

Programowanie w środowiskach zintegrowanych

2019/2020

Kamil Wyżgoł, grupa ISI3

Zadanie programistyczne

1. Opis programu

- ❖ **Nazwa:** Terminarz
- ❖ **Narzędzie realizacji:** Visual Studio - C# (WinForms z elementami WPF)
- ❖ **Użyty framework:** .NET Framework 4.5 (wersja 4.5 dla większej kompatybilności)
- ❖ **Opis:** Program pozwala na tworzenie oraz zarządzanie listami wydarzeń.
- ❖ **Główne cechy:**
 - ➔ umożliwia personalizację profilu,
 - ➔ umożliwia utworzenie grupy,
 - ➔ umożliwia przeglądanie profili innych użytkowników należących do tej samej grupy,
 - ➔ umożliwia dostęp do wydarzeń w ramach grupy lub konkretnego użytkownika,
 - ➔ powiadamia o zbliżających się wydarzeniach (w zależności od zapisanych ustawień).
- ❖ **Pakiet realizujący połączenie z bazą danych:**
Oracle.ManagedDataAccess
- ❖ **Link do filmu prezentującego działanie:**
<https://www.youtube.com/watch?v=KfEJOE9pSkI>

Wszystkie grafiki użyte w programie pochodzą ze strony: <https://icons8.com/>

2. Fragmenty kodu związane z bazą danych:

❖ Połączenie z bazą danych

```
public static void DatabaseConnection()
{
    try
    {
        string conString = "User Id=root; password=12345;" +
            "Data Source=127.0.0.1:1521/oltpstud; Pooling=false;";
        con = new OracleConnection();
        con.ConnectionString = conString;
        con.Open();
    }
    catch (Exception exc)
    {
        con = null;
    }
}
```

❖ Funkcja realizująca zapytania

```
public static OracleDataReader QueryResult(string queryCommand)
{
    try
    {
        OracleCommand cmd = con.CreateCommand();
        cmd.CommandText = queryCommand;
        OracleDataReader reader = cmd.ExecuteReader();
        return reader;
    }
    catch (Exception exc)
    {
        MessageBox.Show("Błąd podczas wysyłania zapytania do bazy  
danych. Sprawdź swoje połączenie internetowe.");
        return null;
    }
}
```

❖ Funkcja realizująca komendy DML

```
public static bool dmloperation(List<string> cmdList)
{
    try
    {
        OracleTransaction transaction = con.BeginTransaction();
        OracleCommand cmd = con.CreateCommand();
        cmd.Transaction = transaction;

        try
        {
            foreach (string cmdText in cmdList)
            {
                cmd.CommandText = cmdText;
                cmd.ExecuteNonQuery();
            }
            transaction.Commit();
            return true;
        }
        catch (Exception exc)
        {
            transaction.Rollback();
            MessageBox.Show("Błąd podczas próby zapisu do bazy danych.  
Sprawdź swoje połączenie internetowe.", "Komunikat");
            return false;
        }
    }
    catch (Exception exc)
    {
        MessageBox.Show("Błąd podczas próby zapisu do bazy danych.  
Sprawdź swoje połączenie internetowe.", "Komunikat");
        return false;
    }
}
```

3. Schematy użytych tabel (oraz definicje użytych sekwencji)

- ❖ **CREATE SEQUENCE project_users_seq
INCREMENT BY 1
START WITH 1
NOCYCLE;**
- ❖ **CREATE SEQUENCE project_events_seq
INCREMENT BY 1
START WITH 1
NOCYCLE;**
- ❖ **CREATE SEQUENCE project_groups_seq
INCREMENT BY 1
START WITH 1
NOCYCLE;**
- ❖ **CREATE TABLE project_users(user_id NUMBER(6)
CONSTRAINT p_user_id_pk PRIMARY KEY,
u_login VARCHAR2(25) NOT NULL,
u_password VARCHAR2(25) NOT
NULL, first_name VARCHAR2(25),
last_name VARCHAR2(25), email
VARCHAR2(50), informations
VARCHAR2(400));**

