SOCIAL MEDIA POST MANAGER

DOKUMENTACJA

1 Tytuł projektu

Nazwa aplikacji: Social Media Post Manager

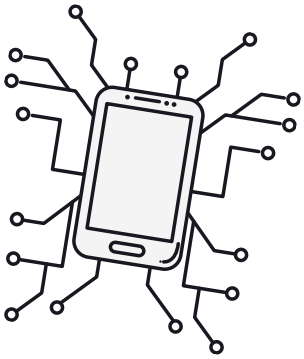
Krótki opis: Aplikacja konsolowa do zarządzania postami w mediach społecznościowych z wykorzystaniem systemu RBAC (Role-Based Access Control).

2 Opis projektu

Cel projektu: Stworzenie aplikacji konsolowej, która symuluje działanie programu do zarządzania treściami na trzech różnych platformach społecznościowych tj. Facebook, Instagram oraz Reddit. Ze względu na fakt, iż program opiera się na systemie RBAC, użytkownik programu w zależności od posiadanej roli będzie miał dostęp do różnych funkcjonalności.
Aplikacja umożliwia zarządzanie postami na platformach społecznościowych (Facebook, Instagram, Reddit) za pomocą konsolowego interfejsu. Dzięki zastosowaniu RBAC, użytkownicy mają różne poziomy dostępu – od przeglądania po pełną administrację treścią. Użytkownik może tworzyć, edytować, usuwać i przeglądać posty w zależności od przypisanej roli.

Grupa docelowa: osoby korzystające z mediów społecznościowych, a także administratorzy takich serwisów.

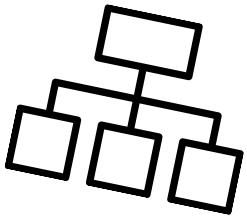
3 Technologie



Język programowania: C#
Środowisko: .NET 8
IDE: Visual Studio 2022
Dodatkowe biblioteki użyte w programie:

- System.IO
- System.Security.Cryptography
- System.Text

4 Struktura katalogów



```
C:.
├── Dokumentacja_Projektu_CSharp.pdf
├── krol_bartosz_kwaskiewicz_nataniel.sln
├── README.md
├── struktura.txt
└── krol_bartosz_kwaskiewicz_nataniel
    ├── krol_bartosz_kwaskiewicz_nataniel.csproj
    ├── krol_bartosz_kwaskiewicz_nataniel.sln
    ├── Program.cs
    ├── bin
    │   └── Debug
    │       └── net8.0
    │           ├── admin_logs.txt
    │           ├── app_logs.txt
    │           ├── facebook.txt
    │           ├── instagram.txt
    │           ├── krol_bartosz_kwaskiewicz_nataniel.deps.json
    │           ├── krol_bartosz_kwaskiewicz_nataniel.dll
    │           ├── krol_bartosz_kwaskiewicz_nataniel.exe
    │           ├── krol_bartosz_kwaskiewicz_nataniel.pdb
    │           ├── krol_bartosz_kwaskiewicz_nataniel.runtimeconfig.json
    │           ├── reddit.txt
    │           └── users.txt
    ├── classes
    │   ├── ConsoleHelper.cs
    │   ├── IUserActions.cs
    │   ├── Logger.cs
    │   ├── SocialMediaPostManager.cs
    │   ├── SystemManager.cs
    │   └── User.cs
    └── obj
        ├── krol_bartosz_kwaskiewicz_nataniel.csproj.nuget.dgspec.json
        ├── krol_bartosz_kwaskiewicz_nataniel.csproj.nuget.g.props
        ├── krol_bartosz_kwaskiewicz_nataniel.csproj.nuget.g.targets
        ├── project.assets.json
        ├── project.nuget.cache
        └── Debug
            └── net8.0
                ├── .NETCoreApp,Version=v8.0.AssemblyAttributes.cs
                ├── apphost.exe
                ├── krol_bartosz_kwaskiewicz_nataniel.AssemblyInfo.cs
                ├── krol_bartosz_kwaskiewicz_nataniel.AssemblyInfoInputs.cache
                ├── krol_bartosz_kwaskiewicz_nataniel.assets.cache
                ├── krol_bartosz_kwaskiewicz_nataniel.csproj.BuildWithSkipAnalyzers
                ├── krol_bartosz_kwaskiewicz_nataniel.csproj.CoreCompileInputs.cache
                ├── krol_bartosz_kwaskiewicz_nataniel.csproj.FileListAbsolute.txt
                ├── krol_bartosz_kwaskiewicz_nataniel.dll
                ├── krol_bartosz_kwaskiewicz_nataniel.GeneratedMSBuildEditorConfig.editorconfig
                ├── krol_bartosz_kwaskiewicz_nataniel.genruntimeconfig.cache
                ├── krol_bartosz_kwaskiewicz_nataniel.GlobalUsings.g.cs
                ├── krol_bartosz_kwaskiewicz_nataniel.pdb
                └── ref
                    └── krol_bartosz_kwaskiewicz_nataniel.dll
```

