

# Sprawozdanie z projektu:

## Programowanie Komputerów

### Chat społecznościowy



**Politechnika  
Śląska**

Kamil Nocoń gr.1 INF AEI

26.05.2023

# 1. Konsumencki opis projektu

---

Główne założenia aplikacji:

- Projekt służy do obsługi serwisu internetowego Alterra.
- Menu logowania wraz ze sprawdzaniem danych.
- Obsługa i zapis chatu grupowego.
- Dodatkowe funkcjonalności zależne od typu konta.

Po uruchomieniu programu, aplikacja wita użytkownika w serwisie oraz uruchamia interfejs logowania. Prosi o podanie loginu użytkownika (nickname) oraz hasła, dokonuje walidacji (sprawdzenia poprawności danych) i w przypadku znalezienia zgodnych danych, uruchamiane jest menu (zależne od typu konta). (rys. 1.1)

W przypadku podania nieprawidłowych danych, użytkownik zostaje o tym powiadomiony, a aplikacja umożliwia ponowne podanie danych wejściowych. (rys. 1.2)

```
Witaj w serwisie internetowym Alterra!  
-----<>-----  
ZALOGUJ SIE:  
Podaj login do swojego konta:  
Melania21  
Podaj haslo do swojego konta:  
!mela1234
```

Rysunek 1.1

```
Witaj w serwisie internetowym Alterra!  
-----<>-----  
ZALOGUJ SIE:  
Podaj login do swojego konta:  
Melania21  
Podaj haslo do swojego konta:  
zleHaslo  
Podane dane sa nieprawidlowe!  
Podaj login do swojego konta:
```

Rysunek 1.2

Typy kont użytkowników:

- Użytkownicy podstawowi;
- Moderatorzy;
- Administratorzy;

Każde konto w serwisie ma przypisany (baza danych) jeden z trzech dostępnych typów kont. Typ konta determinuje wyświetlone dla użytkownika menu, dostępne po zalogowaniu. Każdy z typów użytkowników posiada różne opcje – funkcjonalności w programie.

a. Użytkownicy podstawowi:

Po zalogowaniu użytkownikowi wyświetla się menu z 4 możliwymi do podjęcia akcjami (rys. 2)

```
UDALO SIE ZALOGOWAC! Witaj Melania21
-----Panel sterowania-----
Jaka operacje chcesz wykonac:
-----<>-----
1. Wejdz w chat.
2. Pokaz liste uzytkownikow chatu.
3. Pokaz informacje o profilu.
4. Wyjdz z aplikacji.
```

Rysunek 2

Funkcjonalność nr 1 – Wejść w chat:

Główne zadanie i funkcja programu, dostępna dla każdego konta, wyświetla (z archiwum) dotychczasową konwersację grupową oraz umożliwia wysłanie nowej wiadomości z aktualnie zalogowanego konta. Pozostałe konta po zalogowaniu się będą mogły odczytać nowe wiadomości i również na nie odpowiedzieć. (rys. 3)

```
// ABY WYJSC Z KONWERSACJI NAPISZ /esc
< Azymut > Hejka wszystkim!
< Cieciel > Siemaneczko
< Bielan > No heyy, to działa;
< Bielan > siemkaaa
< Azymut >
Zaczynamy Zabawe za 30min
< Azymut > Zaczynamy Zabawe za 30min
Pozostalo 15min
```

Rysunek 3

Budowa pojedynczej wiadomości:

< nickname > Treść wiadomości; (rys. 4 – przykładowa wiadomość)

< Azymut > Przykładowa Wiadomosc!

Rysunek 4

Nickname jest podświetlony na różny kolor w zależności od typu konta użytkownika.

Legenda kolorów:

Zielony – użytkownicy podstawowi np. < Melania21 >

Niebieski (cyjan) – moderatorzy np. < Bielan >

Czerwony – Administratorzy np. < Cieciel >

W celu wyjścia z chatu należy wpisać komendę „/esc” – po jej nadaniu, stan chatu jest zapisywany, a użytkownik wraca do swojego menu.

