СПЕЦИФІКАЦІЯ ПРОГРАМНИХ ВИМОГ (SRS)

Веб-застосунок «SecurePass» для безпечного зберігання та управління паролями

Версія: 1.0.1

Дата: 01 червня 2025 р.

Автор: Лева Анастасія Василівна

1 ВСТУП

1.1 Призначення

Цей документ є Специфікацією Програмних Вимог (SRS) для веб-застосунку «SecurePass». Він детально описує функціональні та нефункціональні вимоги до системи, її дизайн, поведінку та інші характеристики, необхідні для успішної розробки та впровадження.

Метою SecurePass є створення надійного, безпечного та зручного інструменту для управління паролями та конфіденційними даними користувачів.

1.2 Область застосування продукту

Веб-застосунок «SecurePass» є менеджером паролів, призначеним для безпечного зберігання, генерації та управління обліковими даними користувачів. Він вирішує актуальні проблеми, пов'язані з використанням слабких або повторюваних паролів, ненадійним зберіганням та ризиком компрометації облікових даних у сучасному цифровому середовищі. Продукт спрямований на мінімізацію ризиків, пов'язаних з найпоширенішими кібератаками, такими як перебір паролів, фішинг та витоки даних.

1.3 Межі

Включений функціонал:

* Управління парольними записами: створення, зберігання, перегляд, редагування, видалення, пошук та сортування.
* Автентифікація: реєстрація нового користувача, стандартний вхід, двофакторна автентифікація (2FA) через email, авторизація через Google-акаунт.
* Безпека даних: шифрування AES-256 для записів, хешування scrypt для майстер-пароля, захист від атак перебору (Brute Force).
* Допоміжні інструменти: генератор паролів з різними рівнями складності.
* Експорт всіх збережених даних у форматі CSV.
* Налаштування акаунту: зміна майстер-пароля, увімкнення/вимкнення 2FA, автоматичний вихід, нічний режим, видалення акаунту.

Виключений функціонал (плани на майбутнє):

* Реалізація функцій імпорту даних облікових записів.
* Розробка мобільних версій застосунку (для iOS та Android).
* Розробка розширень для популярних веб-браузерів.
* Інтеграція з хмарними сервісами для синхронізації та резервного копіювання даних.

1.4 Посилання

1. Cherednichenko, O., Kyrychenko, I., Nechvolod, K., Smelyakov, K., Dolhanenko, O. Data Security Analysis in EMM Systems. CEUR Workshop Proceedings. 2024. Vol. 3668. P. 133–144. Doi: <https://ceur-ws.org/Vol-3668/>.
2. Korchenko, Anna; Ivanchenko, Yevheniya; Koshkina, Natalia; Kuznetsov, Oleksandr; Potiy, Oleksandr; Onoprienko, Viktor; Bobukh, Vsevolod. STANDARDIZATION OF SYSTEMS, COMPLEXES AND MEANS OF CRYPTOGRAPHIC PROTECTION OF INFORMATION FOR APPLICATION IN POSTQUANTUM ENVIRONMENT. 2023.
3. Дерев’янко Я.А., Горбенко І.Д. Hash-based cryptography, its security and feasibility in modern cryptosystems. Radiotekhnika. 2023. DOI: 10.30837/rt.2023.2.213.01.
4. Кириченко І. В., Терещенко Г. Ю., Груздо І. В. Застосування симетричних алгоритмів в блокчейні // Бионика интеллекта. – 2020. – № 1(94). – С. 71–75. DOI: https://doi.org/10.30837/bi.2020.1(94).11.
5. Sharafaldin, Iman; Lashkari, Arash Habibi; Ghorbani, Ali A. Toward generating a new intrusion detection dataset and intrusion traffic characterization 1 // 4th International Conference on Information Systems Security and Privacy, ICISSP 2018, Funchal, Madeira. 2018. С. 108–116. DOI: 10.5220/0006639801080116.
6. Єсіна М. В., Кузнецова К. О. Analysis of methods and algorithms for generating key data for FALCON-like electronic signature algorithms. Radiotekhnika. 2022. pp. 83–87.
7. Горбенко І.Д., Єсіна М.Б. Аналіз алгоритму направленого шифрування NTRU PRIME IIT UKRAINE з урахуванням відомих атак. J. RadioTecnika. 2018. v. 191. p. 11–23.
8. Skanda C., Srivatsa B., Premananda B.S. Secure Hashing using BCrypt for Cryptographic Applications // 2022 IEEE North Karnataka Subsection Flagship International Conference, NKCon 2022, Vijaypur. 2022. DOI: 10.1109/NKCon56289.2022.10126956.
9. Jubur, Mohammed; Shrestha, Prakash; Saxena, Nitesh. An In-Depth Analysis of Password Managers and Two-Factor Authentication Tools. ACM Computing Surveys. 2025. Vol. 57, Iss. 524. Article No. 128. DOI: 10.1145/3711117.
10. Старіченко В. С. Порівняльний аналіз архітектури обробки сповіщень на стороні back-end : синхронного REST API та асинхронного підходу з Apache Kafka / В. С. Старіченко, О. Ф. Лановий // Радіоелектроніка та молодь у XXI столітті : матеріали 29-го Міжнар. молодіж. форуму, 16–19 квітня 2025 р. – Харків : ХНУРЕ, 2025. – Т. 6 – С. 324–326.
11. IEEE 830-1993 - IEEE Recommended Practice for Software Requirements Specifications.
12. Закон України «Про основні засади забезпечення кібербезпеки України».
13. ISO/IEC 18031:2005 – Information technology — Security techniques — Random bit generation.
14. Конвенція про кіберзлочинність (Будапештська конвенція).
15. ISO/IEC 27001 - Information technology — Security techniques — Information security management systems — Requirements.

