МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

ЯКИМІВ Андрій Романович

Алгоритми для оцінки рівня безпеки облікових записів користувачів у соціальних мережах / Algorithms for assessing the security level of user accounts in social networks

спеціальність: 125 – Кібербезпека

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм -21
А.Р. Якимів

Науковий керівник
д.т.н., професор М.М.Касянчук

Кваліфікаційну роботу допущено
до захисту:

«____» _____ ___2023 р.

Завідувач кафедри

_____ В.В.Яцків

ТЕРНОПІЛЬ – 2023

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь "магістр"
спеціальність:125 – Кібербезпека
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ В.В.Яцків
"____"_____20__ р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ЯКИМІВ Андрій Романович
(прізвище, ім'я по-батькові)

1. Тема кваліфікаційної роботи
Алгоритми для оцінки рівня безпеки облікових записів користувачів у соціальних мережах / Algorithms for assessing the security level of user accounts in social networks

керівник роботи: д.т.н., проф. М.М.Касянчук

затверджені наказом по університету «__» _____ 2022 року № ___

2. Строк подання студентом закінченої кваліфікаційної роботи:
1 грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити
- ідентифікувати загрози інформаційної безпеки для персональних сторінок користувачів соціальних мереж;
- проаналізувати засоби контролю безпеки для сторінок користувачів у соціальних мережах;
- дослідити особливості налаштування параметрів безпеки сторінок користувачів у соціальних мережах;
- розробити алгоритми оцінки та підвищення рівня безпеки сторінок користувачів у соціальних мережах;
- розробити програмне забезпечення для підвищення рівня інформаційної безпеки персональних сторінок користувачів у соціальних мережах.

5. Перелік графічного матеріалу у роботі.

- структурна схема програмного забезпечення;
- схема процесу обробки інформації елементами програми;
- модель класів;
- алгоритм функціонування програми;
- архітектура програмного забезпечення;
- діаграма послідовностей програмного забезпечення;
- структура API.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання: 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Особливості захисту облікових записів користувачів у соціальних мережах	12.2022 р. – 03.2023 р.	
2	Алгоритмічне забезпечення для оцінки рівня безпеки облікових записів користувачів соціальних мереж	03.2023 р. – 05.2023 р.	
3	Програмне забезпечення для підвищення рівня безпеки облікових записів користувачів у соціальних мережах	05.2023 р. – 11.2023 р.	

Студент _____ Якимів А.Р.
(підпис)

Керівник роботи _____ д.т.н., проф. Касянчук М.М.
(підпис)

АНОТАЦІЯ

Випускна кваліфікаційна робота на тему „Алгоритми для оцінки рівня безпеки облікових записів користувачів у соціальних мережах” на здобуття освітнього ступеня «Магістр» зі спеціальності 125 „Кібербезпека” освітньо-професійної програми «Кібербезпека» написана обсягом 87 сторінок і містить 19 ілюстрацій, 12 таблиць, 2 додатоки та 35 джерела за переліком посилань.

Метою випускної кваліфікаційної роботи є підвищення рівня безпеки користувачів у соціальних мережах.

Методи дослідження - методи експертних оцінок, методи об'єктно-орієнтованого аналізу і проектування, емпіричні методи тестування програмного забезпечення ..

Результати дослідження: Робота зосереджується на аналізі та підвищенні захищеності персональних сторінок у соціальних мережах. У першій частині вивчаються загрози для користувацьких облікових записів, аналізуються існуючі засоби контролю та параметри безпеки. Поставлено задачі дослідження. У другій частині розглядаються алгоритми для оцінки рівня безпеки. Здійснюється експертна оцінка та розроблюється алгоритмічне забезпечення для обчислення цього рівня. Також вивчається архітектура програмного забезпечення. У третій частині описується створене програмне забезпечення для підвищення безпеки облікових записів у соціальних мережах. Обґрунтовується вибір засобів розробки, висвітлюється організація графічного інтерфейсу, особливості реалізації та проводиться тестування програми.

Ключові слова: СОЦІАЛЬНІ МЕРЕЖІ, БЕЗПЕКА, СТОРІНКИ КОРИСТУВАЧІВ, ОБЛІКОВІ ЗАПИСИ, КОТРОЛІ БЕЗПЕКИ.

ABSTRACT

The graduate work on the topic „ Algorithms for assessing the security level of user accounts in social networks” for Master’s degree on speciality 125 "Cybersecurity " is written on 87 pages and contains 19 illustrations, 12 tables, 2 appendixs and 35 references.

The aim of the graduation qualification work is to enhance the security level of users on social networks.

Research methods include expert assessment methods, object-oriented analysis and design methods, and empirical software testing methods.

Research results: The work focuses on the analysis and improvement of the security of personal pages on social networks. The first part examines threats to user accounts, analyzes existing security controls, and explores security parameters. Research objectives are established. The second part discusses algorithms for assessing security levels. Expert assessment is carried out, and algorithmic provisions for calculating this level are developed. The software architecture is also studied. The third part describes the developed software for enhancing the security of user accounts on social networks. The choice of development tools is justified, the organization of the graphical interface is highlighted, implementation features are outlined, and program testing is conducted.

Keywords: SOCIAL NETWORKS, SECURITY, USER PAGES, ACCOUNTS, SECURITY CONTROLS.

ЗМІСТ

ВСТУП

Соціальні мережі представляють собою інтерактивні комп'ютерно-опосередковані технології, які сприяють створенню та обміну інформацією, ідеями, кар'єрними інтересами та іншими вираженнями через віртуальні спільноти [1]. Сучасне Інтернет-середовище використовують для користування соціальними мережами понад мільярд людей у всьому світі [2].

Останні роки характеризуються стрімким розвитком соціальних мереж, які швидко вплетаються в повсякденне життя людей. Люди все більше визнають переваги використання соціальних мереж, які охоплюють різні сфери людської діяльності, включаючи навіть торгівлю товарами у соціальних мережах [3-4].

Проте, разом із безліччю переваг існують і певні недоліки, що можуть мати серйозні наслідки. Більшість користувачів соціальних мереж вважають, що їх безпека гарантована, але насправді кожен з них може стати жертвою злому особистих сторінок через недотримання звичайних правил безпеки, розроблених розробниками кожної соціальної мережі.

Кіберзлочинці використовують соціальні мережі як інструмент для пошуку та збору інформації про своїх потенційних жертв, часто використовуючи непрацюючі облікові записи. Це може призвести до поширення загроз та дезінформації, а тому кожен користувач повинен дотримуватись правил користування соціальними мережами та слідувати порадам експертів з безпеки даних [5-7].

Проблеми, пов'язані із захистом інформації, є актуальними для фахівців з інформаційної безпеки і звичайних користувачів. Незважаючи на технологічний прогрес у сфері захисту інформації, вразливість особистих сторінок в соціальних мережах продовжує зростати. Серед загроз для користувачів виділяються впровадження шкідливого коду через недостовірні джерела, атаки через веб-додатки, соціально-інженерні атаки, а також загрози цілісності та конфіденційності даних.

Отже, розробка алгоритмів для оцінки та підвищення рівня безпеки облікових записів користувачів у соціальних мережах, є вкрай важливою та актуальною задачею.

Мета дослідження – підвищення рівня безпеки користувачів у соціальних мережах.

Для досягнення поставленої мети потрібно розв’язати наступні взаємопов’язані **завдання**:

- ідентифікувати загрози інформаційної безпеки для персональних сторінок користувачів соціальних мереж;
- проаналізувати засоби контролю безпеки для сторінок користувачів у соціальних мережах;
- дослідити особливості налаштування параметрів безпеки сторінок користувачів у соціальних мережах;
- розробити алгоритми оцінки та підвищення рівня безпеки сторінок користувачів у соціальних мережах;
- розробити програмне забезпечення для підвищення рівня інформаційної безпеки персональних сторінок користувачів у соціальних мережах.

Об’єкт дослідження – захист конфіденційних даних користувачів у соціальних мережах.

Предмет дослідження – алгоритми для оцінки рівня безпеки облікових засобів коритувачів у соціальних мережах.

Методи дослідження – методи експертних оцінок, методи об'єктно-орієнтованого аналізу і проектування, емпіричні методи тестування програмного забезпечення.

Наукова новизна отриманих результатів:

- розроблено математичну модель для показника рівня безпеки сторінок користувачів у соціальних мережах у якій враховано контролів безпеки, що дало змогу отримати числове значення показника рівня безпеки сторінки користувача

Практичне значення. У процесі виконання роботи розроблено програмне забезпечення для підвищення рівня безпеки сторінок користувачів у соціальних мережах.

Публікації та апробація.

1. Якимів А.Р. Архітектура програмного забезпечення для оцінки рівня захисту користувачів у соціальних мережах // Збірник матеріалів школи-семінару молодих вчених і студентів «Комп'ютерні інформаційні технології», Тернопіль., 2023. С. 24-25.

2. Якимів А.Р. Аналіз контролів безпеки персональних сторінок користувачів у соціальних мережах // Збірник матеріалів школи-семінару молодих вчених і студентів «Комп'ютерні інформаційні технології», Тернопіль., 2023. С. 31-32.

1 ОСОБЛИВОСТІ ЗАХИСТУ ОБЛІКОВИХ ЗАПИСІВ КОРИСТУВАЧІВ У СОЦІАЛЬНИХ МЕРЕЖАХ

1.1 Загрози для персональних сторінок користувачів соціальних мереж

Інформаційна безпека визначається як стан захищеності систем обробки і зберігання даних, який забезпечує конфіденційність, доступність і цілісність інформації. Це охоплює комплекс заходів, спрямованих на запобігання несанкціонованому доступу, використанню, розголошенню, руйнуванню, внесенню змін, ознайомленню, перевірці, запису чи знищенню [8].

Загроза інформаційної безпеки визначається як потенційна можливість порушення режиму інформаційної безпеки [6,7]. Атака на інформаційну систему виникає внаслідок навмисної реалізації загроз, а особи, які це роблять, вважаються зловмисниками.

Інцидент інформаційної безпеки відбувається, коли виникає одна чи кілька небажаних чи неочікуваних подій, що можуть значно вплинути на безпеку інформації [8]. Зазвичай загроза виникає через наявність вразливостей у системах захисту, наприклад, через неконтрольований доступ до персональних комп'ютерів або мобільних пристроїв.

Історія розвитку інформаційних систем демонструє, що нові вразливості постійно з'являються, але засоби захисту також активно розвиваються. Оновлення програмного забезпечення, такі як ті, які випускає фірма Microsoft, призначені для усунення вразливостей. Однак цей підхід має обмежену ефективність через проміжок часу між виявленням загрози і її усуненням, під час якого зловмисник може завдати шкоди [8].

Відповідно до цього, більш прийнятним є випереджаючий підхід до захисту, який передбачає розробку механізмів захисту від можливих, передбачуваних і потенційних загроз. Цей підхід використовується в науковій роботі.

Важливо враховувати, що деякі загрози можуть виникати в результаті випадкових помилок або техногенних явищ. Найнебезпечнішими вважаються загрози, що виникають внаслідок ненавмисних дій людей.

Знання можливих загроз інформаційної безпеки та уразливих місць системи захисту є важливим для вибору ефективних заходів забезпечення безпеки.

Соціальні мережі на сьогоднішній день представляють собою значний об'єкт ризиків для інформаційної безпеки, і фахівці з цієї галузі виокремлюють різноманітні загрози, зокрема:

Облікові записи друзів: Взаємодія з обліковими записами "друзів" може бути ризикованою, оскільки довіра до цих осіб може призвести до можливості зловмисникам використовувати ці облікові записи для атак.

Використання скорочувачів URL-адрес: Послуги скорочення URL-адрес можуть приховувати справжні адреси сайтів, створюючи можливість для розповсюдження шкідливого програмного забезпечення через маскування. Ця загроза залишається актуальною, навіть при застосуванні покращених методів виявлення спаму та загроз.

Використання одних і тих самих імен та паролів: Атака, відома як "Daisy Chain," полягає в тому, щоб зламати облікові записи соціальних мереж і використовувати їх для проникнення в корпоративні ресурси через співробітників.

Веб-атаки: Хакери можуть використовувати соціальні мережі для організації атак через вразливості браузерів, XSS / CSRF-атаки та інші методи. Їх мета - проникнення в інформаційну систему користувача та установка шкідливого коду.

Витік інформації та компрометація поведінки співробітників: Соціальні мережі можуть використовуватися для витоку важливої інформації компанії або компрометації репутації через публікації від невдоволених або неналежно поводяться співробітників. Системи захисту даних та аналізу інтернет-публікацій використовуються для запобігання цим загрозам.

Враховуючи розвиток соціальних мереж та їх вплив на користувачів, безпека в цьому контексті стає критично важливою, і застосування заходів захисту є необхідністю для уникнення негативних наслідків.

Загрози інформаційної безпеки для персональних сторінок користувачів соціальних мереж кожна класифікувати наступним чином:

1) за складовими інформаційної безпеки:

- доступність: зловмисники можуть спробувати блокувати чи обмежувати доступ користувача до його сторінки.
- цілісність: зміна, видалення або спотворення інформації на сторінці користувача.
- конфіденційність: несанкціонований доступ до приватної інформації, витік конфіденційних даних;

2) за компонентами соціальних мереж:

- фінансова інформація: крадіжка фінансових даних користувачів, які пов'язані із соціальним профілем;
- персональні дані: несанкціонований доступ та використання особистої інформації;
- приватна інформація: оголошення особистих подробиць або використання їх для шахрайства;

3) за характером впливу:

- випадкові або навмисні: атаки можуть бути випадковими (наприклад, через вразливості) або навмисні (зловмисницькі дії для отримання користі);

4) за розташуванням джерела загроз:

- всередині або поза соціальною мережею: загрози можуть виникати від інших користувачів всередині мережі чи зовнішніх зловмисників, що спрямовують атаки на соціальну платформу.

У роботі проведено аналіз конкретних кібератак:

1) атака через вразливість facebook (2018):

- ознаки атаки: викрадення ключів доступу до 50 мільйонів сторінок;
- характер впливу: навмисна атака;
- розташування джерела загроз: всередині соціальної мережі;
- наслідки: велика кількість користувачів стала уразливою до несанкціонованого доступу;

2) хакерське вторгнення в облікові записи youtube (2018):

- ознаки атаки: додавання хакерами інформації до заголовків роликів та оголошення про "перевірку безпеки";
- характер впливу: навмисна атака;
- розташування джерела загроз: всередині соціальної мережі;
- наслідки: негативний вплив на репутацію компаній та втрата контролю над обліковими записами;

3) витік інформації з облікових записів instagram (2017):

- ознаки атаки: оприлюднення у відкритому доступі особистих даних користувачів;
- характер впливу: навмисна атака;
- розташування джерела загроз: всередині соціальної мережі;
- наслідки: збільшення загрози для конфіденційності особистої інформації користувачів;

4) масова розсилка троянського вірусу через facebook messenger (2017):

- ознаки атаки: поширення шкідливого програмного забезпечення через вірусні файли;
- характер впливу: навмисна атака;
- розташування джерела загроз: всередині соціальної мережі;
- наслідки: компрометація безпеки комп'ютерів користувачів та витік облікових даних;

5) фішингові атаки на службовців США через соціальні мережі (2017):

- ознаки атаки: Відсилання фішингових повідомлень зі шкідливими посиланнями;
- характер впливу: Навмисна атака;
- розташування джерела загроз: Зовнішнє джерело;
- наслідки: Контроль над пристроями службовців та потенційна загроза для національної безпеки;

6) шахрайська операція через соціальні мережі (липень 2017):

- ознаки атаки: Використання фейкової особистості для поширення троянського програмного забезпечення;
- характер впливу: Навмисна атака;
- розташування джерела загроз: Всередині соціальної мережі;
- наслідки: Дистанційне керування комп'ютерами та можливий збір конфіденційної інформації;

7) тіньова схема фінансових злочинів через соціальні мережі (серпень 2016):

- ознаки атаки: Полювання на передплатників банків і пропозиції фіктивних фінансових послуг;
- характер впливу: Навмисна атака;
- розташування джерела загроз: Зовнішнє джерело;
- наслідки: Фінансові втрати та погіршення репутації користувачів;

8) шкідлива програма Hammertoss (липень 2015):

- ознаки атаки: Пошук профілів за заданими критеріями та розсилка шкідливого програмного забезпечення;
- характер впливу: Навмисна атака;
- розташування джерела загроз: Зовнішнє джерело;
- наслідки: Кібератаки на важливі установи та урядові органи.

У таблиці 1.1 подано результати аналізу та класифікації за ознаками розглянутих вище кібератак.

