІТ, БАЗА ДАНИХ, ООП, .NET, МОВА C#

Метою роботи є розробка програми «Хмарне сховище даних», яка буде надавати користувачу файли збережені у хмарному сховищі.

В результаті отримана програма, що дозволяє зберігати на віддаленому сервері файли та їх характеристики, такі як: назва, тип, об’єм, дата завантаження, дата видалення, ідентифікаційний номер користувача який володіє файлом, стан файлу: чи є цей файл улюбленим або видаленим. Є можливість зберігати нові файли у сховищі, завантажувати на персональний комп’ютер, видаляти, сортувати та редагувати їх.

В процесі розробки використано середовище розробки Microsoft Visual Studio 2022, фреймворк WPF, платформа .NET 8.0, мова програмування C#, платформа хмарних обчислень Azure.

**ЗМІСТ**

[ВСТУП 6](#_Toc167044521)

[1 ОПИС ВИМОГ 7](#_Toc167044522)

[1.1 Сценарії 8](#_Toc167044523)

[1.2 Функції 13](#_Toc167044524)

[2 ПРОЕКТУВАННЯ ПРОГРАМИ 22](#_Toc167044525)

[2.1 Об’єктна модель програми. 22](#_Toc167044526)

[2.2 Реалізація деяких функцій програми. 29](#_Toc167044527)

[2.2.1 Основні функції класу «MainWindow». 29](#_Toc167044528)

[2.2.2 Основні функції класу «SignUpWindow». 31](#_Toc167044529)

[2.2.3 Основні функції класу «Email». 33](#_Toc167044530)

[2.2.4 Основні функції класу «Storage». 36](#_Toc167044531)

[2.3 Формат даних. 43](#_Toc167044532)

[2.4 UML діаграма послідовності. 44](#_Toc167044533)

[3 ІНСТРУКЦІЯ КОРИСТУВАЧА 45](#_Toc167044534)

[3.1 Встановлення програми. 45](#_Toc167044535)

[3.2 Робота з програмою. 45](#_Toc167044536)

[ВИСНОВОК 50](#_Toc167044537)

[ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 51](#_Toc167044538)

**ВСТУП**

У цій курсовій роботі розглянуто розробку хмарного сховища файлів з використанням платформи хмарних обчислень Microsoft Azure. Основним завданням проекту є створення додатка, що забезпечує зберігання, управління та доступ до файлів, розміщених на сервері Azure. Для управління метаданими файлів та інформацією про користувачів використовується база даних Microsoft Azure SQL Database.

Сучасні підприємства та індивідуальні користувачі стикаються з необхідністю зберігання великих обсягів файлів різного типу, таких як документи, зображення, відео тощо. Використання хмарних сервісів дозволяє забезпечити високу доступність, надійність та безпеку збережених даних. У цьому проекті хмарне сховище надає зручний та масштабований інструмент для зберігання файлів.

Розроблений додаток надає користувачам можливість завантажувати файли на сервер, організовувати видаляти, редагувати та відновляти їх, здійснювати пошук. Крім того, система може повідомляти користувача про процес завантаження файли з хмари на комп’ютер та навпаки.

Ключовим аспектом розробки є використання об’єктно-орієнтованого програмування (ООП). Завдяки принципам ООП, таким як інкапсуляція, наслідування та поліморфізм, додаток є гнучким та масштабованим. Це дозволяє легко вносити зміни до початкового коду та адаптувати систему до нових вимог. Взаємодія з базою даних здійснюється через Entity Framework, що спрощує роботу з даними та забезпечує ефективну взаємодію між додатком та базою даних.

Важливо зазначити, що хмарна платформа Azure пропонує широкий спектр сервісів для забезпечення безпеки та захисту даних, таких як шифрування, контроль доступу та резервне копіювання. Це забезпечує високу надійність та безпеку зберігання файлів, що є критично важливим для користувачів.

Таким чином, розробка хмарного сховища файлів з використанням Microsoft Azure у рамках цієї курсової роботи є актуальним та важливим завданням, що сприяє розвитку навичок об’єктно-орієнтованого програмування та забезпечує ефективне управління великими обсягами файлів у хмарному середовищі.

**1 ОПИС ВИМОГ**

1.1 Сценарії

**Сценарій 1. Вхід користувача до свого облікового запису**

*Основний сценарій*

1. Користувач заповнює поля у вікні входу і натискає кнопку "Log In".

2. Програма перевіряє дані користувача.

3. Користувач бачить головне вікно програми.

*Додатковий сценарій*

1. Користувач заповнює поля у вікні входу і натискає кнопку "Log In".

2. Програма перевіряє дані користувача та бачить що щось не співпадає.

3. Користувач бачить повідомлення, що дані введені невірно.

**Сценарій 2. Запам’ятовування користувача на комп’ютері**

*Передумова*

Користувач заповнив поля у вікні входу.

*Основний сценарій*

1. Користувач ставить галочку у клітинці “Remember me” та натискає кнопку "Log In".

2. Програма перевіряє дані користувача та запам'ятовує користувача.

*Додатковий сценарій*

1. Користувач ставить галочку у клітинці “Remember me” та натискає кнопку "Log In".