1.5 Означення та абревіатури

2FA – Two-Factor Authentication

AES – Advanced Encryption Standard

API – Application Programming Interface

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

ISO – International Organization for Standardization

JSON – JavaScript Object Notation

MFA – Multi-Factor Authentication

OTP – One-Time Password

REST – Representational State Transfer

SMTP – Simple Mail Transfer Protocol

TCP/IP – Transmission Control Protocol/Internet Protocol

UI – User Interface

UML – Unified Modeling Language

UX – User Experience

1. ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

«SecurePass» є самостійним веб-застосунком, розробленим на основі клієнт-серверної архітектури. Він функціонує як окремий додаток, взаємодіючи з декількома зовнішніми сервісами та компонентами:

* Сервер Бази Даних (MySQL): Для постійного зберігання даних користувачів та парольних записів
* Email Server: Для надсилання електронних листів (верифікаційні коди, коди 2FA, сповіщення про зміну пароля тощо)
* Google Auth Server: Для надання можливості авторизації через облікові записи Google
* Веб-браузер користувача: Виступає клієнтською частиною, через яку відбувається вся взаємодія з застосунком

Застосунок позиціонується як комплексне та безпечне рішення для підвищення особистої та корпоративної цифрової безпеки.

2.2 Функції продукту

«SecurePass» надає наступні ключові функції для задоволення потреб користувачів у сфері безпечного зберігання та керування обліковими даними:

* Управління парольними записами: Користувачі можуть створювати, зберігати, переглядати, редагувати та видаляти записи облікових даних для різних веб-ресурсів. Запис включає поля: назва сайту, логін, email, пароль, URL-адреса, номер телефону, секретне слово, нікнейм, custom ID, попередній пароль, резервний email, підказка для пароля.
* Пошук та організація даних: Реалізовано функції пошуку збережених записів за назвою сайту або логіном, а також сортування списку парольних записів за визначеними критеріями (наприклад, за датою додавання чи за алфавітом).
* Генерація паролів: Вбудований інструмент дозволяє створювати криптографічно стійкі та унікальні паролі з різними параметрами складності: "Легкий", "Середній", "Складний", "Парольна фраза" та "PIN-код".
* Автентифікація та авторизація: Застосунок пропонує повноцінну систему реєстрації та входу, підтримує двофакторну автентифікацію (2FA) через email та можливість авторизації через Google-акаунт.
* Захист даних: Для чутливих даних (паролі, секретні слова, підказки, попередні паролі) використовується симетричне AES-шифрування. Майстер-пароль користувача хешується за допомогою стійкого алгоритму scrypt.
* Захист від атак перебору: Впроваджено механізм тимчасового блокування форми входу після кількох невдалих спроб (наприклад, після трьох).
* Експорт даних: Користувачі мають можливість експортувати всі свої збережені дані у форматі CSV.
* Налаштування акаунту: Користувач має повний контроль над своїм профілем, включаючи зміну майстер-пароля, увімкнення/вимкнення 2FA, керування опцією автоматичного виходу, перемикання нічного режиму та видалення акаунту.

2.3 Характеристики користувачів

Цільовою аудиторією веб-застосунку «SecurePass» є широке коло користувачів, які прагнуть підвищити свою цифрову безпеку. До них відносяться:

* Активні інтернет-користувачі, що мають велику кількість облікових записів.
* Користувачі, що бажають підвищити рівень своєї цифрової безпеки та відмовитися від практики повторного використання паролів.
* Особи, які часто забувають або використовують однакові чи прості паролі для різних сервісів.
* Користувачі, які потребують простого та інтуїтивного інструменту для генерації складних та унікальних паролів.

Застосунок розроблений таким чином, щоб бути зрозумілим та зручним у використанні навіть для недосвідчених користувачів без глибоких технічних знань.

2.4 Загальні обмеження

Тип застосунку: «SecurePass» є виключно веб-застосунком і не передбачає фізичних компонентів або автономних десктопних/мобільних версій у поточній реалізації.

Для доступу та повноцінного функціонування застосунку SecurePass користувачу необхідне стабільне підключення до Інтернету. Без активного з’єднання з мережею робота програми неможлива.

Застосунок сумісний із популярними сучасними веб-браузерами. Його коректна робота гарантована у браузерах Google Chrome версії не нижче 80, Mozilla Firefox версії не нижче 78, Safari версії не нижче 14 та Microsoft Edge версії не нижче 80. Важливо, щоб браузер підтримував сучасні веб-стандарти, зокрема HTML5, CSS3 та JavaScript.

Для реєстрації у SecurePass, а також для використання двофакторної автентифікації (2FA), користувач повинен мати діючий email-акаунт. Якщо користувач обирає вхід через Google, йому необхідно мати активний обліковий запис Google.

Користувач несе повну відповідальність за збереження свого майстер-пароля до SecurePass або даних для входу через Google. У разі втрати цих даних доступ до облікового запису може бути втрачено.

2.5 Припущення й залежності

Для коректної взаємодії користувача із застосунком передбачається наявність стабільного підключення до мережі Інтернет. Без доступу до Інтернету використання функцій застосунку неможливе.

Вважається, що на пристрої користувача встановлений веб-браузер, який працює справно та підтримує необхідні веб-стандарти, такі як HTML5, CSS3 та JavaScript. Це є обов’язковою умовою для забезпечення коректного відображення інтерфейсу та функціональності застосунку.

Функціонування застосунку залежить від постійної доступності серверної інфраструктури, зокрема сервера, на якому розміщено бекенд-систему та базу даних. У разі відсутності доступу до сервера робота застосунку буде порушена.

Робота окремих функцій, зокрема надсилання електронних листів для підтвердження реєстрації чи двофакторної автентифікації (2FA), залежить від доступності зовнішніх сервісів. Застосунок повинен мати постійний доступ до SMTP-сервера для відправки повідомлень. Крім того, функція авторизації через Google можлива лише за наявності стабільного зв’язку з сервером автентифікації Google (Google Auth Server).

Припускається, що для роботи застосунку налаштовано SMTP-сервер, який забезпечує надсилання одноразових кодів підтвердження (OTP) користувачам.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача (UI)

Інтерфейс SecurePass є критично важливим для забезпечення зручності, ефективності та довіри користувачів до рішення, що працює з їхніми найчутливішими даними.

