

Serwis Społecznościowy – Dokument Projektowy

Opis Projektu

Serwis społecznościowy, umożliwiający:

- Rejestrację / Logowanie
- Upload zdjęć
- Dodawanie użytkowników do znajomych
- Dodawanie postów
- Ocenę postów
- Tworzenie grup użytkowników
- Dodawanie komentarzy do postów
- Czat z innymi użytkownikami

Motywacja

Serwis rozwiązuje problem komunikacji ze znajomymi, tworzenia grup o określonym przeznaczeniu oraz wymianie informacji oraz treści pomiędzy użytkownikami.

Technologie

W projekcie wykorzystywane będą następujące technologie:

- ASP.NET Core (C#) – backend
- Angular (TypeScript) – frontend
- MySQL - baza danych

Przedział czasowy

Przedział czasowy wymagany do realizacji projektu to 16/10/2021 – [wstawić czas końcowy]

Role członków zespołu

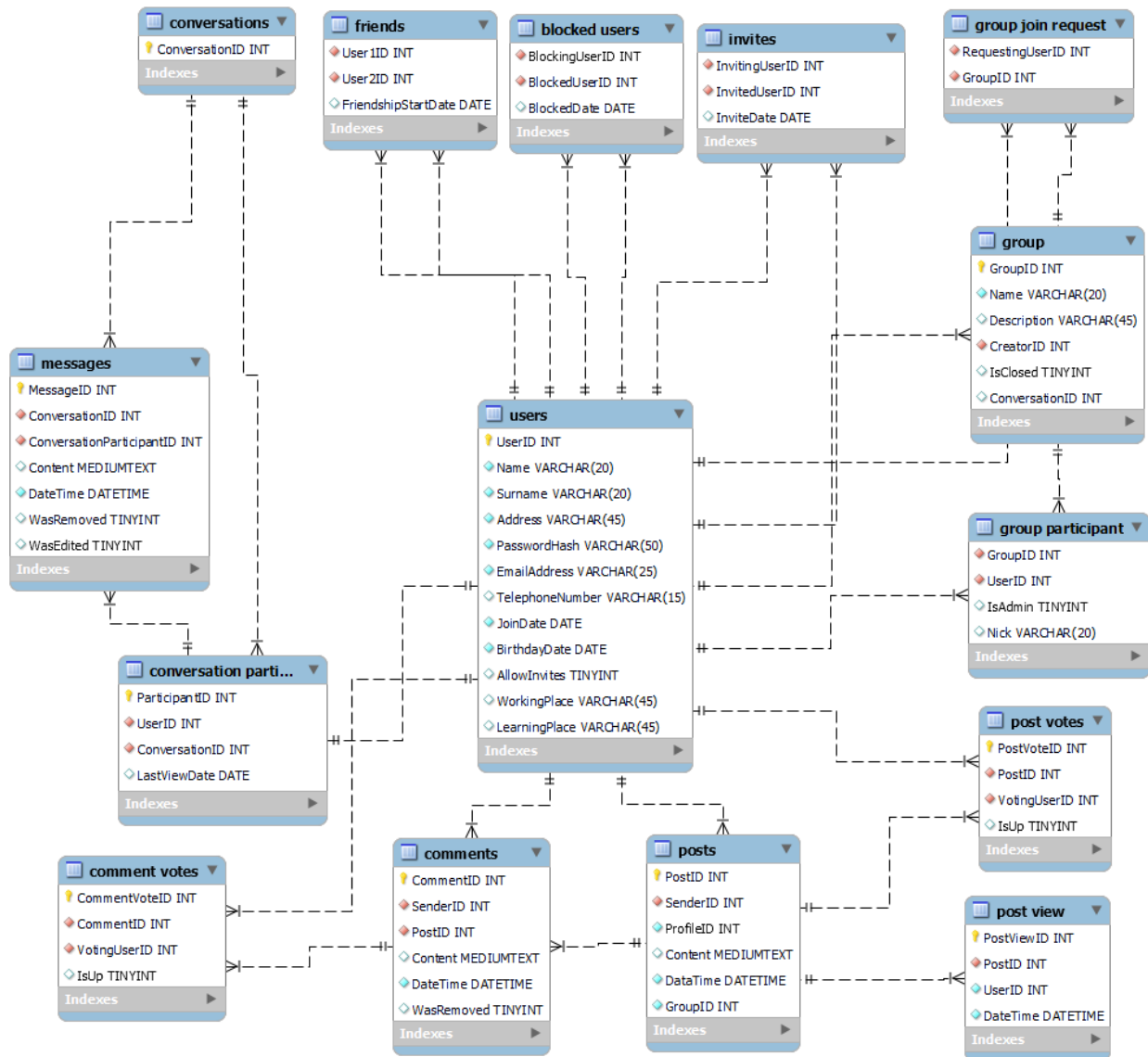
- Tomasz Celiński – Lider projektu, backend developer / tester
- Damian Tomczyszyn – backend developer / tester
- Artur Żelazko – frontend developer / tester
- Dima Nimchynskyi – frontend developer / tester