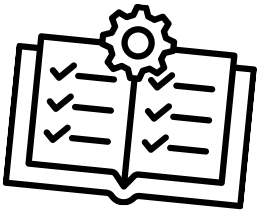
Pliki facebook.txt, instagram.txt oraz reddit.txt zawierają posty opublikowane przez użytkowników w następującym formacie:
Nazwa użytkownika;[dd.mm.rrrr hh:mm:ss];treść posta

Przykładowa linijka:
admin;[18.04.2025 16:40:46];test

Plik users.txt zawiera dane o użytkownikach w następującym formacie:
Nazwa użytkownika;Hasło(zahashowane);Rola

Przykładowa linijka:
Tomek;36H2MozQ+XgE63R6TT6Cycoq013Vf76wXzVXftVdGcs=;user

5 Instrukcja instalacji i uruchomienia



Wymagania systemowe:
- Windows 10/11
- .NET SDK 8.0 lub wyższy

Sposoby na uruchomienie aplikacji:

1. **Używając pliku aplikacji przez eksplorator plików (pliku .exe)**
   a. Pobierz i rozpakuj folder krol_bartosz_kwaskiewicz_nataniel.zip
   b. Udaj się do folderu głównego projektu o nazwie krol_bartosz_ kwaskiewicz_nataniel
   c. Odnajdź plik krol_bartosz_ kwaskiewicz_nataniel.exe
   d. Uruchom go klikając na niego dwa razy
2. **Używając wiersza poleceń**
   a. Pobierz i rozpakuj folder krol_bartosz_kwaskiewicz_nataniel.zip
   b. Uruchom wiersz poleceń
   c. W wierszu poleceń za pomocą komendy cd przenieś się do folderu krol_bartosz_ kwaskiewicz_nataniel\bin\Debug\net8.0
   d. Wpisz nazwę pliku .exe tj. krol_bartosz_ kwaskiewicz_nataniel.exe
3. **Używając IDE (np. Visual Studio)**
   a. Pobierz i rozpakuj folder krol_bartosz_kwaskiewicz_nataniel.zip
   b. Udaj się do folderu głównego projektu o nazwie krol_bartosz_ kwaskiewicz_nataniel
   c. Uruchom plik z rozszerzeniem .sln za pomocą IDE
   d. W IDE skompiluj program (w przypadku Visual Studio przycisk F5)

Uruchomienie przez Git/GitHub:
git clone https://github.com/Komputerr/ProjektPO.git
cd ProjektPO\krol_bartosz_kwaskiewicz_nataniel
dotnet build
dotnet run

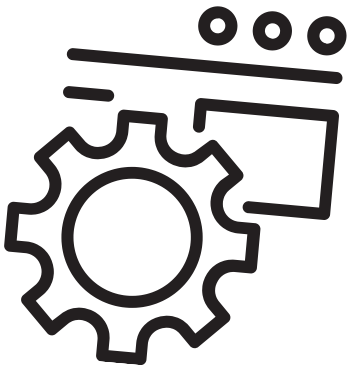
6 Opis działania aplikacji

Logowanie: użytkownik wybiera logowanie, rejestrację lub dostęp jako gość.