1. FR-UI-1.1 Зрозумілість та простота: Інтерфейс повинен бути максимально зрозумілим та простим для широкого кола користувачів, незалежно від їхнього технічного рівня.
2. FR-UI-1.2 Інтуїтивна навігація: Навігація по застосунку має бути інтуїтивно зрозумілою, дозволяючи користувачам легко знаходити потрібні функції (зберігання записів, генератор, налаштування) та виконувати основні операції (додавання, редагування, пошук) за мінімальну кількість кроків.
3. FR-UI-1.3 Візуальні підказки: Система повинна надавати візуальні підказки (наприклад, tooltips при наведенні на поля введення), повідомлення про успіх або помилку, а також інформативні модальні вікна для підтвердження критичних дій.
4. FR-UI-1.4 Мінімалістичний дизайн: Дизайн інтерфейсу сприятиме його простоті та ефективності, уникаючи зайвих елементів декору та надмірної інформації, що допоможе користувачам зосередитися на ключових функціях.
5. FR-UI-1.5 Інтеграція безпеки в дизайн: Безпека та захист конфіденційності даних повинні бути глибоко інтегровані у дизайн інтерфейсу. Це включає:
   * Візуальні індикатори відповідності рівню безпеки при генерації пароля.
   * Обережне поводження з відображенням паролів (наприклад, приховування за "оком" з опцією показати).
   * Надання попереджень про ризики (наприклад, при експорті даних).
6. FR-UI-1.6 Персоналізація: Повинна бути можливість перемикання між світлою та темною темами («нічний режим») для комфортної роботи в різних умовах освітлення.

3.1.2 Апаратний інтерфейс

FR-HW-1.1 Відсутність прямих вимог: Веб-застосунок «SecurePass» не висуває специфічних вимог до апаратного інтерфейсу на стороні клієнта.

FR-HW-1.2 Залежність від клієнтського пристрою: Функціонування залежить від можливостей кінцевого пристрою користувача (ПК, ноутбук, смартфон, планшет) коректно запускати та підтримувати сумісний веб-браузер.

3.1.3 Програмний інтерфейс

FR-SW-1.1 Бекенд-фреймворк: Бекенд застосунку реалізовано на мові програмування Python із використанням фреймворку Flask.

FR-SW-1.2 Система управління базами даних (СУБД): Для зберігання даних використано реляційну СУБД MySQL.

FR-SW-1.3 ORM (Object-Relational Mapping): Взаємодія з базою даних здійснюється за допомогою бібліотеки SQLAlchemy.

FR-SW-1.4 Криптографічні бібліотеки: Для шифрування (AES) та хешування (scrypt) використовуються бібліотеки Cryptography та Werkzeug Security відповідно.

FR-SW-1.5 OAuth клієнт: Для авторизації через Google-акаунт використовується бібліотека Authlib.

FR-SW-1.6 Фронтенд-технології: Користувацький інтерфейс розроблено з використанням HTML5, CSS3 та JavaScript.

FR-SW-1.7 Інші Python-бібліотеки: PyMySQL, python-dotenv, Flask-Migrate.

3.1.4 Комунікаційний протокол

FR-COM-1.1 Веб-трафік: Взаємодія між клієнтом та сервером реалізована за допомогою протоколу HTTP/HTTPS для забезпечення конфіденційності та цілісності даних під час передачі.

FR-COM-1.2 Формат обміну даними: Обмін даними між клієнтом та сервером здійснюється у форматі JSON.

FR-COM-1.3 Email комунікації: Для надсилання електронних листів (верифікація email, 2FA коди, сповіщення) використовується протокол SMTP.

FR-COM-1.4 Зв'язок з базою даних: Зв'язок з базою даних MySQL здійснюється по протоколу TCP/IP (порт 3306).

3.1.5 Обмеження пам'яті

FR-MEM-1.1 Серверна інфраструктура: Обмеження пам'яті визначаються доступною серверною інфраструктурою, на якій розгорнуто застосунок.

FR-MEM-1.2 Кешування: Для тимчасового зберігання даних (зокрема кодів 2FA) використовується in-memory кешування, що має враховувати доступну пам'ять сервера.

3.1.6 Операції

Цей підрозділ описує основні дії, які може виконувати система.

FR-OP-1.1 Реєстрація користувача: Система повинна дозволяти створення нового облікового запису.

FR-OP-1.2 Авторизація користувача: Система повинна забезпечувати процес входу користувача в обліковий запис.

FR-OP-1.3 Створення запису: Система повинна дозволяти користувачеві створювати новий запис облікових даних.

FR-OP-1.4 Редагування запису: Система повинна дозволяти користувачеві редагувати існуючий запис облікових даних.

FR-OP-1.5 Видалення запису: Система повинна дозволяти користувачеві видаляти існуючий запис облікових даних.

FR-OP-1.6 Перегляд запису: Система повинна дозволяти користувачеві переглядати деталі збереженого запису.

3.1.7 Функції продукту (деталізовані функціональні вимоги)

Цей розділ є розширенням та деталізацією функцій, згаданих у розділі 2.2, з прив'язкою до конкретних вимог.

3.1.7 Функції продукту (деталізовані функціональні вимоги)

FR-FUNC-1.1 Система управління парольними записами:

* FR-FUNC-1.1.1 Створення: Користувач повинен мати можливість створити новий запис з полями: site\_name (обов'язково), login (обов'язково), password (обов'язково), email, site\_url, phone\_number, secret\_word, nickname, custom\_id, old\_password, backup\_email, password\_hint.
* FR-FUNC-1.1.2 Перегляд: Авторизований користувач повинен мати можливість переглядати деталі будь-якого збереженого запису. Конфіденційні поля повинні дешифруватися та відображатися, з можливістю приховати/показати їх вміст.
* FR-FUNC-1.1.3 Редагування: Користувач повинен мати можливість змінювати дані в існуючому записі.
* FR-FUNC-1.1.4 Видалення: Користувач повинен мати можливість видалити запис облікових даних.
* FR-FUNC-1.1.5 Пошук: Система повинна підтримувати пошук записів за site\_name, login, email, site\_url, nickname, custom\_id.
* FR-FUNC-1.1.6 Сортування: Система повинна надавати можливість сортування списку записів за site\_name (за алфавітом) та date\_added (за датою додавання).

