ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «**РОЗРОБКА ДЕСКТОПНОГО ДОДАТКУ ДО ГЕНЕРАЦІЇ ТА ЗБЕРІГАННЯ ПАРОЛІВ КОРИСТУВАЧІВ**»

Виконав: студент 4 курсу, групи ПД–41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Візер А. М.
(прізвище та ініціали)

Керівник Негоденко О.В.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Київ –2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально–науковий інститут інформаційних технологій

Кафедра Інженерія програмного забезпечення
Ступінь вищої освіти – «Бакалавр»
Спеціальність – 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерія програмного забезпечення
_____Негоденко О.В.____.
“____”_____2021 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

ВІЗЕР АНТОНІЙ МИКОЛАЙОВИЧ.

1.Тема роботи: «Розробка десктопного додатку до генерації та зберігання паролів користувачів».

Керівник роботи: Негоденко О.В кандидат технічних наук, доцент,

затверджені наказом вищого навчального закладу від «12» березня 2021 року № 65.

2.Строк подання студентом роботи «01» червня 2021 року .

3.Вхідні дані для роботи:

Менеджери паролів

Алгоритм генерації надійних паролів

Розробка додатку за допомого Microsoft Visual Studio, C#, WPF, XML.

.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1. Аналіз та характеристика програмного забезпечення

4.2. Концепція

4.3. Аналіз існуючих розробок

4.4. Аналіз отриманої інформації

5. Перелік графічного матеріалу

5.1. Мета, об’єкт, Предмет

5.2. Актуальність

5.3. Програми аналоги

5.4. Вимоги до нового додатку

5.5. Діаграми UML

5.6. Засоби розробки

5.7. Інтерфейс додатку

5.8. Зберігання даних

5.9. Висновки

Дата видачі завдання: «19» квітня 2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір джерел інформації	19.04.2021	Виконано
2	Вимоги до встановленого додатку	23.04.2021	Виконано
3	Оцінка якості тестування до систем	29.04.2021	Виконано
4	Концепція та архітектура додатку	02.05.2021	Виконано
5	Вступ, висновки, реферат	05.05.2021	Виконано
6	Розробка презентації	06.05.2021	Виконано
7	Перед захист диплому	11.05.2021	
8	Задача роботи	01.06.2021	

Студент _____. Візер А. М. .
(підпис) (прізвище та ініціали)

Керівник роботи _____. Негоденко О.В .
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 45 с., 25 рис., 15 джерел.

Ключеві слова: ПАРОЛІ, МЕНЕДЖЕР ПАРОЛІВ, ГЕНЕРАТОР ПАРОЛІВ, MICROSOFT VISUAL STUDIO.

Об'єкт дослідження – підвищення безпеки користувачів у всесвітній мережі інтернет.

Предмет дослідження – засоби для генерації та зберігання паролів користувачів.

Мета роботи – підвищення безпеки користування мережі інтернет шляхом розробки програмного забезпечення для генерації та зберігання паролів користувачів.

Методи дослідження – метод теорії інформації, метод оптимального управління, обробка та аналіз інформації.

У відповідності з поставленою метою для вирішення технічної проблеми в роботі вирішено такі завдання:

- Аналіз вимог до паролів та методів їх зберігання.
- Аналіз технічні засобів для розробки додатку та вибір оптимальних рішень для розробки додатку.
- На основі результатів виконаних досліджень розроблено додаток для зберігання та генерації паролів користувачів.

Упровадження розробленого застосунку підвищить надійність паролів користувачів та їх безпеку в інтернеті.

В роботі виконано аналіз існуючих застосунків для операційної системи Windows.

Проаналізовано можливість середовища розробки Visual Studio . Розроблено логіку додатку та зручність користування для користувачів.

Галузь використання – завдяки вільному доступу, додатком може користуватись будь-який користувач, який турбується про свою безпеку в інтернеті та хоче надійно зберігати свої паролі.

ЗМІСТ

СПИСОК ВИКОРИСТАНИХ СКОРОЧЕНЬ

XML – розширювана мова розмітки

IDE – інтегроване середовище розробки

UML – уніфікована мова моделювання

AES – Алгоритм шифрування даних

БД – база даних

СУБД – система управління базами даних

ВСТУП

Актуальність дослідження. На сьогоднішній день паролі набули широкого розповсюдження їх використовують для захисту облікових записів на сайтах, в соц-мережах онлайн банкінгу та великої кількості інших додатків та сервісів. Тому через постійний ріст загроз злому паролів їх складність зростає з кожним роком. Через необхідність у великій кількості надійних паролів, які складно запам’ятати користувачі зневажають безпекою створюючи слабкі паролі або використовують один пароль на декількох сайтах або зберігають паролі у незахищеному вигляді. Для вирішення цієї проблеми можна використовувати спеціальні технічні засоби для збереження та генерації паролів.

Мета роботи. Виходячи з актуальності засобів для збереження паролів, метою даної роботи є створення додатку для генерації паролів користувачів який буде генерувати надійні паролі та зберігати їх у захищеному вигляді.

Для виконання поставленої мети слід виконати наступні завдання:

1.Проаналізувати переваги та недоліки інших менеджерів паролів.

2. Розробити вимоги до нового додатку на основі аналізу переваг та недоліків існуючих додатків.

3. Проаналізувати технічні засоби, що використовуються для розробки та обрано необхідні для створення надійного додатку.

4. Спроектувати та розробити новий додаток на основі аналізу потреб користувачів.

Об'єкт дослідження – підвищення безпеки користувачів у всесвітній мережі інтернет.

Предмет дослідження – засоби для генерації та зберігання паролів користувачів.

Мета роботи – підвищення безпеки користування мережі інтернет шляхом розробки програмного забезпечення для генерації та зберігання паролів користувачів.

Методи дослідження – метод теорії інформації, метод оптимального управління, обробка та аналіз інформації.

Практичне значення одержаних результатів. Додаток який допоможе користувачами мережі інтернет підвищити надійність паролів.

Особистий внесок. Вдосконалено алгоритм збереження даних за допомогою алгоритму шифрування AES.

Результати роботи. Матеріали дипломного проекту можуть сприяти вдосконаленню алгоритмів збереження та генерації паролів а також підвищенню безпеки користувачів в мережі інтернет.

АНАЛІЗ ЗАХИСТУ ДАНИХ ЗА ДОПОМОГО ПАРОЛІВ

1.1 Використання паролів.

Паролі використовувалися з давніх часів. Ще в римській імперії вартові запитували пароль у бажаючих війти до території, що охороняється, після отримання відповіді дозволяли прохід тільки в тому випадку, якщо пароль було вказано вірно.

Пізніше паролі, які використовувалися у військових цілях, стали включати не тільки пароль, а і контр пароль; наприклад, в перші дні битви за Нормандію десантники 101-ї повітряно-десантної дивізії США використовували пароль - спалах - який представлявся як виклик і відповідав правильною відповіддю - гром. Виклик та відповідь міняли кожні три дні.

Паролі у комп’ютерних системах почали використовувати майже спочатку їх існування. Compatible Time-Sharing System (CTSS), операційна система, введена в MIT в 1961 році, вважається першою комп'ютерною системою з реалізацією входу в систему за допомого паролю. В CTSS була команда LOGIN, яка запитувала пароль користувача для входу в систему, потрібно було ввести вірний пароль.

На сьогоднішній день паролі набули широкого розповсюдження, вони використовуються для захисту операційних систем, систем платежів, почтових клієнтів, сайтів, соціальних та мереж.[1]

З поширенням паролів, як способів захисту та із збільшенням загроз злому паролів, були розроблені політики паролів.

Політика паролів це – набір правил, спрямованих на підвищення комп'ютерної безпеки, спонукаючи користувачів використовувати сильні паролі і використовувати їх належним чином. Політика паролів часто є частиною офіційних правил організації і вивчається в рамках навчання робітників з питань безпеки. Політика паролів може носити рекомендаційний характер, або комп'ютерні системи змушують користувачів її дотримуватися[2].

Типові компоненти політики паролів:

- Довжина пароля. Зазвичай це мінімальна кількість символів яку має містити пароль. Типовою довжиною пароля є мінімум 8 символів;
- Склад пароля. Зазвичай політики пропонують або змушують користувачів використовувати великі та малі літери, цифри, спеціальні символи. В деяких політиках забороняється використовувати для паролів особисті дані, номера телефонів, дати і так далі;
- Список не допустимих паролів. До таких списків входять паролі, які відповідають іншим пунктам політики, але не повинні використовуватися через те, що вони мають низький рівень надійності, наприклад, їх легко підібрати завдяки перебору або вони були розкриті раніше;
- Термін дії пароля. Деякі політики встановлюють обмеження часу придатності паролю.

Зазвичай політики паролів – це компроміс між безпекою і зручністю використання паролів. Чим складніше політика паролів, тим складніше її дотримуватися через те, що користувачеві складно запам'ятати або створити відповідний пароль.

Надійність паролю – це міра ефективності пароля проти атак методом підбору (метод «грубої сили»). У своїй звичайній формі він оцінює, скільки спроб в середньому потрібно зловмисникові, який не має прямого доступу до паролю, щоб правильно його вгадати. Надійність пароля залежить від довжини, складності та непередбачуваності. [3]

Атака грубою силою це – один з найпоширеніших і найпростіших методів атаки на паролі. При атаці методом грубої сили зловмисник систематично перевіряє всі можливі паролі і парольні фрази, поки не буде знайдений правильний.

У комп'ютерній індустрії прийнято визначати надійність пароля в термінах інформаційної ентропії, яка вимірюється в бітах і є концепцією теорії інформації. Замість кількості припущень, необхідних для точного знаходження пароля, наводиться логарифм цього числа за основою 2, який зазвичай називають кількістю

«біт ентропії» в паролі. Пароль з ентропією в 42 біта, обчисленої таким чином, буде таким же надійним, як рядок довжиною 42 біта, згенерований випадковим чином. Іншими словами, для пароля з ентропією 42 біта буде потрібно 2^{42}(4,398,046,511,104) спроб при методі «грубої сили». Таким чином, за рахунок збільшення ентропії пароля на один біт кількість необхідних спроб подвоюється, що вдвічі ускладнює завдання зловмисника. В середньому доведеться спробувати половину можливої кількості паролів, перш ніж знайдеться правильний.

Даний метод оцінки надійності паролів не можна використовувати для оцінки паролів створених людиною тому що люди не вміють досягати достатньої ентропії для створення надійних паролів. Відповідно до одного з досліджень в якому прийняли участь півмільйона користувачів, середня ентропія пароля була оцінена в 40,54 біта. [2] Таким чином, в одному аналізі більше 3 мільйонів восьмисимвольних паролів буква «e» використовувалася понад 1,5 мільйона разів, а буква «f» - лише 250 000 разів. При рівномірному розподілі кожен символ використовувався б приблизно 900 000 раз. Найчастіше використовується цифра «1», тоді як найбільш поширені букви - це a, e, o, i, r. [4] Користувачі рідко в повній мірі використовують великі набори символів при формуванні паролів. Наприклад, результати злому, отримані за допомогою фішингової схеми MySpace в 2006 році, виявили 34 000 паролів, з яких тільки 8,3% використовували змішаний регістр, числа і символи. Тому для злому паролів, які були створені людиною частіше використовують атаку по словнику.

Перебір за словником – це форма атаки грубою силою підчас якої перебираються не всі варіанти, а лише найбільш вірогідні. Список значень для атаки називається словником. Словник зазвичай формується із паролів, які були скомпрометовані або простих патерів, які часто використовуються користувачами.

1.2 Генератор випадкових паролів

Генератор випадкових паролів – це програмне забезпечення, програма або апаратний пристрій, який приймає вхідні дані від випадкового або псевдовипадкового генератора чисел і автоматично генерує пароль. Випадкові паролі можна згенерувати вручну, використовуючи прості джерела випадковості, такі як гральні кістки або монети, або вони можуть бути згенеровані за допомогою комп'ютера.

Генератори випадкових паролів зазвичай виводять рядок символів зазначеної довжини. Це можуть бути окремі символи з деякого набору символів, або слова з деякого списку слів, що утворюють парольну фразу. Програму можна налаштувати так, щоб отриманий пароль відповідав необхідній політиці паролів, наприклад, завжди створюючи поєднання цифр, букв і спеціальних символів. Така політика зазвичай трохи знижує силу генерованих паролів, тому що символи більше не вибираються незалежно.

Сила випадкового пароля від атаки грубою силою, може бути обчислена шляхом обчислення інформаційної ентропії випадкового процесу, який справив його. Якщо кожен символ в паролі створюється незалежно і з однаковою ймовірністю, ентропія в бітах визначається формулою:

$$H = L\log_2 N \tag{1.1}$$

Де H – інформаційна ентропія що виміряються у бітах; L – кількість можливих символів; N – кількість символів в паролі[2].

В таблиці 1.1 представлена ентропія на символ для різних наборів символів розрахована за допомогою формули 1.1.

Таблиця 1.1 – ентропія на символ для різних наборів символів

Набір символів	Кількість символів	Ентропія на символ
Арабські цифри (0-9)	10	3.322 біта
Шістнадцятирічні числа (0-9, A - F)	16	4.000 біта
Латинський алфавіт, чутливий до регістру (a–z або A–Z)	26	4.700 біта
Латинський алфавіт, чутливий до регістру (a – z, A – Z)	52	5.700 біта
Буквено-цифрові регістри, чутливі до регістру (a – z, A – Z, 0–9)	62	5.954 біта
Усі символи ASCII для друку, крім пробілу	94	6.555 біта
Усі розширені символи ASCII для друку	218	7.768 біта

Щоб знайти довжину L, необхідну для досягнення бажаної надійності H, з паролем, вибраним випадковим чином з набору з N символів, обчислюється:

$$L = \frac{H}{\log_2 N} \tag{1.2}$$

Де L – довжина пароля, H – надійність паролю, N – кількість доступних символів[2].