RBAC:
- admin – dodawanie, edytowanie, przeglądanie oraz usuwanie wszystkich postów
- user – dodawanie, edytowanie, przeglądanie oraz usuwanie własnych postów
- guest – przeglądanie postów

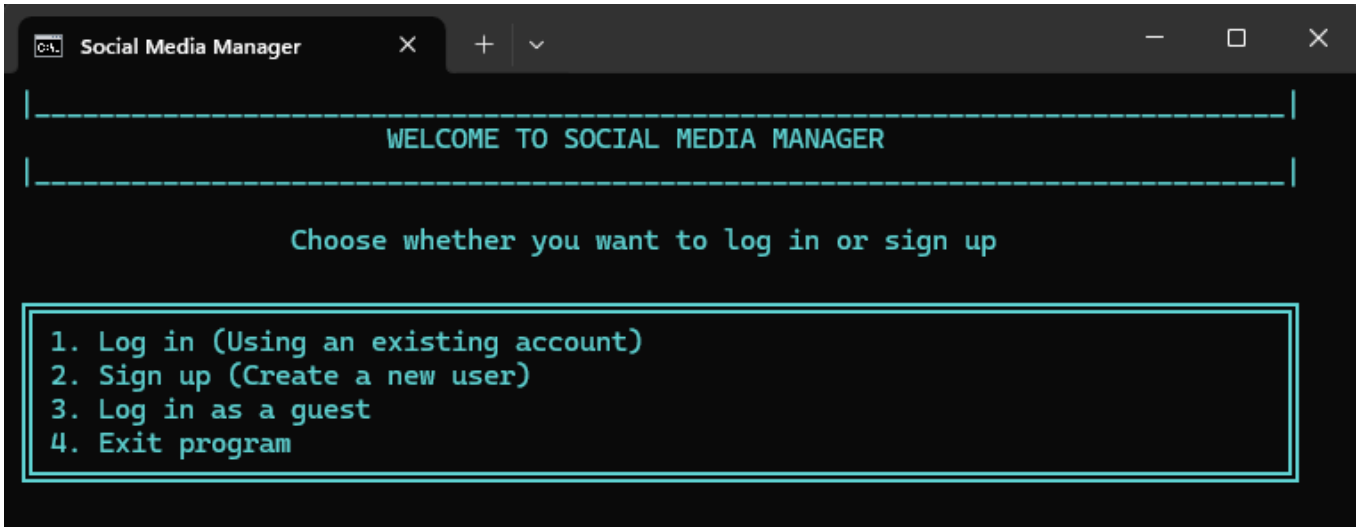
Funkcje aplikacji:

1 - dostępne tylko dla roli admin;
2 - dostępne dla roli admin oraz użytkownik;
3 - dostępne dla roli admin, użytkownik oraz gość;