Milestoney, Epiki, Historyjki, Taski

- Milestone 1
 - Wybór technologii
 - Wybór stosu technologicznego
 - Wybór programów do tworzenia diagramów / schematów
 - Sprawy organizacyjne
 - Stworzenie szkieletu dokumentu projektowego
 - Określenie środowiska deweloperskiego
 - Utworzenie repozytorium
 - Wybór tematyki projektu
- Milestone 2
 - Przygotować opisy / wizualizacje routów
 - Opisać routy
 - Stworzyć mockupy
 - Przygotowanie schematów / diagramów cz1
 - Wstępny schemat bazy danych
 - Diagramy stanów
 - Zaktualizować dokument projektowy
 - Diagram klas – frontend cz 1
 - Analiza najczęściej wykonywanych zapytań do bazy danych
 - Diagram klas – backend
 - Zaktualizować schemat bazy danych
 - Przygotowanie schematów / diagramów cz2
 - Diagram klas – frontend cz 2
 - Normalizacja schematu bazy danych pod kątem najczęściej wykonywanych zapytań
 - Diagram sekwencji – frontend
 - Diagram sekwencji – backend
- Milestone 3
 - Przygotować scenariusze testowe

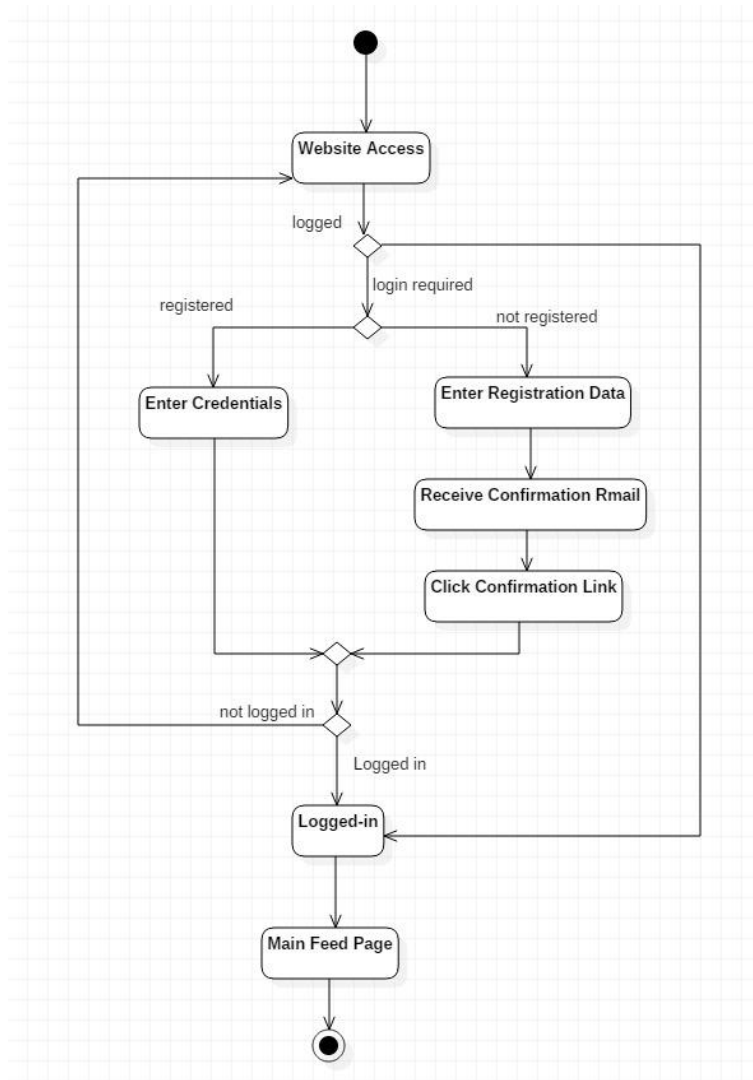
Dokumentacja Techniczna

Schemat bazy danych:

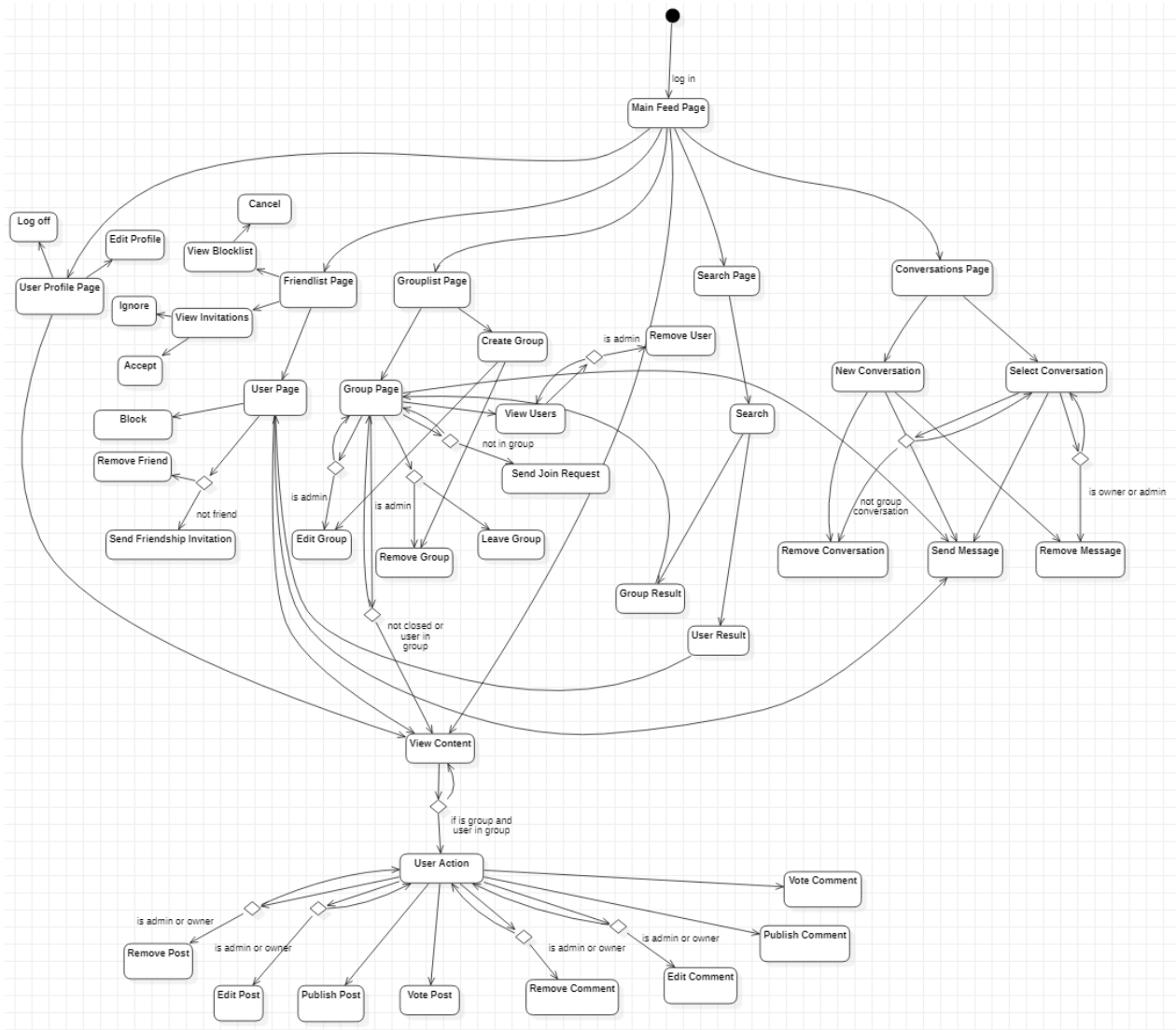


Diagramy stanów:

Logowanie:

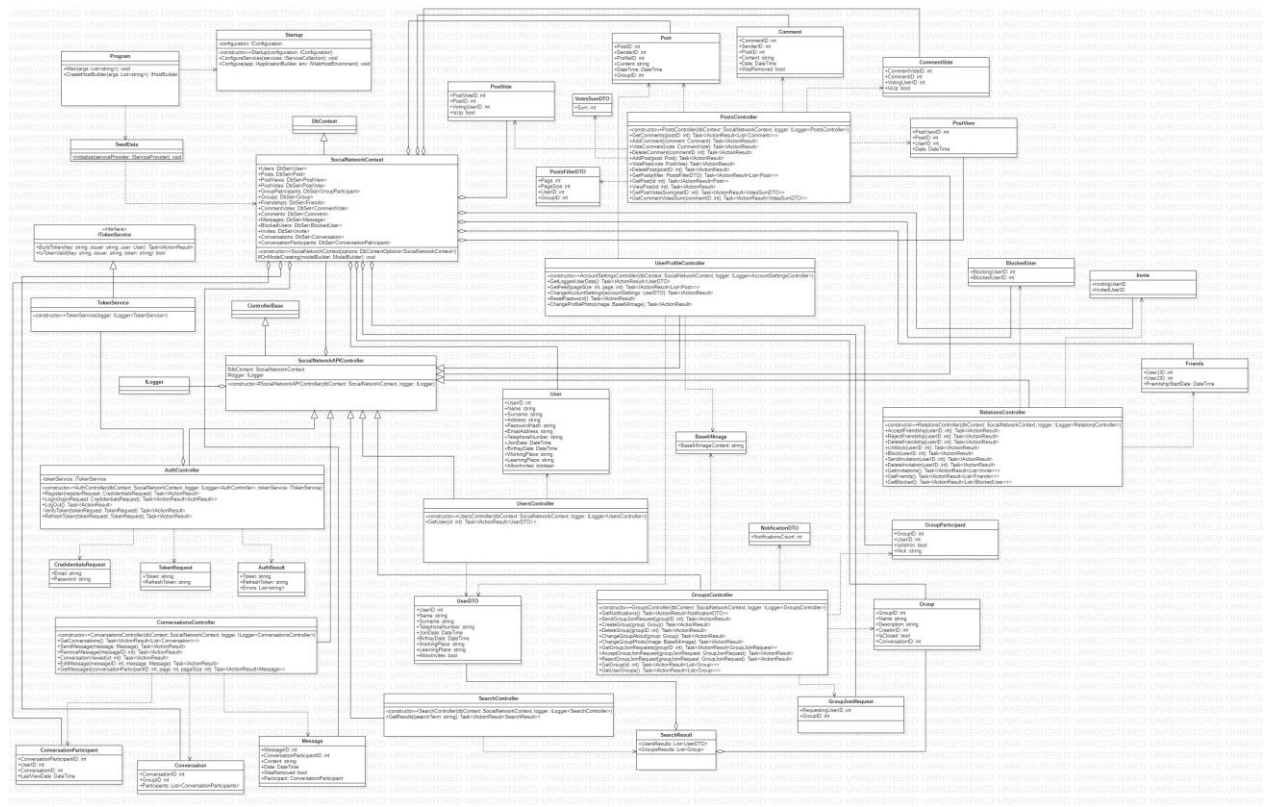


Main feed:



Diagramy klas

Backend:



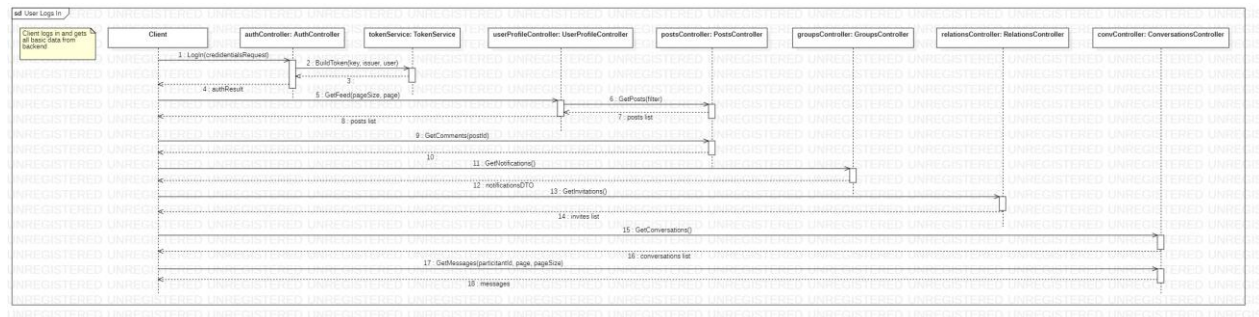
```

classDiagram
    class MainFeedToggleBar {
        +constructor+MainFeedToggleBar(APIService)
        +onUserProfileClick() void
        +onCreatePostClick() void
    }
    class Post {
        +constructor+Post(APIService)
        +onShareClick() void
        +onCommentSectionClick() void
        +onPostSettingsClick() void
    }
    class UserFeedToggleBar {
        +constructor+UserFeedToggleBar(APIService)
        +onSendMessageClick() void
        +onSendMessageRequestClick() void
        +onProfileClick() void
        +onLogoutClick() void
        +onBlockUserClick() void
    }
    class ProfileAboutSection {
        +constructor+ProfileAboutSection(APIService)
        +onEditClick() void
        +onSendMessageClick() void
        +onWorkspaceClick() void
    }
    class GroupToggleBar {
        +constructor+GroupToggleBar(APIService)
        +onJoinGroupClick() void
        +onLeaveGroupClick() void
        +onJoinRequestClick() void
    }
    class ChatList {
        +constructor+ChatList(APIService)
        +onChatClick() void
        +onEditClick() void
        +onProfileClick() void
        +onGroupClick() void
    }
    class UserActionsButtonsPanel {
        +onMainFeedClick() void
        +onUserProfileClick() void
        +onSendMessageClick() void
        +onSearchClick() void
        +onFriendsListClick() void
        +onCommunitiesClick() void
        +onSettingsClick() void
    }
    class LoginView {
        +constructor+LoginView(APIService)
        +onEmailAndPasswordClick() void
        +onForgotPasswordClick() void
    }
    class SearchView {
        +constructor+SearchView(APIService)
        +onSearchClick() void
        +onFilterClick() void
        +onSortClick() void
        +onRefreshClick() void
    }
    class LocalStorageService {
        +storage+string[]
        +getKey+string, value+string void
        +setItem+string, value+string void
        +removeKey+string void
    }
    class MainFeedToggleBar <|-- UserFeedToggleBar
    class MainFeedToggleBar <|-- ProfileAboutSection
    class MainFeedToggleBar <|-- GroupToggleBar
    class MainFeedToggleBar <|-- ChatList
    class MainFeedToggleBar <|-- UserActionsButtonsPanel
    class MainFeedToggleBar <|-- LoginView
    class MainFeedToggleBar <|-- SearchView
    class Post <|-- ProfileAboutSection
    class Post <|-- GroupToggleBar
    class Post <|-- ChatList
    class Post <|-- UserActionsButtonsPanel
    class Post <|-- LoginView
    class Post <|-- SearchView
    class UserFeedToggleBar <|-- ProfileAboutSection
    class UserFeedToggleBar <|-- GroupToggleBar
    class UserFeedToggleBar <|-- ChatList
    class UserFeedToggleBar <|-- UserActionsButtonsPanel
    class UserFeedToggleBar <|-- LoginView
    class UserFeedToggleBar <|-- SearchView
    class ProfileAboutSection <|-- GroupToggleBar
    class ProfileAboutSection <|-- ChatList
    class ProfileAboutSection <|-- UserActionsButtonsPanel
    class ProfileAboutSection <|-- LoginView
    class ProfileAboutSection <|-- SearchView
    class GroupToggleBar <|-- ChatList
    class GroupToggleBar <|-- UserActionsButtonsPanel
    class GroupToggleBar <|-- LoginView
    class GroupToggleBar <|-- SearchView
    class ChatList <|-- UserActionsButtonsPanel
    class ChatList <|-- LoginView
    class ChatList <|-- SearchView
    class UserActionsButtonsPanel <|-- LoginView
    class UserActionsButtonsPanel <|-- SearchView
    class LoginView <|-- SearchView
    class SearchView <|-- LocalStorageService
  