Мінімальна кількість біт ентропії, необхідних для пароля, залежить від моделі загрози для системи в якій він буде використовуватися. RFC 4086, «Randomness Requirements for Security», опублікований в червні 2005 р, представляє деякі приклади моделей загроз і способи обчислення бажаної ентропії для кожної з них. [5] Їх відповіді варіюються від 29 біт ентропії, необхідної, якщо очікуються тільки онлайн-атаки, і до 96 біт ентропії, необхідної для важливих криптографічних ключів, використовуваних в таких додатках, як шифрування, де пароль або ключ повинні бути безпечними протягом тривалого періоду часу.

1.3 Менеджери паролів

Менеджер паролів – це комп'ютерна програма, яка дозволяє користувачам зберігати, створювати і управляти своїми паролями для локальних додатків і онлайн - сервісів.[6]

Менеджер паролів допомагає в створенні складних паролів та зберіганні таких паролів в зашифрованому вигляді в базі даних.

Типи менеджерів паролів включають:

- локально встановлені програмні додатки
- онлайн-сервіси, доступ до яких здійснюється через портали веб-сайтів
- локально доступні апаратні пристрої, які служать ключами.

Залежно від типу використовуваного менеджера паролів і функціональності, пропонованої його розробниками, зашифрована база даних або зберігається локально на пристрої користувача, або зберігається віддалено через онлайн-службу файлового хостингу. Менеджери паролів зазвичай вимагають, щоб користувач згенерував і запам'ятав один «головний» пароль для розблокування і доступу до будь-якої інформації, що зберігається в їх базах даних. Багато додатків для управління паролями пропонують додаткові можливості, які підвищують зручність і безпеку, такі як зберігання інформації про кредитні картки, а також функція автозаповнення.

1.3.1 Локально встановлені програмні додатки

Зазвичай розміщуються на персональному комп'ютері або мобільному пристрої користувача, наприклад, на смартфонах, у вигляді локально встановленого програмного додатка. Ці додатки можуть працювати в автономному режимі, при цьому база даних паролів зберігається незалежно і локально на тому ж пристрої, що і програмне забезпечення менеджера паролів. В якості альтернативи менеджери паролів можуть пропонувати або вимагати хмарний підхід, при якому

база даних паролів залежить від онлайн-служби файлового хостингу і зберігається віддалено, але обробляється програмним забезпеченням для управління паролями, встановленим на пристрої користувача.

Деякі автономні додатки не потребують доступу до інтернету тому вони захищені від онлайн атак. В деякій мірі повністю автономний менеджер паролів більш безпечний, але менш зручний і функціональний, ніж онлайн-менеджер.

1.3.2 Веб-сервіси

Онлайн-менеджер паролів - це веб-сайт, на якому безпечно зберігаються дані для входу. Вони являють собою веб-версію більш звичайного настільного менеджера паролів.

Перевагами онлайн-менеджерів паролів перед настільними версіями є портативність (їх зазвичай можна використовувати на будь-якому комп'ютері з веб-браузером і підключенням до мережі без установки програмного забезпечення) і знижений ризик втрати паролів в результаті крадіжки або пошкодження одного ПК - хоча такий же ризик існує для сервера, який використовується для зберігання паролів користувачів.

Основними недоліками онлайн-менеджерів паролів є те, що дані користувача зберігаються в хмарі а не на комп’ютері користувача. Оскільки сервери і хмара є об'єктом кібератак. Знову ж, користувачі схильні обходити безпеку для зручності. Ще один важливий фактор – чи використовується одно або двостороннє шифрування.

Деякі онлайн-системи управління паролями, такі як Bitwarden, мають відкритий вихідний код, де вихідний код може піддаватися незалежному аудиту або розміщуватися на власному комп'ютері користувача, а не покладатися на хмару служби.

Використання веб-менеджера паролів є альтернативою методам єдиного входу, таким як OpenID або обліковий запис Microsoft Microsoft (раніше Microsoft Wallet, Microsoft Passport, .NET Passport, Microsoft Passport Network і Windows Live ID), або може служити тимчасовим заходом до знаходження кращого рішення.[7]

1.3.3 Апаратні пристрої

Менеджери паролів на основі токенів повинні мати механізм токенів безпеки, в якому локально доступний апаратний пристрій, такій як смарт-карти або захищені USB-флеш-пристрої, використовується для аутентифікації користувача замість або на додаток до традиційного тексту на основі пароля або іншої двохфакторної системи аутентифікації. Дані, що зберігаються в токені, зазвичай зашифровані, щоб запобігти зондування і несанкціонованого читання даних. Деяким системам токенів як і раніше потрібно програмне забезпечення, завантажене на ПК, а також обладнання (пристрій читання смарт-карт) і драйвери для правильного читання і декодування даних.[8]

Облікові дані захищені за допомогою токена безпеки, тому зазвичай пропонується багатофакторна аутентифікація шляхом об'єднання чогось, що є у користувача, наприклад мобільний додаток, який генерує токен, схожий на віртуальну смарт-карту.

Є кілька компаній, які виробляють спеціальні сторонні пристрої аутентифікації, одним з найпопулярніших є YubiKey. Але тільки деякі сторонні менеджери паролів можуть інтегруватися з цими апаратними пристроями. Хоча це може здатися проблемою, у більшості менеджерів паролів є інші прийнятні варіанти двоетапної перевірки, що інтегруються з такими додатками, як Google Authenticator і вбудованими генераторами TOTP. Хоча сторонні токен-пристрої корисні для підвищення безпеки, вони вважаються додатковими заходами безпеки і зручності і не вважаються необхідними і не критичними для правильного функціонування диспетчера паролів.

1.4 Огляд існуючих менеджерів паролів

Менеджер паролів Google – один з сервісів компанії Google який вбудований в браузер Google Chrome. Для використання потрібно мати обліковий запис Google. На рисунку 1.1 представлений інтерфейс цього додатку.

Крім збереження паролів цей веб додаток має наступні функції:

– Автозаповнення форм з паролями;
– Генерація паролів;
– Перевірка паролів користувачів на вразливості (виявлення слабких паролів та скомпрометованих паролів);
– Сповіщення користувачів якщо їх паролі були скомпрометовані;
– Синхронізація паролів між декількома пристроями.

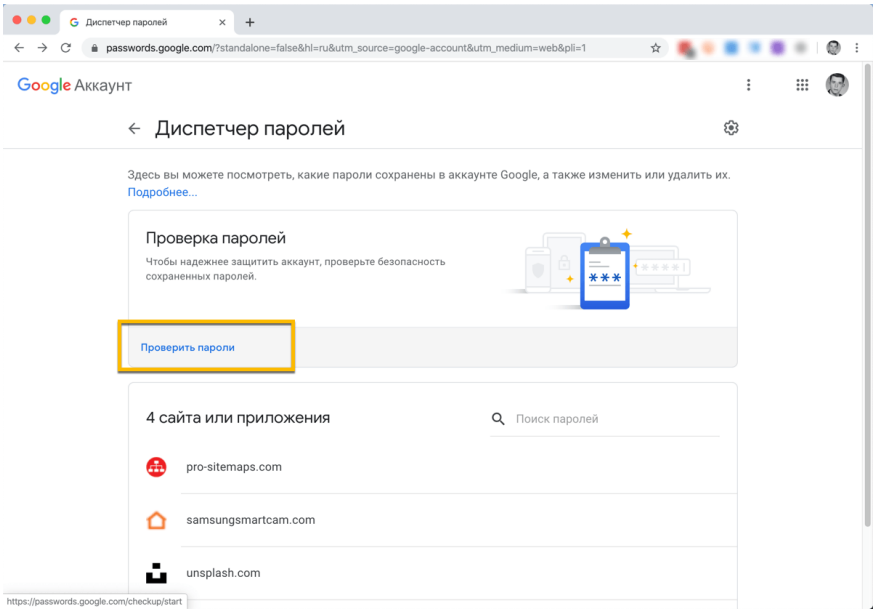


Рисунок 1.1 – Інтерфейс менеджеру паролів Google

Основними перевагами цього веб додатку є зручність, простота використання та доступність на різних платформах.

До недоліків можна віднести залежність від сервісів компанії Google, а також можливість використання цього сервісу лише в браузері Google Chrome. Іще одним недоліком можна вважати орієнтованість на зберігання лише паролів від сайтів та неможливість зберігати інші паролі наприклад від десктопних програм.

KeePass – це безкоштовний менеджер паролів з відкритим вихідним кодом, який допомагає вам безпечно керувати своїми паролями. Ви можете зберігати всі свої паролі в одній базі даних, яка заблокована майстер-ключем. Таким чином, вам

потрібно запам'ятати лише один головний ключ, щоб розблокувати всю базу даних. Інтерфейс програми KeePass представлено на Рис 1.2. [9]

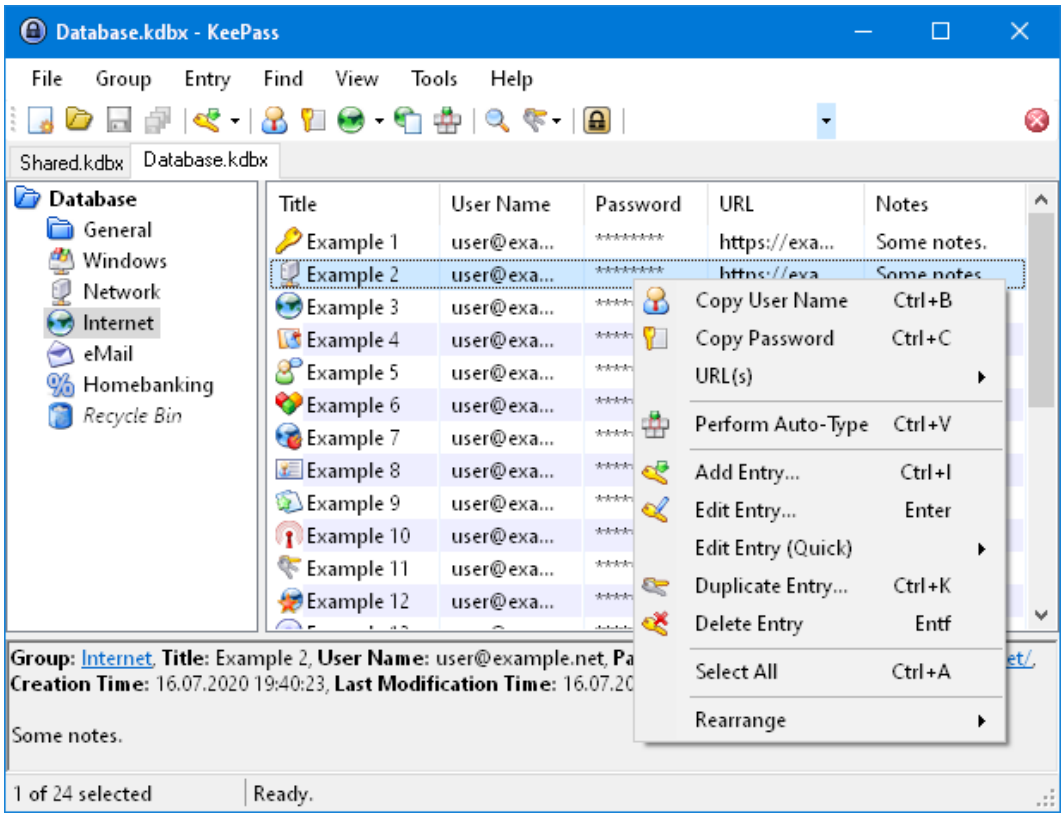


Рисунок 1.2 – Інтерфейс програми KeePass

Перевагами KeePass вважається підтримка багатьох мов та наявність додаткових функцій для безпеки таких як генератор паролів та нагадування для зміни майстер ключа від бази даних.

Серед недоліків додатку можна виділити проблеми з безпекою в програмі. Існує декілька не критичних вразливостей через які зловмисник може отримати доступ до паролів користувача.

LastPass це – менеджер паролів, який зберігає паролі в зашифрованому вигляді на своїх серверах. Стандартна версія LastPass поставляється з веб-інтерфейсом (рис 1.3), але також включає плагіни для різних веб-браузерів і додатків для багатьох смартфонів. LastPass має заповнювач форм, який автоматизує введення пароля і заповнення форм, а також підтримує генерацію

пароля, спільне використання сайту і введення журналу сайту, а також двухфакторную аутентифікацію. [10]

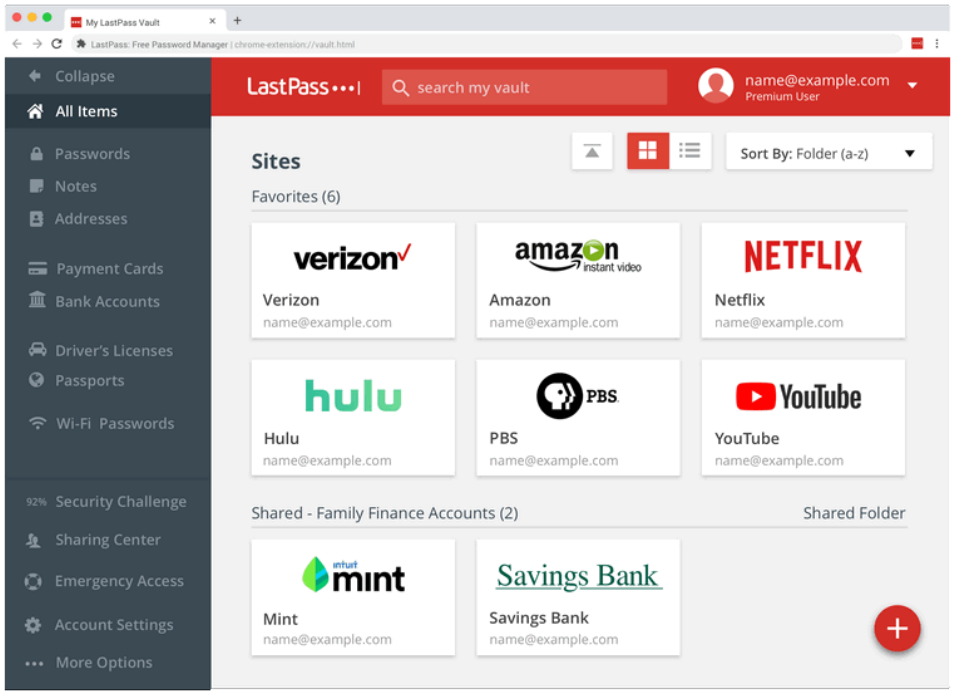


Рисунок 1.3 – Інтерфейс додатку LastPass

LastPass має багато додаткових функцій які допомагають підвищити безпеку даних користувача в інтернеті також до переваг можна віднести зручний інтерфейс програми.