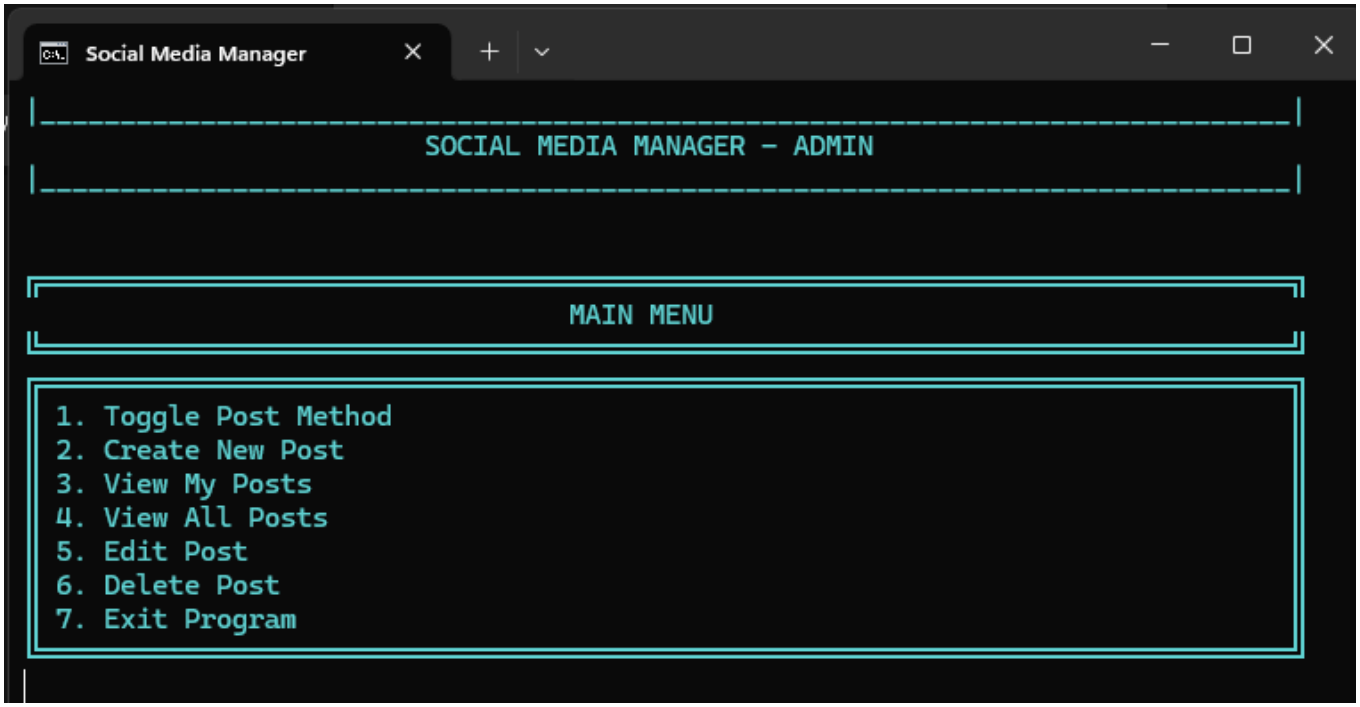


- Konfigurowanie metody publikowania wpisów (2)
- Publikowanie wpisów (2)
- Edycja wpisów (2)
- Usuwanie wpisów na danej platformie
  - Usuwanie wszystkich własnych postów (2)
  - Usuwanie jednego z własnych postów (2)
  - Usuwanie wszystkich postów (1)
  - Usuwanie jednego z wszystkich postów (1)
- Przeglądanie wszystkich postów z danej platformy (3)
- Przeglądanie własnych postów z danej platformy (2)
- Przeglądanie logów aplikacji (1)

