Bezpieczeństwo systemów informatycznych – Manager Haseł
Plan realizacji

Grzegorz Nieużyła
Dawid Karolewski

Z uwagi na mnogość kont na różnorakich portalach posiadanych przez statystycznego użytkownika Internetu, konieczne jest tworzenie różnych i silnych haseł aby uniknąć przechwycenia kont po wyciekach danych. Aby ułatwić ten proces tworzone są narzędzia pozwalające na automatyczne generowanie i bezpieczne przetrzymywanie haseł.

1 Opis projektu

Tematem projektu jest zaprojektowanie i implementacja aplikacji desktopowej pełniącej funkcję managera haseł. Końcowa aplikacja ma pozwalać na:

- Przechowywanie haseł/sekretów w odpowiednio zabezpieczonej, lokalnej bazie danych.
- Ustawienie głównego hasła za pomocą którego będą szyfrowane wszystkie wprowadzone dane
- Generowanie trudnych haseł o konfigurowalnej złożoności, długości i zbiorze znaków aby umożliwić działanie na stronach o różnych zasadach tworzenia haseł.
- Ustawienie okresu ważności haseł, wyświetlanie przypomnień o konieczności zmiany w formie alertów.
- Wyświetlanie złożoności hasła przy manualnym wpisywaniu haseł
- Wklejanie hasła do schowka systemowego
- Integrację z przeglądarkami internetowymi opartymi na Chromium za pomocą rozszerzenia.

2 Harmonogram

- 03.11 - Przedstawienie planu i rozpoczęcie fazy implementacji.
- 14.11 - Prototyp aplikacji - możliwość zapisywania haseł w zabezpieczonej bazie danych.
- 24.11 - Przedstawienie 75% implementacji.
- 15.12 - Skończenie fazy implementacji - rozpoczęcie fazy testów.
- 15.01 - Złożenie gotowego projektu.

3 Technologie

- Python - logika biznesowa.
- Javascript - plugin do przeglądarki internetowej.
- PyQt - graficzny interfejs użytkownika dla wersji desktopowej.
- PyCryptodome - operacje kryptograficzne.