Недоліками програми є неможливість зберігати дані локально. Всі дані користувачів зберігаються на серверах компанії, що вимагає підключення до інтернету без якого програмою не можна користуватися. В компанії було декілька інцидентів з безпекою, але перевірити справжній рівень захищеності не можливо так як всі дані користувачів зберігаються на серверах.

Аналіз програм аналогів показав, шо більшість проблем з безпекою виникають при передачі або збережені паролів в інтернеті. Також було виявлено, що деякі додатки мають ускладнений інтерфейс через наявність різних функцій, які напряму не відносяться до збереження паролів. Узагальнене порівняння переваг та недоліків представлено у таблиці 1.2.

Таблиця 1.2 – Порівняння аналогів.

Назва додатку	Основні переваги	Основні недоліки
Менеджер паролів Google	Легкий у використані та доступний.	Не можливість зберігання даних про паролі від десктопних додатків. Використання лише у якості розширення до браузера.
KeePass	Підтримка великої кількості мов. Можливість зберігання даних локально. Підтримка багатьох систем	Важка синхронізація між платформами. Проблеми з безпекою.
LastPass	Велика кількість функцій. Зручний інтерфейс користувача	Збереження даних користувача лише на серверах компанії. Проблеми з безпекою

1.5 Аналіз вимог до майбутнього програмного додатку

При розробці нового додатку було вирішено зробити акцент на простоті використання та портативності додатку. Створений менеджер паролів повинен зберігати паролі користувачів в зашифрованому виді, а також мати функціонал для генерації надійних паролів користувачів. Додаток повинен бути портативним, тобто легко переходити від однієї системи до іншої для цього додаток повинен бути невеликого розміру та містити в собі мінімальний функціонал який необхідний для зручного користування програмою.

Отже можна виділити наступні функціональні вимоги:

- Збереження даних про паролі користувачів;
- Редагування даних про паролі;

- Шифрування файлів користувачів;
- Генерація паролів користувачів;
- Підтримка декількох файлів для збереження паролів;
- Портативність.

В таблиці 1.3 представлено порівняння майбутнього додатку з аналогами.

Таблиця 1.3 – Порівняння додатку з аналогами

Назва властивості	Назва додатку			
	Менеджер паролів Google	KeePass	LastPass	Додаток що розробляється
Збереження даних	Локально та в хмарі.	Локально	В хмарі	Локально
Підтримка кількох сховищ	Один користувач одне сховище	Так	Один користувач одне сховище	Так
Тип додатку	Розширення до браузеру	Десктопний додаток	Веб сервіс	Портативний десктопний додаток
Генератор паролів	Без можливості вказувати параметри	Дозволяє генерувати паролі з потрібними параметрами	Дозволяє генерувати паролі з потрібними параметрами	Дозволяє генерувати паролі з потрібними параметрами

ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНІЧНИХ ЗАСОБІВ

2.1 Вибір засобів для розробки

2.1.1 Платформа .NET Framework

Платформа .NET Framework – це технологія, яка підтримує створення і виконання веб-служб і додатків Windows. Платформа .NET Framework складається з загальномовного середовища виконання (середовища CLR) і бібліотеки класів .NET Framework. Основою платформи .NET Framework є середовище CLR. Середовище виконання можна вважати агентом, який керує кодом під час виконання і надає основні служби, такі як управління пам'яттю, управління потоками і віддалену взаємодію. При цьому середовищем накладаються умови статичної типізації та інші види перевірки точності коду, що забезпечують безпеку і надійність. Фактично основним завданням середовища виконання є управління кодом. Бібліотека класів є комплексною об'єктно-орієнтованої колекцією повторно використовуваних типів, які застосовуються для розробки додатків - починаючи з звичайних додатків, що запускаються з командного рядка, і додатків з графічним інтерфейсом (GUI) і закінчуючи додатками, що використовують останні технологічні можливості ASP.NET, такі як веб-форми і веб-служби XML.[11]

Середовище CLR управляє пам'яттю, виконанням потоків, виконанням коду, перевіркою безпеки коду, компіляцією і іншими системними службами. Ці кошти є внутрішніми для керованого коду, який виконується в середовищі CLR.

Бібліотека класів платформи .NET Framework являє собою колекцію типів, які тісно інтегруються із середовищем CLR. Бібліотека класів є об'єктно-орієнтованою. Вона надає типи, від яких керований код користувача може успадковувати функції. Це не тільки спрощує роботу з типами .NET Framework, але і скорочує час вивчення нових засобів платформи .NET Framework. Крім того, компоненти незалежних виробників можна легко поєднувати з класами платформи .NET Framework.

Для розробки програмного додатку під Windows було вирішено обрати саме платформу .NET Framework тому що дана платформа має велику бібліотеку класів та підтримку більшості продуктів компанії Microsoft у тому числі й Windows.[12]

2.1.2 Вибір мови програмування

В платформу .NET Framework вбудовані такі мови програмування:

- C#;
- F#;
- J#;
- VB.NET;
- JScript .NET;
- C++/CLI.

Серед них можна виділити дві мови програмування C# та C ++/CLI .

C++ це мова програмування загального призначення, створена Б'ярном Страуструпом як розширення мови програмування C. Згодом мова значно розширилася, і сучасний C++ тепер має об'єктно-орієнтовані, універсальні і функціональні можливості додатково до засобів для низькорівневого маніпулювання пам'яттю.

C++/CLI – це мовна специфікація, створена Microsoft, яка замінює Managed Extensions для C++. Це повна версія, яка спрощує застарілий синтаксис C++ і забезпечує взаємодію з мовами Microsoft .NET, такими як C#. C++/CLI був стандартизований Ecma як ECMA-372. В даний час він доступний в Visual Studio 2005, 2008, 2010, 2012, 2013, 2015, 2017 і 2019 років, включаючи випуски Express.

C# – це об'єктно-орієнтована мова програмування. Розроблено в 1998-2001 роках групою інженерів компанії Microsoft під керівництвом Андерса Хейлсберг і Скотта Вільтаумота як мову розробки додатків для платформи Microsoft .NET Framework.

C# відноситься до сім'ї мов з C-подібним синтаксисом, з них його синтаксис найбільш близький до C++ і Java. Мова має статичну типізацію, підтримує

поліморфізм, перевантаження операторів (в тому числі операторів явного і неявного приведення типу), делегати, атрибути, події, змінні, властивості, узагальнені типи і методи, ітератори, анонімні функції з підтримкою замикань, LINQ, виключення, коментарі в форматі XML.[13]

2.1.2.1 Порівняння C# та C++/CLI

Порівняння двох мов можна почати з продуктивності роботи на обраній платформі. За допомого C++ можна досягти більшої продуктивності та швидкості виконання за рахунок написання більш низкорівневого коду. При написані програми на C# важко досягти такого рівня продуктивності.

Можливості для розробки для платформи Windows вибрані мови програмування мають доступ до бібліотек .NET Framework. Тому можна сказати, що можливості для розробки в них рівні.

Зручність розробки. C# краще підходить для швидкої розробки під Windows за рахунок синтаксису та об’єкто орієнтованості. Розробка на C#, особливо на початку розробки, швидше ніж на C++.

2.1.2.2 Обґрунтування вибору

Проаналізувавши порівняння C# та C++/CLI було вирішено обрати C# оскільки він краще підходить для розробки додатків для платформи Windows.

До інших переваг даної мови програмування можна віднести:

- Доступність всіх необхідних інструментів;
- Автоматичний сбірник мусора;
- Підтримка великої кількості бібліотек для роботи з платформою Windows.

2.1.3 Вибір інструментів для розробки інтерфейсу користувача

Для розробки інтерфейсу на платформі .NET Framework стояв вибір між двома технологіями Windows Forms та WPF.

Windows Forms – інтерфейс програмування додатків (API), відповідальний за графічний інтерфейс користувача і є частиною Microsoft .NET Framework. Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за допомогою створення обгортки для Win32 API в керованому коді.

Але Windows Forms на сьогоднішній день вважаться застарілою технологією їй на заміну прийшла технологія WPF.

Windows Presentation Foundation (WPF) – графічна підсистема (аналог WinForms), яка починаючи з .NET Framework 3.0 в складі цієї платформи. Має пряме відношення до XAML. WPF разом з .NET Framework 3.0 вбудована в Windows Vista, а також доступна для установки в Windows XP Service Pack 2 і Windows Server 2003.[14]

Це перше реальне оновлення технологічного середовища призначеного для інтерфейсу користувача з часу випуску Windows 95. Воно включає нове ядро для заміни GDI і GDI+, використовувані в Windows Forms. WPF є високорівневим об'єктно-орієнтованим функціональним шаром, що дозволяє створювати двовимірні та тривимірні інтерфейси.

Отже, серед двох технологій було обрано WPF, оскільки це більш нова технологія, яка розвивається по сьогоднішній день. До інших переваг WPF відносяться:

- можливість ефективно відокремити призначений для користувача інтерфейс від логіки;
- вбудована функція розкадровки та моделювання анімації;
- прив'язка даних набагато краще, ніж з додатком WinForms;
- дозволяє обробляти великі набори даних, оскільки має вбудовану функцію «віртуалізації призначеного для користувача інтерфейсу»;
- пропонує шаблони даних і елементів управління, які забезпечують гнучке моделювання інтерфейсу користувача на моделях даних;
- підтримує тривимірну графіку, щоб інтерфейс користувача виглядав по-справжньому особливим;

– підтримує різні типи мультимедіа, такі як відео, 3D-контент і анімацію.

2.1.4 Середовище розробки

Найкращим середовищем розробки для платформи .NET Framework можна вважати Microsoft Visual Studio тому що компанія Microsoft розробляла його для розробки продуктів для своїх платформ.

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включає інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних інструментів. Дані продукти дозволяють розробляти як консольні додатки, так і ігри та програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Решта вбудованих інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).[15]

2.1.5 Система контролю версій.

Системи контролю версій – це категорія програмних інструментів, які допомагають записувати зміни, внесені у файли, шляхом відстеження змін, внесених в код.

В якості системи контролю версій було обрано Git, а для збереження вихідного коду проекту було обрано сервіс GitHub.

Git – це безкоштовна розподілена система керування версіями з відкритим вихідним кодом, призначена для швидкої і ефективної обробки всього, від невеликих до дуже великих проектів. Git простий в освоєнні, займає мало місця і має швидку продуктивність.[16]

GitHub, Inc. – провайдер інтернет-хостингу для розробки програмного забезпечення і контролю версій за допомогою Git. Він пропонує розподілену систему контролю версій і управління вихідного коду функції (SCM) в Git. А також надає ряд функцій для контролю своїх проектів та спільної розробки.

2.2 Вибір засобів для збереження даних

2.2.1 Вибір сховища для даних

Серед доступних варіантів збереження даних розглядалися два варіанти: реляційна база даних та файл формату XML.

База даних являє собою організовану сукупність даних, які, як правило, зберігаються і доступні в електронному вигляді. Там, де бази даних більш складні, вони часто розробляються з використанням формальних методів проектування і моделювання.

Система управління базами даних (СУБД) - це програмне забезпечення, яке взаємодіє з кінцевими користувачами, додатками і самою базою даних для збору і аналізу даних. Програмне забезпечення СУБД додатково включає в себе основні засоби, що надаються для адміністрування бази даних. Загальна сума бази даних, СУБД і пов'язаних додатків може називатися «системою баз даних». Часто термін

«база даних» також використовується для загального позначення будь-якої СУБД, системи бази даних або програми, пов'язаною з базою даних.

Вчені-інформатики можуть класифікувати системи управління базами даних відповідно до підтримуваних ними моделей баз даних. Реляційні бази даних стали домінуючими в 1980-х роках. Ці моделі представлені у вигляді рядків і стовпців в серії таблиць, і в переважній більшості використовується SQL для запису і запиту даних. У 2000-х роках стали популярними нереляційні бази даних, що отримали назву NoSQL, тому що вони використовують різні мови запитів.

Extensible Markup Language (XML) – це мова розмітки, яка визначає набір правил для кодування документів в форматі, який зручний для читання людиною і комп'ютером.

Цілі розробки XML підкреслюють простоту, універсальність і зручність використання. Це текстовий формат даних з сильною підтримкою Unicode для різних людських мов. Хоча дизайн XML орієнтований на документи, ця мова широко використовується для представлення довільних структур даних, наприклад, використовуваних в веб-сервісах.

Існує кілька систем схем, які допомагають у визначенні мов на основі XML, в той час як програмісти розробили безліч інтерфейсів прикладного програмування (API) для допомоги в обробці даних XML.

Оскільки, однією із вимог є підтримка багатьох файлів з даними перевагу було надано XML, тому що на відміну від класичної бази даних він дозволить створювати безліч різних файлів з даними користувачів.[17]

2.2.2 Алгоритм шифрування

На сьогодні одним із най надійніших алгоритмів шифрування AES – 256.

Advanced Encryption Standard (AES), також відомий під своїм початковим назвою Rijndael є специфікацією для шифрування електронних даних, встановлених в США Національним інститутом стандартів і технологій в 2001.