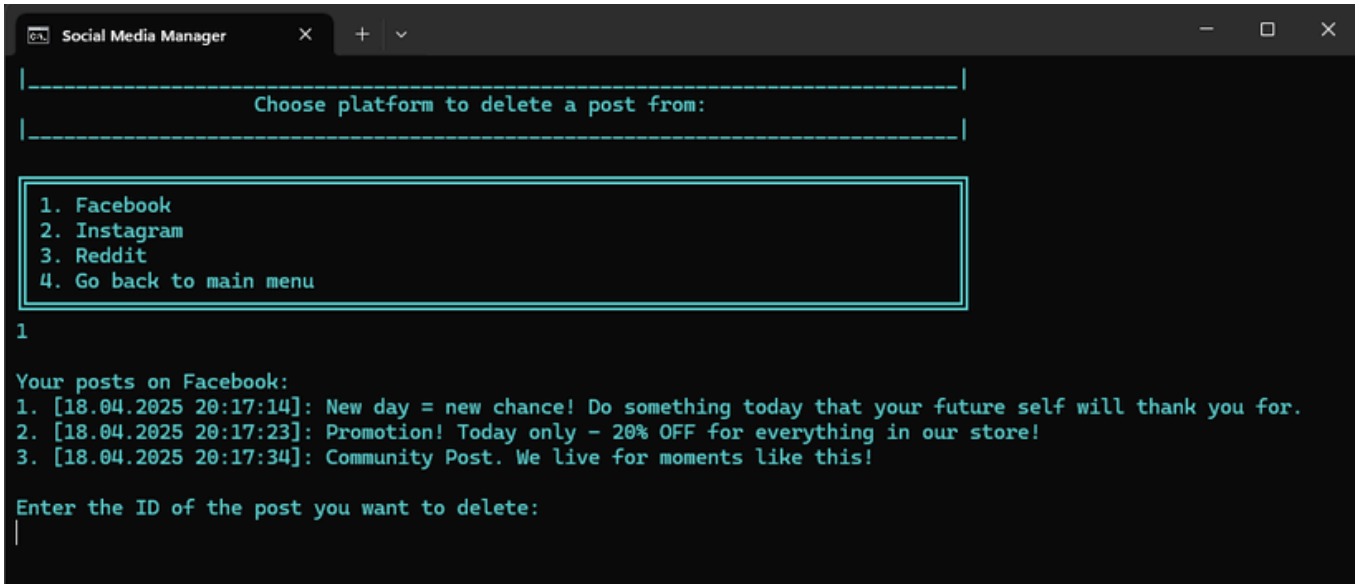
7 Zrzuty ekranu



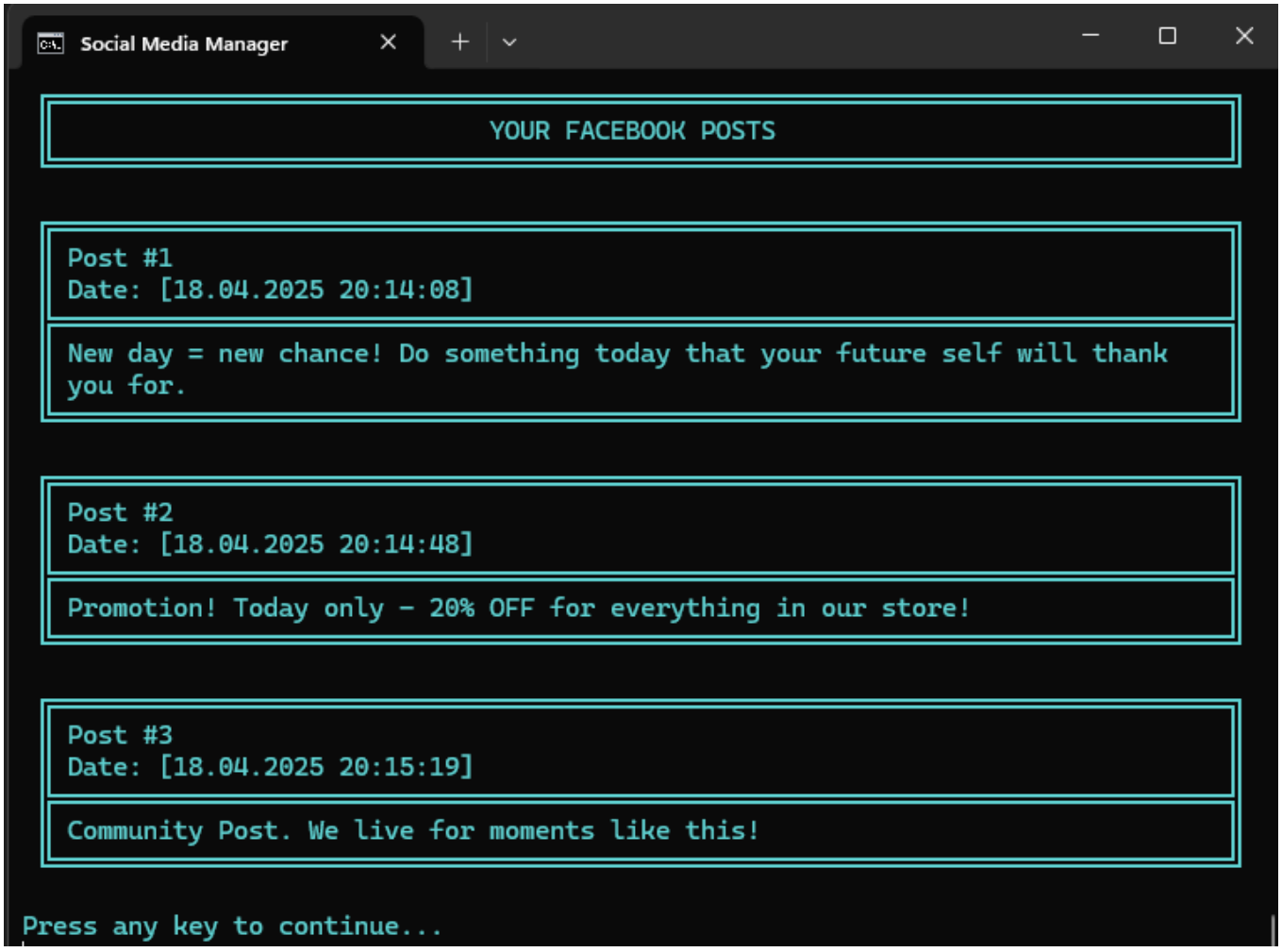
Zrzut 1: Widok w trakcie wejścia do aplikacji.