Таблиця 1.1 - Класифікація за ознаками загроз кібератак, проведених з використанням соціальних мереж

Класифікація загрози / ознаки	Номер атаки							
	1	2	3	4	5	6	7	8
За складовими інформаційної безпеки								
- Доступність	+			+	+	+		
- Цілісність		+		+	+	+		+
- Конфіденційність	+	+	+	+	+	+	+	+
За компонентами соціальної мережі, на які націлені загрози								
- Фінансова інформація				+	+	+	+	
- Персональні дані	+	+	+	+	+	+	+	+
- Приватна інформація	+	+	+	+	+	+		+
- Дані додатків				+	+	+		
За характером впливу								
- Випадкові								
- Навмисні	+	+	+	+	+	+	+	+
За розташуванням джерела загроз								
- Всередині соціальної мережі				+	+	+	+	
- Поза соціальною мережею	+	+	+					+

Аналіз таблиці 1.1 показує, що більшість атак були спрямовані на персональні дані та приватну інформацію користувачів соціальних мереж. Тому саме на захист цих компонентів соціальним мереж будуть спрямовані дослідження в науковій роботі.

Рейтинг найбільш популярних соціальних мереж у світі за 2022 та 2023 рік подано на рисунку 1.1.

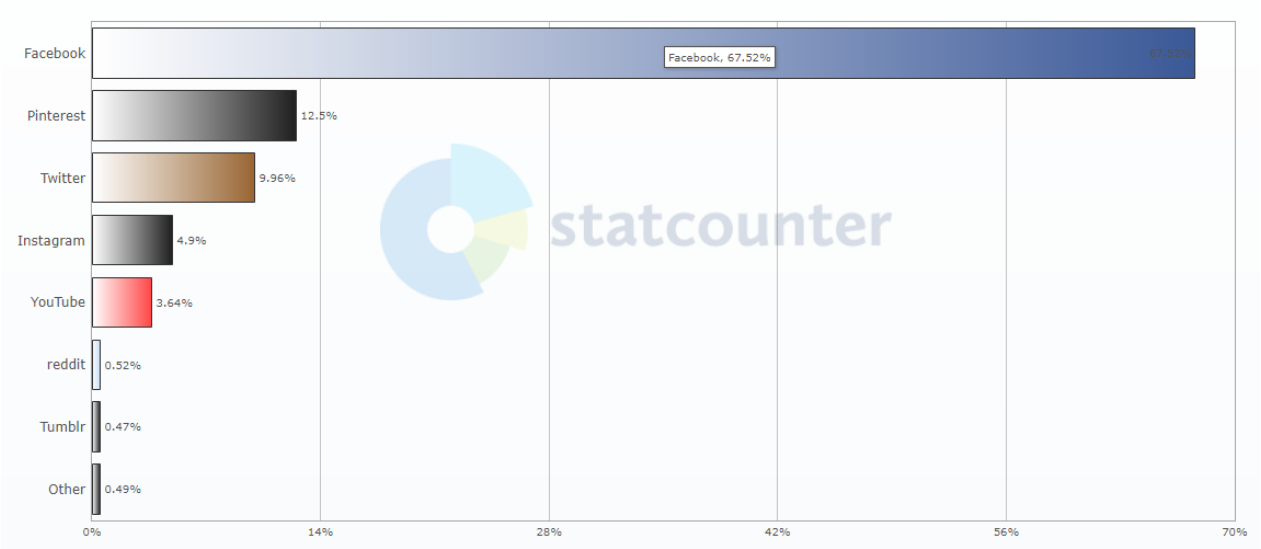

Рисунок 1.1 – Рейтинг соціальних мереж у світі

Всі соціальні мережі можна розділити за різними критеріями. Існують мережі, спрямовані на пошук знайомих, такі як однокласники, колеги або однокурсники. Також існують бізнес-мережі, призначені для пошуку роботи, партнерів та професійного спілкування в бізнес-середовищі. Різні мережі спеціалізуються на відео-, аудіо- або фото контенті, а також на конкретних темах, як от мережа Хабрахабр, яка об'єднує інформаційні технології.

Також важливо враховувати різницю в рівні доступності інформації. Багато мереж відкриті для громадськості, але існують і закриті, до яких можна потрапити лише за запрошенням. З географічної точки зору мережі можна класифікувати як світові, державні або регіональні.

Щодо рівня розвитку, соціальні мережі можна поділити на веб 1.0 (перші мережі з базовим функціоналом), веб 2.0 (сучасні мережі з широким функціоналом для спілкування) і веб 3.0 (мережі майбутнього, які вирішують конкретні проблеми і підвищують лояльність користувачів).

Важливо відзначити, що соціальні мережі є складною соціотехнічною системою, яка глибоко впливає на життя користувачів Інтернету. З великою

кількістю програмних продуктів, призначених для роботи в цих мережах, вони стали неот'ємною частиною щоденного життя.

Великі групи соціальних мереж можуть бути комерційними, професійними чи суспільно-орієнтованими. Їх порівняння та оцінка можуть використовувати різні критерії та методи, що визначають їхні характеристики та призначення.

Таблиця 1.2 - Класифікація соціальних мереж

Ознака класифікації	Вид соціальної мережі
Призначення	1. Для пошуку людей (Facebook, Twitter); 2. Для розваги (Youtube, F3); 3. Для роботи і бізнесу (LingedIn); 4. Для збору новин (Telegram, Facebook, Teitter); 5. Для відео (YouTube); 6. Для аудіо (Last.fm); 7. Для фото (Pinterest, Instagram);
Відкритість	1. Відкриті (FaceBook); 2. Змішані (Instagram); 3. Закриті (PlayboyU.com).
Географічне охоплення	1. Світові (MySpace); 2. Державні (Connect.com.ua); 3. Регіональні.
Рівень розвитку	1. Web 1.0; 2. Web 2.0; 3. Web 3.0.

1.2 Аналіз засобів контролю безпеки сторінок користувачів у соціальних мережах

Спільноти в Інтернеті пропонують користувачам необмежені можливості спілкування, обміну враженнями та інформацією, але разом із тим створюють потенційні загрози для їхньої кібербезпеки. У цьому контексті виникає необхідність вивчення та аналізу засобів контролю безпеки сторінок користувачів у соціальних мережах. Цей аналіз спрямований на визначення ефективності заходів, які вживаються для захисту особистих даних, паролів та іншої конфіденційної інформації користувачів, а також на розгляд можливостей покращення та оптимізації систем безпеки в онлайн-середовищі соціальних мереж.

Безпека облікових записів користувачів у соціальних мережах включає в себе взаємозв’язану сукупність засобів контролю безпеки, які впроваджені на сервері соціальної мережі, а також ряд додаткових контролів, які користувач повинен періодично перевіряти та налаштовувати [5].

Для аналізу засобів контролю безпеки обрано найбільш популярні на сьогодні соціальні мережі: Facebook [11], YouTube [12] та Instagram [13].

У дослідженні [5] виокремлено такі засоби контролю безпеки облікових записів користувачів, які реалізовані на сервері соціальних мереж:

- двоетапна перевірка: надає додатковий рівень безпеки для облікового запису, вимагаючи введення коду або іншого елементу, крім пароля;
- приватний обліковий запис: підвищує рівень конфіденційності облікового запису користувача;
- сповіщення про безпеку та конфіденційність: повідомляє користувача про можливі порушення безпеки чи конфіденційності;
- перевірка авторизованих входів: дозволяє перевірити авторизовані входи та пристрої до облікового запису;

- формування довірених контактів: дозволяє внести "друзів" поручителями для відновлення доступу у випадку порушення безпеки;
- генерування кодів ідентифікації: надає можливість згенерувати коди для двоетапної перевірки;
- перевірка складності пароля: оцінює складність створеного користувачем пароля;
- аналіз витоків пароля є засобом контролю, який дозволяє оцінити, чи стався витік пароля. Для проведення такої перевірки можна скористатися інтернет-сервісом https://haveibeenpwned.com/ [15]. Якщо пароль виявляється у базі даних цього сервісу, рекомендується змінити його;
- перевірка витоків даних для адреси електронної пошти є засобом контролю, який оцінює можливі витоки, пов'язані з електронною поштовою скринькою, що використовується для облікового запису користувача соціальної мережі. Для такої перевірки також можна використати інтернет-сервіс https://haveibeenpwned.com/ [15];
- регулярна зміна пароля є засобом контролю, який нагадує користувачеві соціальної мережі про необхідність періодичної зміни пароля відповідно до встановленої політики безпеки;
- перевірка доступу зовнішніх застосунків до облікового запису є засобом контролю, яка дозволяє перевірити, які зовнішні додатки мають доступ до інформації з облікового запису користувача соціальної мережі.

Результати порівняльного аналізу для засобів контролю безпеки персональних облікових записів користувачів у соціальних мережах подано в таблиці 1.3.

Таблиця 1.3 - Порівняльний аналіз засобів контролю безпеки облікових записів у соцмережах

Засоби контролю безпеки	Реалізація у соціальній мережі		
	Facebook	YouTube	Instagram
Двоетапна перевірка	+	+	+
Приватний обліковий запис	+	+	+
Сповіщення про безпеку та конфіденційність	+	+	-
Перевірка авторизованих входів	+	+	-
Формування довірених контактів	+	-	-
Генерування кодів ідентифікації	+	+	+
Перевірка складності пароля	+	+	+
Перевірка витоків пароля	-	-	-
Перевірка витоків даних для адреси електронної скриньки	-	-	-
Регулярна зміна пароля	-	-	-
Перевірка доступу зовнішніх застосунків до облікового запису	+	+	-

Аналіз таблиці 1.3 вказує на те, що ряд засобів контролю безпеки, які значно впливають на захист облікового запису користувача в соціальних мережах, не мають реалізації в жодній з них. Найбільша кількість засобів контролю забезпечена у соціальній мережі Facebook, в той час як найменша кількість таких засобів виявлена у соціальній мережі Instagram.

Детальний аналіз цих відмінностей може вказати на можливі шляхи вдосконалення безпеки в соціальних мережах та сприяти створенню більш єдиної та ефективної системи захисту особистих даних користувачів.

1.3 Аналіз параметрів безпеки сторінок користувачів у соціальних мережах

У даному розділі розглянемо особливості конфігурації параметрів безпеки особистих сторінок у соціальних мережах Facebook, YouTube та Instagram. Важливо відзначити, що обрані соціальні платформи надають користувачам можливість самостійно встановлювати та керувати рівнем безпеки свого облікового запису.

У соціальній мережі Facebook для налаштування безпеки особистої сторінки, користувачу необхідно перейти в розділ "Безпека та авторизація" та ручним чином встановити параметри безпеки (рисунок 1.2).

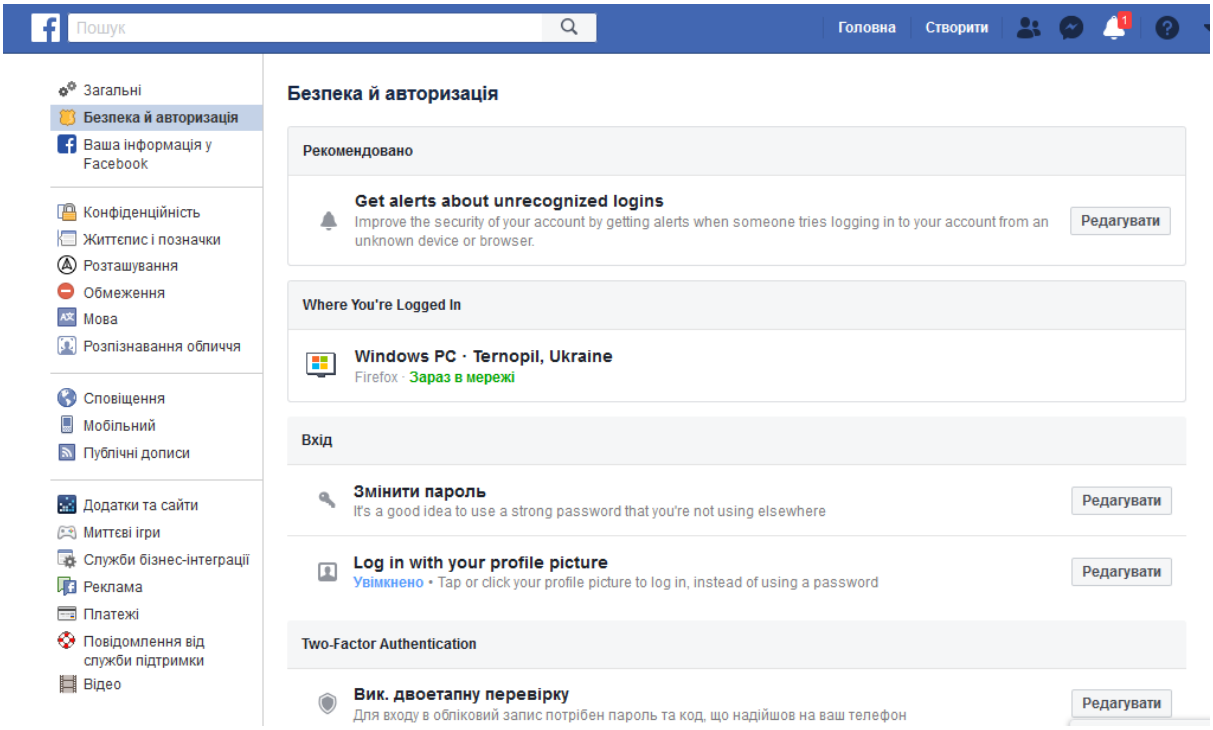

Рисунок 1.2 - Сторінка налаштування безпеки та авторизації у мережі Facebook

Перевагою такого типу конфігурації є можливість налаштовувати всі засоби для контролю, які розроблені на стороні сервера, на одній сторінці. Для соціальної мережі YouTube, налаштування контролів безпеки

здійснюється у єдиному центрі для управління безпекою та іншими налаштуваннями Google, який доступний за адресою https://myaccount.google.com/ (рисунок 1.3).

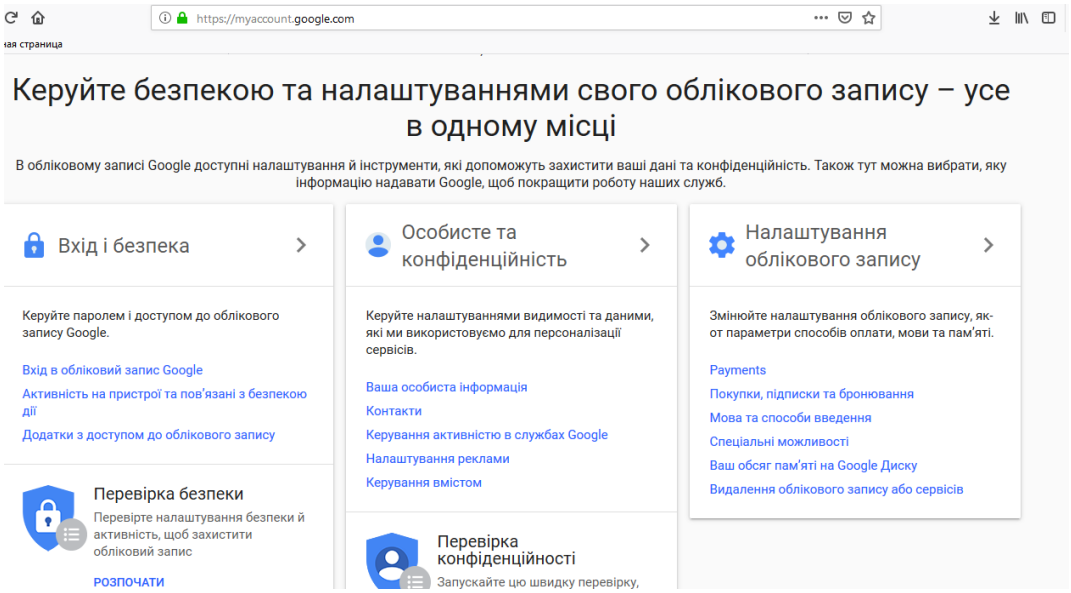

Рисунок 1.3 - Сторінка налаштування безпеки та авторизації для облікового запису Google

У системі Google також розроблено механізм перевірки та налаштування основних параметрів безпеки облікового запису. Для доступу до нього у розділі "Безпека" слід вибрати опцію "Перевірка безпеки" та вибрати "Розпочати". На центральному порталі безпеки Google (https://safety.google/security) розроблено рекомендації для користувачів, які допомагають істотно підвищити рівень безпеки їхнього облікового запису (рисунок 1.4).

Для налаштування контролів безпеки для соціальної мережі Instagram необхідно перейти в розділ “Налаштування” (рисунок 1.5).