- ❖ CREATE TABLE project_settings(user_id
NUMBER(6) CONSTRAINT
p_settings_user_id_pk PRIMARY KEY
CONSTRAINT p_settings_user_id_fk REFERENCES
project_users(user_id) ON DELETE CASCADE,
minimize_enable VARCHAR2(3) NOT NULL,
alarm_enable VARCHAR2(3) NOT NULL,
alarm_type VARCHAR2(10), alarm_time
NUMBER(5));

- ❖ CREATE TABLE project_login_history(user_id
NUMBER(6) CONSTRAINT p_login_history_user_id_fk
REFERENCES project_users(user_id) ON DELETE
CASCADE, login_date TIMESTAMP WITH TIME ZONE
NOT NULL);

- ❖ CREATE TABLE project_groups(group_id NUMBER(6)
CONSTRAINT p_group_id_pk PRIMARY KEY,
admin_id NUMBER(6) CONSTRAINT
p_groups_admin_id_fk REFERENCES
project_users(user_id) ON DELETE CASCADE NOT
NULL,
group_name VARCHAR2(25) NOT NULL,
description VARCHAR2(400));

- ❖ CREATE TABLE project_membership(user_id
NUMBER(6) CONSTRAINT p_membership_user_id_fk
REFERENCES project_users(user_id) ON DELETE
CASCADE, group_id NUMBER(6) CONSTRAINT
p_membership_group_id_fk REFERENCES
project_groups(group_id) ON DELETE CASCADE,
edit_permission VARCHAR2(5) NOT NULL,
CONSTRAINT p_membership_pk PRIMARY KEY(user_id,
group_id));

- ❖ CREATE TABLE project_events(event_id NUMBER(6)
CONSTRAINT p_event_id_pk PRIMARY KEY,
event_name VARCHAR2(25),
event_date TIMESTAMP WITH TIME ZONE NOT
NULL, description VARCHAR2(400), event_type
VARCHAR2(5) NOT NULL,
user_id NUMBER(6) CONSTRAINT p_events_user_id_fk
REFERENCES project_users(user_id) ON DELETE
CASCADE,
group_id NUMBER(6) CONSTRAINT
p_events_group_id_fk REFERENCES
project_groups(group_id) ON DELETE CASCADE);

4. Przykład programistycznego wykonania:

❖ Zapytania:

```
string cmd = string.Format("SELECT g.group_name,  
g.description, m.edit_permission FROM  
project_groups g JOIN project_membership m ON  
g.group_id = m.group_id WHERE g.group_id = {0}  
AND m.user_id = {1}", this.groupId,  
Utilities.UserId);  
OracleDataReader reader =  
Utilities.QueryResult(cmd);  
Przykład odczytania danych:  
reader.Read(); reader.GetString(0);
```

❖ Komendy DML

```
string cmd = string.Format("UPDATE  
project_membership SET edit_permission = 'user'  
WHERE group_id={0} AND user_id={1}", groupId,  
listView.SelectedItems[0].SubItems[2].Text);  
List<string> cmdList = new List<string>();  
cmdList.Add(cmd);  
bool result = Utilities.dmlOperation(cmdList);
```

5.Screeny programu

Logowanie



Login:

Hasło:




Nie posiadasz konta?






Rejestracja




* Login:


* Hasło: 

Imię:

Nazwisko:

E-mail:





Terminarz

Wydarzenia

Profil

Grupy

Ustawienia

Pomoc

Informacje

Wyloguj

Wyjście

Dzisiaj

Data: 30.06.2020

Godzina: 22:38:44

Tryb:

Dzisiejsze wydarzenia

Nazwa	Godzina	Data	Typ	Nazwa grupy
Egzamin		30.06.2020	Grupowe	Studenci
Test po		30.06.2020	Osobiste	-
wydarzenie test	20:22	30.06.2020	Osobiste	-
wydarzenie test 323	20:23	30.06.2020	Osobiste	-
nazwa1	20:56	30.06.2020	Osobiste	-
12345	21:05	30.06.2020	Osobiste	-
1234567	21:09	30.06.2020	Osobiste	-
abcd	21:11	30.06.2020	Osobiste	-