FR-FUNC-1.2 Генератор паролів:

FR-FUNC-1.2.1 Рівні складності: Генератор повинен дозволяти вибір рівня складності:

* + "Легкий" (малі літери, цифри, довжина ≥ 8),
  + "Середній" (великі/малі літери, цифри, довжина ≥ 12),
  + "Складний" (великі/малі літери, цифри, спецсимволи, довжина ≥ 8),
  + "Парольна фраза" (випадкові слова, розділені дефісом, 2–10 слів),
  + "PIN-код" (тільки цифри, 4–10 цифр).
* FR-FUNC-1.2.2 Параметри: Користувач повинен мати можливість вказати довжину пароля/кількість слів/цифр в межах, визначених для кожного типу.
* FR-FUNC-1.2.3 Копіювання: Згенерований пароль повинен бути доступний для копіювання в буфер обміну.

FR-FUNC-1.3 Автентифікація:

* FR-FUNC-1.3.1 Реєстрація: Нові користувачі повинні мати можливість зареєструватися, надавши ім'я, email та майстер-пароль (мінімум 8 символів). Email повинен бути унікальним.
* FR-FUNC-1.3.2 Верифікація Email: Після реєстрації на email користувача надсилається одноразовий код для верифікації облікового запису.
* FR-FUNC-1.3.3 Вхід за логіном/паролем: Користувач повинен мати можливість увійти за допомогою зареєстрованого email та майстер-пароля.
* FR-FUNC-1.3.4 Двофакторна автентифікація (2FA): Якщо 2FA увімкнено, після успішного введення логіна/пароля, на email користувача надсилається 6-значний код для підтвердження входу.
* FR-FUNC-1.3.5 Повторне надсилання коду: Користувач повинен мати можливість повторно надіслати верифікаційний код (для реєстрації) або 2FA код (для входу) після певного періоду часу.
* FR-FUNC-1.3.6 Авторизація через Google: Користувачі повинні мати можливість авторизуватися за допомогою свого Google-акаунта.

FR-FUNC-1.4 Захист акаунту:

* FR-FUNC-1.4.1 Захист від Brute-Force: Система повинна блокувати вхід на 10 хвилин після 3 невдалих спроб введення пароля.
* FR-FUNC-1.4.2 Автоматичний вихід: Система повинна автоматично завершувати сесію користувача після 10 хвилин бездіяльності.

FR-FUNC-1.5 Налаштування акаунту:

* FR-FUNC-1.5.1 Зміна майстер-пароля: Користувач повинен мати можливість змінити свій майстер-пароль, що вимагає введення поточного пароля та підтвердження через OTP, надісланий на email.
* FR-FUNC-1.5.2 Керування 2FA: Користувач повинен мати можливість увімкнути або вимкнути 2FA.
* FR-FUNC-1.5.3 Керування автоматичним виходом: Користувач повинен мати можливість увімкнути або вимкнути автоматичний вихід.
* FR-FUNC-1.5.4 Керування нічним режимом: Користувач повинен мати можливість увімкнути або вимкнути нічний режим інтерфейсу.
* FR-FUNC-1.5.5 Видалення акаунту: Користувач повинен мати можливість безповоротно видалити свій акаунт та всі пов'язані з ним дані, підтвердивши дію поточним паролем.

3.1.8 Припущення й залежності (для конкретних функцій):

* FR-DEP-1.1 Робота SMTP-сервера: Комунікація для 2FA, верифікації email та зміни пароля залежить від коректної роботи SMTP-сервера та налаштувань EMAIL\_HOST, EMAIL\_PORT, EMAIL\_USER, EMAIL\_PASS у файлі .env.
* FR-DEP-1.2 Доступність Google OAuth: Функціонал авторизації через Google залежить від стабільної роботи Google Auth Server та правильної конфігурації GOOGLE\_CLIENT\_ID та GOOGLE\_CLIENT\_SECRET.
* FR-DEP-1.3 Ключ шифрування: Шифрування/дешифрування даних залежить від наявності та коректності змінної ENCRYPTION\_KEY\_BASE64 у файлі .env.
* FR-DEP-1.4 Робота кеш-сервісу: Система кешування (CacheService) використовується для тимчасового зберігання OTP-кодів і вимагає стабільної роботи.

3.2 Властивості програмного продукту (Атрибути якості)

3.2.1 Надійність:

* FR-REL-1.1 Цілісність даних: Система повинна забезпечувати цілісність усіх збережених даних, запобігаючи їх пошкодженню або втраті. Надійність також підвищується за рахунок регулярних резервних копій (хоча це адміністративна, а не програмна функція).
* FR-REL-1.2 Відмовостійкість: Система повинна бути стійкою до непередбачених збоїв, мінімізуючи втрату даних та забезпечуючи швидке відновлення (використання транзакцій бази даних).
* FR-REL-1.3 Логування помилок: Система повинна активно логувати помилки та попередження для забезпечення можливості діагностики та підтримки.

3.2.2 Доступність:

* FR-AVA-1.1 Доступність сервісу: Застосунок повинен бути доступним 24/7 для авторизованих користувачів через веб-інтерфейс, за умови стабільної роботи серверної інфраструктури та інтернет-з'єднання.
* FR-AVA-1.2 Автоматичний вихід: Для підвищення безпеки система надає опцію автоматичного виходу користувача після періоду бездіяльності.