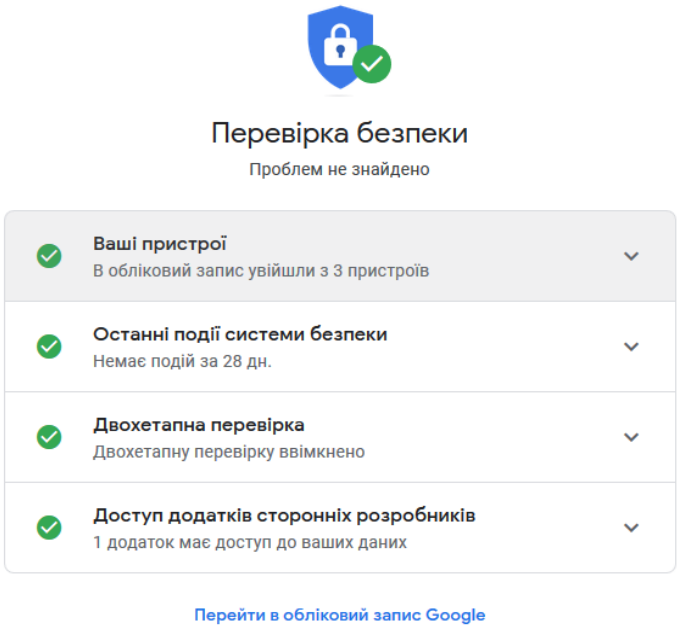

Рисунок 1.4 - Сторінка перевірки та налаштування базових параметрів безпеки облікового запису Google

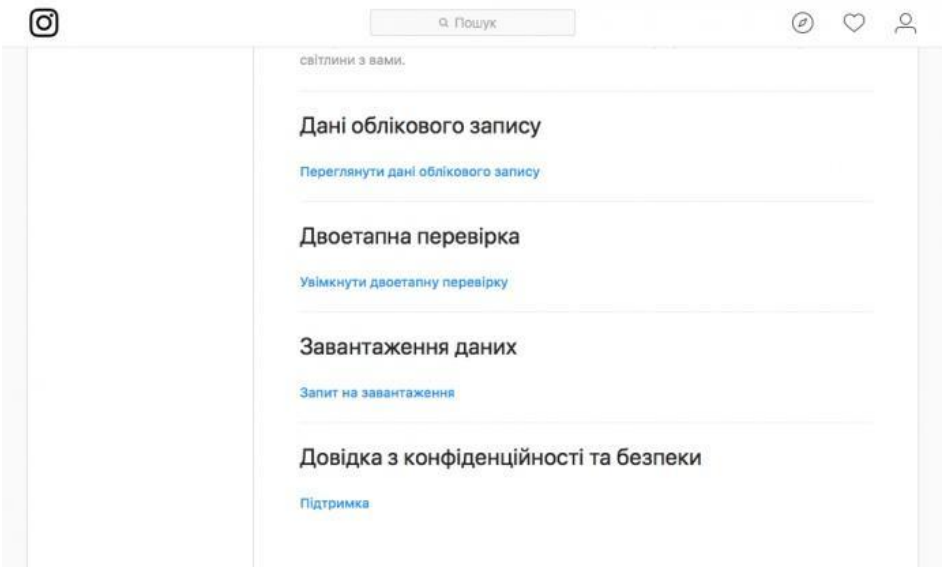

Рисунок 1.5 - Сторінка налаштування у мережі Instagram

Дослідження особливостей налаштування безпеки сторінок користувачів у соціальних мережах, таких як Facebook, YouTube та Instagram, вказує на відсутність комплексного механізму захисту облікових записів. Жодна з соціальних мереж не пропонує інструментів, що дозволяли б

користувачам автоматично перевіряти та налаштовувати засоби контролю безпеки на своїх сторінках.

Існуючі засоби контролю безпеки для сторінок користувачів у соціальних мережах недостатньо ефективні в забезпеченні повноцінного захисту облікових записів. Під час налаштування безпеки не враховуються важливі параметри, такі як перевірка витоків пароля та адреси електронної пошти, на яку зареєстровано обліковий запис, а також складність та регулярність зміни пароля. Зазначені контрольні точки рекомендується періодично перевіряти та налаштовувати відповідно до порад фахівців з інформаційної безпеки [14,17-20].

Однією з ефективних стратегій для створення такого механізму є використання чат-бота, який автоматично допоможе користувачам перевіряти та налаштовувати засоби контролю безпеки на їхніх сторінках у соціальних мережах.

1.4 Постановка задачі дослідження

Аналіз, проведений у параграфі 1.1, виявив, що більшість кібератак спрямовані на особисті дані та приватну інформацію користувачів соціальних мереж. У параграфі 1.2 були розглянуті засоби контролю безпеки для сторінок користувачів у соціальних мережах, таких як Facebook, YouTube та Instagram, і виявлена відсутність ряду важливих контролів безпеки на стороні серверів цих соціальних мереж. Ці контролі безпеки реалізовані сторонніми розробниками та надаються як окремі інтернет-сервіси.

Результати дослідження особливостей налаштування безпеки сторінок користувачів у соціальних мережах Facebook, YouTube та Instagram, представлені у параграфі 1.3, свідчать про відсутність комплексного механізму захисту облікових записів, який дозволяв би користувачам автоматично перевіряти та налаштовувати засоби контролю безпеки на своїх сторінках.

Аналіз літератури [5-8,14,17-20] підкреслює, що в теорії та практиці захисту персональних сторінок користувачів у соціальних мережах існують невирішені проблеми, такі як відсутність класифікації загроз та відсутність моделей оцінки рівня безпеки, які враховують рекомендації фахівців із інформаційної безпеки.

З урахуванням вищезазначених недоліків, науковою проблемою, яка потребує вирішення, стає розробка математичного та програмного забезпечення для підвищення рівня безпеки персональних сторінок користувачів у соціальних мережах.

2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНКИ РІВНЯ БЕЗПЕКИ ОБЛІКОВИХ ЗАПИСІВ КОРИСТУВАЧІВ СОЦІАЛЬНИХ МЕРЕЖ

2.1 Експертна оцінка рівня безпеки персональних сторінок користувачів у соціальних мережах

У кваліфікаційній роботі, яка стосується оцінки контролів безпеки персональних сторінок користувачів у соціальних мережах, використовується метод безпосередньої оцінки [21]. Цей метод включає процедуру визначення узагальнюючого показника на основі первинних показників, які характеризують якість функціонування та оптимізацію складних систем.

Для оцінки контролів безпеки, експертами в галузі інформаційної безпеки, які працюють на кафедрі кібербезпеки Західноукраїнського національного університету, виставлялися оцінки від одного до десяти за кожним показником. Це може включати такі аспекти, як ефективність перевірки витоків пароля, налаштування безпеки адреси електронної пошти, регулярність зміни пароля та інші.

Далі за кожним показником бали додавались, а потім визначався середній узагальнюючий показник за допомогою вагових коефіцієнтів. Вагові коефіцієнти визначалися з урахуванням значимості кожного показника, і ця процедура може ґрунтуватися на експертній думці або об'єктивних розрахунках [21,22].

Цей підхід дозволяє систематизувати та узагальнити оцінки експертів для отримання комплексного уявлення про рівень безпеки персональних сторінок користувачів у соціальних мережах.

$$C_i = \frac{\sum_{i=1}^{n} C_i}{N} \tag{2.1}$$

де N – кількість опитаних експертів,

C_i – сума балів для кожного засобу контролю безпеки.

Результати експертної оцінки контролів безпеки персональних сторінок користувачів у соціальних мережах зведено в таблиці 2.1.

Таблиця 2.1 - Результати експертної оцінки контролів безпеки

Засоби контролю безпеки	Експертні бали					Середній показник
	1	2	3	4	5	
Двоетапна перевірка	10	10	10	10	10	10
Приватний обліковий запис	9	8	10	9	10	9,2
Сповіщення про безпеку та конфіденційність	9	9	7	10	9	8,8
Перевірка авторизованих входів	8	9	8	8	9	8,4
Формування довірених контактів	6	7	8	5	8	6,8
Генерування кодів ідентифікації	5	6	6	6	4	5,4
Перевірка складності пароля	8	10	7	9	9	8,6
Перевірка витоків пароля	7	8	8	10	9	8,4
Перевірка витоків даних для адреси електронної скриньки	10	9	9	8	10	9,2
Регулярна зміна паролю	9	8	7	7	8	7,8
Перевірка доступу зовнішніх застосунків до облікового запису	9	9	8	9	9	8,8

Як показано в таблиці, ключовим показником, що визначає рівень безпеки персональної сторінки користувача в соціальній мережі, є налаштування контролю двоетапної перевірки, тоді як найменш значущим є генерація кодів ідентифікації. Фактичні значення показників, обчислені за формулою 2.1, показують, що максимально можливий бал рівня безпеки персональних сторінок користувачів у соціальних мережах становить 91,4.

Оцінка рівня безпеки персональних сторінок користувачів у соціальних мережах проводиться за допомогою графічного методу, який дозволяє

виділити зони нормального, допустимого та незадовільного рівня [23]. Критичні значення вихідних показників визначаються враховуючи мінімально припустимий рівень безпеки.

Комплексна оцінка рівня безпеки персональних сторінок користувачів у соціальних мережах може бути розрахована як середньозважена величина всіх отриманих значень рівня безпеки Сі згідно з формулою:

$$K = \sum C_i \tag{2.2}$$

Згідно зі значенням К, можна виділити різні рівні безпеки для персональних сторінок користувачів у соціальних мережах:

1) абсолютна безпека: Рівень безпеки, при якому всі контролі безпеки для персональної сторінки користувача в соціальній мережі налаштовані повністю;

2) достатній рівень безпеки: Рівень, при якому налаштовані всі контролі безпеки, які впроваджені на сервері соціальної мережі, а також ряд зовнішніх контролів;

3) допустимий рівень безпеки: Рівень, при якому налаштовані базові контролі безпеки, що впроваджені на сервері соціальної мережі;

4) задовільний рівень безпеки: Рівень, при якому налаштовані критичні контролі безпеки, впроваджені на сервері соціальної мережі;

5) незадовільний рівень безпеки: Критично низький рівень безпеки.

Таблиця 2.2 містить оцінку рівня безпеки відповідно до налаштованих контролів безпеки персональних сторінок користувачів у соціальних мережах. Враховуючи отримані результати, можна сформулювати інтервали, що характеризують рівні безпеки відповідно до налаштованих контролів безпеки, як показано в таблиці 2.3.

Таблиця 2.2 - Оцінка рівня безпеки відповідно до налаштованих контролів безпеки персональних сторінок користувачів у соціальних мережах

Засоби контролю безпеки	Серед-ній бал	Рівень безпеки Facebook					Рівень безпеки YouTube					Рівень безпеки Instagram				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Двоетапна перевірка	10	+	+	+	+	-	+	+	+	+	-	+	+	+	+	-
Приватний обліковий запис	9,2	+	+	+	+	-	+	+	+	+	-	+	+	+	+	-
Сповіщення про безпеку та конфіденційність	8,8	+	+	+	-	-	+	+	+	-	-	-	-	-	-	-
Перевірка авторизованих входів	8,4	+	+	+	-	-	+	+	+	-	-	-	-	-	-	-
Формування довірених контактів	6,8	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-
Генерування кодів ідентифікації	5,4	+	+	+	-	-	+	-	-	-	-	+	-	-	-	-
Перевірка складності пароля	8,6	+	+	+	+	-	+	+	+	+	-	+	+	+	-	-
Перевірка витоків пароля	8,4	+	+	-	-	-	+	+	-	-	-	+	+	-	-	-
Перевірка витоків даних для адреси електронної скриньки	9,2	+	-	-	-	-	+	-	-	-	-	+	-	-	-	-
Регулярна зміна паролю	7,8	+	-	-	-	-	+	-	-	-	-	+	+	-	-	-
Перевірка доступу зовнішніх застосунків до облікового запису	8,8	+	+	-	-	-	+	+	-	-	-	+	+	-	-	-
Максимальний бал рівня безпеки		91,4	74,8	50,8	27,8	0	84,6	62,2	45	27,8	0	67,4	52,8	45	19,2	0

Таблиця 2.3 - Інтервальна оцінка рівнів безпеки

Рівень безпеки	Значення інтервальної оцінки		
	Facebook	YouTube	Instagram
Абсолютна безпека	K=91,4	K=84,6	K=67,4
	100 %	100 %	100%
Достатній рівень безпеки	91,4 < K < 74,8	84,6 < K < 62,2	67,4 < K < 52,8
	100 % < K% < 82 %	100 % < K% < 74 %	100 % < K% < 78%
Допустимий рівень безпеки	74,8 < K < 50,8	62,2 < K < 45	52, 8 < K < 45
	82 % < K% < 55%	74 % < K % < 53 %	78% < K% < 66 %
Задовільний рівень безпеки	50,8 < K < 19,2	45 < K < 27,8	45 < K < 19,2
	55 % < K% < 30 %	53 % < K% < 33%	66% < K% < 28 %
Незадовільний рівень безпеки	27,8 < K < 0	27,8 < K < 0	19,2 < K < 0
	30% < K% < 0 %	33% <K% <0	28 % < K% < 0

2.2 Алгоритми для оцінки рівня безпеки облікових записів користувачів у соціальних мережах

Вхідними даними для алгоритму оцінки та підвищення рівня безпеки сторінок користувачів у соціальних мережах є контролі безпеки та їх середній бал, який обчислено на основі експертної оцінки у параграфі 2.1. Розроблений алгоритм призводить до отримання кількісного результату оцінки рівня безпеки. Робота алгоритму включає 12 етапів, під час яких користувач отримує інструкції та рекомендації щодо підвищення рівня безпеки своєї персональної сторінки. Бали за кожне питання сумуються, що визначає рівень безпеки персональної сторінки користувача.

Таблиця 2.4 відображає схему роботи алгоритму та є основою для програмного забезпечення, яке буде реалізоване у вигляді чат-боту для оцінки та підвищення рівня безпеки сторінок користувачів у соціальних мережах. Ця схема включає в себе зв'язок між контролями безпеки, їх кількісними показниками та запитаннями, які будуть використовуватися в програмній системі.

Алгоритми дій для трьох соціальних мереж (Facebook, YouTube та Instagram) подано в таблиці 2.4. На кожному етапі алгоритму користувачеві персональної сторінки у соціальній мережі надаються рекомендації та присвоюється відповідний бал залежно від його відповіді на запитання. Останнім етапом алгоритму є оцінка рівня безпеки персональної сторінки та надання рекомендацій щодо її підвищення, базуючись на отриманій інтервальній оцінці, поданій в таблиці 2.3.

Таблиця 2.4 - Схема роботи алгоритму

№	Назва контролю безпеки	Серед-ній бал	Зміст запитання	Відповідь на запитання	Дія		
					Facebook	YouTube	Instagram
1	2	3	4	5	6	7	8
1	Двоетапна перевірка	10	Чи налаштована двоетапна перевірка?	Так	Перехід до № 2		
				Ні	Налаштуйте		
2	Приватний обліковий запис	9,2	Чи є обліковий запис персональної сторінки у соціальній мережі приватним?	Так	Перехід до № 3		Перехід до №6
				Ні	Налаштуйте		Налаштуйте
3	Сповіщення про безпеку та конфіденційність	8,8	Чи налаштовано сповіщення про входи до облікового запису із інших пристроїв?	Так	Перехід до № 4		-
				Ні	Налаштуйте		
4	Перевірка авторизованих входів	8,4	Чи виконується перевірка авторизованих входів?	Так	Перехід до № 5	Перехід до №6	-
				Ні	Перевірте	Перевірте	
5	Формування довірених контактів	6,8	Кількість налаштованих довірених контактів	Жодного	Налаштуйте	-	-
				До трьох	Налаштуйте		
				Більше трьох	Перехід до № 6		
6	Генерування кодів ідентифікації	5,4	Чи згенеровано код ідентифікації ?	Так	Перехід до № 7		
				Ні	Налаштуйте		

Продовження табл. 2.4

1	2	3	4	5	6	7	8
7	Перевірка складності пароля	8,6	Чи відповідає пароль вимогам складності?	Так	Перехід до № 8		
				Ні	Змініть пароль		
8	Перевірка витоків пароля	8,4	Чи використовується надійний пароль ?	Так	Перехід до № 9		
				Ні	Змініть пароль		
9	Перевірка витоків даних для адреси електронної скриньки	9,2	Чи були витоки даних пов'язані з електронною адресою на яку зареєстрована персональна сторінка	Так	Змініть e-mail		
				Ні	Перехід до № 10		
10	Регулярна зміна пароля	7,8	Чи регулярно змінюється пароль до облікового запису ?	Так	Перехід до № 11		
				Ні	Змініть пароль		
11	Перевірка доступу зовнішніх застосунків до облікового запису	8,8	Чи є доступ зовнішніх застосунків до облікового запису?	Так	Перехід до № 12		
				Ні	Перевірте		
12	Оцінка рівня безпеки та рекомендації щодо його підвищення						

2.3 Розробка архітектури програмного забезпечення

Архітектура програмного забезпечення визначає сукупність компонентів програми, їх зв'язки та способи організації інформаційного обміну між ними. Проектування програмних сервісів виконується етапно з використанням блочно-ієрархічного підходу, що передбачає розробку продукту частинами.

Перший етап призначений для визначення складових модулів програми. Процес проектування розпочинається з уточнення структури програми, включаючи визначення структурних компонентів та їх взаємодію. Структурна схема відображає цю структуру та взаємодію між частинами.

Структурні компоненти програмного продукту можуть включати програми, підсистеми, бази даних, бібліотеки ресурсів і інше. Структурна схема програми часто показує наявність підсистем та інших компонентів, а розробка її виконується методом покрокової деталізації.