+

Dodaj

Edytuj

×

Usuń

Opis

Terminarz

Wydarzenia

Profil

Grupy

Ustawienia

Pomoc

Informacje

Wyloguj

Wyjście

Login:kamil

Nowe hasło:

Powtórz nowe hasło:

Imię:Kamil

Nazwisko:W

Email:kamil@mail.com

Informacje dodatkowe:

informacje

Zapisz

Terminarz

Wydarzenia

Profil

Grupy

Ustawienia

Pomoc

Informacje

Wyloguj

Wyjście

+

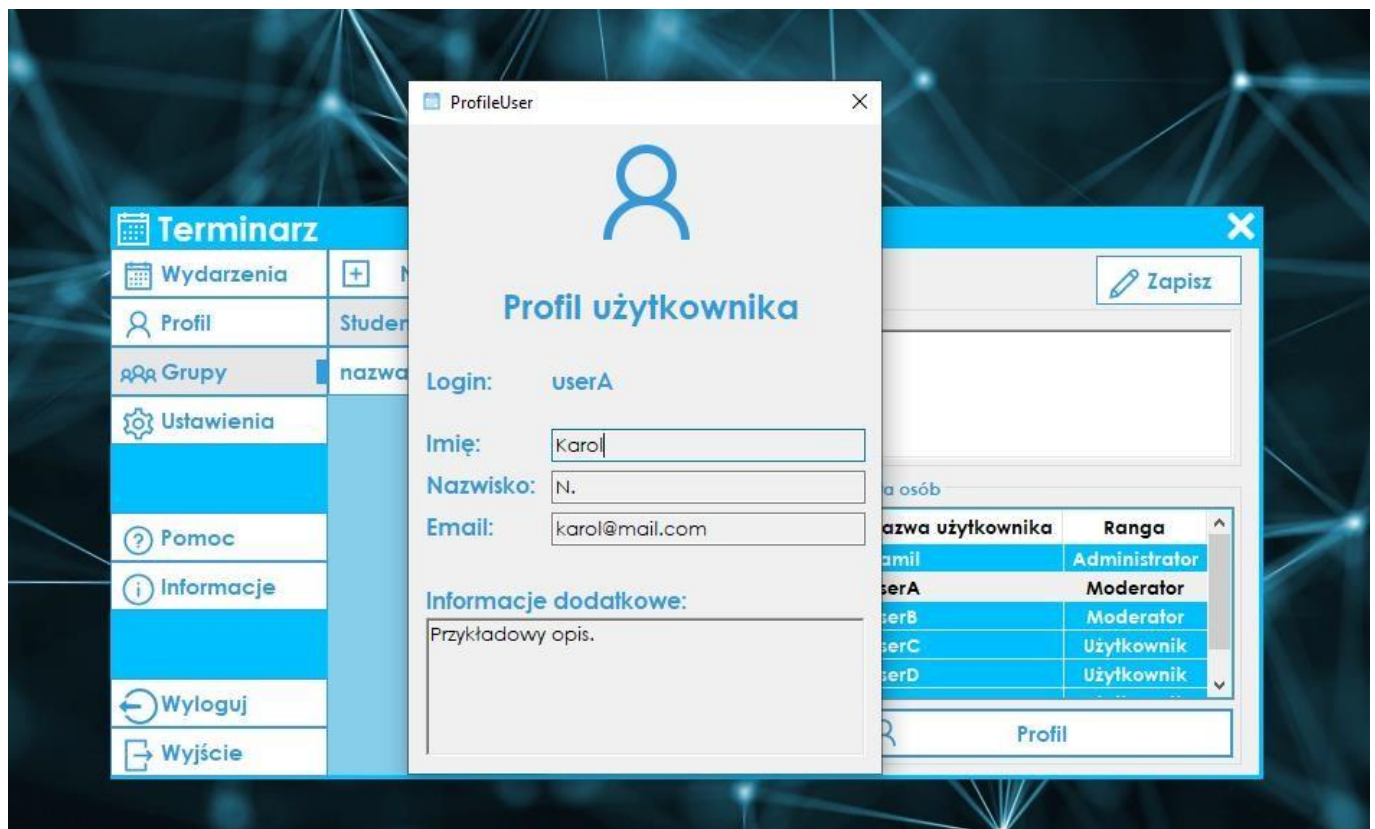
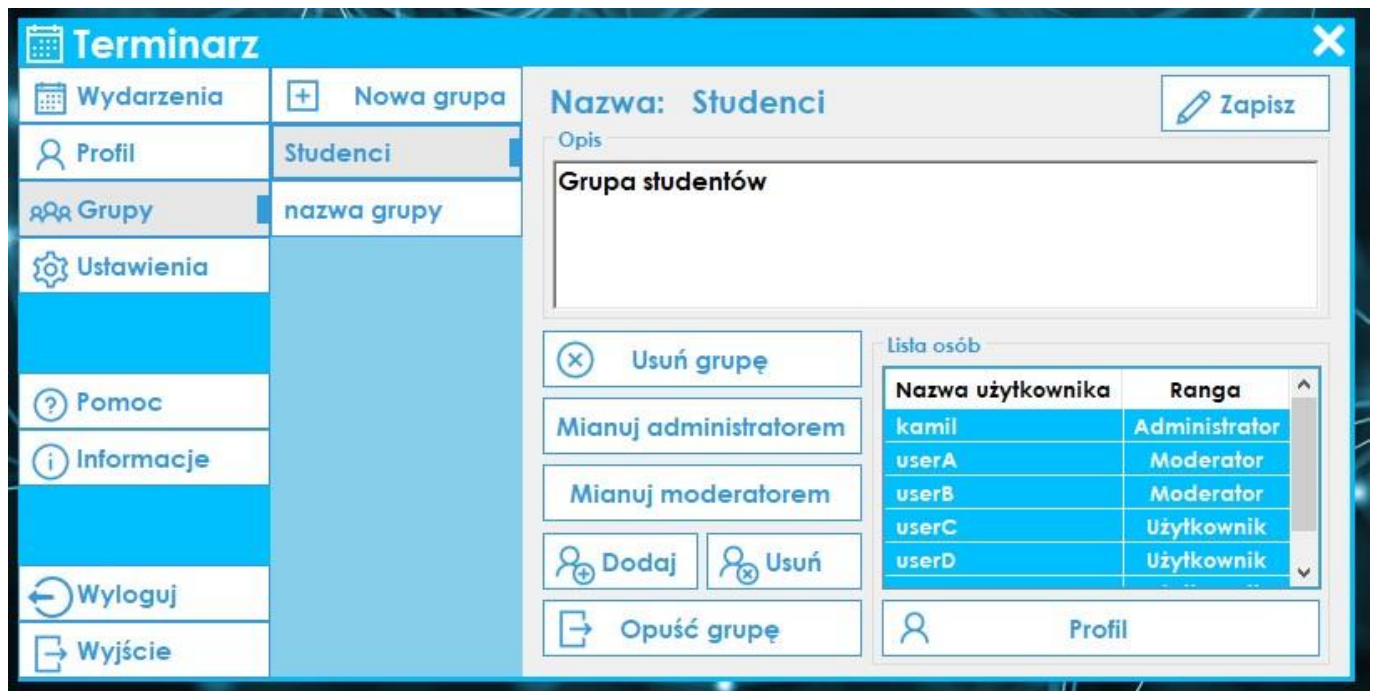
Nowa grupa

