Warszawa, 21.04.2021

Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych

Bezpieczeństwo medycznych systemów informacyjnych (BEMSI)

Projekt: Aplikacja web/mobile – szyfrowany notatnik z przechowywaniem w chmurze

Dokumentacja wstępna

Prowadzący: dr inż. Robert Kurjata

Wykonawcy:

Adamiuk Zuzanna 300444, Depko Kinga 300452

Krakowiak Aleksandra 290292, Rancew Joanna 300465

Spis treści

Ogólna koncepcja

Wstęp i cele

Głównym celem projektu jest stworzenie szyfrowanego notatnika, który będzie umożliwiał bezpieczne przechowywanie danych w chmurze. Operacje aplikacji wykonywane będą zapewniając poufność, za równo w trakcie przechowywania i przesyłania notatek. Na każdym etapie pracy z danymi, stosowane będą odpowiednie zabezpieczenia, by nie były one dostępne dla osób trzecich.

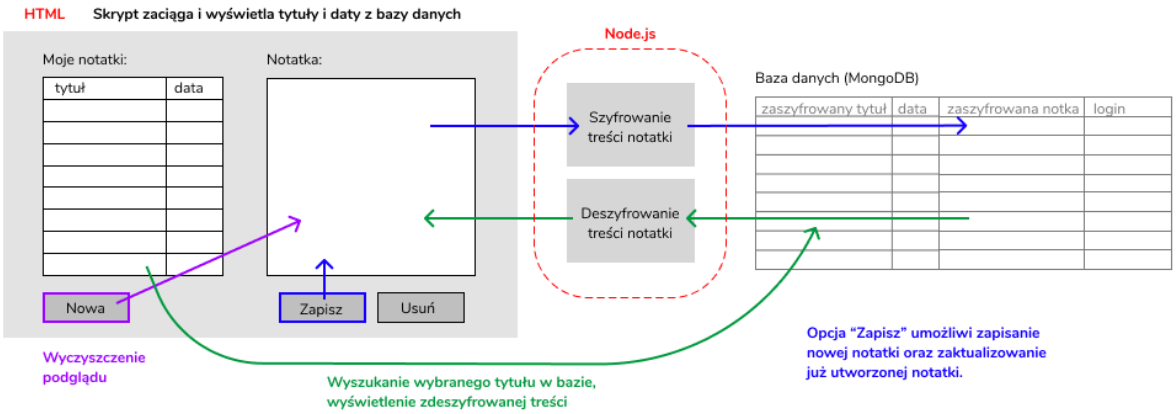
Projekt rozwiązania

Dostęp dla użytkownika

Dostęp do notatnika będzie umożliwiony z poziomu przeglądarki poprzez aplikację webową. Aplikacja będzie wymagała od użytkownika logowania (login oraz hasło) - umożliwi to korzystanie z aplikacji wielu użytkownikom tego samego komputera lub z różnych miejsc. Użytkownik będzie miał możliwość tworzenia notatek, które zostaną odpowiednio zaszyfrowane i zabezpieczone tak, by osoby postronne nie miały do nich dostępu (pozostali użytkownicy aplikacji). W systemie nie będzie możliwości pobrania notatki, by zapewnić przechowywanie ich jedynie w bezpieczny, zaszyfrowany sposób.

Projekt działania:

Działanie naszego rozwiązania przedstawia poniższy schemat:



Rysunek 1. Uproszczony schemat działania aplikacji.

Po uruchomieniu aplikacji będzie wyświetlał się panel logowania – login, hasło oraz opcja rejestracji. Następnie, po pomyślnym logowaniu, wyświetla się tabela „Moje notatki" z rozszyfrowanymi już tytułami notatek danego użytkownika i puste pole „Notatka". Użytkownik może wybrać, która notatka zostanie wyświetlona poprzez wybór tytułu z tabeli, w momencie wyboru, serwer pobiera informacje o wybranym tytule – znajduje notatkę odpowiadającą wybranemu tytułowi, serwer ją pobiera z bazy danych, deszyfruje treść i przesyła na frontend.

Oprócz tego wyświetlone zostaną przyciski:
„Nowa” – czyszczący pole „Notatka”, umożliwiający tworzenie nowej notatki,
„Zapisz” – przesyłający treść pola „Notatka” do serwera, gdzie treść zostaje zaszyfrowana i przesłana do bazy danych – umożliwia to zarówno modyfikację, jak i zapisywanie nowej notatki,
„Usuń” – pozwala na usunięcie wyświetlanej notatki (nie występuje tu proces szyfrowania)

Podczas rejestracji dla nowego użytkownika generowany jest klucz prywatny, który będzie przechowywany lokalnie, w bezpiecznym środowisku.