На рисунку 2.1 можна побачити приклад структурної схеми програмного забезпечення, де зображено різні структурні компоненти, такі як програми, підсистеми, бази даних та бібліотеки ресурсів. Визначення структурних компонентів та їх зв'язків є ключовим етапом у створенні ефективної архітектури програми.

Рисунок 2.2 представляє схему обробки інформації елементами програмного забезпечення. У цій схемі можна побачити, як компоненти взаємодіють між собою та з зовнішнім середовищем для обробки інформації.

HTTP-запити використовуються для збереження та передачі інформації. GET-запити використовуються для отримання даних з віддаленого сервера, і ці дані тимчасово зберігаються в програмному сервісі. Асинхронні HTTP-запити дозволяють виконувати окремі завдання асинхронно, не блокуючи основний потік виконання. Це особливо важливо для графічних програм, де блокування основного потоку може призвести до затримок у відгуку інтерфейсу користувача.

Асинхронність також важлива для веб-додатків при обробці запитів від користувачів та зверненні до баз даних чи мережевих ресурсів. Вона дозволяє

ефективно використовувати час при тривалих операціях, не блокуючи роботу інших частин програми.

Ця асинхронна модель дозволяє оптимізувати роботу програми, забезпечуючи високий рівень продуктивності та відгуку.

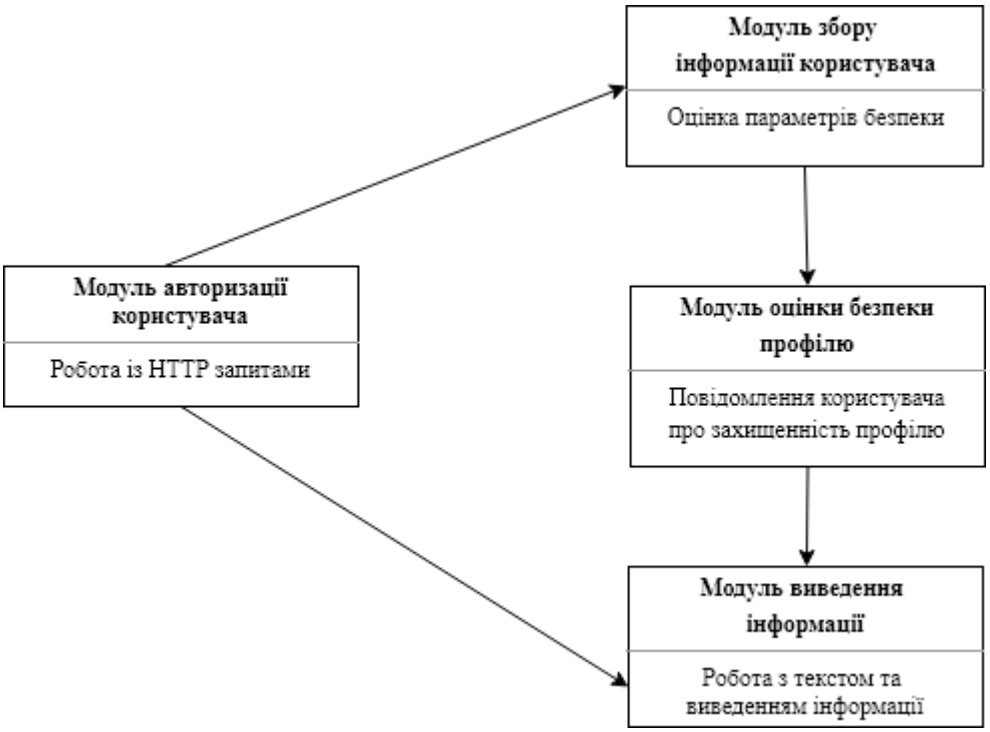

Рисунок 2.1 - Структурна схема програмного забезпечення

При великих запитах до бази даних або інших мережевих операцій, які можуть займати тривалий час, асинхронний метод дозволяє основному потоці продовжувати свою роботу, не чекаючи завершення цих операцій. Це робить веб-додаток більш ефективним і здатним обробляти більше запитів одночасно.

У синхронному додатку, якщо операції, такі як отримання даних з бази даних, виконуються в основному потоці, цей потік буде заблокований, поки операція не завершиться. Це може призвести до затримок у відгуку системи та погіршити взаємодію з користувачем. Асинхронний підхід допомагає уникнути цих проблем і підвищити швидкодію та відгук системи.

Рисунок 2.2 - Схема процесу обробки інформації елементами програми

Для роботи з асинхронними викликами в C# використовуються два ключові слова: async і await. Їх завданням є спростити написання асинхронного коду, і вони використовуються в парі для створення асинхронних методів. Зображення моделі класів, яке відображає зв’язок між класами описаного програмного забезпечення, представлено на рисунку 2.3.

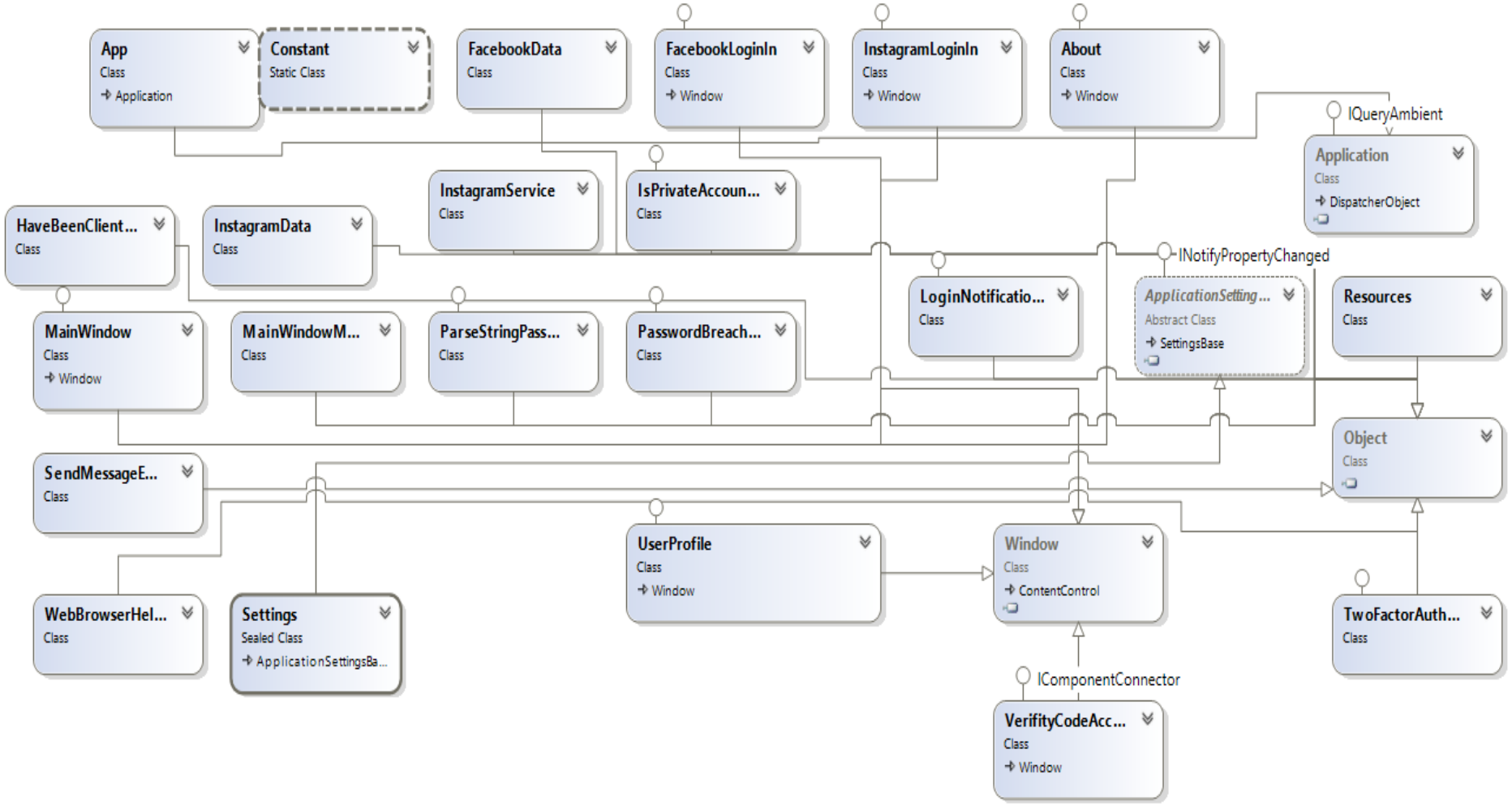

Рисунок 2.3 - Модель класів

Для більш чіткого визначення взаємодії між модулями програми в процесі створення програмного забезпечення, ми розробимо алгоритм, який представлятиме основні етапи обробки даних. Алгоритм - це чітке та точне послідовне завдання виконавцю виконати дії, спрямовані на досягнення конкретної мети чи вирішення поставленої задачі.

Будь-який виконавець, включаючи комп'ютер, обмежений конкретним набором операцій (наприклад, екскаватор копає яму, вчитель викладає, комп'ютер виконує арифметичні дії). Таким чином, алгоритми повинні відповідати наступним вимогам:

- зрозумілість: Виконавець повинен мати змогу розуміти та виконувати кожну команду алгоритму для досягнення поставленої мети;
- визначеність (однозначність): Алгоритм не повинен містити неоднозначних вказівок. Кожна команда повинна мати чітке та однозначне тлумачення, і виконавець повинен точно знати, що робити на кожному кроці;
- дискретність: Алгоритм повинен розбивати виконання завдання на прості, послідовні елементарні дії, які виконуються послідовно. Виконання кожної наступної дії залежить від виконання попередньої;
- масовість: Алгоритм повинен бути спроектований так, щоб розв'язувати широкий спектр задач даного типу, а не лише конкретну задачу;
- результативність: Виконання алгоритму повинно завершуватися отриманням кінцевих результатів. Уникайте ситуацій, які можуть призвести до "зациклення" алгоритму.

Алгоритм функціонування програмного забезпечення відображає послідовності виконання процесів через блок-схеми з найбільш важливими процедурами і функціями (рисунок 2.4).

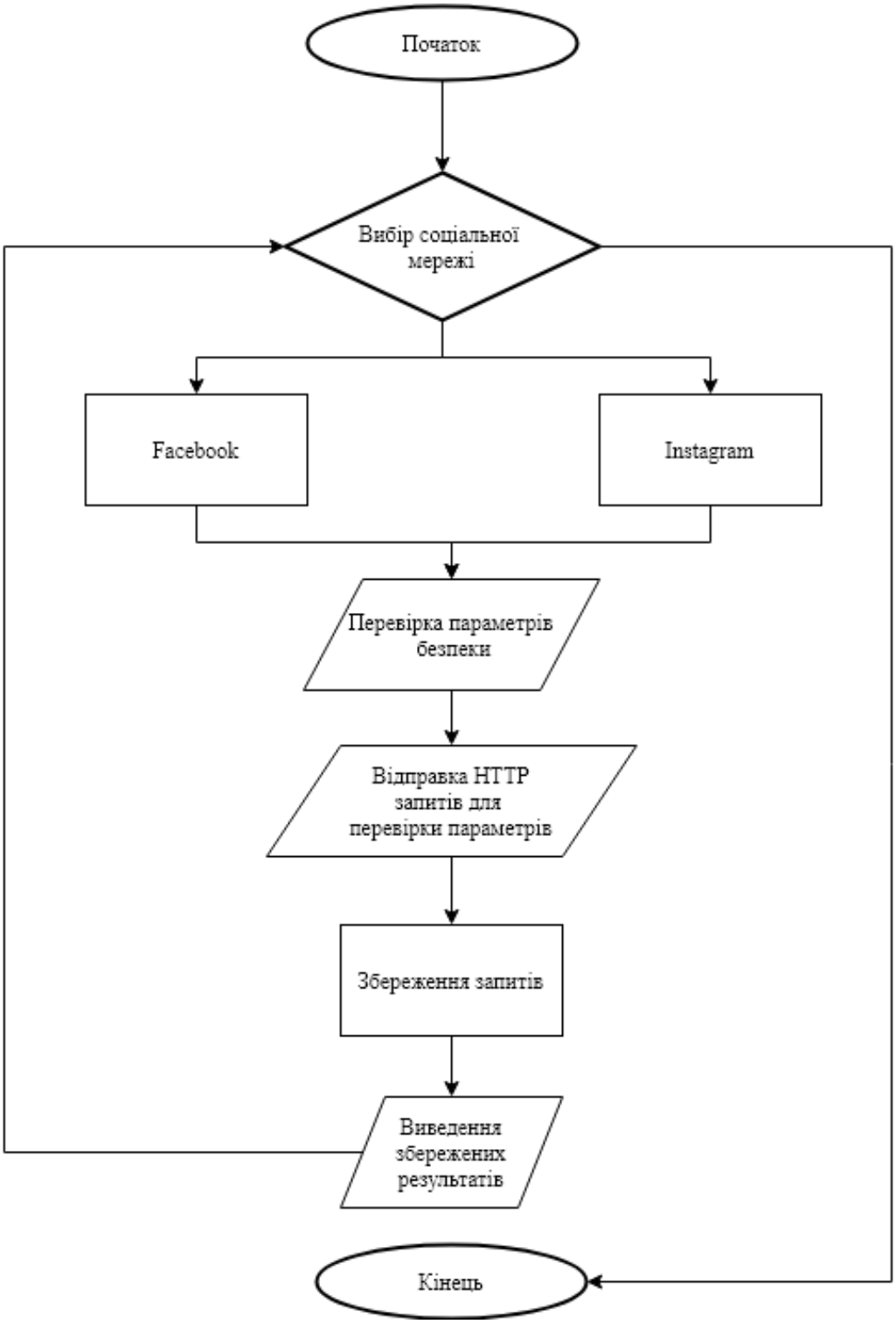

Рисунок 2.4 - Алгоритм функціонування програми

Опишемо основні етапи роботи розробленого алгоритму. Початок алгоритму включає у себе завантаження розробленого програмного сервісу. Далі користувач обирає соціальну мережу, зокрема Facebook або Instagram. Після того як користувач авторизується в обраній мережі, програмний сервіс проводить

перевірку всіх параметрів безпеки в профілі користувача та надсилає HTTP-запити для перевірки цих параметрів.

У випадку успішної перевірки даних, програмний сервіс повідомляє користувача про вразливі точки в безпеці його профілю. Це дає користувачеві можливість покращити захищеність своєї особистої сторінки.

Запропонована архітектура передбачає об'єднання модулів, які працюють в різноманітних середовищах та, відмінно від існуючих, забезпечує можливість автоматично підвищувати рівень захисту сторінок користувачів у соціальних мережах. Це здійснюється з урахуванням зовнішніх контролів безпеки.

На зображенні 2.5 представлена розроблена архітектура програмного забезпечення, яка складається з трьох рівнів:

- рівень інтерфейсу користувача (User Interface): Відповідає за реалізацію графічного інтерфейсу користувача;
- рівень логіки області (Domain Logic): На цьому рівні реалізована структурна та алгоритмічна частина програмного сервісу. Модуль Business Logic Services відповідає за обробку логіки виконання запитів до бази даних. Модуль Process Centric Services управляє логікою виконання запитів до модулів Business Logic Services та Storage Service;
- рівень джерел даних (Datasources): Представлений базами даних та модулями, що взаємодіють з ними. Модуль Storage Services відповідає за аналіз та запис даних з особистої сторінки користувача у базу даних PostgresServer. Local Database Services - це локальні служби баз даних, які використовуються для аналізу вхідних даних у Storage Services та їх подальшого запису в базу даних. Модуль Adapter Services отримує та конвертує дані з серверів, на яких реалізовані API-функції соціальних мереж (Facebook API, Instagram API), а також API функцій сервісу Have I Been Pwned.

.