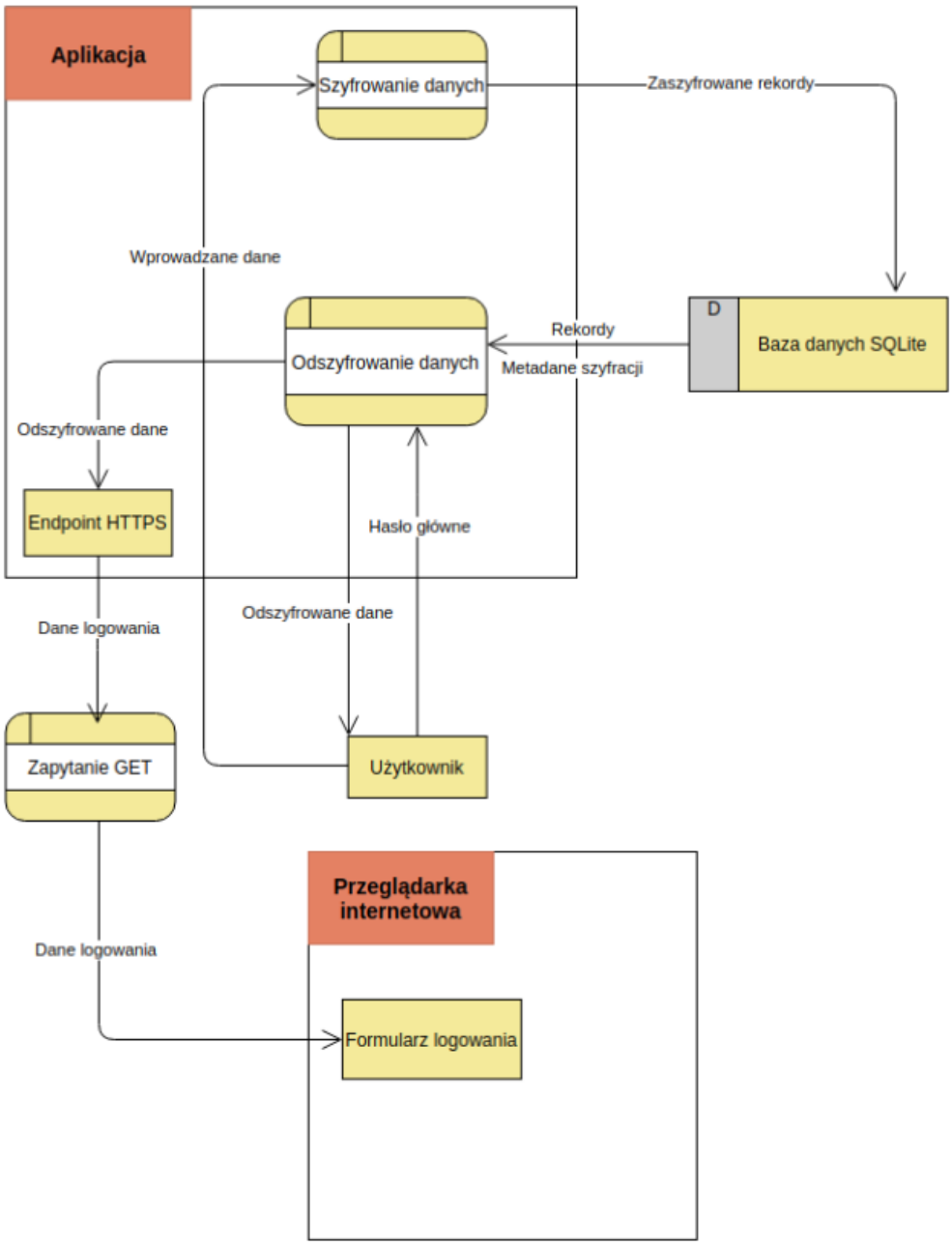
Wybór Pythona podyktowany został przez prostotę i szybkość implementacji, a także dużą dostępność bibliotek pomocniczymi. Kolejnym czynnikiem wyboru następujących technologii były wcześniejsze doświadczenia z tymi narzędziami. Uzyskujemy również multiplatformowość. Javascript jest jedyną technologią pozwalają na implementację rozszerzenia do przeglądarek internetowych.