2. Програма перевіряє дані користувача та їй не вдається запам’ятовати користувача.

3. Користувач бачить повідомлення про помилку.

**Сценарій 3. Реєстрація користувачем нового облікового запису**

*Передумова*

Користувач натиснув кнопку “Sign Up” у вікні входу.

*Основний сценарій*

1. Користувач заповнює поля у вікні входу і натискає кнопку "Sign Up".

2. Програма перевіряє дані користувача.

3. Користувач бачить вікно підтвердження пошти.

*Додатковий сценарій*

1. Користувач заповнює поля у вікні входу і натискає кнопку "Sign Up ".

2. Програма перевіряє дані користувача та бачить що щось не співпадає.

3. Користувач бачить повідомлення, що дані введені невірно.

**Сценарій 4. Підтвердження пошти**

*Передумова*

Користувач пройшов перевірку та натиснув кнопку “Sign Up” у вікні реєстрації.

*Основний сценарій*

1. Користувач дивиться код на пошті та вводить його у відповідне поле.

2. Програма перевіряє код.

3. Користувач бачить головне вікно програми.

*Додатковий сценарій*

1. Користувач дивиться код на пошті та вводить його у відповідне поле.

2. Програма перевіряє код та бачить що він не співпадає з дійсним.

3. Користувач бачить повідомлення, що код неправильний.

*Додатковий сценарій*

1. Користувач дивиться код на пошті та не бачить його.

2. Користувач натискає кнопку “Send again”

3. Програма надсилає новий код.

**Сценарій 5. Загрузка файлу до хмарного сховища**

*Передумова*

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

*Основний сценарій*

1. Користувач натискає кнопку "Add File".

2. Програма виводить користувачу провідник де він може вибрати файл.

3. Користувач вибирає файл та натискає кнопку підтвердження.

4. Програма завантажує файл на хмару.

5. Програма додає запис з інформацією про завантаження до вкладки “Downloads”.

6. Програма оновлюється та відображає усі файли у сховищі.

*Додатковий сценарій*

1. Користувач натискає кнопку "Add File".

2. Програма виводить користувачу провідник де він може вибрати файл.

3. Користувач вибирає файл та натискає кнопку підтвердження.

4. Програма знаходить помилку при спробі завантаження файлу на хмару.

5. Користувач бачить повідомлення про помилку.

**Сценарій 6. Пошук файлів**

*Передумова*

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

*Основний сценарій*

1. Користувач вводить ім’я або тип файлу у рядок пошуку.

2. Програма знаходить усі файли за критеріями пошуку.

3. Користувач бачить знайдені файли.

*Додатковий сценарій*

1. Користувач вводить ім’я або тип файлу у рядок пошуку.

2. Програма не знаходить жодних файлів за критеріями пошуку.

3. Користувач бачить пусте місце.

**Сценарій 7. Видалення файлу**

*Передумова*

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

*Основний сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Delete” або натискає кнопку зі смітником на самому файлі.

2. Програма видаляє файл з хмари.

3. Програма оновлюється та відображає усі файли у сховищі.

4. Програма оновлюється та відображає видалений файл на вкладці “Trash”.

*Додатковий сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Delete” або натискає кнопку зі смітником на самому файлі.

2. Програма знаходить помилку при спробі видалення файлу з хмару.

3. Користувач бачить повідомлення про помилку.

**Сценарій 8. Перейменування файлу**

*Передумова*

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

*Основний сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Rename”

2. Користувач вводить нове ім’я.

3. Програма перевіряє введені дані.

4. Програма змінює назву файлу.

*Додатковий сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Rename”

2. Користувач вводить нове ім’я.

3. Програма перевіряє введені дані та знаходить помилку.

4. Користувач бачить повідомлення, що дані введено невірно.

**Сценарій 9. Додавання файлу до “Улюблених” файлів**

*Передумова*

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

*Основний сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Add to favorites” або натискає кнопку з зіркою на самому файлі

2. Програма додає файл до “Улюблених” файлів.

3. Користувач бачить цей файл на вкладці “Tagged Files”.

*Додатковий сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Add to favorites” або натискає кнопку з зіркою на самому файлі

2. Програмі не вдається додати файл до “Улюблених” файлів.

3. Користувач бачить повідомлення про помилку.

**Сценарій 10. Завантаження файлу на комп’ютер**

*Передумова*

Користувач пройшов аутентифікацію і відкрив головне вікно програми.

*Основний сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Download” або натискає кнопку зі стрілкою на самому файлі

2. Програма виводить користувачу провідник де він може вибрати папку в яку він хоче завантажити файл.

3. Користувач вибирає папку та натискає кнопку підтвердження.

4. Програма завантажує файл на комп’ютер.

6. Програма додає запис з інформацією про завантаження до вкладки “Downloads”.

7. Програма оновлюється та відображає усі файли у сховищі.

*Додатковий сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Download” або натискає кнопку зі стрілкою на самому файлі

2. Програма виводить користувачу провідник де він може вибрати папку в яку він хоче завантажити файл.

3. Користувач вибирає папку та натискає кнопку підтвердження.

4. Програма знаходить помилки при спробі завантаження файлу на комп’ютер.

5. Користувач бачить повідомлення про помилку.