Funkcjonalność nr 2 – Pokaż listę użytkowników chatu:

Opcja umożliwiająca podejrzenie członków (nickname, imię, nazwisko, typ konta) chatu grupowego, w którym aktualnie się znajdujemy. (rys. 5)

```
-----Panel sterowania-----
Jaka operacje chcesz wykonac:
-----<>-----
1. Wejdz w chat.
2. Pokaz liste uzytkownikow chatu.
3. Pokaz informacje o profilu.
4. Wyjdz z aplikacji.
2
Uzytkownicy chatu grupowego:
1. Melania21 Angelika Wariał podstawowy
2. Kony Konrad Ostrowski podstawowy
3. Bielan Wiktoria Bielanik moderator
4. skowronek2115 Janek Skowron podstawowy
5. Cieciel Marcel Swierczek administrator
6. lucyjax Lucja Szczepak podstawowy
7. bjngf Marcin Ciok moderator
8. Kinimod Percy Duda podstawowy
9. Gimpson Daniel Działowy podstawowy
10. Azymut Kamil Noc administrator
11. admin admin admin administrator
```

Rysunek 5

Funkcjonalność nr 3 – Pokaż informację o profilu:

Opcja umożliwiająca podejrzenie informacji jawnych aktualnie zalogowanego użytkownika. (rys. 6– przykład dla konta lucyjax)

```
UDALO SIE ZALOGOWAC! Witaj lucyjax
-----Panel sterowania-----
Jaka operacje chcesz wykonac:
-----<>-----
1. Wejdz w chat.
2. Pokaz liste uzytkownikow chatu.
3. Pokaz informacje o profilu.
4. Wyjdz z aplikacji.
3
Imie: Lucja
Nazwisko: Szczepak
Login: lucyjax
Miasto: Konin
Numer telefonu: 876238450
Email: konieczyna1303@poczta.pl
Rodzaj konta: podstawowy
Numer konta: 6
Status konta (1-ban): 0
```

Rysunek 6

Funkcjonalność nr.4 – Wyjdz z aplikacji:

Ta opcja zapisuje wszystkie wysłane wiadomości i zamyka program.

b. Moderatorzy:

Po zalogowaniu użytkownikowi wyświetla się menu z 5 możliwymi do podjęcia akcjami (rys. 7)

```
UDALO SIE ZALOGOWAC! Witaj Bielan
-----Panel sterowania-----
Jaka operacje chcesz wykonac:
-----<>-----
1. Wejdz w chat.
2. Pokaz liste uzytkownikow chatu.
3. Pokaz informacje o profilu.
4. Zbanuj uzytkownika.
5. Wyjdz z aplikacji.
```

Rysunek 7

Funkcjonalności 1, 2, 3 pokrywają się z podstawowymi użytkownikami.

Funkcjonalność nr.4 – Zbanuj użytkownika:

Opcja umożliwiająca zmianą statusu konta dowolnego użytkownika (nie dotyczy administratorów – nie można zbanować konta administratora). Domyślnie każde konto jest odbanowane i może swobodnie zalogować się do aplikacji. Jednak gdy moderator/administrator zbanują jego konto to uniemożliwia mu zalogowanie się do aplikacji i branie udziału w konwersacji. Aby zbanować konto należy podać Nickname użytkownika oraz potwierdzić chęć nałożenia ograniczenia (rys. 8).

```
4. Zbanuj uzytkownika.
5. Dodaj nowego uzytkownika.
6. Wyjdz z aplikacji.
4
Zmiana statusu konta!
-----<>-----
Podaj nick uzytkownika, ktorego status konta chcesz zmienic:
Melania21
Czy na pewno chcesz zbanowac uzytkownika Melania21? (T/N)
T
Uzytkownik Melania21 zostal zbanowany z aplikacji!
```

Rysunek 8

Użytkownik przy próbie zalogowania się do swojego konta, zostanie poinformowany o braku takiej możliwości, wynikającej z nałożonego bana. (rys. 9)

```
KONTO Melania21 JEST ZBANOWANE!
Podane dane sa nieprawidlowe!
Podaj login do swojego konta:
```