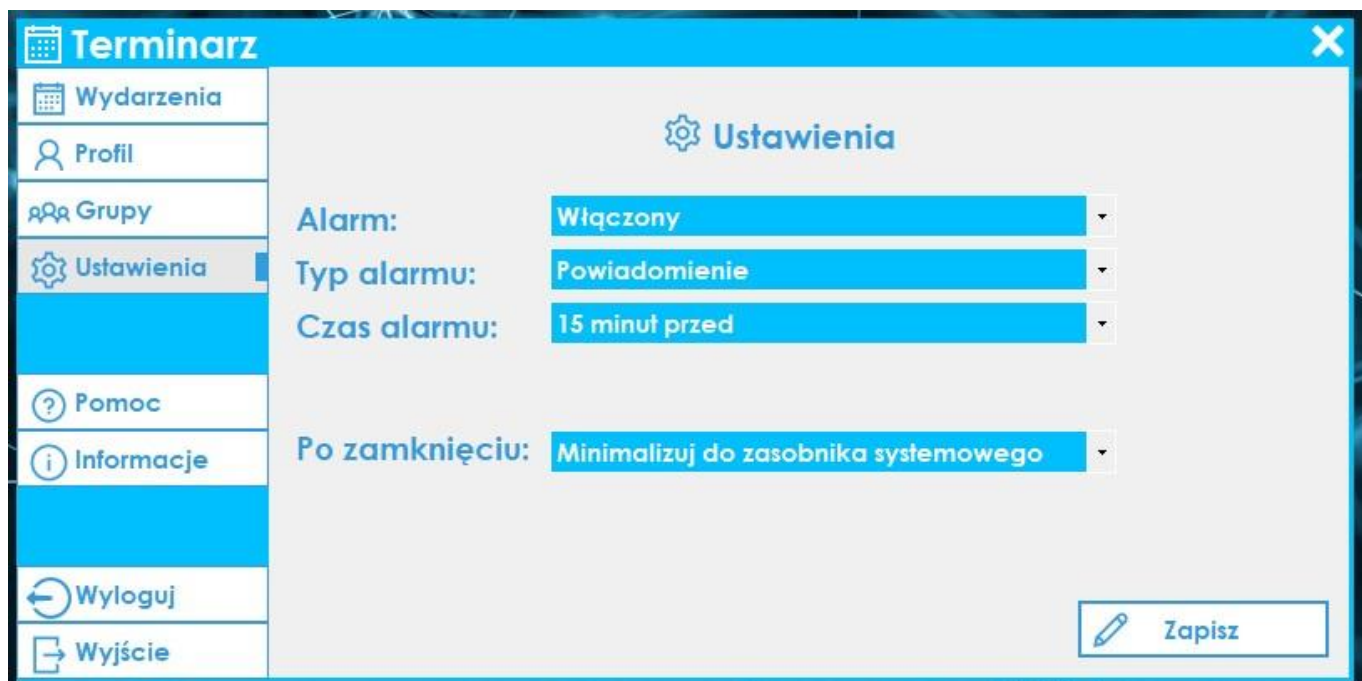
Tworzenie grupy

Nazwa:

Opis:

Utwórz





Terminarz

Wydarzenia

Profil

Grupy

Ustawienia

Pomoc

Informacje

Wyloguj

Wyjście

Nazwa: Terminarz

Opis: Program pozwala na tworzenie oraz zarządzanie listami wydarzeń.

Główne cechy:

- umożliwia personalizację profilu,
- umożliwia utworzenie grupy,
- umożliwia przeglądanie profili innych użytkowników należących do tej samej grupy,
- umożliwia dostęp do wydarzeń w ramach grupy lub konkretnego użytkownika,
- powiadamia o zbliżających się wydarzeniach (w zależności od zapisanych ustawień).

Wykonał: Kamil Wyżgoł, Grupa ISI3

PWŚZ, MSBD

2019/2020

Wszystkie użyte grafiki pochodzą ze strony: <https://icons8.com/>

Terminarz

Wydarzenia

Profil

Grupy

Ustawienia

Pomoc

Informacje

Wyloguj

Wyjście

Dzisiaj

Data: 30.06.2020

Wybór typu wydarzenia

Typ wydarzenia:

Osobiste

Wybierz

Anuluj

				Nazwa grupy
Wydarzenie test	20:22	30.06.2020	Osobiste	Studenci
wydarzenie test 323	20:23	30.06.2020	Osobiste	-
nazwa1	20:56	30.06.2020	Osobiste	-
12345	21:05	30.06.2020	Osobiste	-
1234567	21:09	30.06.2020	Osobiste	-
abcd	21:11	30.06.2020	Osobiste	-

Dodaj

Dodawanie wydarzenia

Podstawowe informacje

Nazwa:

Data:

30.06.2020



Godzina:

22



:

47



Szczegóły

Typ:

Osobiste



Szczegóły



Dodaj



Anuluj