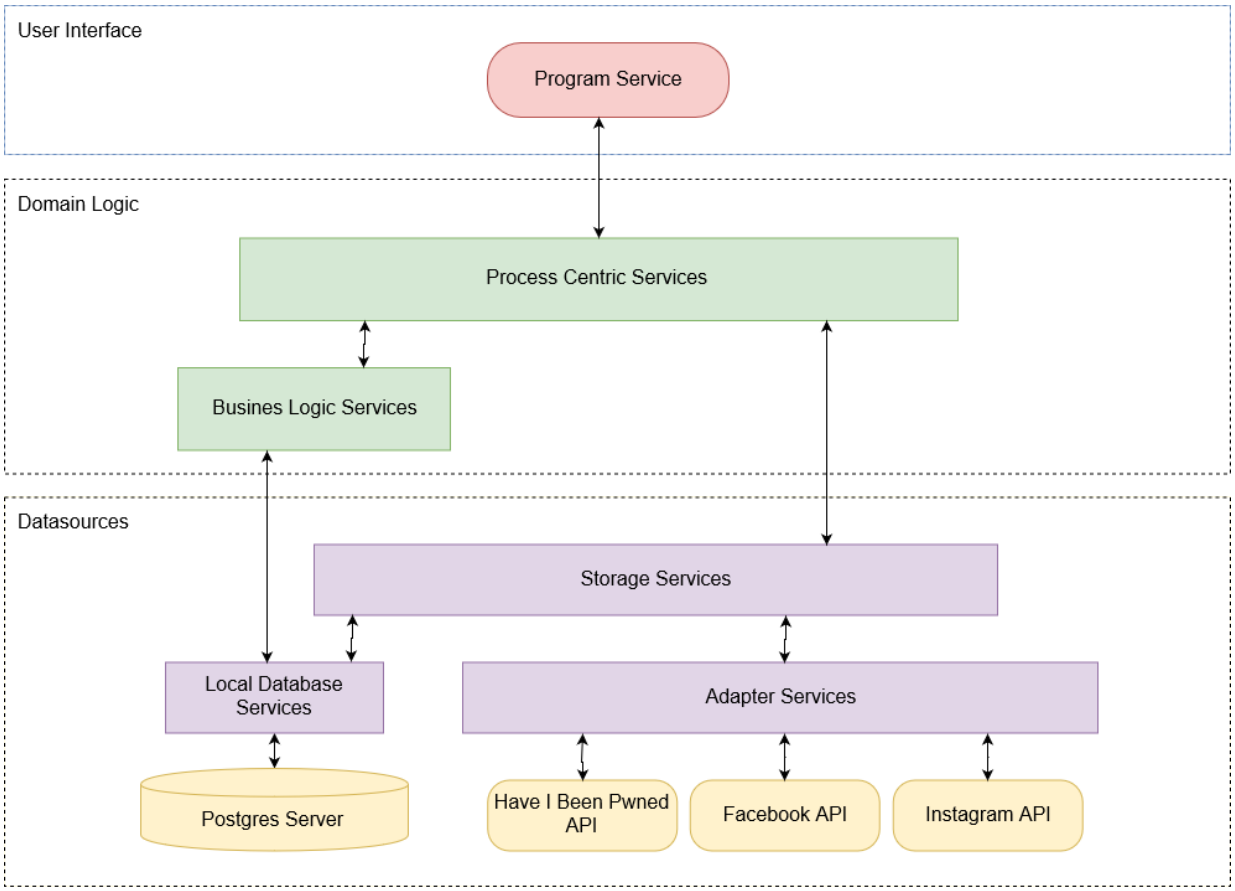

Рисунок 2.5 - Архітектура програмного забезпечення

Діаграма послідовностей, зображена на рисунку 2.6, наочно відображає взаємодії між програмними модулями під час типового використання. Ця діаграма подає об'єкти взаємодії, розташовані у часовій послідовності, та ілюструє послідовні запити, які обмінюються між об'єктами для виконання функціональності сценарію.

У конкретному випадку, протокол використання соціальних медіа Facebook та Instagram включає фреймворк OAuth 2.0, який також є вбудованим у програму. Зазвичай додаток не є ресурсом соціальних медіа, і власник виступає від імені користувача. Додаток запитує доступ до необхідних ресурсів, які контролюються користувачем та розташовані на сервері. У цьому контексті для отримання доступу до захищених ресурсів використовується програма, яка отримує так званий маркер доступу до облікових даних користувача.

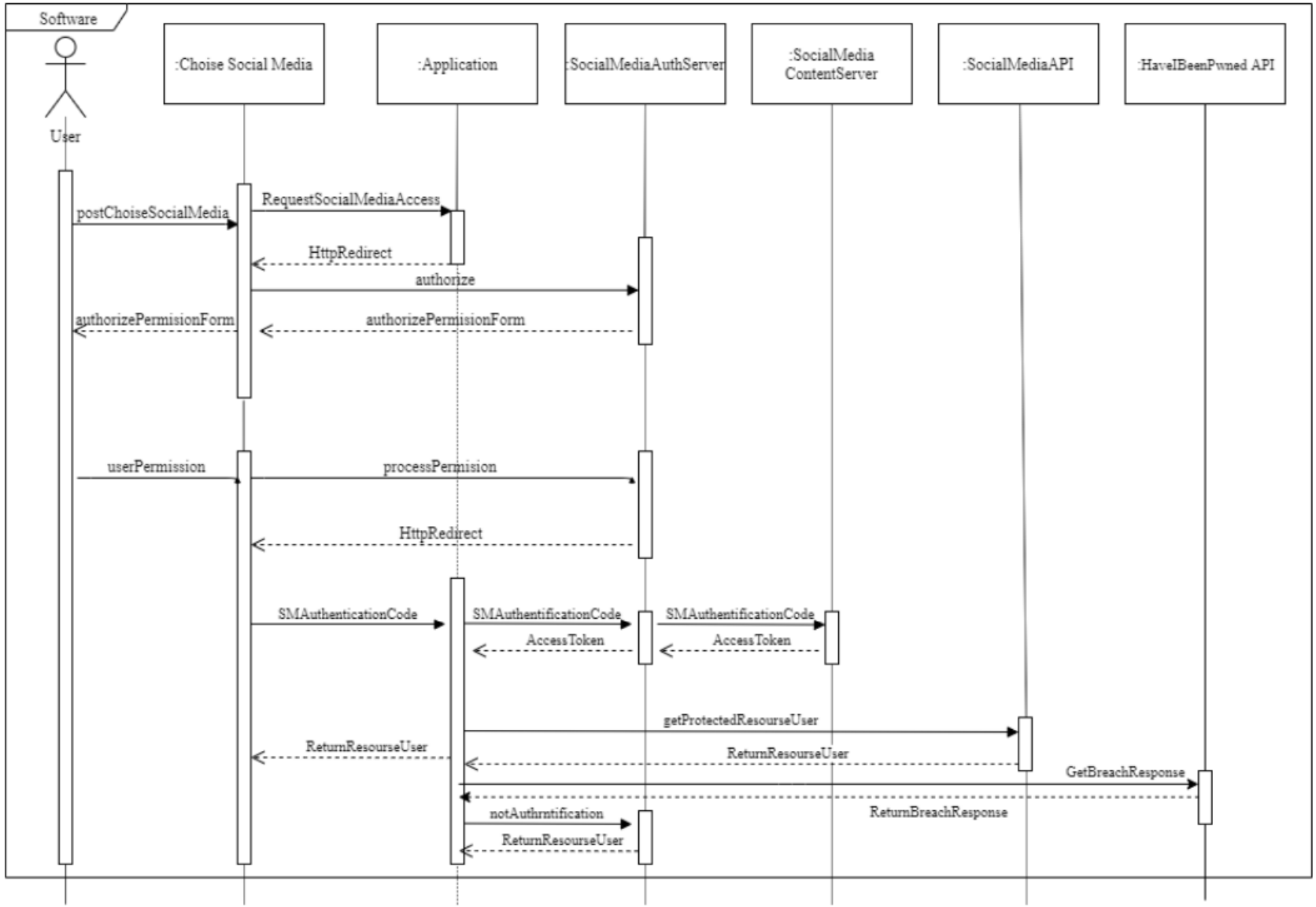

Рисунок 2.6 - Діаграма послідовностей програмного забезпечення

Розроблене програмне забезпечення зареєстроване в соціальних мережах за допомогою ідентифікатора програми та ключа (client_id та client_secret). При отриманні доступу користувача, його перенаправляють на сервер авторизації, використовуючи ідентифікатор програми та URL-адресу програми. Цей URL використовується для перенаправлення користувача після завершення авторизації, після чого користувач отримує форму запиту на дозвіл.

У випадку, якщо програма отримала повноваження від користувача для отримання даних, сервер авторизації повертається до URI із "рядком перевірки", який містить код авторизації. Додаток може змінювати цей код авторизації для отримання маркера доступу OAuth.

Якщо веб-додаток отримав маркер доступу до облікового запису соціальних медіа, він може виконувати авторизовані запити від імені користувача, включаючи використання маркера доступу у API-запитах до Facebook, Instagram та HaveIBeenPwned.

У випадку, якщо користувач не надав авторизацію веб-програмі, запит автоматично перенаправляється на попередньо вказане URI з серверів соціальних мереж, з додаванням параметра помилки для повідомлення про відмову у запиті на авторизацію.

Основні функції програмного забезпечення включають приватну перевірку профілю, перевірку надійності пароля, аналіз паролів на порушення та перевірку порушень електронної пошти. Ключовою частиною програми є запити до сервера веб-сервісу для перевірки коректності даних користувача. Процес відправки запитів розбивається на етапи, включаючи створення об'єкту Web Request за допомогою методу CREATE, в який передається адреса електронного ресурсу чи об'єкту. Для отримання інформації, специфічної для протоколу HTTP, використовуються два класи: HttpWebRequest і HttpWebResponse, які успадковуються від WebRequest і WebResponse відповідно.

API (Application Programming Interface) - це інтерфейс програмування додатків, який включає в себе набір визначень підпрограм, протоколів взаємодії та ресурсів для створення програмного забезпечення. У більш простих термінах, API представляє собою набір чітко визначених методів для взаємодії різних компонентів. Це забезпечує розробникам засоби для ефективної розробки програмного забезпечення. API може бути призначеним для веб-систем, операційних систем, баз даних, апаратного забезпечення та програмних бібліотек.

У таблиці 2.5 наведено коротку характеристику запитів API, яка вказує на основні особливості та властивості, що стосуються взаємодії з API.

Таблиця 2.5 - Зведена таблиця запитів API

API запит	Опис	Текст запиту	Текст відповіді
GET /api	Отримання всіх елементів запиту	Відсутній	Масив елементів запиту
GET /api/{id}	Отримання об’єкту по індексу	Відсутній	Елемент запиту
POST /api/	Додавання нового елементу	Елемент запиту	Елемент запиту
PUT /api{id}/	Оновлення існуючого елементу	Елемент запиту	Відсутній
DELETE /api/{id}	Видалення елементу	Відсутній	Відсутній

Розглянемо детальніше кожен із цих запитів:

- GET (Отримання): Використовується для отримання (або читання) даних ресурсу. У випадку "вдалої" адреси, GET повертає дані ресурсу у форматі XML або JSON разом із кодом стану HTTP 200 (OK). У випадку помилок, зазвичай повертається код 404 (NOT FOUND) або 400 (BAD REQUEST);

- POST (Створення): Цей запит найчастіше використовується для створення нових ресурсів, особливо вкладених ресурсів. Практично він використовується для створення ресурсу, пов'язаного з батьківським ресурсом. POST запит відправляється до батьківського ресурсу, і сервіс встановлює зв'язок між створюваним ресурсом та батьківським ресурсом, призначаючи новий ресурс;
- PUT (Оновлення): Зазвичай використовується для оновлення ресурсу. Тіло PUT-запиту, надісланого до існуючого ресурсу, повинно містити оновлені дані про оригінальний ресурс, будь то повністю або частково;
- DELETE (Видалення): Використовується для видалення ресурсу, ідентифікованого конкретним URI (ID). При успішному видаленні повертається HTTP-код 200 (OK) разом із тілом відповіді, що містить дані видаленого ресурсу. Також можливе використання HTTP-коду 204 (NO CONTENT) без тіла відповіді, що сприяє економії трафіку.

На рисунку 2.7 наведено структуру API.

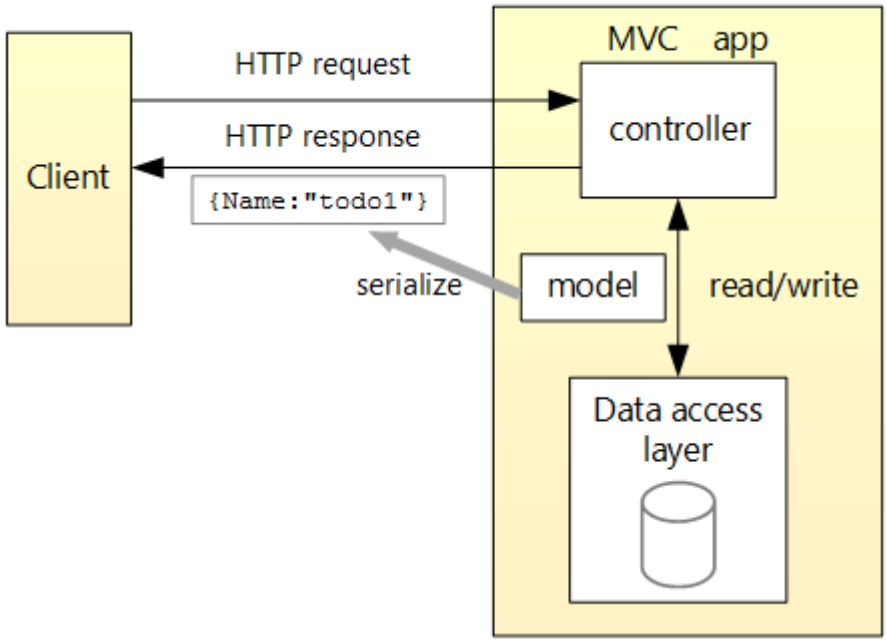

Рисунок 2.7 – Структура API

3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДВИЩЕННЯ РІВНЯ БЕЗПЕКИ ОБЛІКОВИХ ЗАПИСІВ КОРИСТУВАЧІВ У СОЦІАЛЬНИХ МЕРЕЖАХ

3.1 Обґрунтування вибору засобів розробки програмного забезпечення

Для розробки програмного забезпечення для управління запасами була обрана мова програмування C#, технологія Microsoft .NET та середовище розробки Visual Studio Community 2022.

Visual Studio Community 2022 володіє високою гнучкістю та універсальністю у системі проектування. Це означає, що можна легко включати HTML-дескриптори та обробники коду в один чи кілька файлів, не обмежуючи можливостей використання Visual Studio та отримання переваг від корисних властивостей. Розробник може програмувати за допомогою декількох мов програмування в одному проекті, зручно переміщуючи веб-сторінки між C# та Visual Basic.

Це середовище розробки має велику кількість корисних модулів та компонентів для розробки програмних продуктів. Застосовує об’єктно-орієнтований підхід, що гарантує стабільність роботи та легкість у налаштуванні майбутнього програмного сервісу. Розробник отримує широкий спектр можливостей для візуального оформлення інтерфейсу, групування інформації у вікні програмного сервісу та стабільної взаємодії програмного забезпечення із сервером.

Переваги ASP.NET:

- висока швидкість: ASP.NET відзначається високою швидкістю порівняно з іншими технологіями, що базуються на скриптах. Це зумовлено тим, що код на стороні сервера компілюється в одну чи декілька DLL, що покращує продуктивність;
- ефективні користувацькі елементи управління: використання користувацьких елементів управління дозволяє легко виділяти шаблони, такі як меню сайту, що робить розробку більш зручною. У порівнянні з PHP,

ASP.NET надає більше можливостей для організації користувацьких інтерфейсів;

- розширений набір елементів управління та бібліотек класів: ASP.NET має широкий набір доступних елементів управління та бібліотек класів, що сприяє швидкій розробці додатків;
- багатомовність: ASP.NET використовує можливості .NET для написання коду сторінки на різних мовах програмування, таких як VB.NET, C#, J#, що дозволяє розробникам вибирати мову за їхніми вподобаннями;
- можливість кешування: ASP.NET надає можливість кешування всієї сторінки або її частин, що покращує продуктивність та ефективність використання ресурсів;
- розділення візуальної та логічної частини: можливість розділення візуальної та логічної частини («code behind») спрощує роботу з кодом та забезпечує більшу чіткість;
- об’єктно-орієнтована мова: мова програмування для розробки на платформі ASP.NET є повноцінною об’єктно-орієнтованою, що полегшує структурування та розробку коду;
- інтеграція з .NET Framework: ASP.NET постачається разом з .NET Framework, що дозволяє реалізувати всі його можливості та використовувати розробникам чітко структурований набір інструментів для реалізації функціональності;
- повна підтримка Unicode: ASP.NET має повну підтримку стандарту Unicode, що дозволяє створювати багатомовні додатки. У порівнянні, платформа PHP підтримку Unicode має обмежену, а підтримка обробки виключних ситуацій з'явилася лише в PHP 5.

Для організації інтерфейсу з користувачем програмної системи, яка є веб-орієнтованою, визначено використання стилю WUI (Web User Interface), де навігація здійснюється в рамках одного чи декількох вікон з використанням текстових або візуальних гіперпосилань.

Вимоги до апаратних ресурсів будуть залежати від кількості користувачів системи. Для невеликої кількості користувачів (до 1000) рекомендована конфігурація наступна:

- процесор з тактовою частотою не менше 2 ГГц;
- оперативна пам'ять обсягом 1 ГБ;
- жорсткий диск обсягом не менше 250 ГБ, оскільки система потребує значну кількість місця для зберігання бази даних, відео та зображень;
- пропускна здатність мережі повинна бути достатньою для забезпечення швидкості завантаження 0,25 Мб/с для 1000 користувачів. Для цього необхідне з'єднання з пропускною здатністю 250 Мб/с.