**Сценарій 11. Відтворення видаленого файлу з корзини**

*Передумова*

Користувач пройшов аутентифікацію і відкрив головне вікно програми, а саме вкладку з видаленими файлами - Корзину.

*Основний сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Restore” або натискає кнопку зі стрілкою на самому файлі

2. Програма не бачить помилок при спробі відтворення файлу.

3. Користувач бачить відтворений файл у своєму сховищі.

*Додатковий сценарій*

1. Користувач натискає правою кнопкою миші по файлу та вибирає опцію “Restore” або натискає кнопку зі стрілкою на самому файлі

2. Програма знаходить помилки при спробі відтворення файлу.

3. Користувач бачить сповіщення про помилку.

1.2 Функції

**Функція 1. Вхід користувача до свого облікового запису**

Панель вводу даних користувача розташовується на вікні входу до облікового засобу (рис. 1.1) і містить такі текстові поля:

1) Пошта – електронна пошта користувача;

2) Пароль – пароль від облікового запису користувача;

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1.1 – Вікно входу

В текстовому полі для пошти важливо вводити коректні дані які співпадають з шаблоном назв електронних скриньок. Довжина паролю повинна бути не менше 6 символів. Відсутність даних в полі вводу або некоректність цих даних означає, що буде виведено повідомлення з текстом, що саме не так. Повідомлення про помилку вводу буде з'являтися над текстовими полями. Приклади повідомлень про помилку:

* Fill in all the fields.
* Email has been entered incorrectly.
* Password must contain more than 6 symbols.

Під текстовими полями є кнопка “Log In” яка здійснюватиме операцію перевірки даних, оновлення бази даних та перенесення користувача до головного вікна.

**Функція 2. Запам’ятовування користувача на комп’ютері**

На вікні входу до облікового засобу (рис. 1.1) міститься клітинка “Remember me”. Її функція – давати змогу користувачу не вводити свої дані кожного разу при запуску програми. Якщо в даній клітинці стоїть галочка, то при натисканні кнопки “Log In” буде програма запам’ятає користувача та не запросить його дані наступного разу.

**Функція 3. Реєстрація користувачем нового облікового запису**

На вікні входу до облікового засобу (рис. 1.1) міститься кнопка “Sign Up”. При натисканні цієї кнопки відкриватиметься вікно реєстрації (рис. 1.2), яке містить такі текстові поля:

1) Ім’я – ім’я користувача;

2) Прізвище – прізвище користувача;

3) Пошта – електронна пошта користувача;

4) Пароль – пароль від облікового запису користувача;

5) Перевірочний пароль – повторний ввід паролю;

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 1.2 – Вікно реєстрації

В текстовому полі для пошти важливо вводити коректні дані які співпадають з шаблоном назв електронних скриньок. Довжина паролю повинна бути не менше 6 символів. Пароль повинен співпадати с перевірочним паролем. Ім’я та прізвище не повинні перевищувати 15 символів. Відсутність даних в полі вводу або некоректність цих даних означає, що буде виведено повідомлення з текстом, що саме не так. Повідомлення про помилку вводу буде з'являтися над текстовими полями. Приклади повідомлень про помилку:

* Fill in all the fields.
* Email has been entered incorrectly.
* Password must contain more than 6 symbols.
* Name lgth must be between 2 and 15 symbols.
* Surname lgth must be between 2 and 15 symbols.
* Password must contain more than 6 symbols.
* The passwords don't match.

Під текстовими полями є кнопка “Sign Up” яка здійснюватиме операцію перевірки даних та перенесення користувача до вікна підтвердження пошти.

**Функція 4. Підтвердження пошти**

Панель вводу коду з почтової скриньки користувача розташовується на вікні підтвердження пошти (рис. 1.3) і містить текстове поле для вводу коду. Повідомлення про помилку вводу буде з'являтися над текстовим полем. Приклади повідомлень про помилку:

* The code is incorrect! Try again.

Під текстовим полем є кнопка “Send Again” яка надсилає новий код на пошту користувача та блокується на 30 секунд.

Під текстовим полем є кнопка “Confirm” яка здійснюватиме операцію перевірки даних, оновлення бази даних та перенесення користувача до головного вікна.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 1.3 – Вікно підтвердження пошти

**Функція 5. Загрузка файлу до хмарного сховища**

На головному вікні (рис. 1.4) у висувному меню міститься кнопка “Add File”. При натисканні цієї кнопки відкриється провідник де користувач зможе вибрати файл для завантаження на хмару. Після того як файл вибрано він загружається до сховища, оновлюється база даних.

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Рисунок 1.4 – Головне вікно програми

Файл вибраний користувачем на має перевищувати вільний об’єм сховища користувача та має мати унікальне ім’я. Повідомлення про помилку – це окреме вікно-сповіщення.

Процес завантаження додається до журналу завантажень (рис. 1.5).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1.5 – Журнал завантажень

**Функція 6. Пошук файлів**

На головному вікні (рис. 1.4) у правому верхньому кутку міститься значок лупи. При наводці на даний значок розкривається текстове поле пошуку. В текстовому полі регістр не важливий. Користувач вводить фрагмент імені файлу або його розширення. При вводі кожного символу програма оновлює список файлів які бачить користувач.

