Зміст

[Вимоги до системи 2](#_Toc162728805)

[Вимоги до системи в цілому 2](#_Toc162728806)

[Вимоги до процесу розробки 2](#_Toc162728807)

[Планування розробки проекту 2](#_Toc162728808)

[Розробка графічної частини 2](#_Toc162728809)

[Розробка функціоналу 2](#_Toc162728810)

[Об’єднання GUI і функціоналу 3](#_Toc162728811)

[Тестування програмного забезпечення 3](#_Toc162728812)

[Вимоги до структури та функціонування системи 4](#_Toc162728813)

[Робота з віртуальними сейфами 4](#_Toc162728814)

[Автоматичне резервування файлів 4](#_Toc162728815)

[Збереження історії резервування 4](#_Toc162728816)

[Робота з смартфоном та оптичними дисками 5](#_Toc162728817)

[Створення комбінацій клавіш 5](#_Toc162728818)

# Вимоги до системи

## Вимоги до системи в цілому

Програма повинна надавати користувачеві зручний інтерфейс для керування файлами та можливість вибору конкретних файлів і резервування їх в одному з віртуальних сейфів. Інтерфейс цієї програми має бути створений за допомогою технології .Net MAUI Blazor Hybrid, оскільки ця технологія дозволяє використовувати бібліотеку класів Blazor для передавання Views на веб-сайт за потреби, а також можливість масштабування на інші платформи. При необхідності під час розробки варто виділити функціонал програми в окрему C# Class Library.

## Вимоги до процесу розробки

Основний процес розробки буде організований у вигляді Scrum, але пункт «Планування розробки проекту» буде виконаний перед наступними ітераційними пунктами. Розробка буде умовно поділена на такі етапи:

### Планування розробки проекту

Під час цього етапу будуть оцінюватися підходи та вибиратися технології для розробки проекту. Складатися документація до майбутнього проекту та теоретичний приблизний макет проекту.

### Розробка графічної частини

Основна мета цього етапу створити графічну обгортку програми, яка ще не буде виконувати жодних дій пов’язаних з функціоналом резервування файлів, окрім переходу між Views та якимись базовими імплементаціями. Наприклад інтерфейсами сервісів та інше. Результатом має бути визначена у міру можливості графічна частина, якій буде лише необхідно імплементувати функціонал.

### Розробка функціоналу

Отримавши графічну частину, починається написання набору класів, які будуть реалізувати функціонування графічної частини, тобто інтерфейсів які були написані у попередній частині. Результатом є набір класів, що реалізують інтерфейси сервісів та інших сутностей програми.

### Об’єднання GUI і функціоналу

Написавши реалізації для класів та сервісів, потрібно за допомогою dependency injection пов’язати умовний інтерфейс сервісу певного View з його реалізацією. У результаті має бути готова програма, яка використовує інтерфейси, написані у другому етапі, та класи третьої частини, що реалізовують ці інтерфейси.

### Тестування програмного забезпечення

В кінці розробки проводиться тестування та виправляються баги програми. Результат — протестована програма, яка не містить багів.

## Вимоги до структури та функціонування системи

Резервування даних є критично важливою процедурою для забезпечення безпеки та доступності важливої інформації користувача. Люди можуть випадково видалити або пошкодити файли без попередження. Жорсткі диски, флеш-накопичувачі та інше обладнання може вийти з ладу внаслідок випадкових поломок або відмов. Резервне копіювання дозволяє відновити дані, які були збережені на пошкоджених носіях чи випадково видалені.

Програма для створення резервного копіювання даних та роботи з віртуальними сейфами автоматизує низку процесів для зручного та безпечного управління файлами користувачем. Відповідно, програма призначена для організації певного функціоналу, що наведений нижче:

### Робота з віртуальними сейфами

Програма має надавати можливість створювати віртуальні сейфи, у яких можна буде зберігати та резервувати дані. Сейф може бути окремим локальним диском, місцем на локальному диску або флеш накопичувачем. Через інтерфейс можна буде переглянути дані та вміст віртуальних сейфів, а також за потреби добавити файли для резервування чи їх забрати. Також, будуть такі дії з сейфами: видалення, перенесення та створення копій. Самі налаштування сейфу будуть зберігатися у спеціальній папці у каталозі даних програм.

### Автоматичне резервування файлів

У сторінці віртуального сейфу можна буде налаштувати проміжок автоматичного резервування. Наприклад, під час кожного запуску операційної системи. Автоматичне резервування можна буде налаштувати, як і для всіх сейфів на головній сторінці, так і для кожного окремого на власній сторінці сейфу. Пріоритетність така, що якщо є власний проміжок автоматичного резервування, то загальний не виконується, а якщо немає власного, то виконується загальний.

### Збереження історії резервування

Кожна дія над сейфами та файлами у них буде зберігатися в історії, яка буде доступна для перегляду у окремому розділі в сторінці віртуального сейфу. Буде збережено дата, час, файли, що переслалися у сейф, і файли, які не змогли зарезервуватися. Також, історію можна буде очищати чи за допомогою кнопки зберегти у вигляді текстового файлу через діалогове вікно.

### Робота з смартфоном та оптичними дисками

На головній сторінці програми можна буде обрати сторонні дії резервування, а саме резервування з телефона у віртуальний сейф і резервування віртуального сейфу на оптичний диск. З телефона можна буде перенести файли у віртуальний сейф через спеціальну сторінку програми. На оптичний диск через іншу сторінку можна буде записати дані. Програма має попереджати, якщо об’єм диску замалий для резервування.

### Створення комбінацій клавіш

У головній сторінці буде розділ налаштувань, де можна буде налаштувати комбінацію клавіш для швидкого перенесення файлів у поточний відкритий віртуальний сейф чи відкриття поточного відкритого віртуального сейфа у файловому провіднику. Для роботи клавіш має бути відкрита сторінка віртуального сейфу.