AES – це підмножина блочного шифру Rijndael, розроблена двома бельгійськими криптографами, Вінсентом Рейменом і Джоан Деем. Rijndael - це

сімейство шифрів з різними ключами і розмірами блоків. Для AES NIST вибрав трьох членів сімейства Rijndael, кожен з розміром блоку 128 біт, але з трьома різними довжинами ключів: 128, 192 і 256 біт.

AES був прийнятий урядом США. Як стандарт шифрування даних (DES), який був опублікований в 1977 році. Алгоритм, описаний AES, є алгоритмом з симетричним ключем, що означає, що один і той же ключ використовується як для шифрування, так і для дешифрування даних.

AES включений в стандарт ISO / IEC 18033-3. AES вступив в силу в якості стандарту федерального уряду США 26 травня 2002 року після затвердження міністром торгівлі США. AES доступний в багатьох різних пакетах шифрування і є першим (і єдиним) загальнодоступним шифром, схваленим Агентством національної безпеки США (NSA) для надсекретної інформації при використанні в схваленому NSA криптографічному модулі (див. Безпека AES нижче).

AES заснований на принципі проектування, відомому як мережа заміщення-перестановки, ефективний як в програмному, так і в апаратному забезпеченні. На відміну від свого попередника DES, AES не використовує мережу Фейстеля. AES - це варіант Rijndael з фіксованим розміром блоку 128 біт і розміром ключа 128, 192 або 256 біт. Навпаки, Rijndael як такої визначається з розмірами блоків і ключів, які можуть бути будь-яким кратним 32 бітам, мінімум 128 і максимум 256 біт.

ПРОЕКТУВАННЯ ТА РОЗРОБКА ДОДАТКУ

3.1 Проектування додатку

Unified Modeling Language (UML) є універсальною мовою моделювання в області програмної інженерії, яка призначена для забезпечення стандартного способу візуалізації конструкцій системи.

Спочатку створення UML було мотивоване бажанням стандартизувати розрізнені системи позначень і підходи до проектування програмного забезпечення. Він був розроблений в Rational Software в 1994-1995 рр., А їх подальша розробка велася до 1996 р.

У 1997 році UML був прийнятий в якості стандарту Object Management Group (OMG) і з тих пір знаходиться під управлінням цієї організації. У 2005 році UML був також опублікований Міжнародною організацією зі стандартизації (ISO) в якості затвердженого стандарту ISO. З тих пір стандарт періодично переглядався, щоб охопити останню версію UML.[18]

3.1.1 Діаграма класів

У програмній інженерії, діаграма класів в Unified Modeling Language (UML) є тип діаграми статичної структури, яка описує структуру системи, показуючи системи класів, їх атрибути, операції (або методи), і відносини між об'єктами.

Діаграма класів є основним блоком об'єктно-орієнтованого моделювання. Вона використовується для загального концептуального моделювання структури додатка. Діаграми класів також можна використовувати для моделювання даних. Класи на діаграмі класів представляють як основні елементи, взаємодії в додатку, так і класи, які потрібно запрограмувати.

На діаграмі класів(рис 3.1) зображено чотири класи:

- FileEncryptor – клас відповідає за шифрування та збереження файлів;
- PasswordConteiner – клас працює з пралями користувачів;

– Password – клас з даними про пароль;
– PasswordGenerator – клас використовується для генерації паролів.

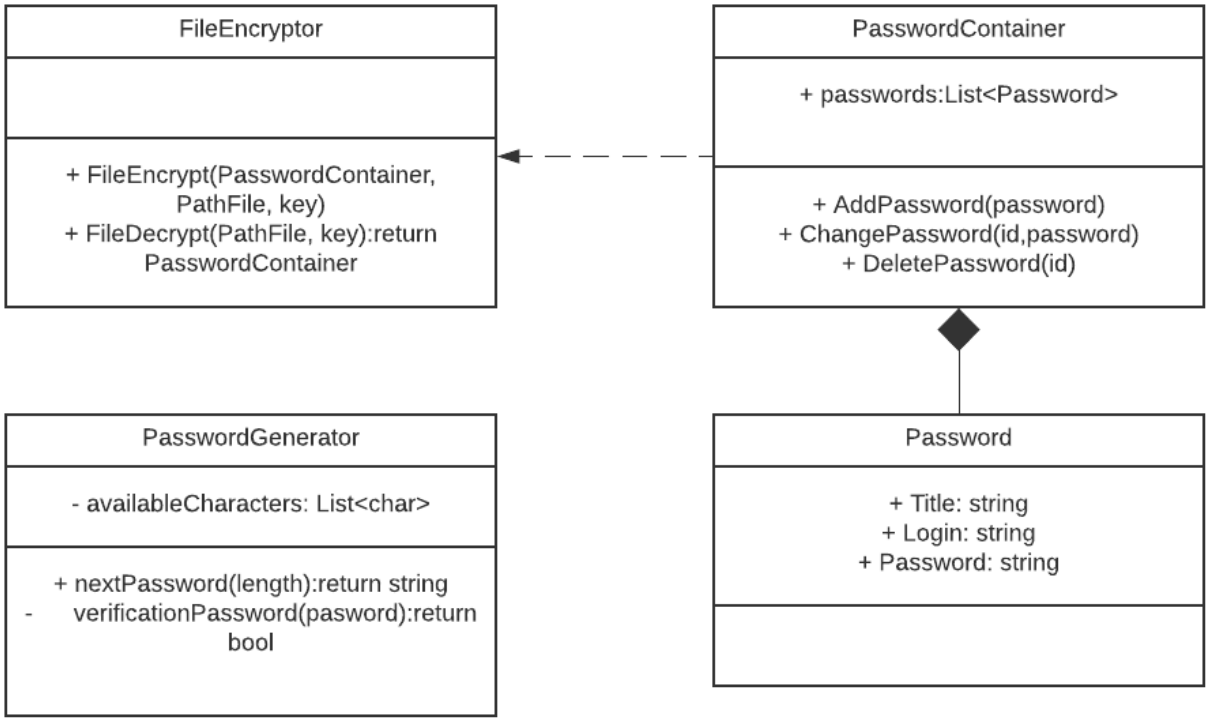


Рисунок 3.1 – Діаграма класів

3.1.2 Діаграма варіантів використання

Діаграма прецедентів є графічним зображенням можливих взаємодій користувача з системою. Діаграма варіантів використання показує різні варіанти використання і різні типи користувачів, які є в системі, і часто супроводжується діаграмами інших типів. Варіанти використання представлені гуртками або еліпсами. Акторів часто зображують у вигляді фігурок.

На діаграмі (рис 3.2) варіантів використання показано, які можливості доступні користувачу при використані додатку:

– створення нового файлу;
– відкриття файлу;
– редагування файлу з паролями;
– генерація випадкового паролю.

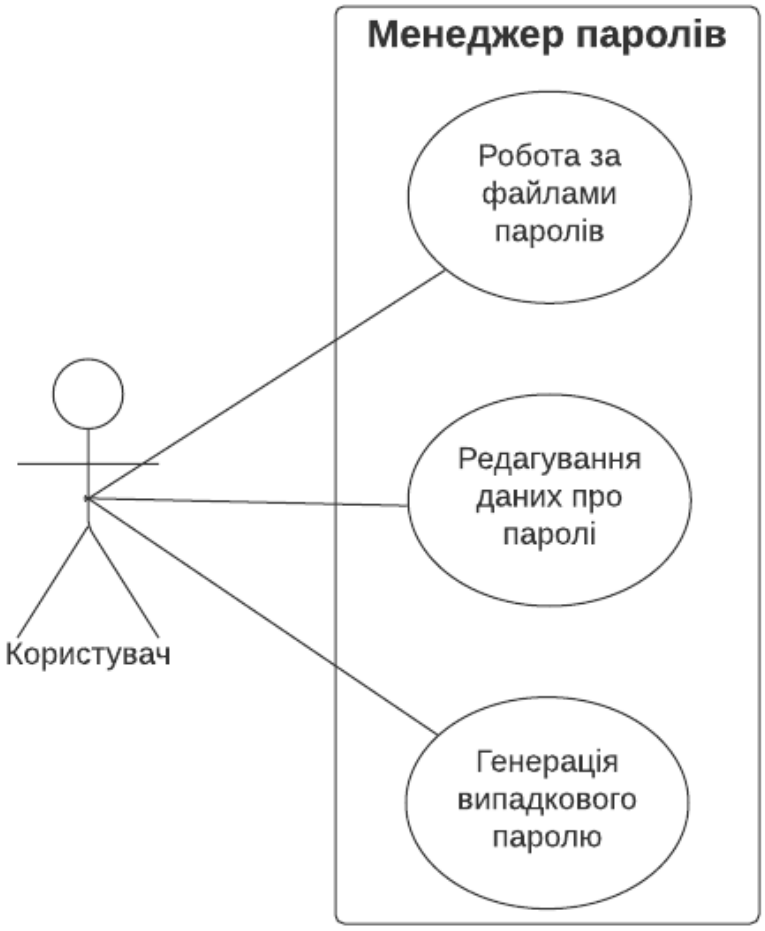


Рисунок 3.2 – Діаграма варіантів використання

3.1.3 Діаграма діяльності

Діаграми діяльності – це графічне представлення робочих процесів у додатку. Діаграми дій призначені для моделювання як обчислювальних, так і організаційних процесів, а також потоків даних, що супроводжують деякі дії. Хоча діаграми дій в основному показують загальний потік управління, вони також можуть включати елементи, що показують потік даних між діями через одне або декілька сховищ даних.

На діаграмі діяльності Рис 3.3 показано в якому порядку виношуються дії з файлом паролів користувача а саме:

1. Відкриття або створення файлу з паролями;
2. Редагування файлу (створення нового паролю або редагування вже існуючих);
3. Збереження файлу за паролями.

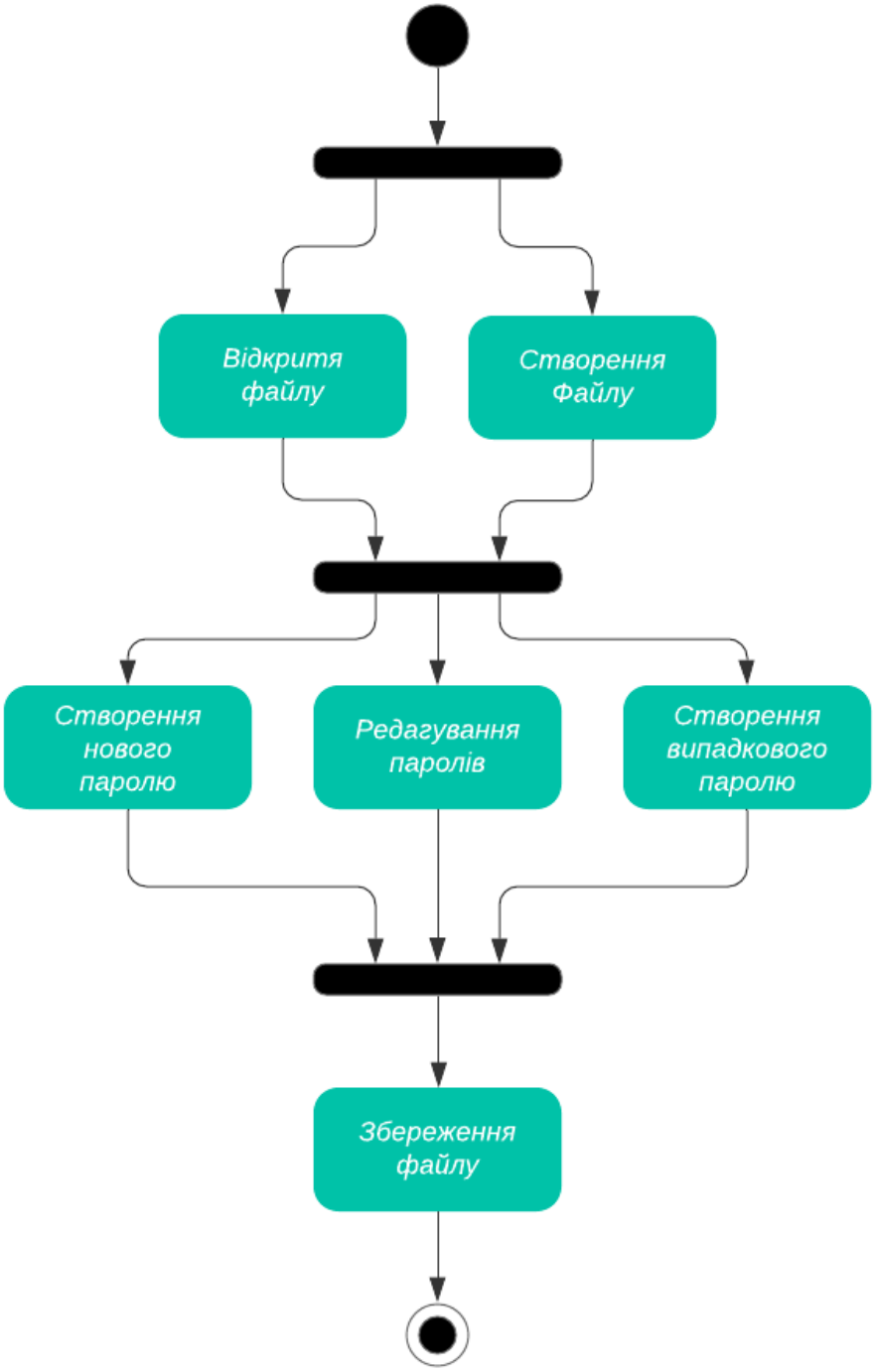


Рисунок 3.3 – Діаграма діяльності

3.2 Розробка додатку

3.2.1 Розробка інтерфейсу програми

Інтерфейс програми було розроблено за допомогою Редактор коду XAML в інтегрованому середовищі розробки Visual Studio містить всі інструменти,

необхідні для створення додатків WPF і UWP для платформи Windows. Редагувати дизайн вікон програми можна як за допомогою коду так і взаємодіючи з візуальним редактором.