**Функція 7. Видалення файлу**

На головному вікні (рис. 1.4) на іконці кожного файлу міститься кнопка зі значком корзинки. Також при натисканні правої кнопки миші на іконці кожного файлу відкривається контекстне меню (рис. 1.6), де є пункт “Delete”. При натисканні однієї з цих кнопок стан файлу змінюється на “видалений” та файл буде відображатись у вкладці “Trash” з іконкою корзини. У сховищі оновиться список файлів, а у користувача звільниться місце для інших файлів. Оновлюється база даних.

Изображение выглядит как текст, снимок экрана, фиолетовый, Прямоугольник

Автоматически созданное описание

Рисунок 1.6 – Контекстне меню файлу

**Функція 8. Перейменування файлу**

У контекстному меню кожного файлу (рис. 1.6) є пункт “Rename”. При виборі цього пункту відкриється діалогове вікно для зміни імені файлу (рис. 1.7). Це вікно містить текстове поле для імені та дві кнопки: Confirm та Cancel. Якщо текстове поле пусте, то виведеться повідомлення: “Enter something!” Кнопка “Confirm” робить перевірку даних оновлює базу даних та змінює ім’я файлу. Кнопка “Cancel” відміняє операцію.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1.7 – Діалогове вікно для зміни імені файлу

**Функція 9. Додавання файлу до “Улюблених” файлів**

На головному вікні (рис. 1.4) на іконці кожного файлу міститься кнопка зі значком зірки. Також в контекстному меню (рис. 1.6) є пункт “Add to favorites”. При натисканні однієї з цих кнопок стан файлу змінюється на “доданий до улюблених” та файл буде відображатись у вкладці “Tagged files” з іконкою зірки. Оновлюється база даних.

**Функція 10. Завантаження файлу на комп’ютер**

На головному вікні (рис. 1.4) на іконці кожного файлу міститься кнопка зі значком стрілки вниз. Також в контекстному меню (рис. 1.6) є пункт “Download”. При натисканні однієї з цих кнопок відкривається провідник де користувач може вибрати папку до якої треба завантажити файл. Коли папку вибрано починається завантаження файлу. Повідомлення про помилку – це окреме вікно-сповіщення.

Процес завантаження додається до журналу завантажень (рис. 1.5).

**Функція 11. Відтворення видаленого файлу з корзини**

На головному вікні (рис. 1.8) у вкладці “Trash” на іконці кожного відселеного файлу міститься кнопка зі значком закругленої стрілки. Також в контекстному меню відселеного файлу (рис. 1.9) є пункт “Restore”. При натисканні однієї з цих кнопок видалений файл відтворюється, а його стан стає “невидалений”. Оновлюється база даних.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1.8 – Вкладка “Trash”

Изображение выглядит как текст, снимок экрана, Шрифт, Прямоугольник

Автоматически созданное описание

Рисунок 1.9 – Контекстне меню видаленого файлу

**2 ПРОЕКТУВАННЯ ПРОГРАМИ**

2.1 Об’єктна модель програми.

В даній програмі містяться форми, та класи, що взаємодіють з цими формами (рисунок 2.1).

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.1 – Архітектура програми.

У цій програмі реалізовані такі основні класи, як «MainWindow», «SignUpForm» та «Storage». Діаграма класів зображена на рисунку 2.2.