3.2.3 Безпека

Безпека є центральною та пріоритетною властивістю SecurePass:

* FR-SEC-1.1 Шифрування конфіденційних даних: Усі чутливі дані (паролі, секретні слова, підказки, попередні паролі) повинні шифруватися за допомогою симетричного алгоритму AES-256 перед збереженням у базу даних.
* FR-SEC-1.2 Захист майстер-пароля: Майстер-пароль користувача не зберігається у відкритому вигляді, а хешується за допомогою алгоритму scrypt з унікальною криптографічною "сіллю", що забезпечує стійкість до атак перебору.
* FR-SEC-1.3 Двофакторна автентифікація (2FA): Реалізовано через надсилання одноразового коду на email користувача для додаткового рівня захисту при вході.
* FR-SEC-1.4 Захист від атак перебору (Brute Force): Система повинна тимчасово блокувати форму входу після трьох невдалих спроб введення пароля на 10 хвилин.
* FR-SEC-1.5 Валідація вхідних даних: Система повинна коректно обробляти помилки введення даних користувачем для запобігання некоректним операціям.
* FR-SEC-1.6 Захист сесії: Сесії користувачів повинні бути захищені через параметри SESSION\_COOKIE\_SECURE=True, SESSION\_COOKIE\_HTTPONLY=True, SESSION\_COOKIE\_SAMESITE='Lax'.

3.2.4 Супроводжуваність:

* FR-MAINT-1.1 Модульність архітектури: Програмне забезпечення має бути реалізоване з використанням багатошарової архітектури (Domain, Repositories, Services, Application Core, Utilities, External) для полегшення розуміння, супроводження та розширення.
* FR-MAINT-1.2 Конфігурованість: Чутливі конфігураційні дані (наприклад, DATABASE\_URL, SECRET\_KEY, EMAIL\_USER) повинні зберігатися у файлі .env для зручності управління та деплойменту.
* FR-MAINT-1.3 Логування: Система повинна вести логування подій та помилок (наприклад, спроби входу, помилки шифрування, помилки БД) для полегшення діагностики та супроводу.

3.2.5 Переносимість:

* FR-PORT-1.1 Крос-браузерність: Веб-застосунок повинен коректно відображатися та працювати в сучасних версіях популярних веб-браузерів.
* FR-PORT-1.2 Незалежність від ОС клієнта: Програмне забезпечення не повинно висувати специфічних вимог до операційної системи користувача — доступ відбувається через браузер.
* FR-PORT-1.3 Технологічна переносимість: Завдяки використанню поширених кросплатформних технологій (Python, Flask, MySQL, HTML5, CSS3, JavaScript), система може бути перенесена на різні серверні платформи.

3.2.6 Продуктивність:

* FR-PERF-1.1 Час відгуку: Система повинна швидко реагувати на дії користувача (реєстрація, вхід, CRUD-операції, генерація паролів, зміна налаштувань).
* FR-PERF-1.2 Швидкість обробки криптографічних операцій: Шифрування та хешування повинні виконуватись ефективно, без помітних затримок.
* FR-PERF-1.3 Ефективність завантаження/сортування: Завантаження та сортування списків парольних записів має відбуватися ефективно навіть при великому обсязі даних.

3.3 Вимоги до бази даних:

* FR-DB-1.1 СУБД: Для зберігання даних використовується реляційна СУБД MySQL.
* FR-DB-1.2 Моделювання: Структура бази даних повинна бути спроєктована з використанням ER-діаграми.
* FR-DB-1.3 Сутності: Основні сутності — "користувачі" (таблиця user) та "записи паролів" (таблиця entries).
* FR-DB-1.4 Зв'язки: Між "user" і "entries" має бути зв'язок "один до багатьох", реалізований через поле user\_id.
* FR-DB-1.5 Нормалізація: Структура БД має відповідати третій нормальній формі для уникнення надлишковості та аномалій.
* FR-DB-1.6 Конфіденційність: Поле enc\_password у таблиці entries повинно зберігати паролі у зашифрованому вигляді (AES).
* FR-DB-1.7 ORM: Взаємодія з базою даних реалізована за допомогою ORM SQLAlchemy.

3.4 Інші вимоги:

* FR-OTHER-1.1 Авторське право: Програмний продукт «SecurePass» захищений авторським правом відповідно до поданих документів на реєстрацію.
* FR-OTHER-1.2 Відповідність стандартам: Розробка враховує принципи стандартів кібербезпеки, зокрема ISO/IEC 27001.
* FR-OTHER-1.3 Навчальне використання: Застосунок може бути використаний навчальним закладом безкоштовно та без обмежень у часі.

4 ДОДАТКОВІ МАТЕРІАЛ

4.1 Діаграми UML

Діаграма прецедентів (Use Case Diagram): Моделює функціональні вимоги системи з точки зору взаємодії акторів із системою (Див. Рисунок 1-2).

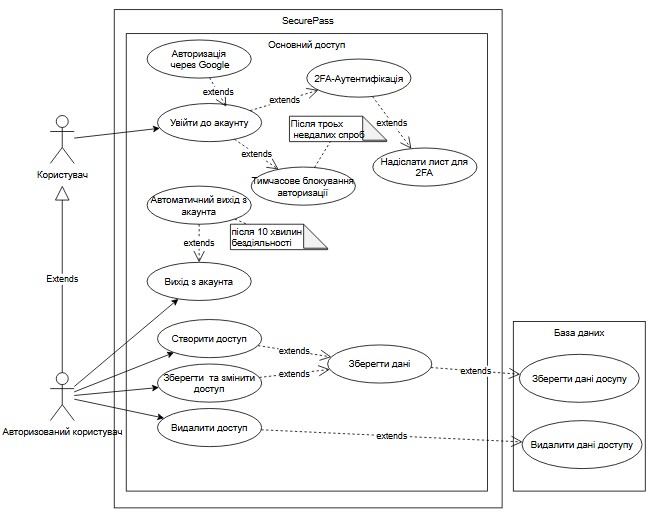


Рисунок 1 – Перша частина діаграми прецедентів веб-застосунку «SecurePass» (основний доступ та безпека)

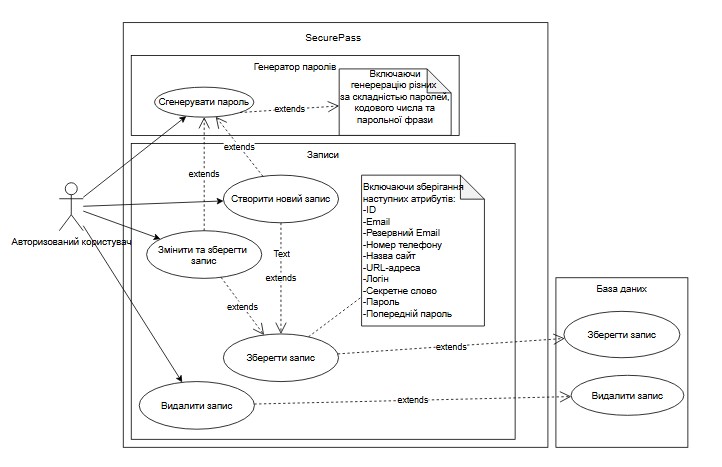


Рисунок 2 – Друга частина діаграми прецедентів веб-застосунку «SecurePass» (управління записами та генерація паролів).