При розробці додатку було створено наступні вікна:

– головне вікно програми.
– вікно редагування паролю.
– вікно відкриття файлу.
– вікно створення файлу.
– вікно довідки.

Для перегляду всіх паролів користувача та взаємодії з файлами було створено головне вікно програми рис 3.4. На цьому віні було створено наступні елементи:

– меню через яке користувач може взаємодіяти з файлами та також відкрити довідку по програмі.
– панель для перегляду всіх користувацьких паролів.
– поля для перегляду обраного паролю.
– кнопки для редагування обраного паролю.

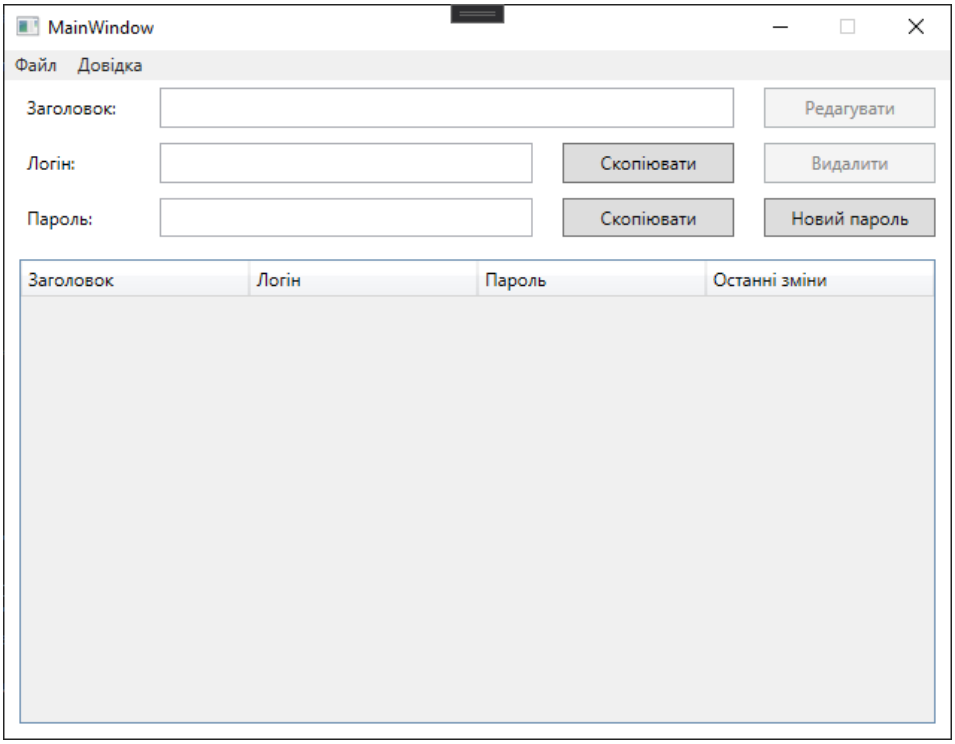


Рисунок 3.4 – Головне вікно програми

Вікно редагування паролю (рис 3.5) будо створено для того щоб користувач міг створювати та змінювати паролі. На нього були додані наступні елементи:

– поля для даних про пароль.
– кнопки управління (зберегти та скасувати).
– текстові поля для відображення інформації про силу обраного паролю.
– кнопка для генерації випадкового паролю.

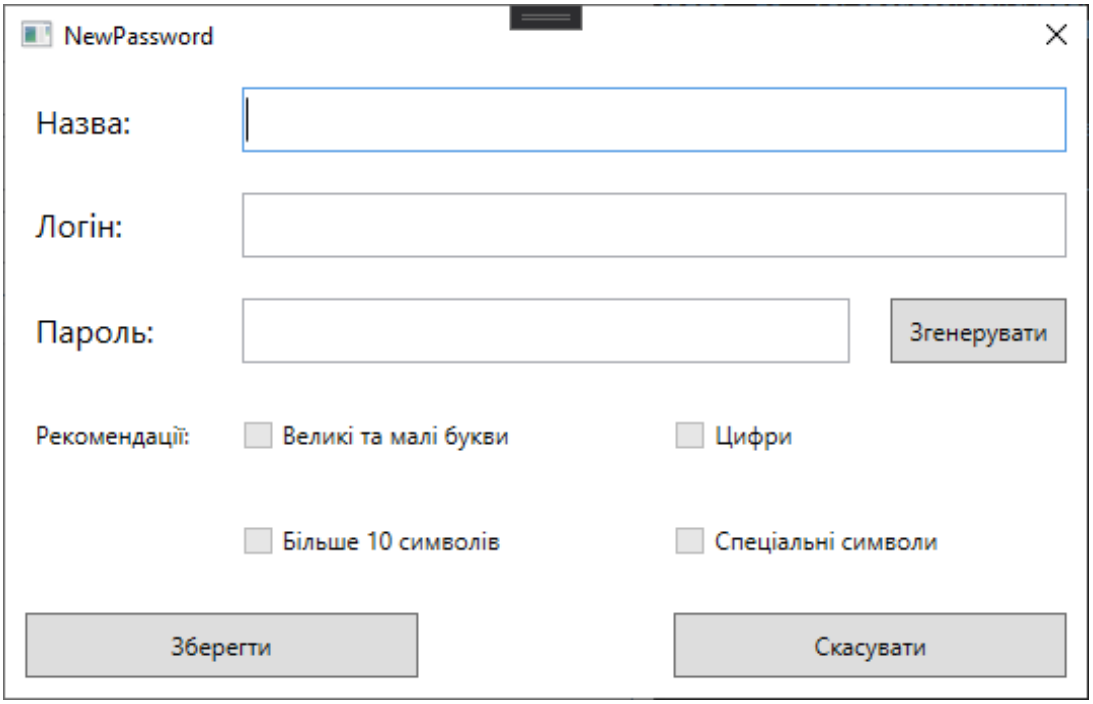


Рисунок 3.5 – Вікно редагування паролю

Вікна створення файлу (Рис. 3.6) та відкриття файлу використовується для роботи з файлами.

Вікно створення файлу містить наступні елементи:

– поле для введення ключа від файлу.
– текстове поле для відображення шляху розташування файлу.
– кнопки управління (скасувати та створити).

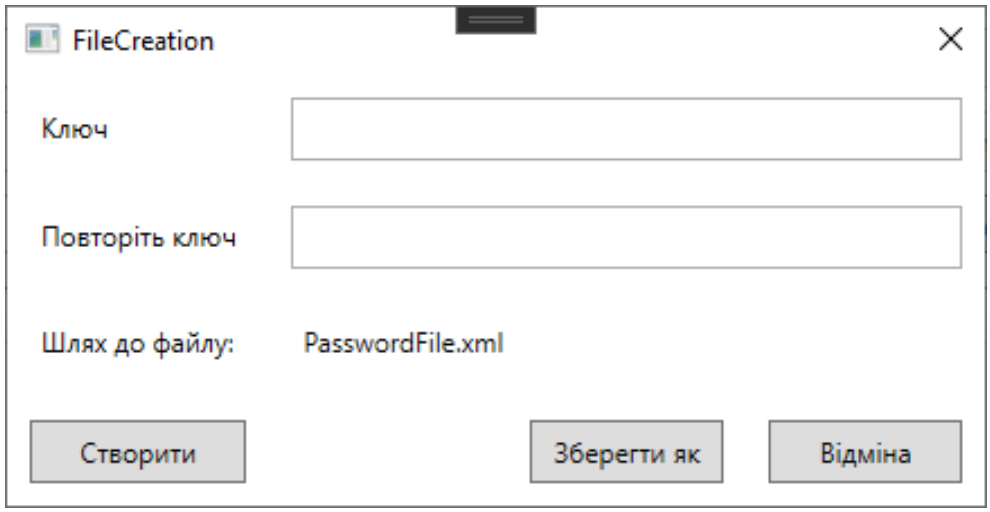


Рисунок 3.6 – Вікно створення файлу

Вікно відкриття файлу (Рис. 3.7) містить наступні елементи:

– поля для створення ключа від файлу.
– текстове поле для відображення шляху розташування файлу.
– кнопки управління (скасувати та створити).
– кнопка для вибору місця для збереження файлу (зберегти як).

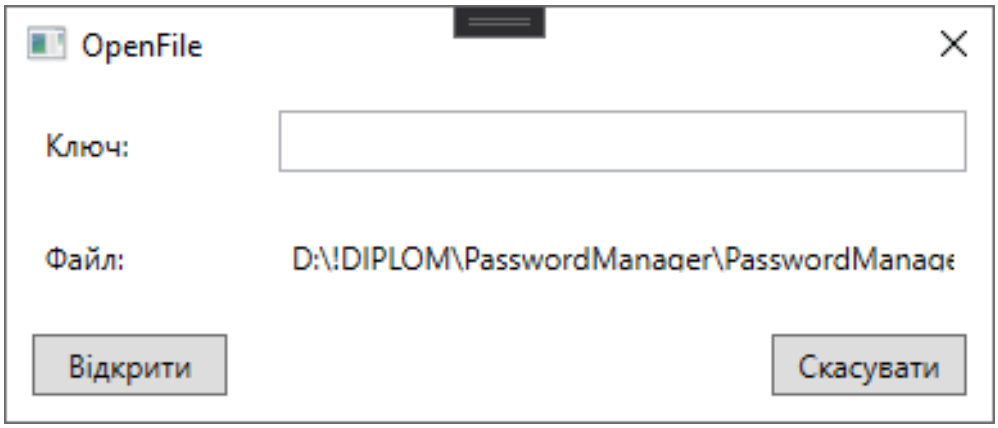


Рисунок 3.7 – Вікно відкриття файлу

Вікно довідки містить в собі інформацію про розробника та його контакні дані зображено на рисунку 3.8

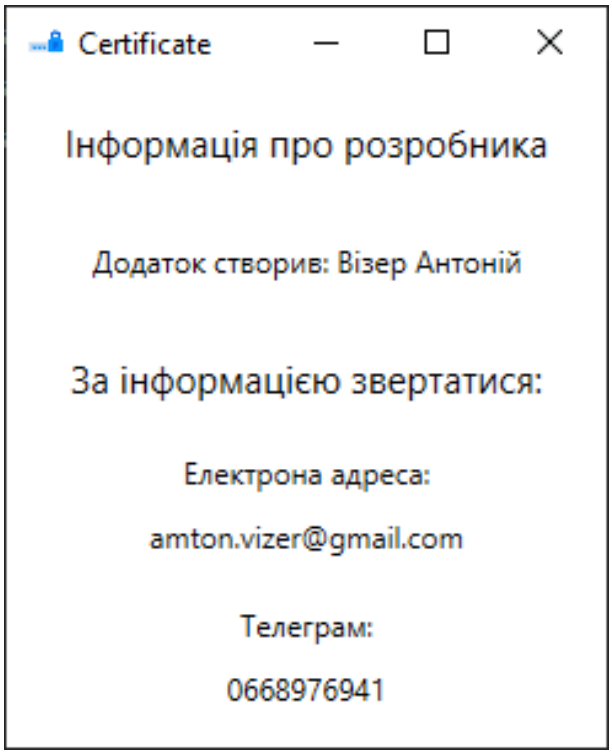


Рисунок 3.8 – Вікно довідки

3.2.2 Зберігання та захист паролів

Для зберігання паролів використано файл формату XML які зашифровані за допомогою алгоритму AES. Формування файлів XML виконується за допомого серіалізації для чого використовується вбудований в .NET Framework клас XmlSerializer. Після чого дані в форматі XML шифруються та записуються в файл. За цей процес відповідає клас FileEncryptor.

На рисунку 3.8 показано код методу fileEncrypt, який записує дані класу PasswordContainer в існуючий або створений файл. Для створення нового зашифрованого файлу виконуються такі дії:

1. Отримуємо від користувача ключ для нового файлу;
2. Отримуємо від користувача шлях для збереження файлу;
3. Створюємо новий екземпляр класу PasswordConteiner;
4. Створюємо екземпляр класу XmlSerializer;
5. Створюємо потік для роботи з файлами;
6. Створюємо екземпляр класу Aes;
7. Встановлюємо ключ шифрування для екземпляру класу Aes;

8. Створюємо потік шифрування та запису;
9. Використовуючи раніше створені класи серіалізуємо та записуємо дані в файл за вказаним адресом.

```
public static void fileEncrypt( PasswordContainer passwordContainer,string PathFile, byte[] key)
{

    XmlSerializer formatter = new XmlSerializer(typeof(PasswordContainer));

    using (FileStream fileStream = new FileStream(PathFile, FileMode.OpenOrCreate))
    {
        using (Aes aes = Aes.Create())
        {
            aes.Key = key;

            byte[] iv = aes.IV;
            fileStream.Write(iv, 0, iv.Length);


            using (CryptoStream cryptoStream = new CryptoStream(fileStream, aes.CreateEncryptor(), CryptoStreamMode.Write))
            {
                using (StreamWriter encryptWriter = new StreamWriter(cryptoStream))
                {
                    formatter.Serialize(encryptWriter, passwordContainer);
                }
            }

        }
    }
}
```

Рисунок 3.8 – Код методу fileEncrypt класу FileEncryptor

Для зчитування й декодування файлу використовується метод fileDecrypt який з файлу отримує дані про екземпляр класу PasswordContainer (Рис 3.9). Для зчитування й декодування зашифрованого файлу виконуються такі дії:

1. Отримуємо від користувача ключ для файлу;
2. Отримуємо від користувача шлях до файлу;
3. Створюємо екземпляр класу XmlSerializer;
4. Створюємо потік для роботи з файлами;
5. Створюємо екземпляр класу Aes;
6. Встановлюємо ключ шифрування для екземпляру класу Aes;
7. Створюємо потік декодування та зчитування файлів;
8. Використовуючи раніше створені класи десеріалізуємо та повертаємо дані з файлу у вигляді класу PasswordContainer.

```
public static PasswordContainer fileDecrypt(string PathFile, byte[] key)
{


    XmlSerializer formatter = new XmlSerializer(typeof(PasswordContainer));

    using (FileStream fileStream = new FileStream(PathFile, FileMode.Open))
    {
        using (Aes aes = Aes.Create())
        {
            byte[] iv = new byte[aes.IV.Length];
            int numBytesToRead = aes.IV.Length;
            int numBytesRead = 0;
            while (numBytesToRead > 0)
            {
                int n = fileStream.Read(iv, numBytesRead, numBytesToRead);
                if (n == 0) break;

                numBytesRead += n;
                numBytesToRead -= n;
            }
            using (CryptoStream cryptoStream = new CryptoStream(fileStream, aes.CreateDecryptor(key, iv),CryptoStreamMode.Read))
            {
                using (StreamReader decryptReader = new StreamReader(cryptoStream))
                {
                    return (PasswordContainer)formatter.Deserialize(decryptReader);
                }
            }
        }
    }
}
```

Рисунок 3.9 - Код методу fileDecrypt класу FileEncryptor

Фали з паролями зберігаються у форматі XML документу у зашифрованому вигляді якщо розшифровувати файл він має наступний вигляд рисунок 3.10. В файлі зберігаються у вигляді блоків даних з інформацією про паролі шо включає:

– Назву сайту або програми від паролю
– Логін
– Пароль
– Дата та час остатніх змін.