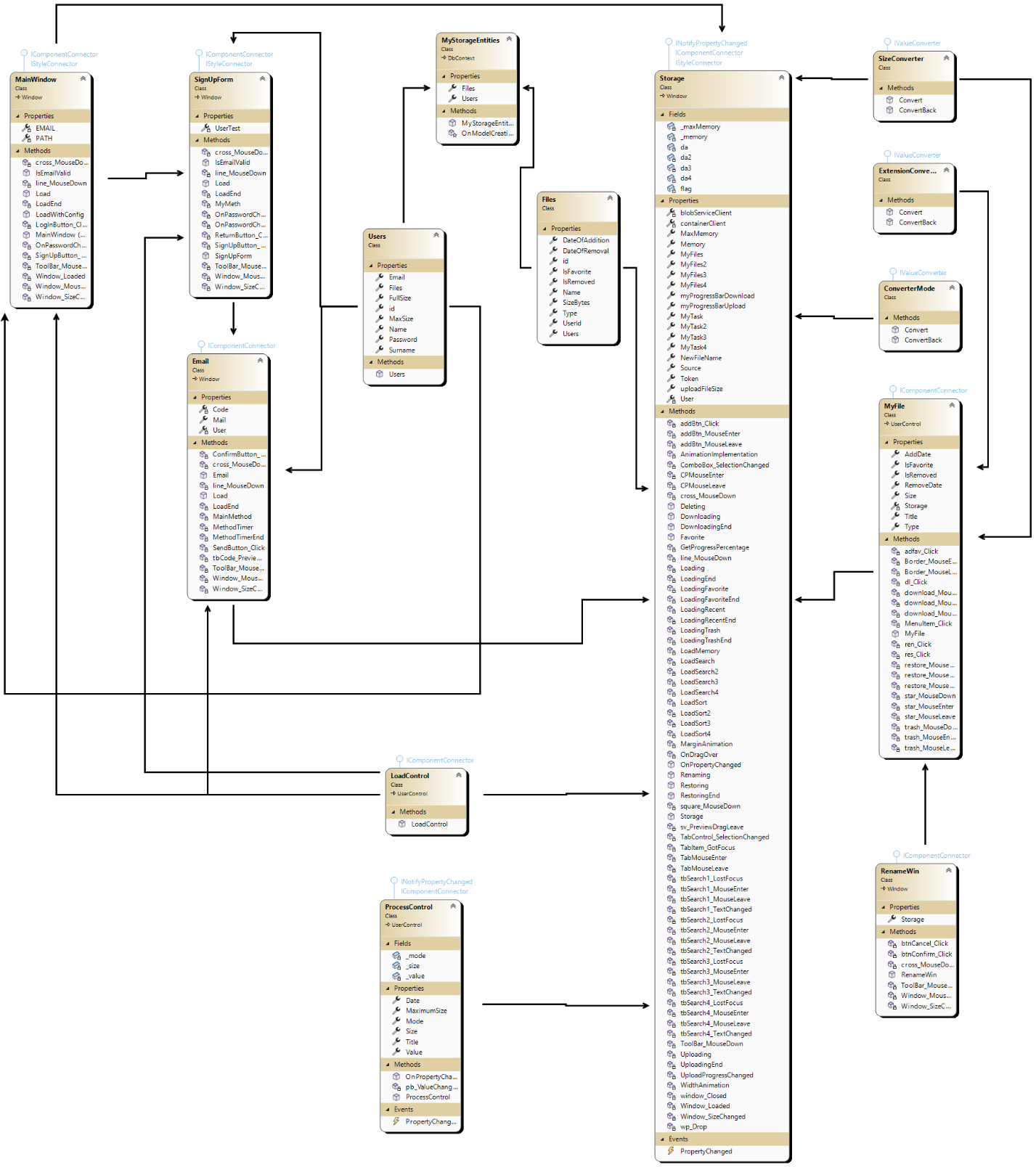


Рисунок 2.2 – Діаграма класів.

Клас «MainWindow» (рис. 2.3) відповідає за вхід користувача до головної сторінки хмарного сховища. Тут відбуваються перевірки на заповненість полів та правильність введених даних. Відбувається доступ до бази даних та перевіряється наявність акаунту.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2.3 – Клас «MainWindow».

Класи «SignUpForm» (рис. 2.4) та «Email» (рис. 2.5) відповідають за успішну реєстрацію та подальший вхід користувача до головної сторінки хмарного сховища. В них реалізуються усі перевірки на заповненість полів та правильність даних для додавання нового запису до таблиці користувачів в базі даних.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 2.4 – Клас «SignUpForm».

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2.5 – Клас «Email».

Клас «Storage» (рисунок 2.6) робить усі необхідні дії на головній сторінці. Всі методи натискання на кнопку, пошуку та сортуванню були закодовані саме в цьому класі.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

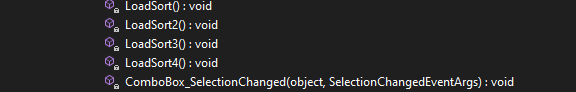


Рисунок 2.6 – Клас «Storage».

Програма використовує Entity Framework для роботи з базами даних. Діаграма на базі моделі зображена на рисунку 2.7.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 2.7 – Діаграма на базі моделі Entity Framework.

Програма реалізовує класи-конвертери такі як: «ConverterMode», «SizeConverter» та «ExtensionConverter».

Клас «ConverterMode» (рисунок 2.8) конвертує операції загрузки даних на хмару та з неї у путь до відповідної гіф-анімації.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.8 – Клас «ConverterMode».

Клас «SizeConverter» (рисунок 2.9) конвертує об’єм пам’яті.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 2.9 – Клас «SizeConverter».

Клас «ExtensionConverter» (рисунок 2.10) конвертує розширення файлу у путь до відповідної іконки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.9 – Клас «ExtensionConverter».

2.2 Реалізація деяких функцій програми.

2.2.1 Основні функції класу «MainWindow».

Подія натискання кнопки “Log In”(рис. 2.10) відповідає за перевірку даних користувача для входу до свого облікового засобу. Виклик методу “Load” (рис. 2.11) відбувається в окремому потоці та відповідає за доступ до бази даних для перевірки даних та відображення анімації завантаження. Виклик методу “LoadEnd” (рис. 2.12) припиняє анімацію завантаження. Якщо користувач поставив галочку до клітинки “Remember me” то буде створено файл з поштою користувача для того щоб при визові події загрузки вікна (рис. 2.13) програма не просила користувача вводити дані заново. Виклик методу “LoadWithConfig” (рис. 2.14) шукає користувача за поштою у файлі “config.txt” та заходить до його облікового запису.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.10 – Подія натискання кнопки “Log In”.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2.11 – Метод “Load”.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.12 – Метод “LoadEnd”.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.13 – Подія завантаження вікна.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.14 – Метод “LoadWithConfig”.

2.2.2 Основні функції класу «SignUpWindow».