Rysunek 9

Każdy administrator/moderator ma również prawo odbanować, znieść karę z dowolnego konta. W tym celu należy wybrać funkcjonalność 4, podać nazwę zbanowanego użytkownika i potwierdzić chęć jego odbanowania. (rys. 10)

```
4. Zbanuj uzytkownika.
5. Dodaj nowego uzytkownika.
6. Wyjdź z aplikacji.
4
Zmiana statusu konta!
-----<>-----
Podaj nick uzytkownika, ktorego status konta chcesz zmienic:
Melania21
Czy na pewno chcesz odbanowac uzytkownika Melania21? (T/N)
T
Uzytkownik Melania21 zostal odbanowany z aplikacji!
```

Rysunek 10

Funkcjonalność nr.5 – Wyjdź z aplikacji:

Ta opcja zapisuje wszystkie wysłane wiadomości i zamyka program.

c. Administratorzy:

Po zalogowaniu użytkownikowi wyświetla się menu z 6 możliwymi do podjęcia akcjami (rys. 11)

```
UDALO SIE ZALOGOWAC! Witaj Azymut
-----Panel sterowania-----
Jaka operacje chcesz wykonac:
-----<>-----
1. Wejdz w chat.
2. Pokaz liste uzytkownikow chatu.
3. Pokaz informacje o profilu.
4. Zbanuj uzytkownika.
5. Dodaj nowego uzytkownika.
6. Wyjdz z aplikacji.
```

Rysunek 11

Funkcjonalności 1, 2, 3, 4 pokrywają się z moderatorami.

Funkcjonalność nr.5 – Dodaj nowego użytkownika:

Opcja umożliwiająca dodanie do chatu przez administratora nowego użytkownika (automatyczne wpisanie go do bazy danych i dodanie do chatu grupowego). Administrator osobiście podaje wszystkie dane nowego użytkownika, wybiera typ konta i potwierdza utworzenie profilu. (rys. 12– dodanie przykładowe użytkownika „Matrix”)

```
5. Dodaj nowego uzytkownika.
6. Wyjdz z aplikacji.
5
Dodaj nowego czlonka chatu grupowego:
imie: Marcin
nazwisko: Mikolowski
nickname: Matrix
haslo: redpill
miasto: Torun
nr_telefonu: 456687426
email: email@poczta.pl
rodzaj_konta: podstawowy
Dodano uzytkownika Matrix do bazy uzytkownikow aplikacji.
```

Rysunek 12

Funkcjonalność nr.6 – Wyjdz z aplikacji:

Ta opcja zapisuje wszystkie wysłane wiadomości i zamyka program.



## 2. Techniczny opis projektu

### 1. Struktura danych:

Struktura danych składa się z dwóch plików tekstowych (bazy danych użytkowników – `uzytkownicy.txt` oraz archiwum wiadomości – `chat.txt`). (rys. 13)



Rysunek 13

W pliku `uzytkownicy.txt` znajdują się separowane spacją dane (w tym login i hasło służące do sesji logowania oraz informacja o banie) dotyczące użytkowników aplikacji. (rys. 14)

```
uzytkownicy.txt - Notepad
File Edit Format View Help
0 imie nazwisko nickname(login) haslo miasto nr_telefonu email rodzaj_konta ban(1-zbanowane)
1 Angelika Wariat Melania21 !mela1234 Katowice 636284255 mela@poczta.pl podstawowy 0
2 Konrad Ostrowski Kony Kongosy37 Nysa 290384890 konygaming@poczta.pl podstawowy 0
3 Wiktoria Bielanik Bielan FadoLove3 Bialystok 457693703 fadoslaw@poczta.pl moderator 0
4 Janek Skowron skowronek2115 Rick2115 Tarnobrzeg 143694446 janek.skowron@poczta.pl podstawowy 0
5 Marcel Swierczek Cieciel Leszy33 Elblag 597556764 pustapuszka@poczta.pl administrator 0
6 Lucja Szczepak lucyjax lux123 Konin 876238450 konieczyna1303@poczta.pl podstawowy 0
7 Marcin Ciok bjngf zaq1@WSX Poznan 196049318 marcin.ciok@poczta.pl moderator 0
8 Percy Duda Kinimod Disney1999! Londyn 853128140 youritspec@poczta.pl podstawowy 0
9 Daniel Dzialowy Gimpson ArmiaAu Bytom 535211376 dany2121@poczta.pl podstawowy 0
10 Kamil Noc Azymut NocnaFuria2 Katowice 555333212 kamsnoc@poczta.pl administrator 0
11 admin admin admin a admin 435567423 admin administrator 0
12 Marcin Mikolowski Matrix redpill Torun 456687426 email@poczta.pl podstawowy 0
```