Діаграма станів (State Diagram): Моделює динамічну поведінку застосунку, зокрема логіку управління доступом користувачів(Див. Рисунок 1-3).

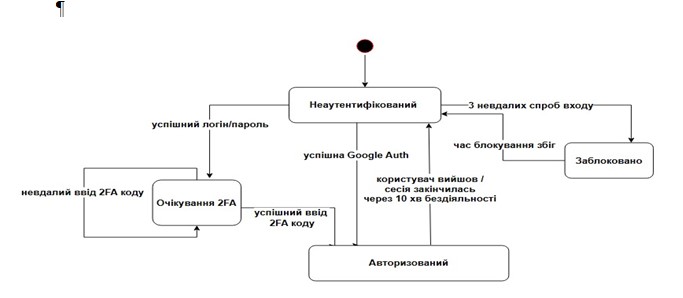


Рисунок 3 – Діаграма станів веб-застосунку «SecurePass».

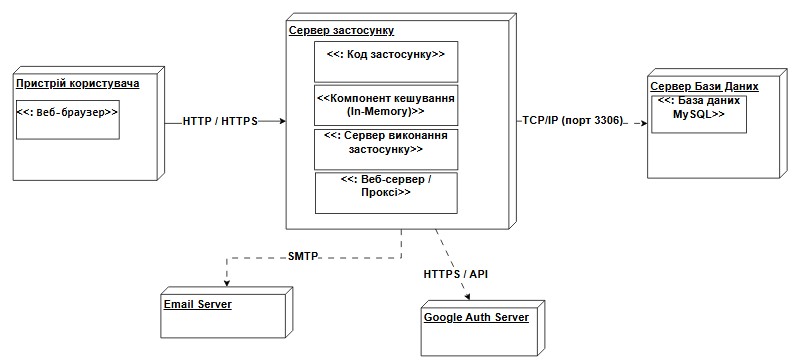
Діаграма розгортання (Deployment Diagram): Візуалізує фізичну структуру та середовище виконання веб-застосунку, включаючи ключові вузли та комунікаційні канали(Див. Рисунок 4).

Рисунок 4 – Діаграма розгортання веб-застосунку «SecurePass».

Діаграма пакетів (Package Diagram): Відображає логічну організацію коду, показуючи, як класи та інші елементи групуються в модулі або пакети(Див. Рисунок 5).

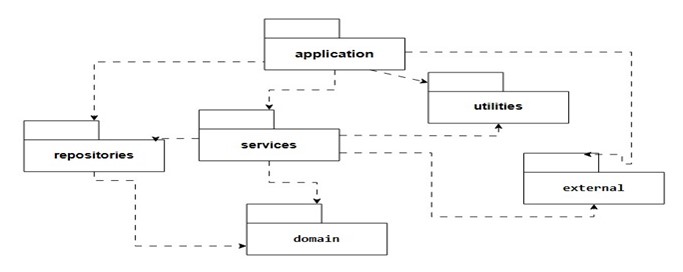


Рисунок 5– Діаграма пакетів веб-застосунку «SecurePass».

Діаграма класів (Class Diagram): Відображає основні типи об'єктів (класи), їхні атрибути, операції та взаємозв'язки між ними, заглиблюючись у внутрішню статичну структуру коду(Див. Рисунок 6).

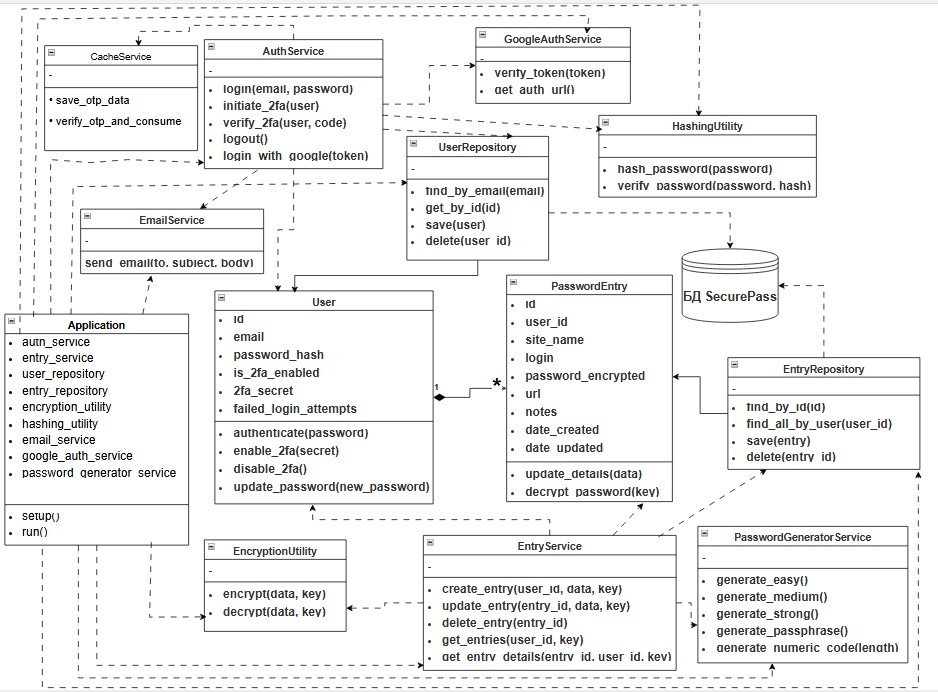


Рисунок 6 – Діаграма класів веб-застосунку «SecurePass».