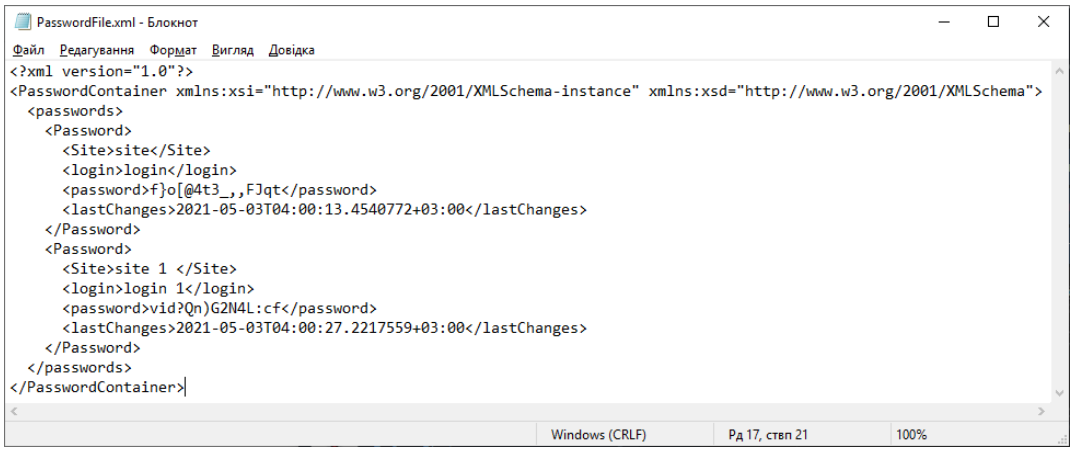


Рисунок 3.10 – Розшифрований файл з паролями.

3.2.3 Генерація паролів

Для генерації надійних паролів було створено клас PasswordGenerator. Цей клас містить в собі всі необхідні елементи для генерації паролів. Він складається з списку всіх доступних символів, які використовуються при генерації та налаштування які символи обов’язково повні бути присутні в паролі.

Щоб згенерувати пароль спочатку потрібно створити екземпляр класу PasswordGenerator та вказати які символи будуть присутні в згенерований цим класом паролях в конструкторі за замовчуванням використовуються всі символи латинського алфавіту великого та малого регістрів а також цифри та всі спеціальні символи доступні для друку. На рисунку 3.11 показано конструктор за замовчуванням класу PasswordGenerator.

```
PasswordGenerator()
{

    availableCharacters = new List<char>();

    for (char i = '!'; i <= '~'; i++)
    {
        availableCharacters.Add(i);
    }

    availabilityCapitalLetter = true;
    availabilityNumeral = true;
    availabilitySpecialSymbols = true;
}
```

Рисунок 3.11 – Конструктор за замовчуванням класу PasswordGenerator.

Після створення екземпляру класу за допомого методу nextPassword (рис 3.12) можна генерувати паролі заданої довжини. При генерації паролю випадково обираються символи з списку доступних символів потім перевіряються на наявність всіх необхідних символів в згенерованому паролі.

```
public string nextPassword(int length)
{
    string password = "";

    Random random = new Random();

    do
    {
        password = "";
        for (int i = 0; i < length; i++)
        {
            password += availableCharacters[random.Next(0,availableCharacters.Count)];
        }
    } while (!verificationPassword(password));

    return password;
}
```

Рисунок 3.12 – Конструктор за замовчуванням класу PasswordGenerator.

3.2.4 Взаємодія між вікнами програми.

При переході від одного вікна до іншого створюється екземпляр класу нового вікна та встановлюються необхідні параметри вікна. Далі визивається метод ShowDialog що виводить на екран нове вікно в режимі діалогу. Після завершення роботи вікна програма аналізує відповідь методу ShowDialog та виконує відповідні дії. На рисунку 3.13 показано взаємодію одного вікна з іншим на прикладі створення нового паролю.

```
private void button_NewPassword_Click(object sender, RoutedEventArgs e)
{
    var NewPasswordWindow = new NewPassword();

    Nullable<bool> result = NewPasswordWindow.ShowDialog();

    if(result == true)
    {
        filePassword.PasswordData.passwords.Add(NewPasswordWindow.password);

    }

    UpdateWindow();

}
```

Рисунок 3.13 – Відкриття вікна створення нового паролю.

3.3 Тестування програмного забезпечення

Тест-кейс - це специфікація вхідних даних, умов виконання, процедури тестування та очікуваних результатів, які визначають один тест, який повинен бути виконаний для досягнення конкретної мети тестування програмного забезпечення, наприклад, для перевірки певних функцій програми або для перевірки на відповідність певним вимогам. Для отримання бажаного покриття тестування програмного забезпечення можна створити набір тест-кейсів.[19]

Таблиця 3.1 –Тест-кейси для тестування функціоналу головного меню

Функціонал головного меню		
Передумова: вхід до головного меню, наявність хоча б одного поля з паролем		
Крок тесту	Дані тесту	Очікуваний результат
Збереження змін у файлі паролей	Натиснення на пункт меню «Файл» «Зберегти»	Збереження змін у файлі паролей
Створення нового запису в файлі паролей	Натиснення на кнопку «Новий пароль»	Відкриття вікна створення нових паролей
Редагування запису в файлі паролей	Натиснення на кнопку «Редагувати пароль»	Відкриття вікна редагування паролей
Видалення запису з файлу паролей	- Обрати рядок з необхідним записом - Натиснути кнопку «Видалити»	Видалення запису у файлі паролей
Копіювання необхідних даних	Натиснути одну з кнопок «Скопіювати»	Текст скопійовано до буферу обміну

Таблиця 3.2 – Тест-кейси для тестування створення нового файлу паролів

Функціонал створення нового файлу паролів		
Передумова: Відкрити вікно створення нового файлу паролів		
Крок тесту	Дані тесту	Очікуваний результат
Введення коректних значень без вказання шляху збереження файлу	- В ведення ключа - Повторне ведення ключа - Натиснути кнопку «Створити»	- Створення нового файлу паролів - Відкриття файлу в головному меню програми
В ведення коректних значень з вказанням шляху збереження файлу	- Ввести ім’я користувача - Ввести електрону пошту - Ввести пароль - Натиснути кнопку «Зберегти як»	- Відкриття діалогового вікна з вибором майбутнього місцеположення файлу - Створення нового файлу паролів - Відкриття файлу в головному меню програми
Введення некоректних значень	- В ведення ключа менше 16 символів - Натиснути кнопку «Створити»	- Повідомлення про помилку - Створення нового файлу заборонено

Таблиця 3.3 – Тест-кейси для тестування функціоналу відкриття файлу.

Функціонал відкриття файлу паролів		
Передумова: Відкрити вікно пошуку файлу паролів		
Крок тесту	Дані тесту	Очікуваний результат
Обрано файл з розширенням .xml та введено вірний ключ	- Пошук та вибір необхідного файлу з розширенням .xml - Ввести ключ до файлу - Натиснути кнопку «Відкрити»	Відкриття файлу в головному меню

Продовження таблиці 3.3 – Тест-кейси для тестування функціоналу відкриття файлу.

Функціонал відкриття файлу паролів		
Передумова: Відкрити вікно пошуку файлу паролів		
Крок тесту	Крок тесту	Крок тесту
Обрано файл з розширенням .xml та введено невірний ключ	- Пошук та вибір необхідного файлу з розширенням .xml - Ввести невірний ключ до файлу - Натиснути кнопку «Відкрити»	- Повідомлення про введення невірного ключа
Обрано файл з невірним розширенням	- Пошук та вибір необхідного файлу	- Повідомлення про помилку розширення файлу

Таблиця 3.4 – Тест-кейси для тестування функціоналу створення нового паролю.

Функціонал створення нового паролю		
Передумова: натиснення кнопки «Новий пароль»		
Крок тесту	Дані тесту	Очікуваний результат
Введення коректних значень без автоматичної генерації паролю	- Заповнити необхідні поля - Переглянути рекомендації щодо складності паролю - Натиснути кнопку «Зберегти»	- Створення нового запису в файлі - Відображення нового паролю в головному меню
Введення коректних значень з автоматичною генерацією паролю	- Заповнити необхідні поля - Натиснути кнопку «Згенерувати» - Натиснути кнопку «Зберегти»	- Створення нового запису в файлі - Відображення нового паролю в головному меню

Продовження таблиці 3.4 – Тест-кейси для тестування функціоналу створення нового паролю.

Функціонал створення нового паролю		
Передумова: натиснення кнопки «Новий пароль»		
Крок тесту	Дані тесту	Очікуваний результат
Введення некоректних значень	- Залишити порожнім хоча б одне поле - Натиснути кнопку «Зберегти»	- Повідомлення про помилку заповнення полів

Таблиця 3.5 – Тест-кейси для тестування функціоналу редагування паролю

Функціонал редагування паролю		
Передумова: натиснення кнопки «Новий пароль»		
Крок тесту	Дані тесту	Очікуваний результат
Зміна введених значень без автоматичної генерації паролю	- Редагувати необхідні поля - Переглянути рекомендації щодо складності паролю - Натиснути кнопку «Зберегти»	- Зміна запису в файлі - Відображення паролю в головному меню
Зміна введених значень з автоматичною генерацією паролю	- Редагувати необхідні поля - Натиснути кнопку «Згенерувати» - Натиснути кнопку «Зберегти»	- Внесення змін до запису в файлі - Відображення паролю в головному меню
Введення некоректних значень	- Залишити порожнім хоча б одне поле - Натиснути кнопку «Зберегти»	- Повідомлення про помилку заповнення полів

3.4 Фінальна збірка та запуск програми.

Після завершення розробки та прохождення усіх тест-кейсів додаток був скомпільований за допомогою інтегрованого середовища розробки Visual Studio. В результаті компіляції був отриманий .exe файл (рис 3.14) за допомогою якого можна запустити додаток. Оскільки додаток є портативним він не потребує встановлення і може бути перенесений на інший комп’ютер

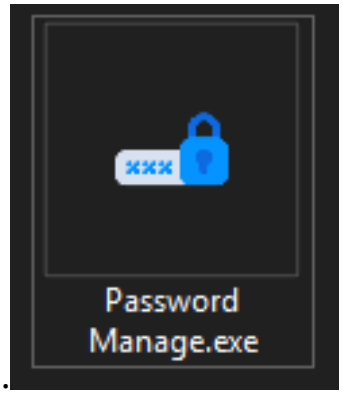


Рисунок 3.14 – Файл запуску додатку

ІНСТРУКЦІЯ КОРИСТУВАЧА

4.1 Почтакок роботи

4.1.1 Перед використанням

Для використання додатку потрібна операційна система Windows 10 та бібліотеки .NET Framework 4.7.2.

Для запуску додатку потрібно розмістити .exe файл в зручному місці після чого запустити його.

4.1.2 Головне вікно

Після запуску перед користувачем відкривається головне вікно програми яке зображене на рисунку 4.1. Спочатку майже всі кнопки керування не активні крім меню в лівій верхній частині програми.