Rysunek 14

W pliku chat.txt znajdują się wszystkie dotychczas wysłane wiadomości (zapis konwersacji grupowej). Każda wiadomość jest poprzedzona, zakodowaną kolorem nazwą nadawcy. Kompilator w trakcie odczytywania wiadomości, korzystające ze zdefiniowanych danych koloruje nickname użytkowników. (rys. 15)

```

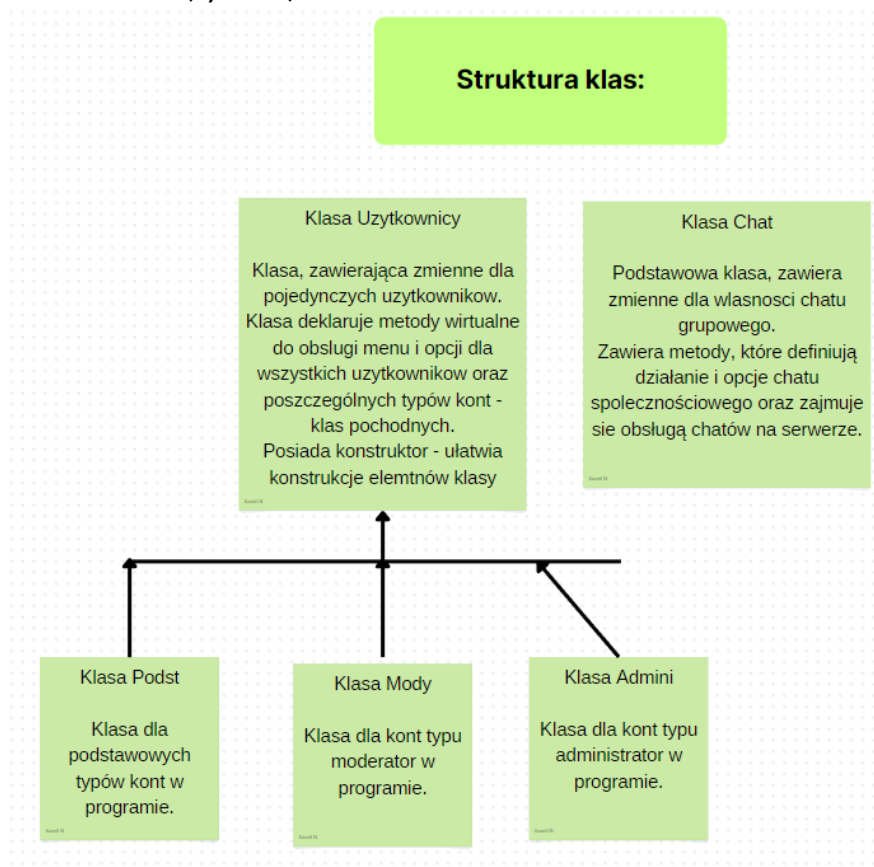
chat.txt - Notepad
File Edit Format View Help
// ABY WYJSC Z KONWERSACJI NAPISZ /esc
[[31m< Azymut > [[0m Hejka wszystkim!
[[31m< Cieciel > [[0m Siemaneczko
[[36m< Bielan > [[0m No heyy, to działa;
[[36m< Bielan > [[0msiemkaaa

```

Rysunek 15

W przypadku chęci stworzenia drugiego i następnego chatu grupowego, wystarczy utworzyć nowy obiekt klasy Chat, a ten będzie zapisywał konwersacje do innego pliku txt, w ten sposób możemy odpalić wiele niezależnych konwersacji grupowych, co zapewnia łatwą skalowalność i możliwość rozwoju aplikacji w przyszłości.