Подія натискання кнопки “Sign Up”(рис. 2.15) відповідає за перевірку даних користувача для входу до свого облікового засобу. Виклик методу “Load” (рис. 2.16) відбувається в окремому потоці та відповідає за доступ до бази даних для перевірки даних та відображення анімації завантаження. Виклик методу “LoadEnd” (рис. 2.17) припиняє анімацію завантаження. Для підтвердження пошти викликається метод “MyMeth” (рис. 2.18)

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.15 – Подія натискання кнопки “Sign Up”.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.16 – Метод “Load”.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.17 – Метод “LoadEnd”.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.18 – Метод “MyMeth”.

2.2.3 Основні функції класу «Email».

Подія натискання кнопки “Confirm”(рис. 2.19) відповідає за перевірку вводу. Виклик методу “Load” (рис. 2.20) відбувається в окремому потоці та відповідає за доступ до бази даних для занесення нового облікового запису, переходу до головного вікна та відображення анімації завантаження . Виклик методу “LoadEnd” (рис. 2.21) припиняє анімацію завантаження. Подія натискання кнопки “Send again”(рис. 2.22) викликає метод “MethodTimer” (рис. 2.23) який запускає таймер на 30 секунд та блокує кнопку “Send again”. Після завершення потоку з таймером викликається метод “MethodTimerEnd” (рис. 2.23), який розблоковує кнопку. Метод “MainMethod” (рис. 2.24) відповідає за надсилання коду на поштову скриньку користувача.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 2.19 – Подія натискання кнопки “Confirm”.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.20 – Метод “Load”.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.21 – Метод “LoadEnd”.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 2.22 – Подія натискання кнопки “Send again”.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.23 – Методи “MrthodTimer” та “MethodTimerEnd”.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2.24 – Методи “MainMethod”.

2.2.4 Основні функції класу «Storage».

В даному класі є декілька основних функцій:

* Функції відображення файлів.
* Функція загрузки файлу на хмару.
* Функція завантаження файлу з хмари на комп’ютер.
* Функції видалення та відновлення
* Функції пошуку.
* Функції сортування

Функції відображення файлів є доволі схожими. Їх основна мета – зчитати файли з бази даних в залежності від обраної вкладки та відобразити у вікні. При події зміни вкладки (рис. 2.25) визивається один з методів відображення. Так як вони схожі, приклад відображення усіх наявних файлів, а саме методи “Loading” та “LoadingEnd” зображено на рисунку 2.26.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок 2.25 – Подія зміни вкладки.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.26 – Методи “Loading” та “LoadingEnd”.

Функція загрузки файлу на хмару спрацьовує при події натискання кнопки “Add File” (рис. 2.27) на головному вікні. Це подія запускає в окремому потоці метод “Uploading” (рис. 2.28) який по завершенню викликає метод “UploadingEnd” (рис. 2.29). В цих методах проходить перевірка на вільне місце на хмарі, оновлення бази даних та завантаження файлу.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.27 – Подія натиснення кнопки “Add File”.

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.28 – Метод “Uploading”.

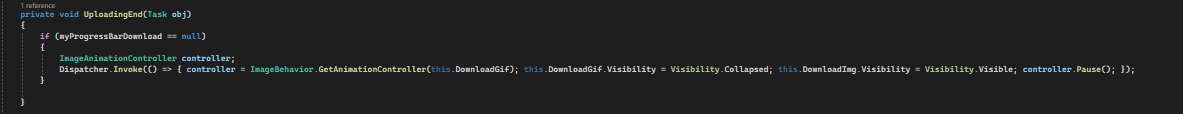


Рисунок 2.29 – Метод “UploadingEnd”.

Функція завантаження файлу з хмари на комп’ютер спрацьовує при події натискання кнопки “Download” у контекстному меню файлу або на самому файлі. Подія зображена на рисунку 2.30. Це подія запускає в окремому потоці метод “Downloading” (рис. 2.31) який по завершенню викликає метод “DownloadingEnd” (рис. 2.32). В цих методах проходить вибір папки користувачем, перевірка та завантаження файлу на комп’ютер.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 2.30 – Подія натиснення кнопки “Download”.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.31 – Метод “Downloading”.

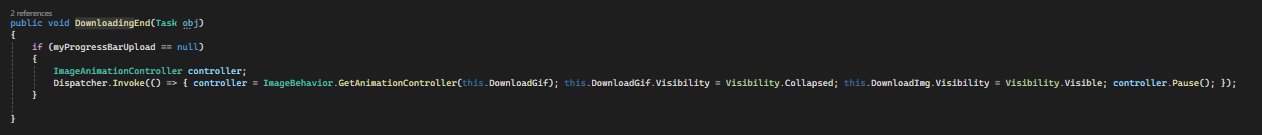


Рисунок 2.32 – Метод “DownloadingEnd”.

Функція видалення файлу з хмари спрацьовує при події натискання кнопки “Delete” у контекстному меню файлу або на самому файлі. Подія зображена на рисунку 2.33. Це подія запускає в окремому потоці метод “Deleting” (рис. 2.34). В цьому методі стан файлу змінюється на “видалений”, файл видаляється з хмари та пересувається до Кошику, де в період одного тижня його можна відновити.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 2.33 – Подія натиснення кнопки “Delete”.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.34 – Метод “Deleting”.