Podczas logowania oraz rejestracji będzie sprawdzane, czy użytkownik o takiej nazwie już istnieje – jeśli tak, użytkownik zostanie poinformowany. (Login jest unikatowy.)

Model zabezpieczeń

Głównym celem zabezpieczeń jest zapewnienie poufności oraz uwierzytelniania dostępu. Bezpieczeństwo zostanie zapewnione poprzez wprowadzenie logowania użytkowników oraz szyfrowanie rekordów w bazie danych (informacji dot. notatek).

Przesyłanie danych

Przesyłanie danych powinno zapewniać poufność oraz integralność danych. Notatki będą chronione za pomocą kombinacji szyfrowania kryptografią symetryczną i HMACa generującego tag uwierzytelniający. Ochrona danych przesyłanych opiera się na tym, że notatka będą szyfrowane za pomocą szeroko stosowanego szyfru blokowego AES-256, używając prywatnego klucza. Notatki będą przesyłane jako zaszyfrowane. Notatki będą także uwierzytelniane za pomocą HMACs, którego wyznaczenie odbywać się będzie z wcześniej zaszyfrowanego tekstu jawnego. Aplikacja zapewniać będzie poprawne szyfrowanie oraz deszyfrowanie.

Przechowywanie danych

Notatki przechowywane będą w nierelacyjnej bazie danych. Aplikacja zapewniać będzie poprawną komunikację z bazą danych. Model przechowywania danych notatki zakłada 5 atrybutów: ID notatki, login autora, tytuł, treść oraz datę utworzenia. Login autora notatki będzie przechowywany jako jawny. Jako zaszyfrowane zostaną przechowywane tytuły, treści oraz daty utworzenia notatek. Takie rozwiązanie zabezpiecza przechowywane dane przed dotarciem do zewnętrznego odbiorcy, nawet przy ich wycieku z bazy. Treść notatki będzie cały czas dostępna i możliwa do odszyfrowania, jednakże tylko poprzez użycie klucza prywatnego. Kolejną korzyścią, jaka płynie z założonego schematu przechowywania danych jest możliwość odzyskania ich mimo utraceniu dostępu do przeglądarki czy konta użytkownika.

Bezpieczne logowanie

Login oraz hasło każdego użytkownika będą przechowywane w nierelacyjnej bazie danych. Nazwy użytkowników będą informacjami jawnymi, zaś hasła będą szyfrowane poprzez Hash z Salt, w celu zapewnienia dodatkowej ochrony (w przypadku powtarzających się haseł). Taka technologia zapewnia możliwość szyfrowania tylko w jedną stronę – hasło w momencie utworzenia jest szyfrowane bez możliwości późniejszego odszyfrowania go. Poprawność wprowadzonego przy logowaniu hasła jest sprawdzana przez porównanie wartości wynikowej funkcji hashującej zastosowanej na wprowadzonym haśle do zaszyfrowanego hasła przechowywanego w bazie.

Proponowane narzędzia

By dobrze rozplanować pracę oraz zapoznać się z możliwościami, postawiłyśmy pewne założenia odnośnie wykorzystywanych technologii. Zakładamy jednak wprowadzanie zmian na dalszych etapach pracy nad projektem zarówno w tym zakresie, jak również udoskonalając model zabezpieczeń.

Praca w zespole

Podstawowym narzędziem, na którym będziemy opierać współpracę w ramach projektu jest GitLab https://gitlab-stud.elka.pw.edu.pl, na którym udostępniane będą efekty pracy na kolejnych etapach projektu, a także komentarze i uwagi, co ułatwi płynną współpracę wśród członków naszego zespołu.

Wykorzystywane technologie

Do napisania backend-u aplikacji wykorzystamy oprogramowanie Node.js z bibliotekami Express, cryptico i bcrypt (funkcje hash, salt), zaś do frontend-u - język HTML z CSS-em. Aby połączyć ze sobą wspomniane warstwy, najprawdopodobniej skorzystamy ze skryptów napisanych w JavaScripcie. Do przechowywania notatek oraz danych użytkowników (2 bazy danych) zostanie wykorzystana nierelacyjna baza danych, do której obsługi wykorzystamy system MongoDB.