## 2. Struktura klas. (rys. 16)



Rysunek 16

Podstawową klasą na której oparty został projekt jest klasa Uzytkownicy. Klasa zawiera zmienne danych każdego pojedynczego konta w programie. Na obiektach tej klasy operują metody, związane z obsługą menu i funkcjonalnościami aplikacji. Klasa posiada swój konstruktor, metody dla wszystkich typów użytkowników do obsługi programu. (rys. 17)

Klasa jest klasą bazową, po której w sposób publiczny (w celu przekazania całej zawartości klasy do jej pochodnych) dziedziczą klasy Podst, Mody i Admini. Klasa zawiera definicje metod wspólnych dla każdego typu użytkownika, jak i deklarację metod wirtualnych, których definicję znajdują się w klasach pochodnych. W celu definicji tych samych metod dla różnych klas pochodnych został wykorzystany polimorfizm, następnie na obiekty klas pochodnych została stworzona tablica wskaźników do wywoływania metody dla obiektu klasy pochodnej. W prywatnych danych klasa przechowuje zmienną hasło.

```
// Klasa Uzytkownicy - Zawiera zmienne dla pojedynczych uzytkownikow. Deklaruje metody wirtualne dla obsługi menu i opcji dla poszczególnych typow kont klas pochodnych
class Uzytkownicy
{
public:
    //Zmienne klasy Uzytkownicy zawierajace dane uzytkownikow aplikacji
    std::string imie, nazwisko, login, miasto, nr_telefonu, email, rodzaj_konta;
    int nr_konta, ban, biezacy_uzytkownik;
    //konstruktor
    Uzytkownicy(int = 0, std::string="imie", std::string="nazwisko", std::string = "login", std::string = "haslo", std::string = "miasto", std::string = "11111111", s
    //Metoda obsługująca system logowania do serwisu (walidacji danych logowania, sprawdzenia statusu konta) oraz ladowania menu.
    int logowanie(std::vector<Uzytkownicy> Users);
    //Funkcjonalnosc nr2 - metoda umozliwiajaca wglad w liste uczestnikow chatu
    void pokaz_uzytkownikow(std::vector<Uzytkownicy> Users);
    //Glowna metoda - odpowiada za wyswietlanie i obsluge menu (w zaleznosci od typu zalogowanego konta) - wywoluje funkcjonalnosc programu.
    void obsluga_menu(Uzytkownicy** tablica_kont, std::vector<Uzytkownicy> Users, Chat chat1);
    //Funkcjonalnosc nr4 (mody, admini) - metoda umozliwiajaca zmiane statusu konta (zbanowanie/odbanowanie uzytkownika).
    void ban_uzytkownika(std::vector<Uzytkownicy> &Users);
    //Funkcjonalnosc nr5 (admini) - umozliwia dodanie, do bazy danych i chatu, nowego czlonka konwersacji - utworzenie nowego konta dowolnego typu.
    void dodaj_uzytkownika(std::vector<Uzytkownicy> &Users);
    //Metoda zapisujaca baze danych uzytkownikow i aktualne zmiany w sesji podczas wylaczania programu.
    void zapisz_klasy(std::vector<Uzytkownicy> Users);
    //wirtualna metoda obsługująca menu i opcje dla poszczegolnego typu konta
    //Virtual void menu() { std::cout << "domyslny" << std::endl; }
    //Funkcjonalnosc nr1 - wirtualna metoda deklarujaca mozliwosc napisania wiadomosc dla poszczegolnych typow kont - zdefiniowana inaczej dla roznych typow kont.
    virtual void napisz_wiadomosc(std::string nickname) {std::cout << "domyslny" << std::endl;}
```