Функція відновлення спрацьовує при події натискання кнопки “Restore” у контекстному меню файлу або на самому файлі. Подія зображена на рисунку 2.35. Це подія запускає в окремому потоці метод “Restoring” (рис. 2.36), а після його завершення метод “RestoringEnd”(рис 2.37). В цих методах стан файлу змінюється на “невидалений”, а сам файл повертається до сховища. Оновлюється база даних.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.35 – Подія натиснення кнопки “Restore”.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.36 – Метод “Restoring”.

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 2.37 – Метод “RestoringEnd”.

Функція пошуку працює в залежності від відкритої вкладки та викликається при події внесення тексту користувачем до текстового поля пошуку. Подія зображена на рисунку 2.38. Подія визиває метод “LoadSearch” (рис. 2.39) в окремому потоці. Цей метод застосовує критерії пошуку до файлів та відображає їх.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2.38 – Подія внесення тексту до поля пошуку.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.39 – Метод “LoadSearch”.

Функція сортування працює викликається при події зміни вибраного режиму сортування. Подія зображена на рисунку 2.40. Подія визиває методи відображення в залежності від вибраної вкладки. Приклад методів відображення є на рисунку 2.26.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 2.40 – Подія зміни вибраного режиму сортування.

2.3 Формат даних.

Файли та база даних програми зберігаються на хмарному сервісі Azure. Основними ресурсами є база даних, сервер та сховище (рис. 2.41).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2.41 – Серверні ресурси програми.

Усі файли користувачів зберігаються у блоб контейнерах. Всередині будуть зберігатися папки імена котрих – поштові адреси користувачів.

Изображение выглядит как текст, линия, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 2.42 – Блоб контейнери.

База даних має дві таблиці – Користувачі (Users) та Файли (Files) (рис. 2.43)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2.42 – База даних.

2.4 UML діаграма послідовності.

Діаграма послідовності зображена на рисунку 2.43.

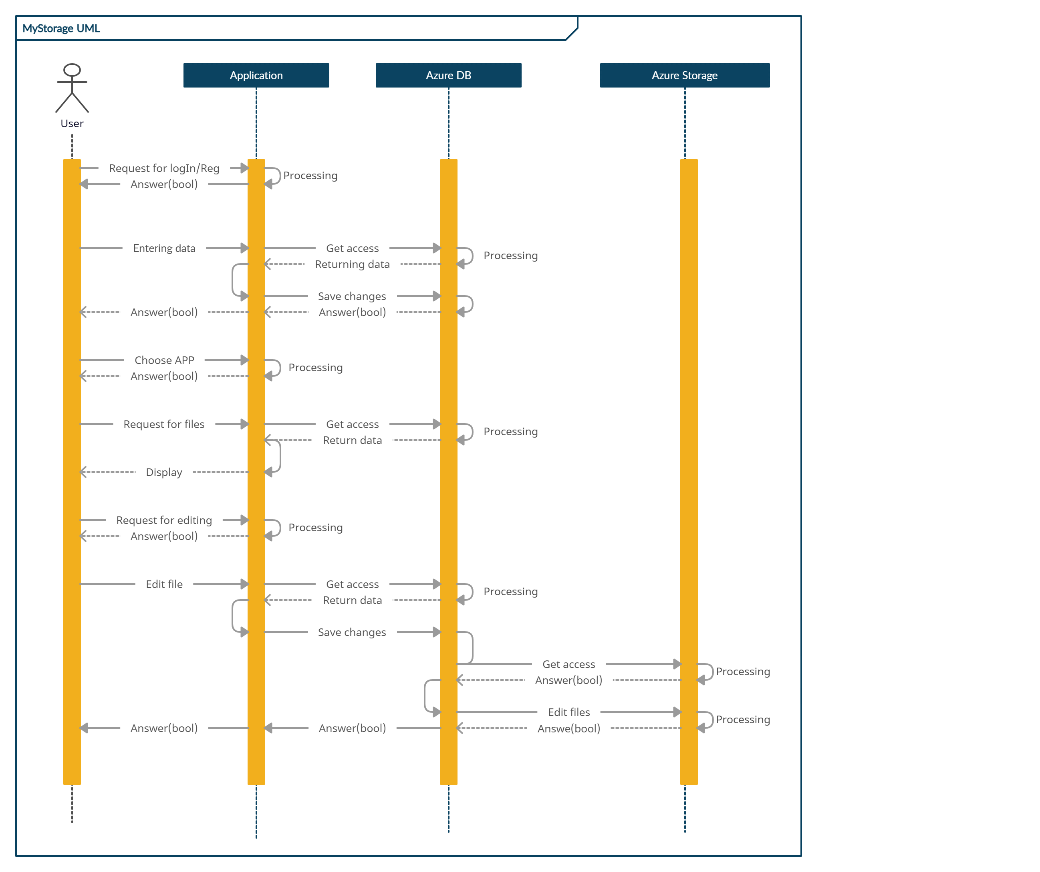


Рисунок 2.43 – Діаграма послідовності.

**3 ІНСТРУКЦІЯ КОРИСТУВАЧА**

3.1 Встановлення програми.