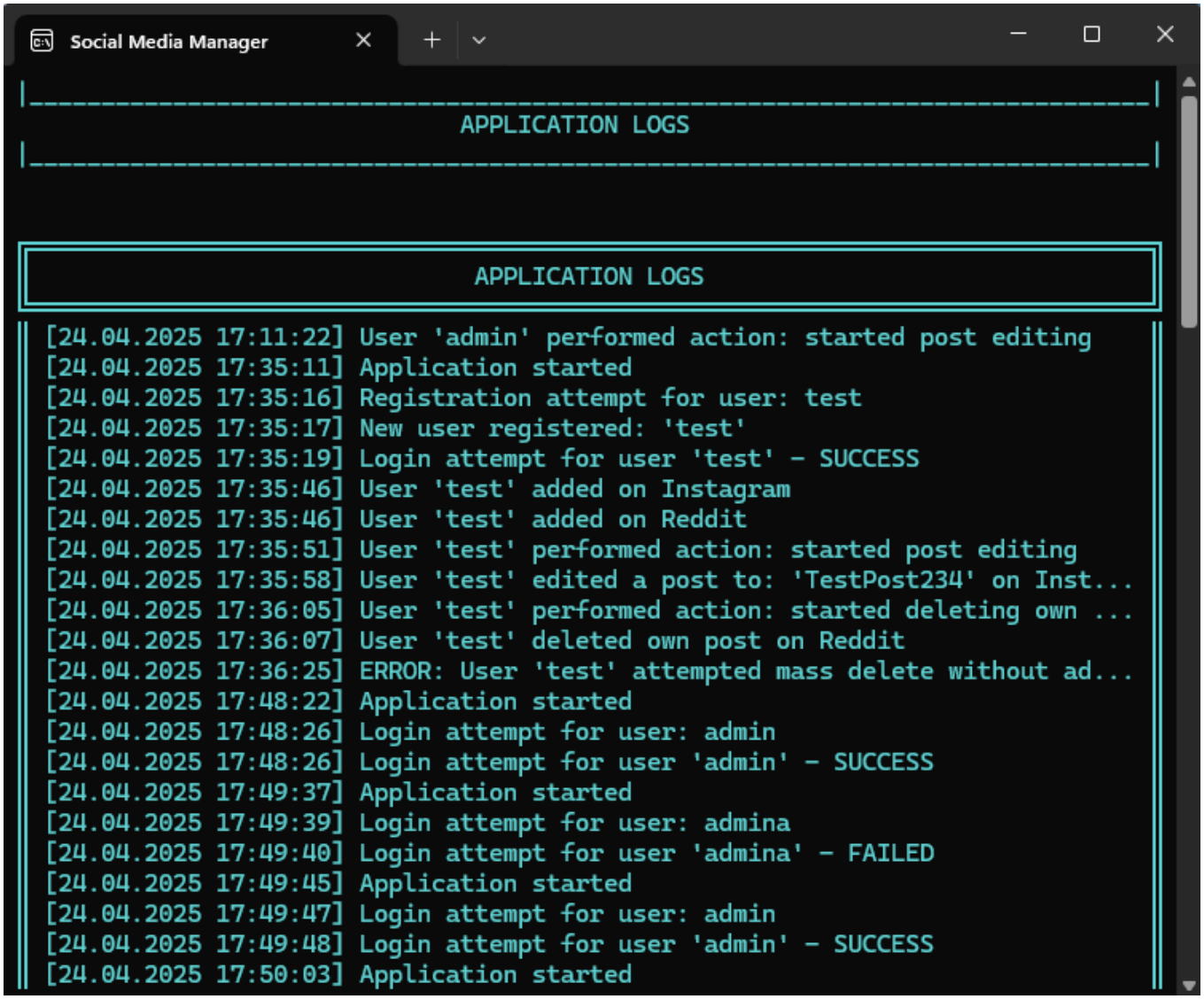
Zrzut 2: Widok głównego menu po zalogowaniu.