Rysunek 17

W celu przekazania danych z klasy został wykorzystany mechanizm zaprzyjaźnienia – klasa przyjaźni się z funkcją załaduj\_klasy(), która odpowiada za odczytanie danych z pliku txt i przekazanie ich dalej. Następnie utworzony zostaje wektor który jest wypełniony obiektami tej klasy przy pomocy konstruktora. Z klasą zaprzyjaźniony jest również operator przeciężenia, który został zdefiniowany dla operatora strumienia wyjściowego – „<<”. Operator ten został wykorzystany przy okazji projektowania funkcjonalności nr 3 – jego wywołanie wyświetla dane aktualnie zalogowanego użytkownika. (rys. 18)

```
protected:
    std::string haslo;
    //funkcja zaprzyjazniona z klasa - wypelnia wektor zawierajacy obiekty tej klasy, poszczegolnymi danymi uzytkownikow z bazy danych (pliku tekstowego).
    friend std::vector<Uzytkownicy> zaladuj_klasy();

    //Funkcjonalnosc nr3 - umozliwia podejrzenie informacji odnosnie swojego konta(info o aktualnie zalogowanym uzytkowniku) - zastosowane jest przeciazanie operatora,
    aby przy pomocy operatora strumienia wyjsciowego wyswietlic wszystkie pozadane dane o zalogowanym uzytkowniku.*/
    friend std::ostream& operator<<(std::ostream& strumien, std::vector<Uzytkownicy> Users);
```

Rysunek 18

Klasy Podst, Mody, Admini dziedziczą po klasie Uzytkownicy. Każda z tych klas zawiera tylko użytkowników swojego typu, zgodnie z analogią nazewnictwa. Klasy te odpowiadają za wywołanie się odpowiednio zadeklarowanej metody w zależności od tego, jaki typ użytkownika jest zalogowany. Każda z nich posiada swój konstruktor. (rys. 19)

```
// Klasa dla uzytkownikow podstawowych - dziedziczy po klasie Uzytkownicy
class Podst :public Uzytkownicy
{
public:
    //konstruktor
    Podst(int = 0, std::string = "imie", std::string = "nazwisko", std::string = "login", std::string = "haslo", s
    //void menu() { std::cout << " MENU Podst" << std::endl;}
    //metoda odpowiadajaca za obsluge wysylania i zapisu wiadomosci przez uzytkownikow podstawowych
    void napisz_wiadomosc(std::string nickname);
};

// Klasa dla moderatorow - dziedziczy po klasie Uzytkownicy
class Mody :public Uzytkownicy
{
public:
    //konstruktor
    Mody(int = 0, std::string = "imie", std::string = "nazwisko", std::string = "login", std::string = "haslo", s
    //void menu() {std::cout << " MENU MODOW" << std::endl;}
    //metoda odpowiadajaca za obsluge wysylania i zapisu wiadomosci przez moderatorow
    void napisz_wiadomosc(std::string nickname);
};

// Klasa dla administratorow - dziedziczy po klasie Uzytkownicy
class Admini :public Uzytkownicy
{
public:
    //konstruktor
    Admini(int = 0, std::string = "imie", std::string = "nazwisko", std::string = "login", std::string = "haslo", s
    //void menu() {std::cout << " MENU ADMINOW" << std::endl;}
    //metoda odpowiadajaca za obsluge wysylania i zapisu wiadomosci przez administratorow
    void napisz_wiadomosc(std::string nickname);
};
```

Rysunek 19

Dodatkowo jest niezależna klasa Chat zawierająca zmienne dla właściwości każdego pojedynczego chatu. Każdy obiekt tej klasy stanowi niezależną konwersację grupową pomiędzy użytkownikami aplikacji (w bieżącej wersji aplikacji istnieje jeden chat). Klasa zawiera metody zajmujące się obsługą chatów. (rys. 20)

```
// klasa Chat - Zawiera zmienne dla chatu. Zawiera metody, ktore definiuja dzialanie i opcje chatu - obsluguja chaty na serwerze.
class Chat
{
    //int ilosc_chatow;
public:
    //metoda klasy Chat ladujaca z archiwum (pliku txt) chat z dotychczasowymi wiadomosciami i wyswietlajaca go uzytkownikowi
    void wyswietl_chat();
};
```

Rysunek 20