Для функціонування програмного забезпечення необхідно мати один комп'ютер, на якому буде встановлений програмний продукт, а також підключення до Інтернету для можливості посилання запитів на сервер.

3.2 Організація графічного інтерфейсу

Інтерфейс користувача – це засіб зручної взаємодії користувача з інформаційною системою, в основному місце взаємодії людини та комп'ютера. Взаємодії відбуваються в просторі, де користувачі співпрацюють із машинами. Зазвичай, головна мета розробки інтерфейсу користувача полягає в створенні простого (самозрозумілого), ефективного та приємного (зручного для користувача) інтерфейсу для управління машиною з метою досягнення бажаного результату.

Існує безліч видів інтерфейсів, серед яких варто відзначити: інтерфейс прямої обробки, графічні інтерфейси користувача (GUI), веб-інтерфейси користувача (WUI), сенсорні екрани, інтерфейси командного рядка, сенсорний інтерфейс користувача, апаратні інтерфейси, уважні інтерфейси користувача, пакетні інтерфейси, розмовні інтерфейси, агенти діалогового інтерфейсу, об'єктно-орієнтовані інтерфейси користувача (OOUI).

Програмне забезпечення буде побудоване на основі графічного інтерфейсу, що є типом інтерфейсу (GUI), дозволяючи користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки. Це відрізняється від текстових інтерфейсів, що базуються на використанні тексту, текстового набору команд та текстової навігації.

Програмне забезпечення буде включати такі компоненти інтерфейсу:

- кнопки, які будуть використовуватися для виклику певних функцій або дій;
- поля для виведення даних, де буде відображатися інформація для користувача;
- поля для введення даних, які дозволяють користувачеві вводити необхідні відомості;
- головне меню, яке містить основні опції та команди для навігації в програмі;
- елементи формування звітності, які дозволяють генерувати різноманітні звіти та аналітичну інформацію.

Опишемо властивості цих компонентів для ефективної реалізації графічного інтерфейсу програми.

Головне вікно програми відповідає за взаємодію та керування іншими частинами програми. У ньому розміщені різні компоненти, які підтримують функціональність програми. На рисунку 3.1 зображені наступні компоненти головного вікна:

- Image (Зображення): цей компонент представляє собою об'єкт зображення однакових розмірів, на якому можна виконувати перенаправлення. На сторінці міститься кілька таких компонентів, кожен з яких має свою унікальну картинку та розташування на сторінці, що надає індивідуальний та зручний інтерфейс;

- Label (Мітка): цей компонент містить об'єкт текстового формату та відображає будь-який текст, який необхідно показати. Використовується для виведення інформації чи позначення елементів;

- Grid (Таблиця): Компонент інкапсулює двовимірну таблицю, де рядки представляють записи, а стовпці - поля набору даних. Кожна колонка компонента Grid описується класом Column, а сукупність колонок стає доступною через властивість columns компонента. Результатом є головне вікно, яке демонструється на рисунку 3.2.

Ці компоненти спільно створюють інтерфейс програми, забезпечуючи відображення і взаємодію з різними елементами та функціоналом.

```
<Grid Margin="20, 35, 20, 20" VerticalAlignment="Center">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="0.5*"/>
        <ColumnDefinition Width="0.5*"/>
        <ColumnDefinition Width="0.5*"/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="50"/>
        <RowDefinition Height="110"/>
    </Grid.RowDefinitions>
    <Label Content="Security FIT"
           Grid.ColumnSpan="3"
           HorizontalAlignment="Center"
           VerticalAlignment="Center"
           FontSize="30"/>
    <Image x:Name="FacebookLogo" Grid.Row="1"
           Grid.Column="0"
           MouseDown="FacebookLogo_OnMouseDown"
           Source="Images/facebook_circle-512.png"
           Height="109"
           VerticalAlignment="Center" />
    <Image x:Name="TwitterLogo"
           Grid.Row="1"
           Grid.Column="2"
           MouseDown="TwitterLogo_OnMouseDown"
           Source="Images/Twitter.png"
           Height="109"
           VerticalAlignment="Center"  />
    <Image x:Name="InstagramLogo"
           Grid.Row="1"
           Grid.Column="1"
           MouseDown="InstagramLogo_OnMouseDown"
           Source="Images/Instagram_logo.png"
           Height="109"
           VerticalAlignment="Center" />
    <Image Source="Images/icon.png"
           Height="100" Margin="14,-115,0,65"
           VerticalAlignment="Center"
           Grid.Column="1"/>
    <Image x:Name="AboutPage"
           Grid.Column="2"
           Source="Images/about.png"
           MouseDown="About_Click"
           HorizontalAlignment="Left"
           Height="50" Margin="124,-125,0,0"
           VerticalAlignment="Top"
           Width="50"/>
    <Image x:Name="ExitImage"
           Grid.Column="2"
           Source="Images/exit.png"
           MouseDown="Exit_Click"
           HorizontalAlignment="Left"
           Height="50"
           Margin="189,-125,0,0"
           VerticalAlignment="Top" Width="50"/>

</Grid>
```

Рисунок 3.1 - Компоненти головного вікна програми програмного забезпечення

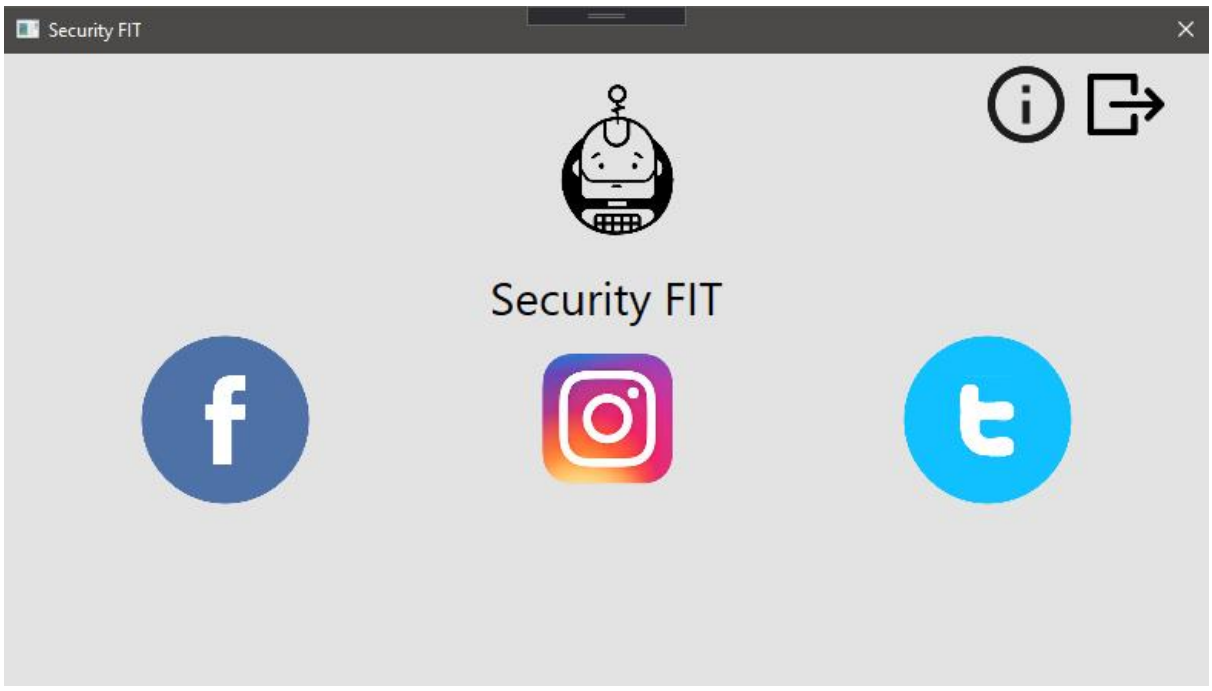

Рисунок 3.2 - Головне вікно програми програмного забезпечення

Вікно авторизації користувача в соціальній мережі (рисунок 3.3) та його інтерфейс (рисунок 3.4) включають наступні компоненти:

- кнопки переходу до соціальних мереж: Кожна кнопка представляє конкретну соціальну мережу, до якої користувач може авторизуватися. Наприклад, кнопки для Facebook, Instagram і HaveIBeenPwnd. Коли користувач клікає на кнопку, його перенаправляє на відповідну сторінку авторизації;

- форма авторизації: Форма, де користувач вводить свої дані для авторизації в обраній соціальній мережі. Зазвичай включає поля для введення ідентифікатора та пароля користувача;

- зображення логотипу соціальної мережі: Додатковий візуальний елемент, який може містити логотип обраної соціальної мережі, щоб користувач був впевнений, що він авторизується в потрібному сервісі.

- кнопка "Назад" або "Скасувати": кнопка, яка дозволяє користувачеві повернутися на попередню сторінку або відмінити процес авторизації, якщо він вирішить це зробити;

```
<Grid VerticalAlignment="Center"
      HorizontalAlignment="Center">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="0.5*"/>
        <ColumnDefinition Width="0.5*"/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="40"/>
        <RowDefinition Height="40"/>
        <RowDefinition Height="40"/>
    </Grid.RowDefinitions>
    <Label Content="Login:"
           VerticalAlignment="Center"/>
    <TextBox x:Name="tbxLogin"
             Height="23"
             Width="220"
             Grid.Column="1"/>
    <Label Content="Password:"
           Grid.Row="1"
           VerticalAlignment="Center"/>
    <PasswordBox Height="23"
                 Width="220"
                 x:Name="pbPassword"
                 PasswordChar="*"
                 Grid.Row="1"
                 Grid.Column="1"/>
    <Button Content="Exit"
            Height="35"
            Margin="0,0,5,0"
            Grid.Row="3"
            Click="OnExit"
            Background="{DynamicResource {x:Static SystemColors.ControlBrushKey}}"/>
    <Button Content="Login"
            Height="35" Grid.Row="3"
            Grid.Column="3"
            Click="OnLoginInButtonClicked"
            Background="{DynamicResource {x:Static SystemColors.ControlBrushKey}}"/>
</Grid>
```

Рисунок 3.3. Компоненти вікна авторизації користувача програми

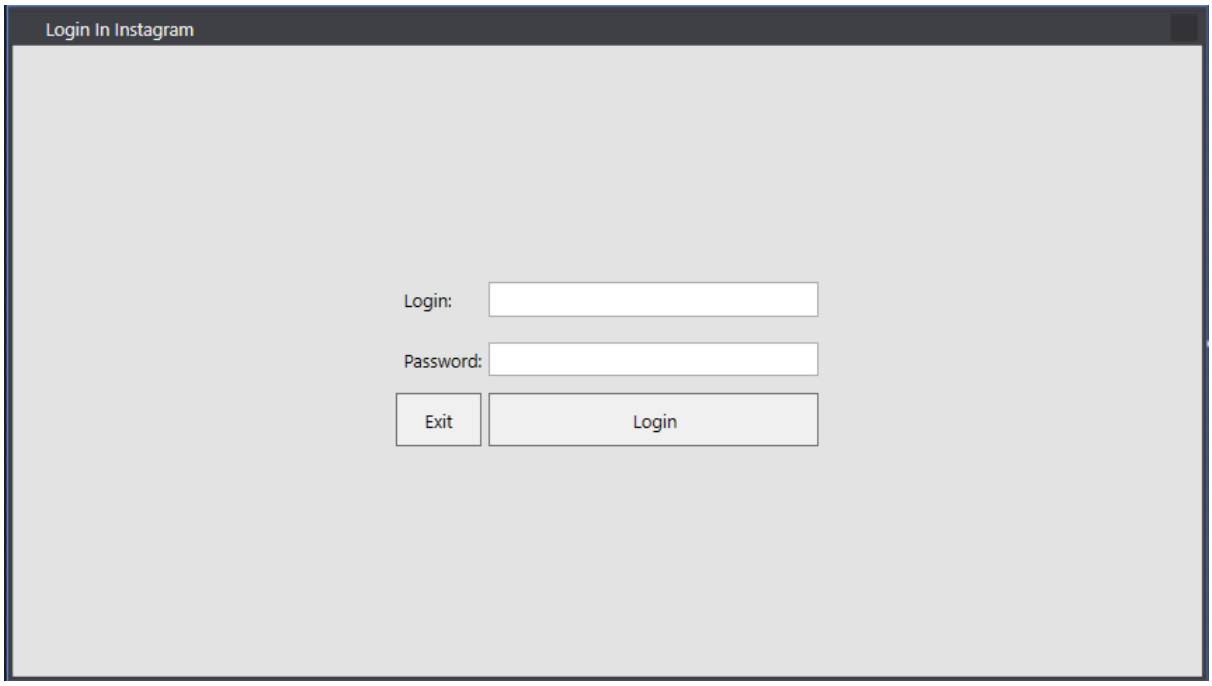

Рисунок 3.4 - Екранна форма «Авторизації користувача»

- інші елементи інтерфейсу: залежно від вимог і дизайну можуть включати інші елементи, такі як текстові підказки, індикатори завантаження, чи інші додаткові компоненти, що поліпшують взаємодію користувача з процесом авторизації.

Ці компоненти спільно створюють інтерфейс авторизації користувача в соціальній мережі, що дозволяє йому безпечно та зручно увійти та мати доступ до своїх особистих даних.

Компоненти вікна авторизації:

- TextBox: цей компонент використовується для введення текстових даних, таких як ідентифікатор користувача чи електронна пошта. Забезпечує користувача текстовим полем, де він може ввести необхідні дані;

- PasswordBox: представляє собою поле для введення паролю користувача. Має атрибут PasswordChar, який дозволяє шифрувати введені символи паролю, забезпечуючи безпеку введених даних;

- Button: цей компонент є кнопкою, яка використовується для запуску процесу авторизації після введення необхідних даних. Може мати текстовий лейбл, який вказує на функцію кнопки, наприклад, "Увійти" або "Авторизуватися".

Ці компоненти використовуються для створення інтерактивного інтерфейсу для користувача, який дозволяє йому введення ідентифікаційних даних та паролю для авторизації в обраній соціальній мережі.

Компоненти вікна "Профілю користувача":

- LabelConntent = «SecuritySettings»: цей компонент є об'єктом текстового поля, яке має текстовий формат "SecuritySettings". Використовується для відображення заголовку частини профілю, яка стосується налаштувань безпеки;

- LabelConntent = «Checkresult»: представляє собою об'єкт текстового поля з текстовим форматом "Checkresult". Використовується для відображення результатів перевірки безпеки в профілі користувача;

- LabelConntent = «Recommendation»: цей компонент є об'єктом текстового поля, яке містить текстовий формат "Recommendation". Використовується для відображення рекомендацій щодо покращення налаштувань безпеки;

- Grid: компонент інкапсулює таблицю із трьома колонками та одинадцятьма стовпцями. Кожен з цих стовпців містить компонент Label, який служить для відображення конкретної інформації користувачеві.

Гнучка структура таблиці дозволяє вміщувати різні компоненти Label для відображення різних аспектів інформації в профілі користувача.

Ці компоненти вікна "Профілю користувача" допомагають структурувати та візуалізувати різні елементи безпеки та рекомендацій для користувача (рисунок 3.5).

```
<Grid Margin="15">
    <Grid.Resources>
        <Style TargetType="Label">
            <Setter Property="FontSize" Value="15"/>
            <Setter Property="HorizontalContentAlignment" Value="Center"/>
        </Style>
    </Grid.Resources>
    <Grid.ColumnDefinitions>
        <ColumnDefinition  Width="*"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
    </Grid.RowDefinitions>

    <Rectangle Grid.Column="0" Grid.ColumnSpan="3">
        <Rectangle.Fill>
            <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                <GradientStop Color="Gray" Offset="0" />
            </LinearGradientBrush>
        </Rectangle.Fill>
    </Rectangle>

    <!--Header Grid Label-->
    <Label Content="Security Setings" FontWeight="Bold"/>
    <Label Content="Check result" Grid.Column="1" FontWeight="Bold"/>
    <Label Content="Recommendation" Grid.Column="2" FontWeight="Bold"/>
```

Рисунок 3.5 - Компоненти вікна «Профіль користувача»

Результат поєднання даних компонентів зображено на рисунку 3.6.