```

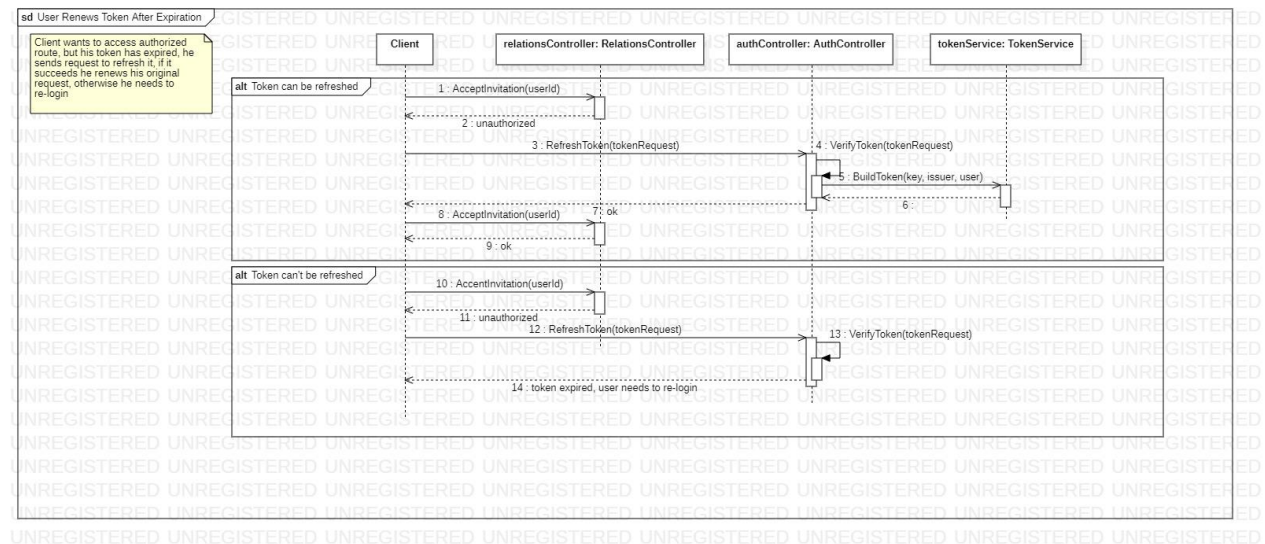
The diagram illustrates the use cases and their relationships for a social media application. The main components are:

- MainFeedToggleBar**: Contains use cases for MainFeedToggleBar (APIService), UserFeedToggleBar (APIService), ProfileAboutSection (APIService), GroupToggleBar (APIService), ChatList (APIService), UserActionsButtonsPanel, LoginView, and SearchView.
- Post**: Contains use cases for Post (APIService), ProfileAboutSection (APIService), GroupToggleBar (APIService), ChatList (APIService), UserActionsButtonsPanel, LoginView, and SearchView.
- UserFeedToggleBar**: Contains use cases for UserFeedToggleBar (APIService), ProfileAboutSection (APIService), GroupToggleBar (APIService), ChatList (APIService), UserActionsButtonsPanel, LoginView, and SearchView.
- ProfileAboutSection**: Contains use cases for ProfileAboutSection (APIService), GroupToggleBar (APIService), ChatList (APIService), UserActionsButtonsPanel, LoginView, and SearchView.
- GroupToggleBar**: Contains use cases for GroupToggleBar (APIService), ChatList (APIService), UserActionsButtonsPanel, LoginView, and SearchView.
- ChatList**: Contains use cases for ChatList (APIService), UserActionsButtonsPanel, LoginView, and SearchView.
- UserActionsButtonsPanel**: Contains use cases for UserActionsButtonsPanel, LoginView, and SearchView.
- LoginView**: Contains use cases for LoginView, SearchView, and LocalStorageService.
- SearchView**: Contains use cases for SearchView and LocalStorageService.
- LocalStorageService**: Contains use cases for LocalStorageService.

Logowanie i pobranie danych (backend)



Odnowienie tokenu (sukces oraz niepowodzenie)



Infrastruktura sprzętowa / zasoby wykorzystywane w projekcie

Opis Środowisk

- Dev - Niestabilne środowisko do wprowadzania zmian mogących zagrozić stabilności systemu
- Test – Środowisko testerskie do przeprowadzania testów na funkcjonalnościach potwierdzonych w fazie dev
- Prod – Środowisko stabilne, na którym działa klient