4.2 ER-діаграма

* Діаграма "сутність-зв'язок" (ERD): Використана для планування та створення бази даних, візуалізуючи сутності системи, їхні атрибути та взаємозв'язки(Див. Рисунок 7).

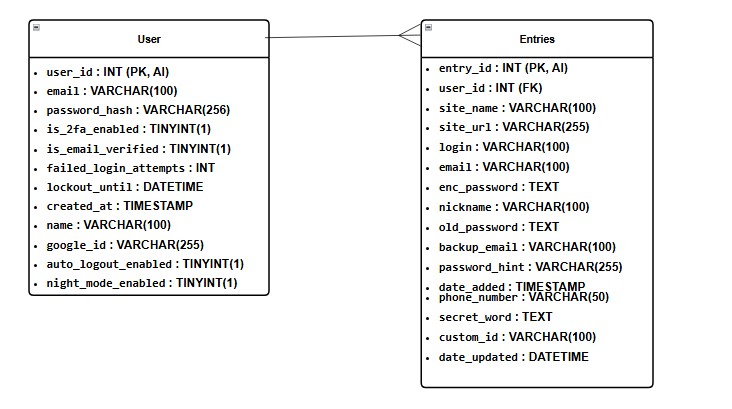


Рисунок 7 – ER-діаграма бази даних «SecurePass».

4.3 Тест-кейси

Для документування процесу тестування розроблено набір тест-кейсів, що охоплюють ключові сценарії використання застосунку (наприклад, реєстрація, створення парольного запису) (Див. Рисунок 8-9).

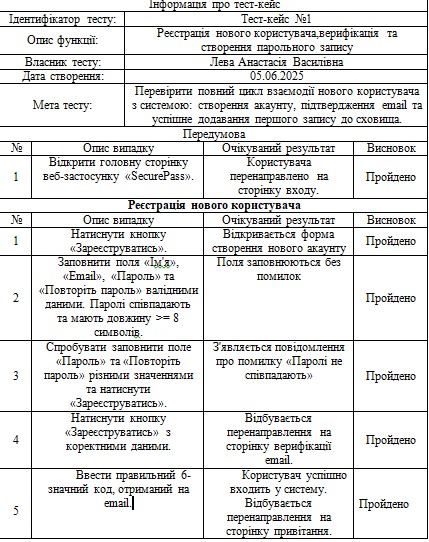


Рисунок 9 – Приклад тест-кейсу №1 для веб-застосунку «SecurePass».



Рисунок 9 – Приклад тест-кейсу №2 для веб-застосунку «SecurePass».

4.4 Фрагменти коду

Включено приклади ключових фрагментів коду, що ілюструють реалізацію важливих механізмів та функцій (наприклад, реєстрація, 2FA, захист від підбору пароля, шифрування, генерація паролів) (Див. Рисунок 11-12).



Рисунок 11 – Приклад фрагменту коду для захисту від підбору пароля в «SecurePass».

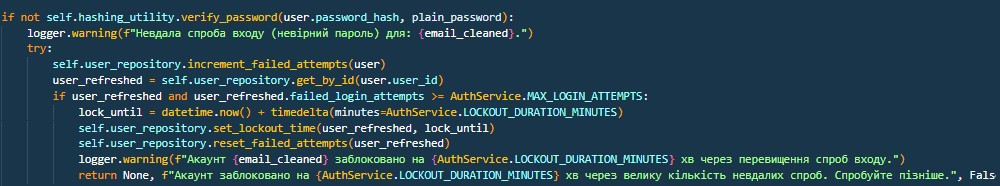


Рисунок 12 – Приклад фрагменту коду для захисту від підбору пароля в «SecurePass».

4.5 Скріншоти інтерфейсу

Документація містить візуальні приклади користувацького інтерфейсу застосунку, включаючи екрани реєстрації, входу, головну панель, форму управління записами, генератор паролів та налаштування аканту (Див. Рисунок 13-15)

.

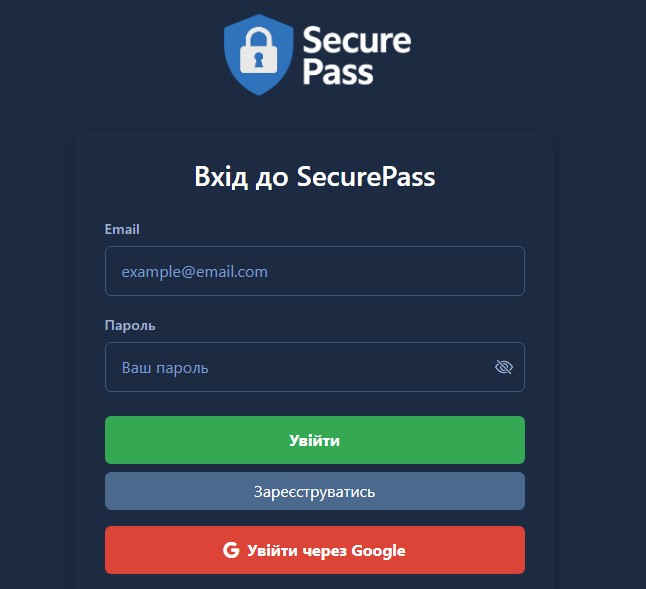


Рисунок 13 – Приклад користувацького інтерфейсу веб-застосунку «SecurePass».