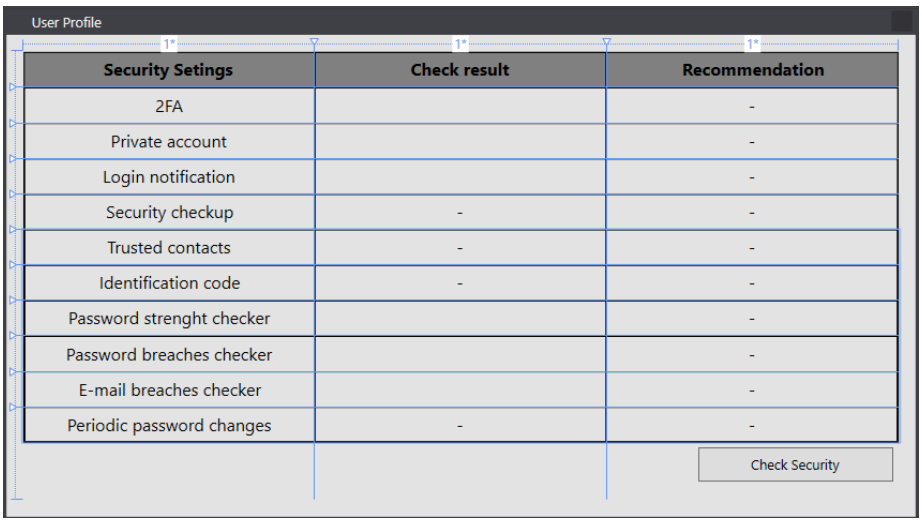

Рисунок 3.6 - Екранна форма «Профіль користувача»

3.3 Особливості реалізації програмного забезпечення

Вхідні та вихідні дані програмного забезпечення для підвищення рівня інформаційної безпеки персональних сторінок користувачів у соціальних мережах буде визначено у цьому параграфі.

Вхідна інформація буде формуватися з даних які клієнт вводить сам під час авторизації у вікні авторизації користувача. Всі ці дані будуть внесенні до пам’яті програмного забезпечення відповідно зберігатиметься у моделі, яка відповідає за збереження цієї інформації. Інформація про користувача вводиться один раз після запуску програми, якщо ж користувач вийшов із цієї програми йому буде необхідно ввести заново ці дані, це зроблено із метою захисту даних користувача.

Наприклад, введення вхідної інформації клієнта здійснюється за допомогою екранної форми «Авторизації користувача» (рисунок 3.7).

У якості вихідної інформації програмного забезпечення будь виступати звіти сервісу у вигляді таблиці із параметрами безпеки, які у користувача налаштовано, а також параметри безпеки, які має можливість перевірити програмне забезпечення.

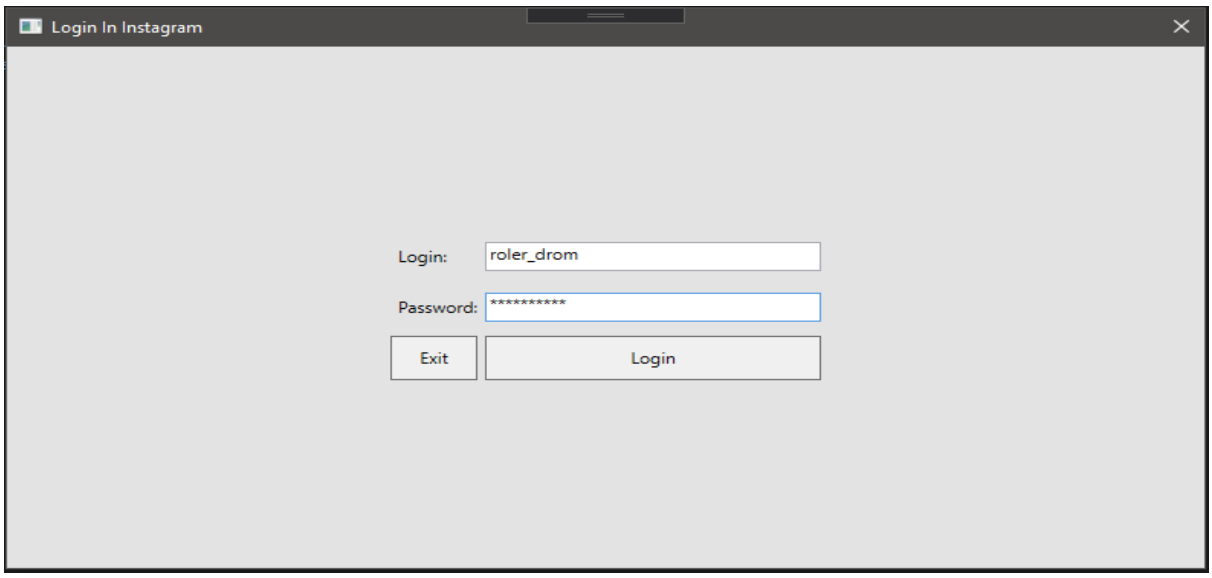

Рисунок 3.7 - Екранна форма «Ідентифікація користувача»

На рисунку 3.8 відображається перелік усіх параметрів безпеки, що перевіряються і до яких має доступ програмне забезпечення.

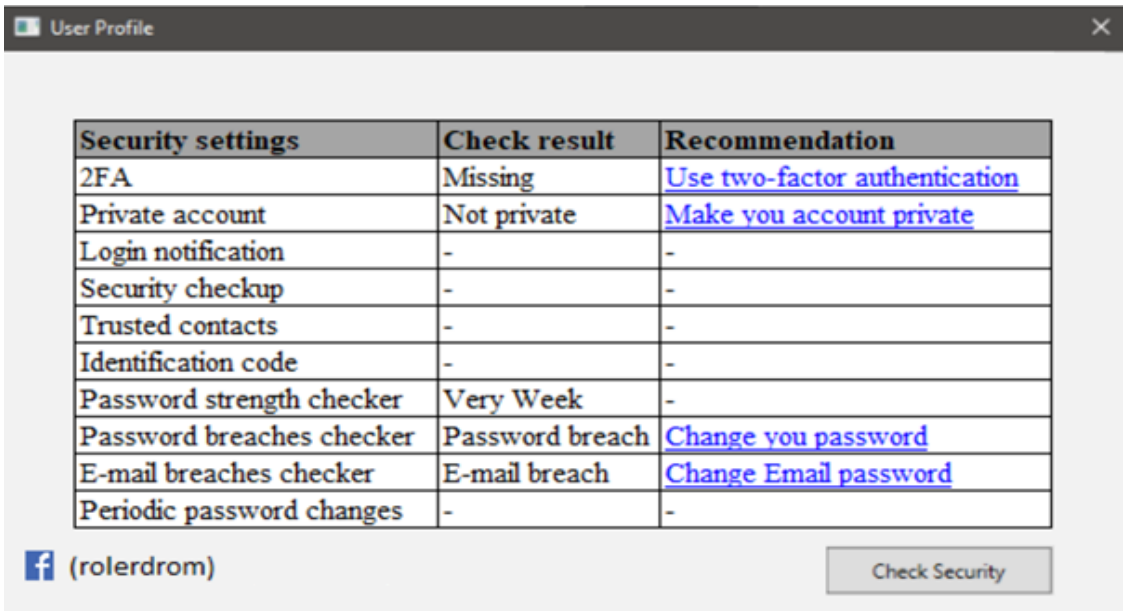

Рисунок 3.8 - Екранна форма «Профіль користувача»

Мова програмування C# надає можливість використовувати функції з динамічною кількістю та типами параметрів. Точна кількість та типи параметрів стають відомими лише під час виклику функції. У програмному продукті використовуються різні функції з різними типами параметрів, що

свідчить про багатофункціональність програмного сервісу. Кожна функція відповідає за конкретний функціонал у програмному сервісі і є невід'ємною складовою програмного забезпечення. Різні функції взаємодіють між собою та утворюють ієрархію функціональності.

У структурному програмуванні застосовується ієрархічний підхід, де окремі структурні одиниці перетворюються на функції, які можуть приймати параметри та повертати результат. Опис функції виконується один раз у певному місці програми, і вона може бути використана багато разів, при цьому лише параметри можуть змінюватися, що дозволяє уникнути повторення однакових операцій.

Наведений у таблиці 3.1 програмний код функцій супроводжується описом логіки їх роботи, що дозволяє краще зрозуміти функціонал програмного сервісу.

Таблиця 3.1 - Функції та їх опис у програмному коді

Назва методу	Програмний код методу
1	2
Функція налаштування сервісу	public HaveIBeenPwnedClientSettings setting = new HaveIBeenPwnedClientSettings() { ApiKey = "AppKey", ApplicationName = "Application Name", TimeOut = TimeSpan.FromSeconds(30)};

Функція перевірки електронної скриньки користувача	private async Task ApiPwnedBeenEmail(string email)
	{
	using (var pwndClient = new AtleX.HaveIBeenPwned.HaveIBeenPwnedClient(setting))
	{var breaches = await pwndClient.GetBreachesAsync(email,BreachMode.ExcludeUnverified);
	if (breaches.Count() != 0)
	{
	listView.Items.Add("Count: " + breaches.Count() + "\n");
	}
	else
	{listView.Items.Add("Good news — no pwnage found!"); }}}

Продовження таблиці 3.1

1	2

Функція перевірки паролю користувача	private async Task ApiPwnedBeenPasssword(string userPassword)
	{using (var pwndClient = new
	AtleX.HaveIBeenPwned.HaveIBeenPwnedClient(setting)
	{var breaches = await
	pwndClient.IsPwnedPasswordAsync(userPassword);
	listView.Items.Add("Password breaches checker: \n");
	if (breaches == true)
	{listView.Items.Add("The password was found");}
	else
	{listView.Items.Add("The password was not found");}}}}

Функція перевірки приватності профілю у соціальній мережі	private void IsPrivateAccount(InstaSession session) {listView.Items.Add("Private Account: \n"); if (session.SessionUser.LoggedInUder.IsPrivate == true) {listView.Items.Add("This is private account!"); } Else {listView.Items.Add("This isn`t private account!"); }}

Функція перевірки надійності паролю користувача	public static PasswordScore CheckStrength(string password)
	{int score = 0;
	if (password.Length < 1)
	return PasswordScore.Blank;
	if (password.Length < 4)
	return PasswordScore.VeryWeak;
	if (password.Length >= 8)
	score++;
	if (password.Length >= 12)
	score++;
	if (Regex.Match(password, @"/\d+/",
	RegexOptions.ECMAScript).Success)
	score++;
	if (Regex.Match(password, @"/[a-z]/",
	RegexOptions.ECMAScript).Success &&
	Regex.Match(password, @"/[A-Z]/",
	RegexOptions.ECMAScript).Success)
	score++;
	if (Regex.Match(password, @"/.[!,@,#,$,%,^,&,*,?,_,~,-
	,£,(,)]/", RegexOptions.ECMAScript).Success)
	score++;

Продовження таблиці 3.1

1	2

Функція для відправки повідомлення користувачеві	public async Task<bool> SendVefityCodeAccount(string securityCode)
	{
	var result = await _instaApi.SendVerifyCode(securityCode);
	_sessionData.LoggedInUder.Pk = result.Value.LoggedInUser.Pk;
	VerifityIsSucceeded = result.Succeeded;
	if (VerifityIsSucceeded)
	{
	await GetCurrentUserAsync();
	await Get2FA();
	}
	return VerifityIsSucceeded;
	}

Функція відображення фото	public ImageSource PhotoProfile()
	{
	BitmapImage logo = new BitmapImage();
	logo.BeginInit();
	logo.UriSource = new Uri(_instagramData.CurrentUser.Value.ProfilePicture);
	logo.EndInit();
	var result = logo;
	return result;
	}

Функція відображення даних користувача	public string FullNameUser()
	{
	return _instagramData.CurrentUser.Value.FullName + " (" + _instagramData.CurrentUser.Value.UserName + ") ";
	}

Функція для ідентифікації поточного користувача	public async Task<IResult<InstaCurrentUser>> GetCurrentUserAsync()
	{
	var _currentUser = await _instaApi.GetCurrentUserAsync();
	_instagramData.CurrentUser = _currentUser;
	return _currentUser;
	}

Продовження таблиці 3.1

1	2

Функція створення серверу для відпраки повідомлення	public async Task<bool> SendMessageToEmail(string toAddressUser, string nameUser)
	{
	var fromAddress = new MailAddress("logincheckfit69@gmail.com", "Login Notification FIT");
	var toAddress = new MailAddress(toAddressUser, nameUser);
	const string fromPassword = "mE-x4PLgp>Y=-,Y";
	const string subject = "Login Notification User in Security FIT";
	const string body = "You are authorized in our application <<Security FIT>>, please do not block such emails";
	bool result;
	var smtp = new SmtpClient
	{
	Host = "smtp.gmail.com",
	Port = 587,
	EnableSsl = true,
	Timeout = 20000,
	DeliveryMethod = SmtpDeliveryMethod.Network,
	UseDefaultCredentials = false,
	Credentials = new NetworkCredential(fromAddress.Address, fromPassword)
	};
	using (var message = new MailMessage(fromAddress, toAddress)
	{
	Subject = subject,
	Body = body
	})
	{ Try {
	await smtp.SendMailAsync(message);
	result = true;
	}
	catch (Exception ex)
	{
	Console.WriteLine(ex.Message);
	result = false;
	}
	}

3.4 Тестування програмного забезпечення

Після завершення розробки програмного забезпечення був проведений функціональний аналіз розробленого продукту та визначена його ефективність. Ефективність роботи програмного забезпечення визначається за певними критеріями та якостями, які характеризують його здатність відповідати вимогам та потребам користувачів.

Поняття якості програмного забезпечення охоплює набір властивостей продукту, що визначають його здатність задовольнити встановлені або передбачувані потреби замовника. Якість може мати різні інтерпретації, залежно від конкретної програмної системи та вимог до неї.

Ключовим критерієм оцінки програмного забезпечення є якість написання коду. Оцінка якості коду може базуватися на різних критеріях, деякі з яких можуть мати значення для людини, а не для комп'ютера. Наприклад, форматування тексту програми може бути неважливим для комп'ютера, але відігравати важливу роль для зручності супроводу. Існуючі стандарти кодування визначають угоди, специфічні для мови програмування, і встановлюють правила з метою полегшення супроводу програмного забезпечення в майбутньому.

Існують різні критерії для визначення якості коду, такі як:

- читабельність коду.
- легкість підтримки, тестування, відлагодження, виправлення помилок, рефакторингу та портування.
- низька складність коду.
- коректність обробки винятків.

Враховуючи ці фактори, можна здійснити об'єктивну оцінку якості програмного забезпечення та його відповідність вимогам індустрійних стандартів.

Відповідно до цих критеріїв було розроблено зведену таблицю та оцінено код написання програмного забезпечення відповідно до цих

факторів. Оцінка факторів написання коду буде відбуватись по десятибальній системі від 1 до 10 (таблиця 3.2).

Таблиця 3.2 - Зведена таблиця факторингу коду програмного сервісу

Фактори оцінки	Оцінка (від 1 до 10)
Прочитність коду	9
Легкість підтримки	8
Низька складність коду	9
Коректність обробки винятків	7

Якість програмного забезпечення також оцінюється за допомогою моделей якості, які можуть мати різну кількість рівнів і частково або повністю співпадати за характеристиками якості. Однією з таких моделей є ISO 9126, яка визначає набір характеристик та атрибутів якості.

Основні характеристики та атрибути якості згідно з ISO 9126 включають:

- функціональність (Functionality): Функціональна придатність (Suitability): Здатність програмного забезпечення виконувати вказані завдання;
- точність (Accuracy): Вірність та точність виконання функцій;
- здатність до взаємодії (Interoperability): Здатність взаємодіяти з іншими системами;
- відповідність стандартам і правилам (Compliance): Співвідношення з встановленими стандартами та правилами;
- захищеність (Security): Рівень захисту від несанкціонованого доступу;

- надійність (Reliability): Зрілість, завершеність (Maturity): Ступінь завершеності та стабільності програми;
- стійкість до відмов (Fault Tolerance): Здатність продовжувати роботу після виникнення помилок;
- відповідність стандартам надійності (Reliability Compliance): Відповідність визначеним стандартам надійності;
- здатність до відновлення (Recoverability): Можливість відновлення роботи після виникнення помилок;
- зручність використання (Usability): Зрозумілість (Understandability): Рівень зрозумілості інтерфейсу та взаємодії;
- зручність навчання (Learnability): Легкість освоєння користувачами програми;
- зручність роботи (Operability): Зручність використання та роботи з програмою;
- привабливість (Attractiveness): Ступінь привабливості та естетичності інтерфейсу;
- відповідність стандартам зручності використання (Usability Compliance): Відповідність визначеним стандартам зручності використання;
- продуктивність (Efficiency): Часова ефективність (Time Behaviour): Швидкість виконання функцій та завдань;
- ефективність використання ресурсів (Resource Utilisation): Оптимальне використання ресурсів комп'ютера;
- відповідність стандартам продуктивності (Efficiency Compliance): Відповідність визначеним стандартам продуктивності;
- зручність супроводу (Maintainability): Аналізованість (Analyzability): Легкість аналізу та змін коду;
- зручність внесення змін (Changeability): Легкість внесення змін у програмний код;
- стабільність (Stability): Ступінь стабільності кодової бази;

- зручність перевірки (Testability): Легкість використання для тестування;
- відповідність стандартам зручності супроводу (Maintainability Compliance): Відповідність визначеним стандартам зручності супроводу;
- переносимість (Portability);
- адаптованість (Adaptability): Здатність програми адаптуватися до різних середовищ використання;
- зручність установки (Installability): Легкість встановлення програми на різних пристроях;
- здатність до співіснування (Coexistence): Можливість співіснування з іншими програмами;
- зручність заміни (Replaceability): Легкість заміни аналогічних продуктів.