4 Architektura

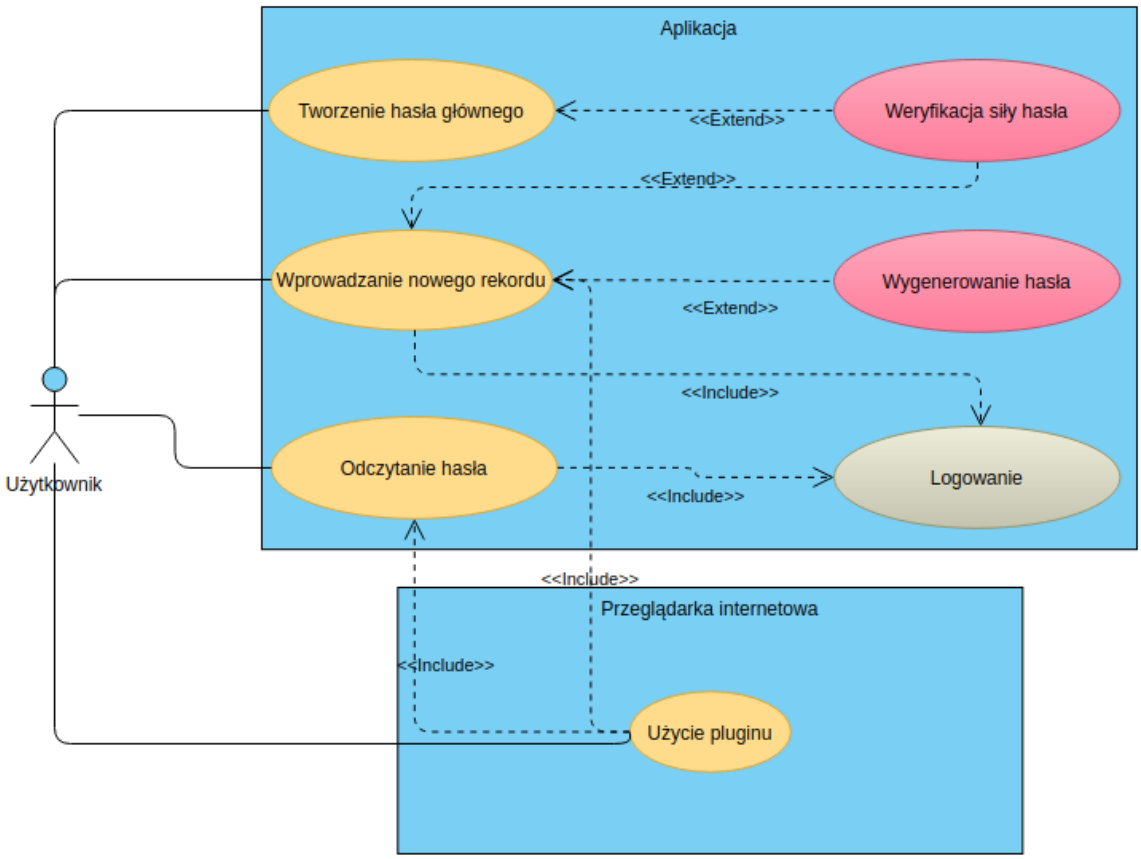
W skład projektu wchodzą dwa komponenty – aplikacja okienkowa i plugin do przeglądarki. Aplikacja służy do zarządzania danymi, umożliwia zmianę lub dodawanie nowych rekordów a plugin do przeglądarki pozwala na automatyczne wpisanie danych do logowania pod warunkiem że aplikacja jest włączona.

Rozszerzenie do aplikacji otrzymuje dane logowania z endpointu HTTPS wystawionego przez aplikację w sieci lokalnej komputera.

Dane użytkownika są składowane w lokalnej bazie danych SQLite, której schemat jest następujący: w tabeli *records* znajdują się wszystkie rekordy tj. indywidualne hasła i dane pomocnicze dla poszczególnych stron internetowych lub aplikacji. Dane właściwe znajdują się w pojedynczej kolumnie *json_record_data* w której znajduje się zaszyfrowany obiekt JSON (dokładny schemat jest przedstawiony na rysunku 4). Dane są zaszyfrowane algorytmem AES, dlatego w tabeli znajduje się także wektor inicjalizacyjny IV konieczny do deszyfracji. Dodana została także tabela *encryption_metadata* w której znajdują się parametry do algorytmu generowania klucza z hasła PBKDF2 tj. sól dodatkowo zabezpieczająca hasło, liczba iteracji, użyta funkcja skrótu oraz długość klucza.



Rysunek 1: Diagram DFD



Rysunek 2: Diagram UML przypadków użycia

records	
record_id	**int**
aes_iv	blob
json_record_data	blob

encryption_metadata	
salt	blob
iterations	int
hmac	text
key_len	int

Rysunek 3: Schemat lokalnej bazy danych

json_record_data	
title	string
website	string
loginUrl	string
login	string
password	string
description	string
modificationDate	integer

Rysunek 4: Schemat pojedynczego rekordu zaszyfrowanego w bazie danych

5 Interfejs - API

API wystawione przez aplikację pozwala na przekazywanie danych logowania do rozszerzenia przeglądarki. Dostępne są następujące endpointy:

- GET v1/api/sites – zwraca listę stron dla których dostępne są dane logowania. Zwracana wartość:

```
[
    "www.example.com",
    ...
]
```

- GET v1/api/password?url={url} – Zwraca dane logowania dla danej strony internetowej. Zwracana wartość:

```
{
    "login" : "...",
    "password": "..."
}
```

- GET v1/api/createpassword?url={url}&?login={login} – Generuje i zwraca dane logowania dla danej strony internetowej. Zwracana wartość:

```
{
    "password": "..."
}
```