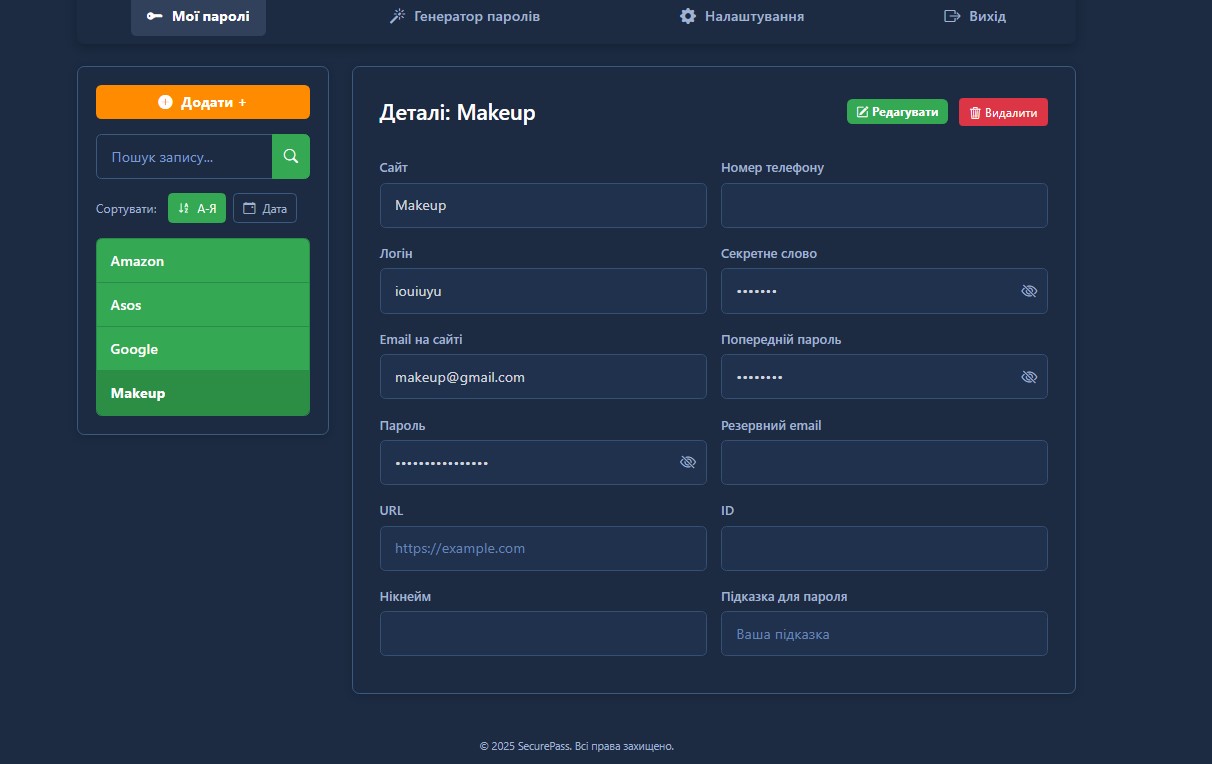


Рисунок 14– Приклад користувацького інтерфейсу веб-застосунку «SecurePass».

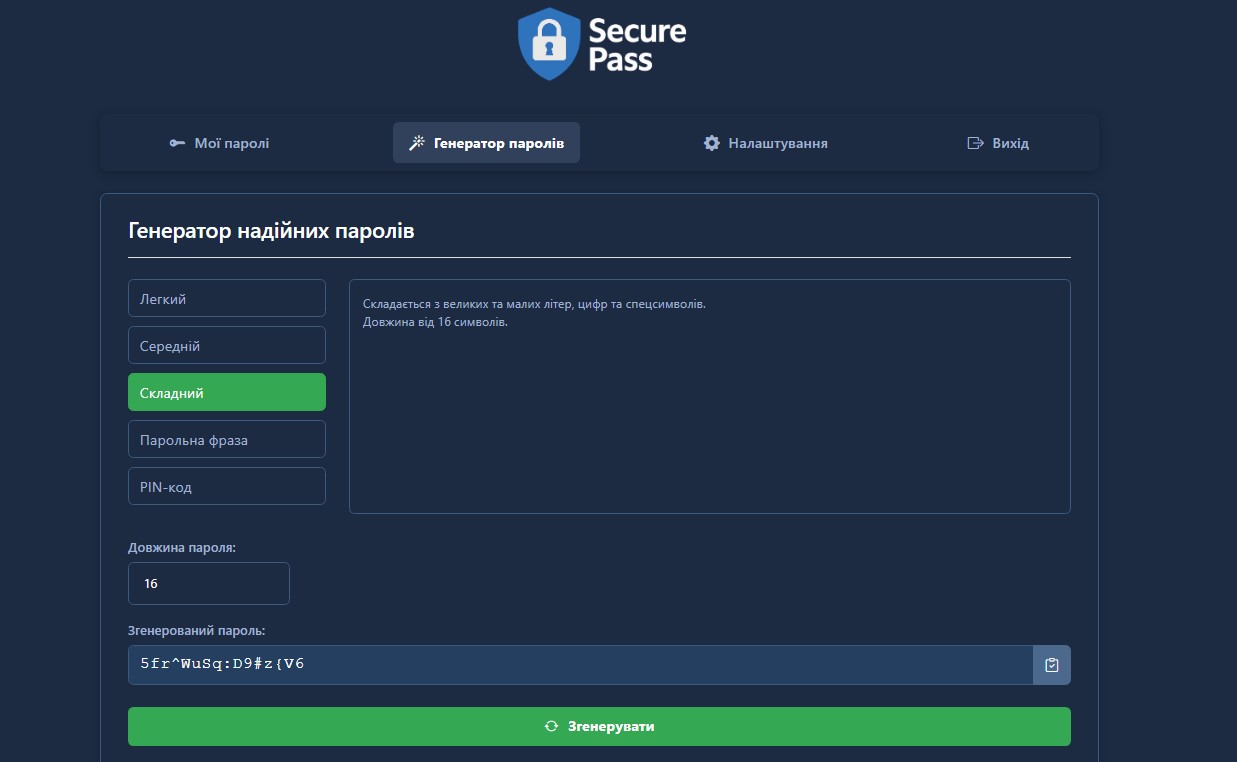


Рисунок 15 – Приклад користувацького інтерфейсу веб-застосунку «SecurePass».