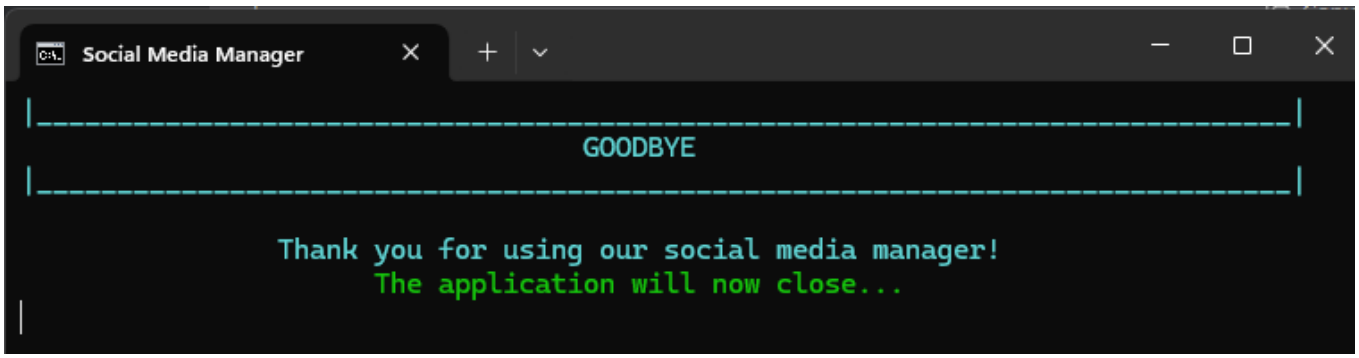
Zrzut 3: Widok w trakcie usuwania postów.



Zrzut 4: Widok w trakcie przeglądania postów.



Zrzut 5: Widok w trakcie przeglądania logów.

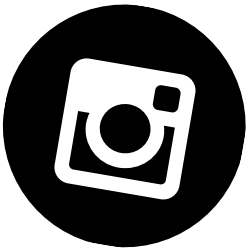


Zrzut 6: Widok w trakcie wychodzenia z aplikacji.

