Luchowski Norbert  
Olesiński Radosław

Smyrak Kacper

Wojtas Przemysław

# Architektura

Architektura typu client-server z SPA frontendem (Vue.js) oraz monolitycznym backendem (ASP.NET) komunikującym się przez REST API.

# Stos technologiczny

# Backend

C# .NET 8 LTS,  
EntityFramework, OAuth2 (jesli szyfrowanie nie bedzie zadowalajace), do komunikacji SignalR (na pewno jest dla real-time), libsodiumb, kontenery, albo do skonteneryzowania całej aplikacji albo tylko dla bazy danych.

Testy: Xunit, Moq, TestServer, testcontainers

Logowanie i autoryzacja przy pomocy ciasteczek

Odpowiednie endpointy:

* Security: logowanie, rejestracja, reset hasła,szyfrowanie
* Raporty: generowanie danych o walorachpdf/xlsx po filtracji lub sortowaniu)
* Walory: zarządzanie walorami, dodawanie,edytowanie pól nowych, CRUD
* Kolekcje: zarządzanie kolekcjamiwalorów, CRUD
* Użytkownik: tworzenie/usuwanie zaproszeń,prywatność kolekcji, zmiana   
  danych,zarządzanie znajomymi, prywatność kolekcji,zmiana danych
* WaloryCudze: mozliwosc udostepnieniawlasnych walorow innym, czyli uzytkownikbedzie mogl poszukac czy ktos nieudostepnil juz wlasnej
* Kategorie: podobnie co do walorów zpodziałem na dwa, CRUD
* CudzeKolekcje: wyswietlanie cudzychkolekcji z paginacja, sortowanie, tworzenie komentarzy
* Powiadomienia: akceptacja zaproszeń,usuniecie powiadomienia
* +Komunikator

# Baza Danych

PosgreSQL, dużo możliwości, relacyjna, coś nowego w porównaniu do MySQL. Może stworzenie dwóch baz 1. do rozmów, 2. do portalu.

Encje:   
- użytkownik (dane personalne, email, hasło, przyjaciele, kolekcje)  
- kolekcje (informacje na jej temat, walory, komentarze, status kto może oglądać)  
- komentarz (kto, kiedy, treść)  
- kategoria   
- walor (kogo, treść, kategoria) planujemy użyć pola JSON z treścią, oraz drugą encje dla publicznych walorów? Gdzie będą polubienia może komentarze   
- Powiadomienia: dwa typy, powiadomienie z zaproszeniem, nieprzeczytane wiadomosci

# Frontend

## Framework główny: Vue.js 3

* Lekki, szybki, z reaktywnym systemem danych
* Łatwo skalowalny projekt dla SPA (Single Page Application)
* Składnia oparta na komponentach (łatwo zarządzać widokami walorów, kategorii, znajomych itd.)

## Najważniejsze używane biblioteki:

|  |  |  |
| --- | --- | --- |
| **Zastosowanie** | **Biblioteka** | **Opis** |
| Stylowanie | Tailwind CSS | Szybkie tworzenie responsywnych interfejsów. |
| Formularze | VeeValidate + Yup | Walidacja danych wejściowych |
| Routing | Vue Router | Nawigacja po podstronach |
| Stan Globalny | Pinia | Zarządzanie stanem zalogowanego użytkownika, kolekcji itd. |
| Komunikacja z API | Axios | Obsługa żądań HTTP |
| Eksport danych | JsPDF, SheetJS | Eksport danych do PDF lub xlsx |

## Zakładki / Podstrony

Login (domyślny bez użytkownika zalogowanego)

* Służy do zalogowania się lub stworzenia konta

Dashboard (domyślny jeśli użytkownik jest zalogowany)

* Główna strona
* Podgląd ostatnio dodanych walorów
* Skróty do ulubionych kategorii
* Notyfikacje

Kolekcja

* Widok listy walorów (z filtrami, sortowaniem)
* Możliwość przełączania między kategoriami
* Edycja / usuwanie / dodawanie nowego waloru

Kategorie

* Lista dostępnych kategorii
* Dodanie nowej kategorii z specjalnymi polami

Znajomi

* Lista znajomych
* Możliwość wysłania zaproszenia
* Podgląd kolekcji znajomego

Chat

* End to end chat
* Powiadomienia o nowych wiadomościach

Ustawienia konta

* Zmiana hasła, e-maila itd.
* Zarządzanie prywatnością

Eksport

* Wybór formatu (PDF/xlsx)
* Filtry, które mają zostać zastosowane przed eksportem

## Inne pomysły (oryginalne):

* Light / Dark mode lub/i pełny wybór motywu (niekoniecznie jak w czarnych fordach)
* Tryb ograniczonego widoku dla innych użytkowników (w stylu ograniczenia profilu jak na Facebook)

## End 2 End communication

W zależności od postępu prac zamierzamy dodać e2e communication w czacie, problemem jest brak signalProtocol w c#, brak gotowej implementacji zmusza nas do stworzenia samemu jakiejś hybrydy (co okaże się ciężkim rozwiązaniem). Docolewo planujemy przechowywać klucz w localstorage, do deszyfrowania, w razie zapomnienia hasła - recovery key. Aplikacja jest w stanie przechowywać klucz, natomiast przeglądarka ma to utrudnione, istnieją poważne ryzyka wykradnięcia itp. Do przedyskutowania