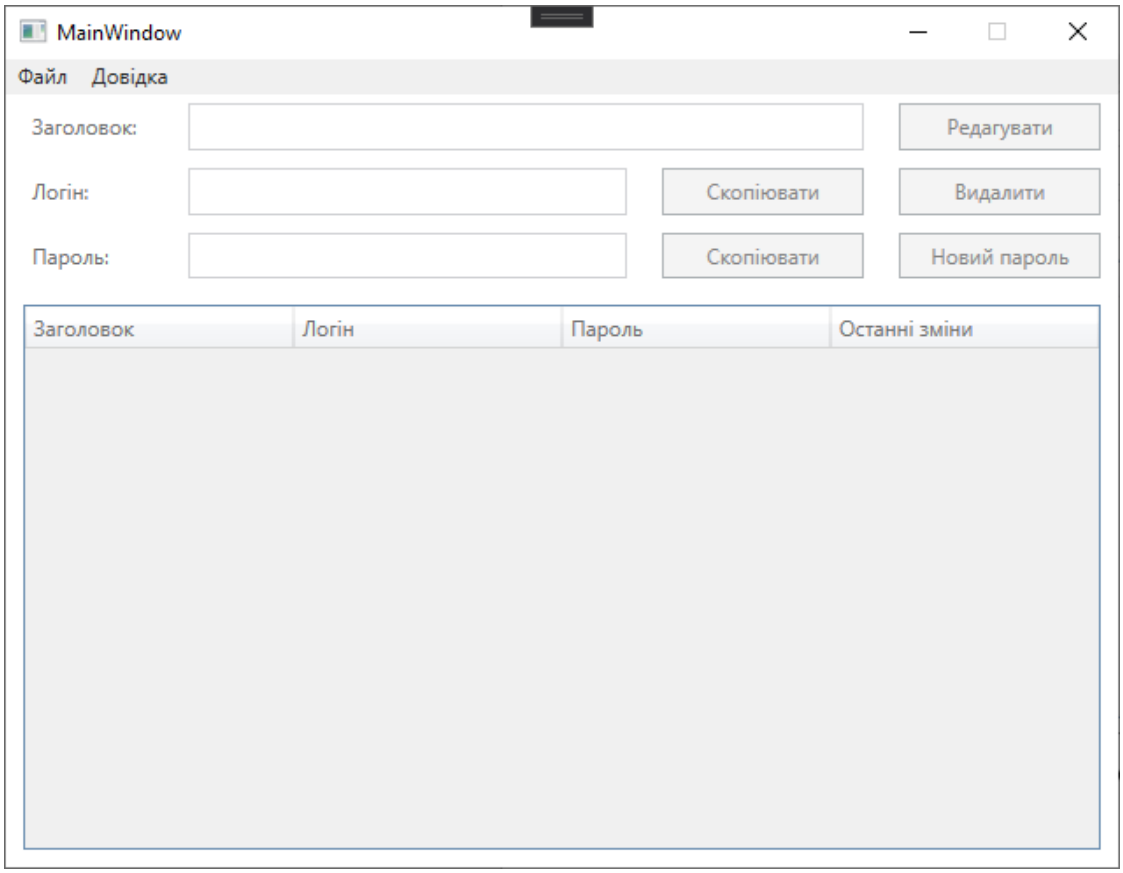


Рисунок 4.1 – Головне вікно програми

4.2 Робота з файлами паролів

Для продовження роботи потрібно створити або відкрити файл для цього потрібно в меню файл (рис 4.2) вибрати відповідний пункт меню далі відкриється відповідне вікно.

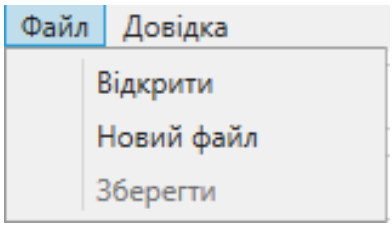


Рисунок 4.2 – Меню файл

4.2.1 Вікно створення нового файлу

Вікно створення файлу дозволяє створити новий файл з паролями. Для створення файлу потрібно у відповідних полях вказати ключ довжиною не менше 16 символів який буде використовуватися в якості майстер-ключа до файлу з паролями.

За допомогою кнопки «Зберегти як» користувач може вказати місце збереження файлу з паролями за замовчуванням файл зберігається в папку з якої була запущена програма.

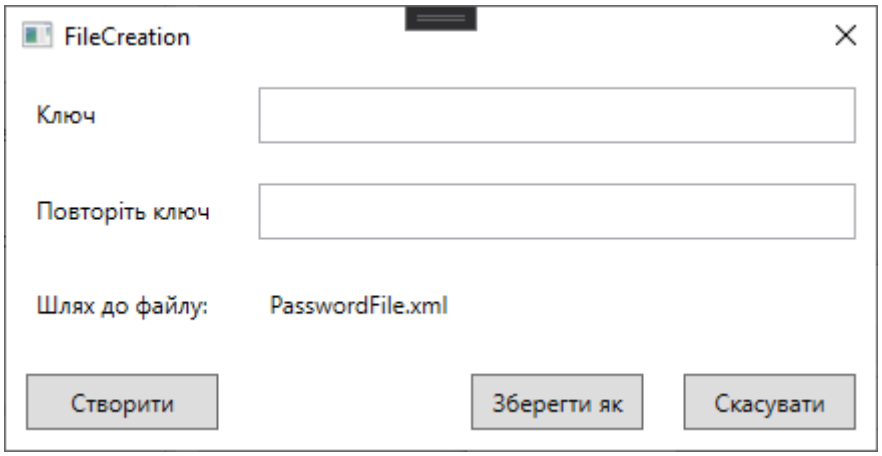


Рисунок 4.3 – Вікно створення нового файлу

4.2.2 Вікно відкриття файлу

Вікно відкриття файлу дозволяє відкрити раніше створений файл. Перед відкриттям вікна у користувача запитують місце знаходження файлу який він хоче відкрити рисунок 4.4.

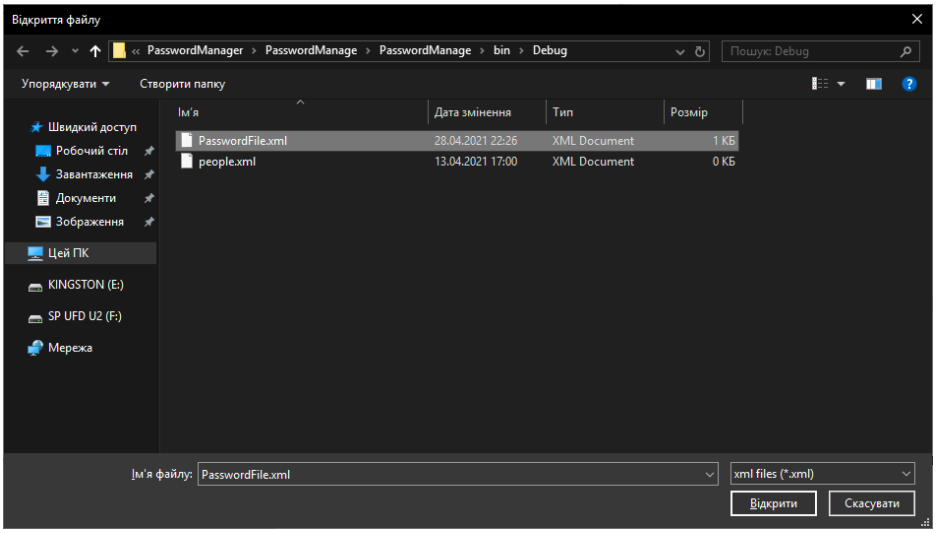


Рисунок 4.4 –Вибір файлу

Після вибору файлу з’явиться вікно відкриття файлу рисунок 4.5 для продовження потрібно у відповідне поле ввести ключ від обраного файлу. Шлях до файлу буде вказано у полі Файл.

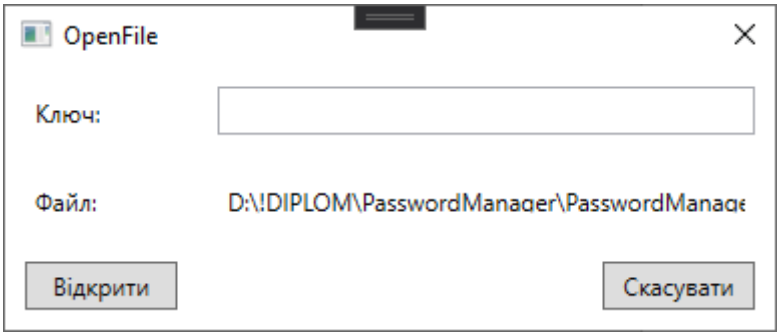


Рисунок 4.5 – Вікно відкриття файлу.

Якщо пароль в ведений не вірно буде показано повідомлення «Неправильний пароль».

4.3 Редагування даних в фалі паролів

4.3.1 Створення видалення та редагування паролів

Після відкриття або створення нового файлу повертаємося до головного вікна. На головному вікні відображається таблиця з усіма паролями користувача де вказано заголовок, логін, пароль та дату останніх змін. Також на головному вікні відображається поля з інформацією про обраний пароль для зручного копіювання.

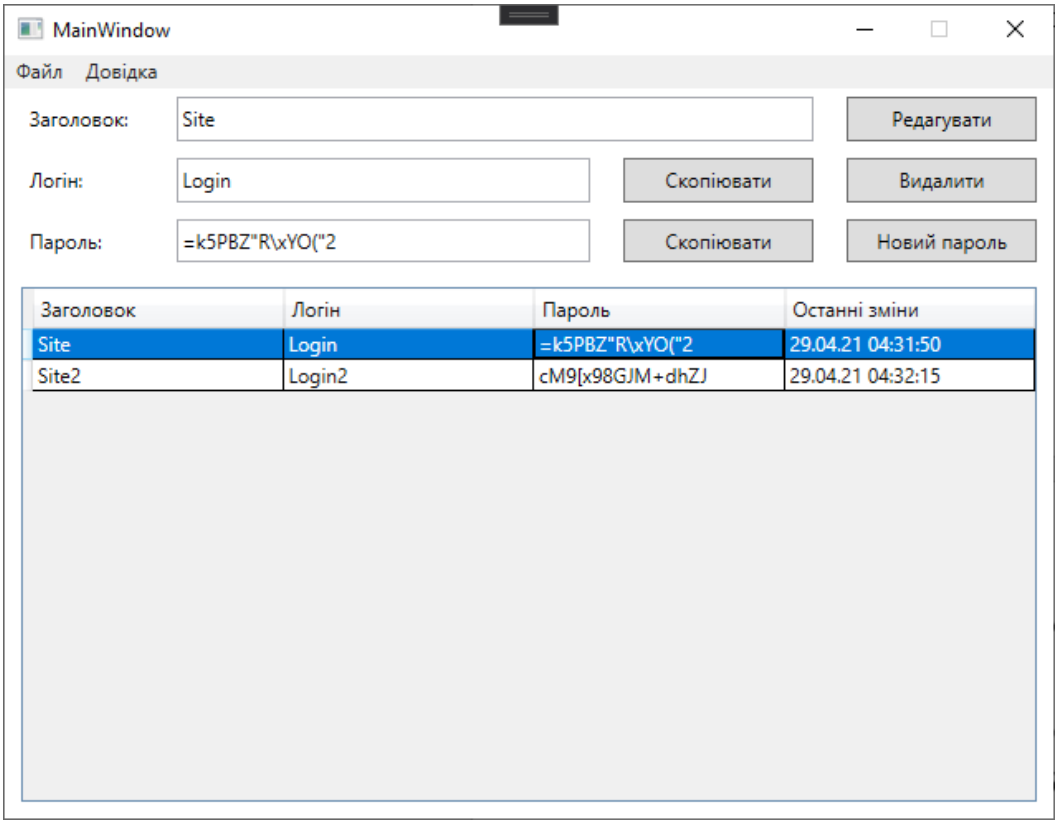


Рисунок 4.6 – Головне вікно програми з даними про паролі.

На головному екрані доступні наступні кнопки:

- кнопка новий пароль, яка відкриває вікно створення нового паролю.
- кнопка видалити, яка видаляє обраний пароль.
- кнопка редагувати, яка відкриває вікно редагування.
- кнопка скопіювати – додає дані з відповідного паля до буферу обміну.

Після внесення змін до файлу необхідно зберегти зміни за допомогою Файл –> Зберегти.

4.3.2 Вікно створення та редагування паролю

Для створення або редагування паролів необхідно заповнити відповідні поля у вікні рис 4.7. Поле Заголовок відповідає назві сайту або програми від якої буде збережено пароль. Поля пароль і логін зберігають дані про пароль і логін. За допомогою кнопки Згенерувати можна створити випадковий пароль.

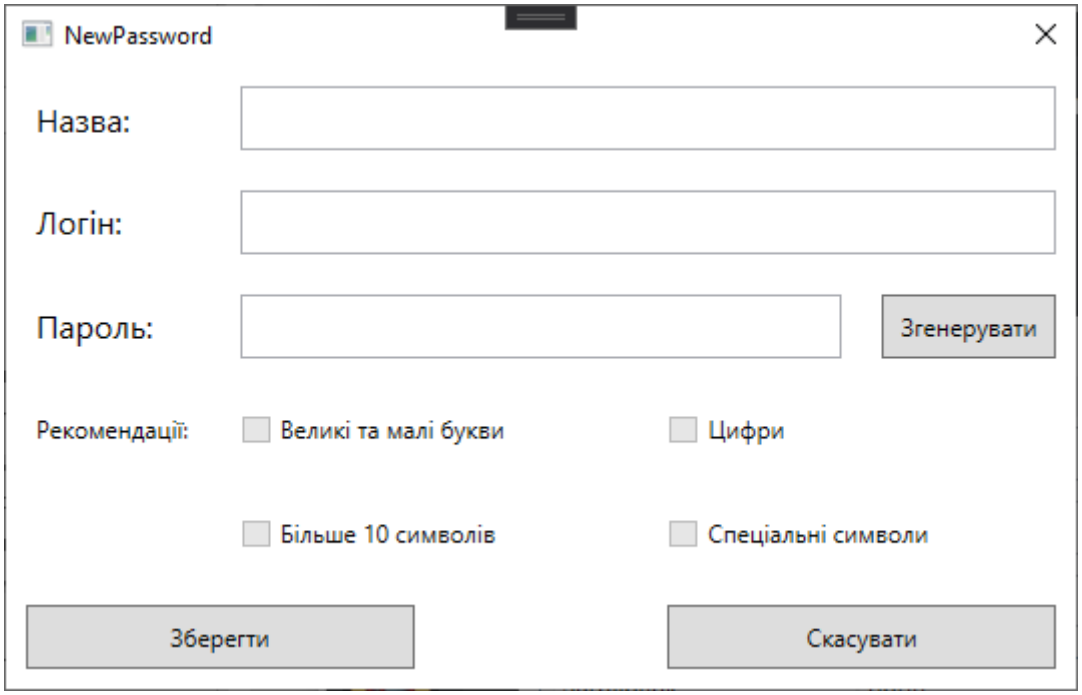


Рисунок 4.7 – Вікно збереження та редагування паролю

Підчас створення паролю вікно містить рекомендації, які підвищують безпеку створених паролів.

Для збереження даних та повернення на головну сторінку потрібно натиснути кнопку зберегти.