8 Przykłady użycia

1. Standardowy użytkownik (np. influencer)

1. Uruchom aplikację.
2. Wybierz opcję 2. Sign up.
3. Wprowadź dane (podane dane są przykładowe):
    a. Login: anna_bloger
    b. Hasło: SecurePass123!
4. Zaloguj się na nowe konto.
5. W menu głównym wybierz 1. Toggle Post Method → 1. Add → 1. Facebook, 2. Instagram.
6. Wybierz 2. Create New Post i dodaj treść:
    a. "Mój pierwszy post! #witam"
7. Sprawdź posty: 4. View All Posts → 1. Facebook.



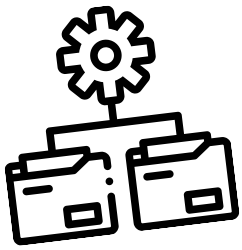
2. Administrator (np. moderator społeczności)

1. Uruchom aplikację.
2. Wybierz opcję 1. Log in.
3. Wprowadź dane:
    - Login: admin
    - Hasło: admin
4. Przejrzyj posty: 4. View All Posts → 3. Reddit.
5. Usuń post: 6. Delete Post → 4. Delete one of all posts (admin).
6. Sprawdź logi: 7. View Application Logs.

3. Gość (np. nowy użytkownik testujący)

1. Uruchom aplikację.
2. Wybierz opcję 3. Log in as guest.
3. Przejrzyj posty: 1. View All Posts → 2. Instagram.

9 Struktura danych i klasy



Klasy:

- User – przechowuje dane użytkownika: login, hasło (SHA256), rola
- SystemManager – funkcje odpowiedzialne za rejestrację/logowanie użytkowników oraz zapisywanie tych danych do pliku users.txt
- SocialMediaPostManager - wszystkie operacje na postach, interakcja z plikami facebook.txt, instagram.txt oraz reddit.txt
- Program - logika aplikacji
- Logger – zapisywanie logów aplikacji do pliku oraz ich odczyt
- ConsoleHelper - metody odpowiadające za szatę graficzną aplikacji

Interface:

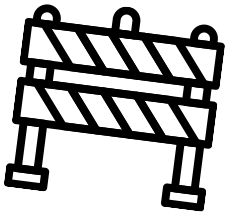
- IUserActions – definiuje metody używane w klasie SocialMediaPostManager

10 Testowanie

Aplikacja została przetestowana manualnie przez deweloperów. Każda funkcjonalność została dokładnie zbadana pod kątem występowania wszelkich błędów oraz niesprawności.

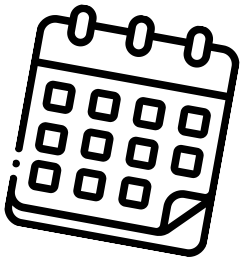
Brak testów jednostkowych (możliwość przeprowadzenia takowych np. przy użyciu xUnit w przyszłości)

11 Problemy i ograniczenia



- Dane są przechowywane w plikach tekstowych (brak bazy danych)
- RBAC działa, ale można rozszerzyć system o więcej ról/uprawnień
- Aplikacja konsolowa ogranicza możliwości jeżeli chodzi o tworzenie interfejsu graficznego - lepiej byłoby stworzyć taką używając aplikacji Windows Forms
- Brak praktycznych umiejętności prowadzenia testów np. jednostkowych

12 Plany rozwoju



- Zastosowanie bazy danych (np. SQLite)
- Interfejs graficzny (np. WPF) lub przeniesienie się na aplikację typu Windows Forms
- Przeprowadzenie odpowiednich testów jednostkowych
- Możliwość dodania obrazów do postów (szczególnie w przypadku Instagrama)
- Dodanie interakcji z postami w postaci komentarzy, polubień, tzw. up oraz down vote'ów itp.
- Zarządzanie komentarzami tj. edycja, usuwanie

13 Autorzy

Imiona i nazwiska autorów: Bartosz Król, Nataniel Kwaśkiewicz

Email:

- bartosz.krol@uczen.zsk.poznan.pl
- nataniel.kwaskiewicz@uczen.zsk.poznan.pl