Відповідно до атрибутів, визначених стандартом, розроблено зведену таблицю (таблиця 3.3), в якій подані оцінки програмного забезпечення за аспектом модульності.

Таблиця 3.3 - Модульна оцінка програмного забезпечення

Моделі оцінки	Оцінка
Функціональність (functionality)	7
Надійність (reliability)	7
Зручність використання (usability)	9
Продуктивність (efficiency)	8
Зручність супроводу (maintainability)	8
Переносимість (portability)	7

У таблиці 3.4 подано результати порівняльного аналізу параметрів безпеки реалізовані у соціальних мережах та програмному сервісі.

Таблиця 3.4 - Порівняльний аналіз параметрів безпеки реалізованих у програмному забезпеченні

Параметри безпеки сторінок	Реалізація у соціальній мережі		Реалізація у програмному сервісі	
	Facebook	Instagram	Facebook	Instagram
1	2	3	4	5
Двоетапна перевірка	+	+	-	+
Приватний обліковий запис	+	+	-	+
Сповіщення про безпеку та конфіденційність	+	-	-	-
Перевірка авторизованих входів	+	-	-	-
Формування довірених контактів	+	-	-	-

Оцінки проводилися за десятибальною шкалою. Зазначені оцінки відображають рівень відповідності кожного атрибуту.

ВИСНОВКИ

У кваліфікаційні робота було отримано наступні результати:

1. Виявлено, що більшість кіберзагроз спрямовані на особисті дані та конфіденційну інформацію користувачів соціальних мереж.

2. Підкреслено відсутність важливих засобів контролю безпеки на серверах провідних соціальних мереж, що вказує на необхідність подальших заходів з підвищення безпеки.

3. Обгрунтовано необхідність створення комплексного механізму захисту облікових записів, який дозволяв би користувачам автоматично перевіряти та налаштовувати засоби контролю безпеки сторінок.

4. Розроблено математичну модель для формування узагальнюючого показника рівня безпеки сторінок користувачів, враховуючи вплив кожного налаштованого контролю безпеки.

5. Доведено, що найбільш значущим показником безпеки є налаштування контролю двоетапної перевірки, а найменш значущим – генерація кодів ідентифікації.

6. Реалізовано програмний сервіс, який автоматично перевіряє параметри безпеки персональних сторінок у соціальних мережах, забезпечуючи підвищення захищеності з урахуванням зовнішніх контролів безпеки.

7. Обґрунтовано вибір програмних засобів для розробки, спрямованих на підвищення інформаційної безпеки персональних сторінок у соціальних мережах.

8. Реалізовано програмний сервіс, що автоматично перевіряє параметри безпеки користувацьких

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Most popular social networks world wide as of October 2022, ranked by number of active users [Електронний ресурс]. – Режим доступу до ресурсу: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number -of-users/.
2. До 50 млн акаунтів в Facebook зламані невідомими хакерами [Електронний ресурс] – Режим доступу до ресурсу : https://znaj.ua/techno/176920-pid-zagrozoyu-50-mln-koristuvachiv-facebook -poperediv-pro-strashnu-nebezpeku.
3. Up to six million Instagram account saffected by data breach [Електронний ресурс] // TheStack. – 2021. – Режим доступу до ресурсу : https://thestack.com/security/2017/09/04/up-to-six-million-instagram-accounts affected -by-data-breach/.
4. У соціальній мережі Facebook через Messenger поширюється вірус-троян [Електронний ресурс] // Детектор Медіа. – 2017. – Режим доступу до ресурсу : https://detector.media/infospace/article/132950/2017-12-15-u-sotsialnii-merezhi-facebook -cherez-messenger-poshiryuetsya-virus-troyan/
5. http://facebook.com/
6. https://www.youtube.com
7. https://www.instagram.com
8. Computational Social Networks: Security and Privacy. [Електронний ресурс] / [M. Salama, M. Panda, Y. Elbarawy та ін.] // Computationa lSocial Networks. – 2012. – Режим доступу до ресурсу : http://njtech.findplus.cn
9. https://haveibeenpwned.com
10. https://strongpasswordgenerator.com/
11. Скрута Г.В. Забезпечення інформаційної безпеки у соціальних мережах / Скрута Г.В., Шкарупа І.В., Нікуліщев Г.І. // Актуальні задачі та досягнення у галузі кібербезпеки: Всеукраїнська науково-практична

конференція студентів і молодих вчених, 23-25 листопада 2016 р. : матеріали конф. – Кропивницький, 2016. – С.79.

12. Карманний Є. В. Підходи до захисту інформації при користуванні соціальними мережами [Електронний ресурс] / Є. В. Карманний, С. О. Ковжога. – 2015. – Режим доступу до ресурсу : http://dspace.nlu.edu.ua/bitstream/ 123456789/8420/1/Karmannuy_Kovgoa.pdf.
13. Методологія експертного оцінювання : конспект лекцій / Уклад. Новосад В.П., Селіверстов Р.Г. – К. : Вид-во НАДУ, 2017. – 48 с.
14. Підходи до нормування економічних показників / Р.В. Волощук // Індуктивне моделювання складних систем : Зб. наук. пр. — К.: МННЦ ІТС НАН та МОН України, 2009. – Вип. 1. – С. 17-25
15. Оцінка рівня економічної безпеки підприємства / А.М. Ткаченко, О.Л. Резніков // Вісник економічної науки України. — 2010. — № 1 (17). — С. 101-106.
16. Матвієнко М. П. Математична логіка та теорія алгоритмів : навч. посіб. студ. для вищих навч. закладів / М. П. Матвієнко, С. П. Шаповалов ; М-во освіти і науки України, Сумський держ. ун-т. – Київ : Ліра-К, 2015. – 211 с.
17. Вигерс К.И. Розробка вимог до ПЗ. – М.: редакція Microsoft, 2004. – 575 с.
18. https://manychat.com/
19. Carley K. Destabilizing networks / K. Carley, J. Lee,D. Krackhardt // Connections. – Vol. 24, № 3. – 2002. –P. 79-92.
20. Дзюндзюк В.Б. Віртуальні співтовариства: потенційна загроза для національної безпеки / В.Б. Дзюндзюк //
21. Shantanu Ghosh. Top seven social media threats.[Electronic resourse] / Shantanu Ghosh // Electronic data. –[Computerweekly, 2017]. Mode of access: World Wide Web:http://www.computerweekly.com/tip/Top-seven-social-mediathreats. Accessed 07 March 2017.
22. C. Fuchs. Social Media: A Critical Introduction - 2nd edition. - 2017. – 400p.

23. S. Alarm, K. El-Khatib “Phishing Susceptibility Detection through Social Media Analytics” In Proceedings of the 9th International Conference on Security of Information and Networks, Newark, NJ, USA, 20–22 July 2016, pp. 61–64.
24. M. Egele, G. Stringhini, C. Kruegel, G. Vigna “Towards detecting compromised accounts on social networks” IEEE Trans. Dependable Secure Computer. 2017, pp. 447–460.
25. K. Smith (2019) 122 Amazing Social Media Statistics and Facts [Online]. Available: https://www.brandwatch.com/ blog/amazingsocial-media-statistics-and-facts/
26. M. Jarir Kanji (2019). Massive breach leaves 267 million Facebook users' data exposed [Online]. Available: www.windowscentral.com/ massive-data-breach-leaves-267-million-facebook-users-data-exposed
27. M. Isaac, S. Frenkel (2018). Facebook Security Breach Exposes Accounts of 50 Million Users [Online]. Available: www.nytimes.com/2018/09/28/technology/ facebook-hack-databreach.html
28. R. McLeish (2017). YouTube accounts hacked by online security group [Online]. Available: http://www.smh.com.au/technology/ technology-news/youtube-accounts-hacked-by-online-security-group20170414-gvl31k.html
29. S. Clark (2017). Up to six million Instagram accounts affected by data breach [Online]. Available: https://thestack.com/security/2017/ 09/04/up-to-six-million-instagram-accountsaffected-by-data-breach/
30. K. Thomas, A. Moscicki (2019) New research: How effective is basic account hygiene at preventing hijacking [Online]. Available: https://security.googleblog.com/2019/05/new-research-how-effectiveis-basic.html
31. Cyber Security Guidelines for Securing Social Media Accounts (2018) [Online]. Available: https://www.qcert.org/sites/default/files/public/documents/guidelines_ for_securing_social_media_accounts.pdf

32. Identity Awareness, Protection, and Management Guide (2018). [Online]. Available: https://www.dla.mil/Portals/104/Users/230/98/
33. T. Hunt (2019). FAQs: Need to know something about Have I Been Pwned. [Online]. Available: https://haveibeenpwned.com/
34. Facebook for developers. Facebook API [Online]. Available: https://developers.facebook.com
35. Instagram Developer Documentation [Online]. Available: https://www.instagram.com/developer/

ДОДАТОК А

ЛІСТИНГ ПРОГРАМНОГО КОДУ

```
publicclassWebBrowserHelperControler
   {

publicstaticreadonly DependencyProperty UrlProperty =
         DependencyProperty.RegisterAttached("Url", typeof(string), typeof(WebBrowserHelperControler),
new PropertyMetadata(OnUrlChanged));

publicstaticstring GetUrl(DependencyObject dependencyObject)
      {
return (string)dependencyObject.GetValue(UrlProperty);
      }

publicstaticvoid SetUrl(DependencyObject dependencyObject, string body)
      {
         dependencyObject.SetValue(UrlProperty, body);
      }
privatestaticvoid OnUrlChanged(DependencyObject d, DependencyPropertyChangedEventArgs e)
      {
var browser = d as WebBrowser;
if (browser == null) return;

         Uri uri = null;

var s = e.NewValue asstring;

if (s != null)
         {
var uriString = s;

            uri = string.IsNullOrWhiteSpace(uriString) ? null : new Uri(uriString);
         }
elseif (e.NewValue is Uri)
         {
            uri = (Uri)e.NewValue;
         }

         browser.Source = uri;
      }
   }
publicclassIsPrivateAccountConverter : IValueConverter
   {
publicobject Convert(object value, Type targetType, object parameter, CultureInfo culture)
      {
if ((bool)value)
         {
return"Private";
         }
else
         {
return"Not Private";
         }
      }

publicobject ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
      {
thrownew NotImplementedException();
```

```
        }
    }
publicobject Convert(object value, Type targetType, object parameter, CultureInfo culture)
        {
if ((bool)value)
            {
return"Message sent";
            }
else
            {
return"Message not sent";
            }
        }

publicobject ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
        {
thrownew NotImplementedException();
        }
private IInstaApi _instaApi;
public IInstaApi InstaApi
        {
get => _instaApi;
set { _instaApi = value; }
        }
private IResult<InstaCurrentUser> currentUser;
public IResult<InstaCurrentUser> CurrentUser
        {
get => currentUser;
set { currentUser = value; }
        }
private IResult<TwoFactorLoginInfo> _twoFactorLogin;
public IResult<TwoFactorLoginInfo> TwoFactorLogin
        {
get => _twoFactorLogin;
set { _twoFactorLogin = value; }
        }
private HaveIBeenPwnedClientSettings _haveBeenSettings { get; set; }
private HaveIBeenPwnedClient _haveBeenClient { get; set; }
privateint _count { get; set; }
publicHaveBeenClientService()
        {

            _haveBeenSettings = new HaveIBeenPwnedClientSettings
            {
                ApiKey = "api-key",
                ApplicationName = "Security FIT"
            };
            GetBeenPwnedClient(_haveBeenSettings);
        }

public HaveIBeenPwnedClient GetBeenPwnedClient(HaveIBeenPwnedClientSettings settings)
        {
            _haveBeenClient = new HaveIBeenPwnedClient(settings);
return _haveBeenClient;
        }

publicasync Task<int> GetCountBreachUserEmail(string email)
        {
var result = await _haveBeenClient.GetBreachesAsync(email, BreachMode.All);
if (result.Count() != 0)
                _count = result.Count();
return _count;
```

```
    }

publicasync Task<bool> GetCheckPasswordUser(string password)
    {
returnawait _haveBeenClient. IsPwnedPasswordAsync(password);
    }
privatebool isSucceeded;
privatebool VerifityIsSucceeded;
private UserSessionData _sessionData;
private InstagramData _instagramData;
private IInstaApi _instaApi;
publicInstagramService(UserSessionData sessionData, InstagramData instagramData)
    {
      _sessionData = sessionData;
      _instagramData = instagramData;
    }

publicenumPasswordScore
    {
      Blank = 0,
      VeryWeak = 1,
      Weak = 2,
      Medium = 3,
      Strong = 4,
      VeryStrong = 5
    }

public UserSessionData GetSessionData(string username, string password)
    {
      _sessionData = new UserSessionData();

      _sessionData.UserName = username;
      _sessionData.Password = password;


return _sessionData;
    }
publicasync Task<bool> GetLoginAsync(string username, string password)
    {

var delay = RequestDelay.FromSeconds(2, 2);
try
      {
        _instaApi = InstaApiBuilder.CreateBuilder()
          .SetUser(_sessionData)
          .UseLogger(new DebugLogger(LogLevel.All))
          .SetRequestDelay(delay)
          .Build();
      }
catch (Exception ex)
      {
        Console.WriteLine(new DebugLogger(LogLevel.All));
        MessageBox.Show(ex.Message);
      }

if (!_instaApi.IsUserAuthenticated)
      {
var request = await _instaApi.LoginAsync();

        isSucceeded = request.Succeeded;

if (isSucceeded)
```

```
            {
await GetCurrentUserAsync();
await Get2FA();
            }
else
            {
await CheckVerify();
            }
          }

return isSucceeded;
      }

publicasync Task CheckVerify()
      {
var verifyStep = await _instaApi.GetVerifyStep();
await _instaApi.ChooseVerifyMethod(Convert.ToInt32(verifyStep.Value.StepData.Choice));
      }

publicasync Task<bool> SendVefityCodeAccount(string securityCode)
      {
var result = await _instaApi.SendVerifyCode(securityCode);
          _sessionData.LoggedInUder.Pk = result.Value.LoggedInUser.Pk;
          VerifityIsSucceeded = result.Succeeded;

if (VerifityIsSucceeded)
          {
await GetCurrentUserAsync();
await Get2FA();
          }

return VerifityIsSucceeded;
      }
public PasswordScore CheckStrength(string password)
      {
int score = 0;

if (password.Length < 1)
return PasswordScore.Blank;
if (password.Length < 4)
return PasswordScore.VeryWeak;

if (password.Length >= 8)
            score++;
if (password.Length >= 12)
            score++;
if (Regex.Match(password, @"/\d+/", RegexOptions.ECMAScript).Success)
            score++;
if (Regex.Match(password, @"/[a-z]/", RegexOptions.ECMAScript).Success &&
           Regex.Match(password, @"/[A-Z]/", RegexOptions.ECMAScript).Success)
            score++;
if (Regex.Match(password, @"/.[!,@,#,$,%,^,&,*,?,_,~,-,£,(,)]/", RegexOptions.ECMAScript).Success)
            score++;

return (PasswordScore)score;
      }

publicasync Task<IResult<InstaCurrentUser>> GetCurrentUserAsync()
      {

var _currentUser = await _instaApi.GetCurrentUserAsync();
```

```
            _instagramData.CurrentUser = _currentUser;

return _currentUser;
        }
publicasync Task<bool> Get2FA()
        {
var _twoFactor = await _instaApi.GetTwoFactorInfoAsync();

            _instagramData.TwoFactorLogin = _twoFactor;

return _twoFactor.Succeeded;
        }
public ImageSource PhotoProfile()
        {
            BitmapImage logo = new BitmapImage();
            logo.BeginInit();
            logo.UriSource = new Uri(_instagramData.CurrentUser.Value.ProfilePicture);
            logo.EndInit();

var result = logo;
return result;
        }
publicstring FullNameUser()
        {
return _instagramData.CurrentUser.Value.FullName + " ( " + _instagramData.CurrentUser.Value.UserName + " ) ";
        }
    }
var  fromAddress = new MailAddress("logincheckfit69@gmail.com", "Login Notification FIT");
var toAddress = new MailAddress(toAddressUser, nameUser);
conststring fromPassword = "password";
conststring subject = "Login Notification User in Security FIT";
conststring body = "You are authorized in our application <<Security FIT>>, please do not block such emails";
bool result;
var smtp = new SmtpClient
            {
                Host = "smtp.gmail.com",
                Port = 587,
                EnableSsl = true,
                Timeout = 20000,
                DeliveryMethod = SmtpDeliveryMethod.Network,
                UseDefaultCredentials = false,
                Credentials = new NetworkCredential(fromAddress.Address, fromPassword)
            };
using (var message = new MailMessage(fromAddress, toAddress)
            {
                Subject = subject,
                Body = body
            })
            {
try
                {
await smtp.SendMailAsync(message);
                    result = true;
                }
catch (Exception ex )
                {
                    Console.WriteLine(ex.Message);
                    result = false;
                }
            }
return result;
        }
```

ДОДАТОК Б

КОПІЇ ПУБЛІКАЦІЙ