ВИСНОВКИ

Робота присвячена розробці додатку для генерації паролів користувачів який збільшує рівень береки користувачів в мережі інтернет.

Проведено дослідження котрі обґрунтовують актуальність роботи та наукову новизну. А саме рівень надійності паролів та рівень безпеки їх зберігання серед користувачів всесвітньої мережі інтернет, проаналізовано інші додатки та виявлено їх недоліки.

Враховуючи переваги та недоліки існуючих застосунків, було проаналізовано вимоги та спроектовано новий портативний додаток та його структурні елементи. Спроектований додаток відповідає усім виявленим вимогам.

Проведено аналіз існуючих програмних засобів для розробки додатків для Windows обрано оптимальні. Таким чином для розробки додатку було обрано мову програмування C# та середовище розробки Visual Studio а для розробки інтерфейсу користувача WPF.

Для перевірки правильності роботи додатку та відповідності усім вимогам було створено тест кейси.

Розроблено програмний додаток який генерує та зберігає дані про паролі користувачів з урахуванням усіх виявлених вимог. Розроблений додаток може використовувати кожен користувач просто розмістивши файл запуску у зручному місці. За допомогою розробленого додатку користувачі можуть зберігати та генерувати паролі.

Результати досліджень бакалаврської роботи апробовані всеукраїнських науково-технічних конференціях:

«Застосування програмного забезпечення в інфокомунікаційних технологіях», м. Київ, 12.02.2021 р.

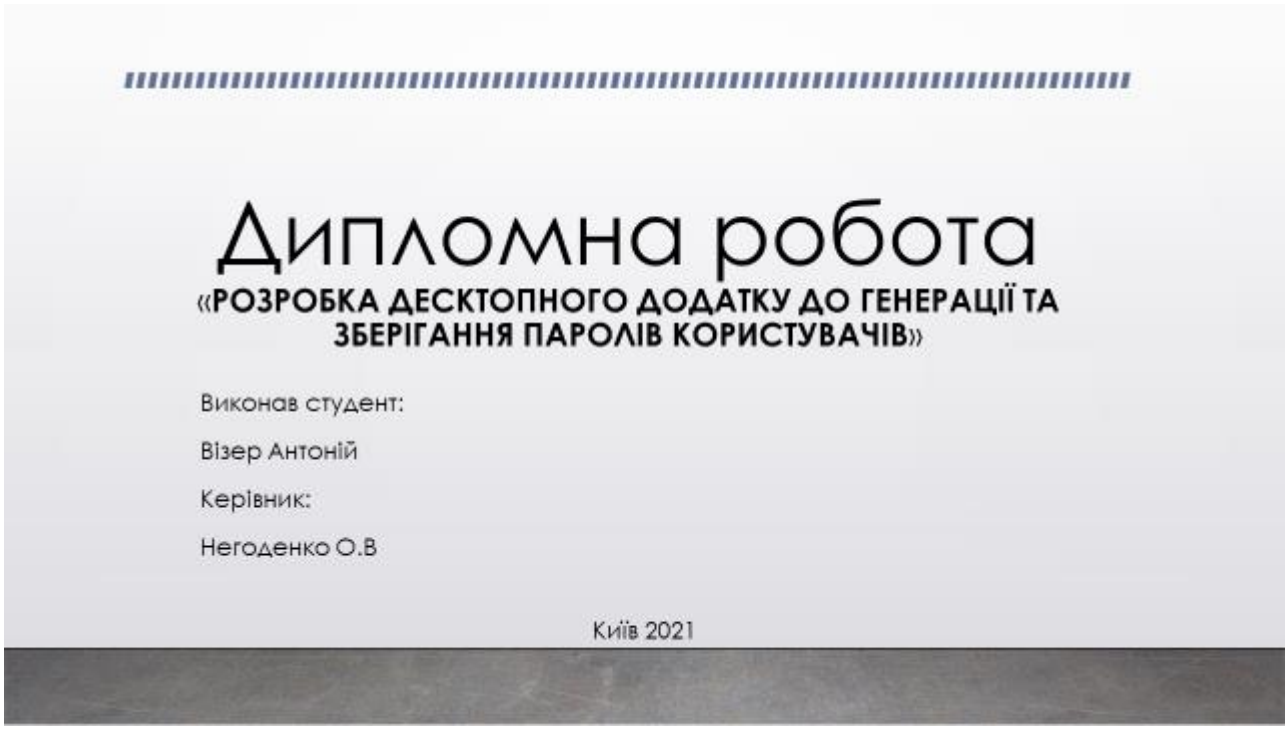
«Сучасний стан та перспективи розвитку IOT», м. Київ, 3.04.2020 р.

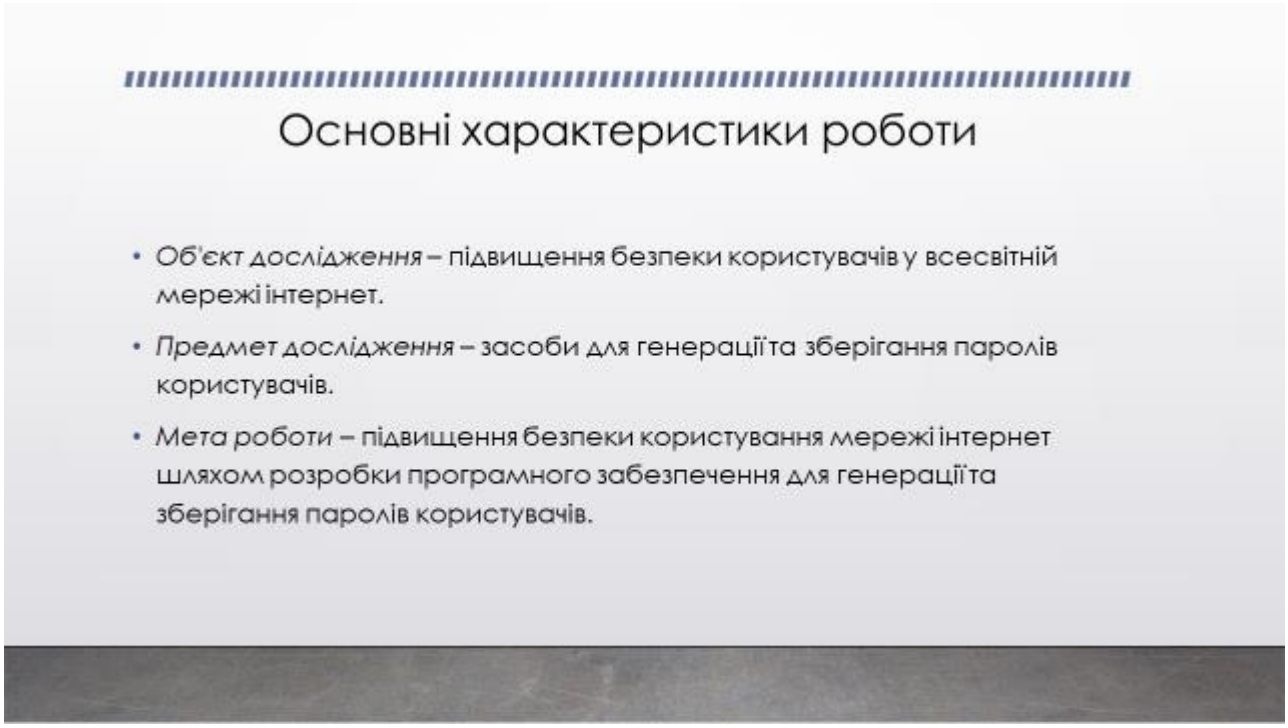
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бобала Ю. Я. Горбатого І. В. Інформаційна безпека. - Видавництво Львівської політехніки - с. 231 - 535
2. Wayback Machine internet archive [Електронний ресурс] – Режим доступу до ресурсу:https://web.archive.org/web/20040712152833/http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63v6_3_3.pdf.
3. Cyberresecurity & Infrastructure security agency [Електронний ресурс] – Режим доступу до ресурсу:https://us-cert.cisa.gov/ncas/tips/ST04-002.
4. Burnett Mark, Perfect Passwords – Rockland Massachusetts: Syngress Publishing – p. 181.
5. tools.ietf.org [Електронний ресурс] – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc4086.
6. Wayback Machine internet archive [Електронний ресурс] – Режим доступу до ресурсу:https://web.archive.org/web/20170227140027/http://www.businessinsider.com/how-to-use-password-manager-store-protect-yourself-hackers-lastpass-1password-dashlane-2017-2.
7. searchsecurity.techtarget.com [Електронний ресурс] – Режим доступу до ресурсу: https://searchsecurity.techtarget.com/definition/single-sign-on.
8. okta.com [Електронний ресурс] – Режим доступу до ресурсу: https://www.okta.com/identity-101/what-is-token-based-authentication/
9. keepass.info [Електронний ресурс] – Режим доступу до ресурсу: https://keepass.info/help/base/faq_tech.html.
10. lastpass.com [Електронний ресурс] – Режим доступу до ресурсу: https://www.lastpass.com/how-lastpass-works.
11. devblogs.microsoft.com [Електронний ресурс] – Режим доступу до ресурсу: https://devblogs.microsoft.com/dotnet/announcing-the-net-framework-4-8/
12. Richter D. CLR via C#. Programming on Microsoft .NET Framework 4.5 in C# - p. 40 – 150.
13. Herbert Schildt C# 4.0 The Complete Reference - p. 42 – 136.

14. Metthew MacDonald Pro Wpf 4.5 in C#: Windows Presentation Foundation in .Net 4.5 (Professional Apress) - p. 34 – 107.
15. Bruce Johnson Professional Visual Studio 2017 p. 131 – 159.
16. Scot Chacon, Ben Straud. Pro Git book – p. 10 – 25
17. support.microsoft.com [Електронний ресурс] – Режим доступу до ресурсу: https://is.gd/byP42B
18. Grady Brooch, James Rumbaugh, Ivan Jacodson. The Unified Modeling Language User Guide (Object Technology Series) p. 103 - 284.
19. Cem Kaner, Jack L.Falk. Testing computer software p. 123 - 412.
20. The World's First Computer Password? It Was Useless Too [Електронний ресурс] – Режим доступу до ресурсу: https://www.wired.com/2012/01/computer-password/
21. Passwords Evolved: Authentication Guidance for the Modern Era Too [Електронний ресурс] – Режим доступу до ресурсу: https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/
22. The 20 Most Hacked Passwords in the World [Електронний ресурс] – Режим доступу до ресурсу: https://www.safetydetectives.com/blog/the-most-hacked-passwords-in-the-world/
23. The Current State Of Authentication [Електронний ресурс] – Режим доступу до ресурсу: https://www.smashingmagazine.com/2016/06/the-current-state-of-authentication-we-have-a-password-problem/
24. Bruce Schneier. Secrets and Lies: Digital Security in a Networked World

Додаток


Дипломна робота
«РОЗРОБКА ДЕСКТОПНОГО ДОДАТКУ ДО ГЕНЕРАЦІЇ ТА ЗБЕРІГАННЯ ПАРОЛІВ КОРИСТУВАЧІВ»
Виконав студент:
Візер Антоній
Керівник:
Негоденко О.В
Київ 2021


Основні характеристики роботи
• Об'єкт дослідження – підвищення безпеки користувачів у всесвітній мережі інтернет.
• Предмет дослідження – засоби для генерації та зберігання паролів користувачів.
• Мета роботи – підвищення безпеки користування мережі інтернет шляхом розробки програмного забезпечення для генерації та зберігання паролів користувачів.

Актуальність

На сьогоднішній день паролі набули широкого розповсюдження їх використовують великої кількості додатків та сервісів. Тому через постійний ріст загроз злому паролів їх складність зростає з кожним роком. Через необхідність у великій кількості надійних паролів, які складно запам'ятати користувачі зневажають безпекою використовують слабкі паролі або один пароль на декількох сайтах та зберігають паролі у незахищеному вигляді. Тому виникають проблеми з безпекою. Для підвищення безпеки можна використовувати спеціальні засоби таки як менеджери паролів.

Програми аналоги

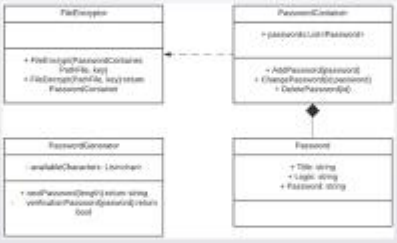
Серед програм аналогів можна виділити:

- KeePass
- LastPass
- Менеджер паролів Google

Вимоги до нового додатку

- Збереження даних про паролі користувачів.
- Редагування даних про паролі.
- Шифрування файлів користувачів.
- Генерація паролів користувачів.
- Підтримка декількох файлів для збереження паролів.
- Портативність.

Діаграми UML



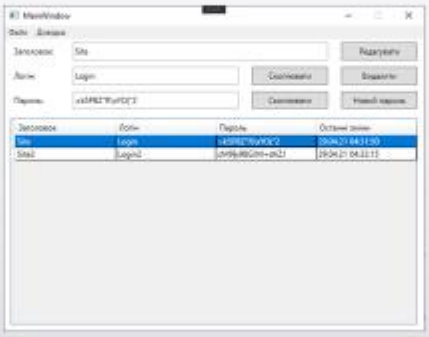
Засоби розробки

Visual Studio 2019

Microsoft .NET

Microsoft WPF

Інтерфейс додатку



Зберігання даних

Дані про паролі зберігаються в серіалізації в XML. Після чого шифруються за допомогою алгоритму RSA.

Висновки

- У даній роботі було розроблено додаток для генерації та зберігання паролів користувачів.
- Модернізовано метод збереження даних в XML файл за допомогою алгоритму шифрування.
- В подальшому досліджені можливі покращення зручності користування додатком наприклад додаванням функції автозаповнення форм з паролями та покращенням інтерфейсу користувача. Окрім цього можлива модернізація додатку в області безпеки.