Кроки встановлення програми мають наступний вигляд:

* запускаємо файл «MyStorage\_v02.exe» та вибираємо шлях в будь-яку директорію, де буде розташована наша папка проекту (наприклад на робочий стіл);
* відкриваємо папку «MyStorage\_v02» й виконуємо наступні дії: папка «bin» → папка «Debug»;
* запускаємо програму подвійним натисканням лівої клавіши мишки на файл «MyStorage\_v02.exe».

3.2 Робота з програмою.

Запускаємо програму шляхом, описаним в пункті 3.1. Відкривається вікно входу (рис. 3.1.)

Изображение выглядит как текст, снимок экрана, дизайн, Шрифт

Автоматически созданное описание

Рисунок 3.1 – Вікно входу

Так як ми ще не зареєструвалися, то необхідно перейти до пункту реєстрації, натиснувши на формі входу кнопку «Sign Up». Але за потреби вийти з програми, натиснувши на значок «Х» в правому верхньому кутку вікна.

Вікно реєстрації наведено на рисунку 3.2, де користувач обов’язково повинен вписати всі дані, які потребує програма. Після коректного вводу інформації програма перенесе користувача до вікна підтвердження пошти, або вкаже на поле, яке необхідно дозаповнити або виправити.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 3.2 – Вікно реєстрації

Після успішної реєстрації, програма переносить користувача до форми підтвердження поштової адреси (рис. 3.3), де користувач вводить код надісланий на його почтову скриньку. Якщо код не прийшов користувач може запросити його знову натиснувши кнопку “Send again”.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 3.3 – Вікно підтвердження пошти

Після підтвердження пошти користувач потрапляє до головного вікна програми (рис. 3.4). Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 3.4 – Головне вікно програми

Додавання файлів до сховища:

Користувач може загрузити файли на хмару натиснувши кнопку “Add file”, яка розташована по центру висувного меню. Після того як кнопку натиснуту користувач побачить провідник де йому треба вибрати файл для завантаження на хмару. Стежити за процесом завантаження можна на вкладці “Downloads”.

Скачування файлів:

Користувач може скачати файл натиснувши кнопку зі стрілкою на самому файлі або натиснувши правою кнопкою миші по файлу та вибравши відповідний пункт у контекстному меню. Після натискання кнопки треба лише вибрати директорію куди буде скачано файл. Стежити за процесом завантаження можна на вкладці “Downloads”.

Видалення файлів:

Користувач може видалити файл натиснувши кнопку зі смітником на самому файлі або натиснувши правою кнопкою миші по файлу та вибравши відповідний пункт у контекстному меню. Після видалення файл опиниться у вкладці “Trash”. Після цього на вкладці “Trash” файл можна або повністю видалити натиснувши смітник, або відновити натиснувши кнопку зі стрілкою (рис. 3.5).

Изображение выглядит как текст, снимок экрана, Шрифт, Прямоугольник

Автоматически созданное описание

Рисунок 3.5 – Контекстне меню видаленого файлу

Змінити ім’я файлу:

Користувач може змінити ім’я файлу натиснувши правою кнопкою миші по файлу та вибравши відповідний пункт у контекстному меню. Після натискання користувач побачить вікно для зміни імені файлу (рис. 3.6).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3.6 – Діалогове вікно для зміни імені файлу

Пошук файлів:

Користувач може шукати файли за ім’ям та розширенням ввівши критерії пошуку до відповідного поля на головному вікні (рис. 3.4).

Сортування файлів:

Користувач може сортувати файлами вибравши критерій сортування у випадаючому списку, що розташований зліва від пошуку (рис. 3.4).

**ВИСНОВОК**

Під час виконання курсової роботи розроблена програма «Хмарне сховище», яка дозволяє зберігати файли у хмарному сховищі та керувати ними. В ході виконання роботи були реалізовані різні методи: вивід інформації, видалення, завантаження на сервер та з нього, аналіз даних, і т. д.

Дана курсова робота є прикладом використання спрощеного варіанту хмарного сховища, де зберігають файли.

В результаті написання цієї курсової роботи була створена повноцінна програма, з усіма необхідними можливостями. Всі данні розроблені на програмній мові С#, а також вивчені та розібрані різні аспекти об’єктно-орієнтованого програмування, розробка програм у середовищі .NET.

Також були опрацьовані окремі фази розробки програмного забезпечення.

В ході виконання курсової роботи отримано  досвід розробки програм, вивчені стандарти програмного забезпечення, а також планування ходу розробки.

Звісно ця програма не ідеальна, тому, на подальше, її можна вдосконалити ще багатьма ідеями.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Set Up Microsoft Azure SQL Server and SQL Database (Step-By-Step Tutorial). URL: <https://www.youtube.com/watch?v=6joGkZMVX4o>

2. Material Design In XAML. URL: <http://materialdesigninxaml.net/>

3. Entity Framework documentation hub. URL: <https://learn.microsoft.com/en-us/ef/>

4. Полное руководство по языку программирования С# 12 и платформе .NET 8. URL: <https://metanit.com/sharp/tutorial/>

5. Документация по .NET. URL: <https://learn.microsoft.com/ru-ru/dotnet/>