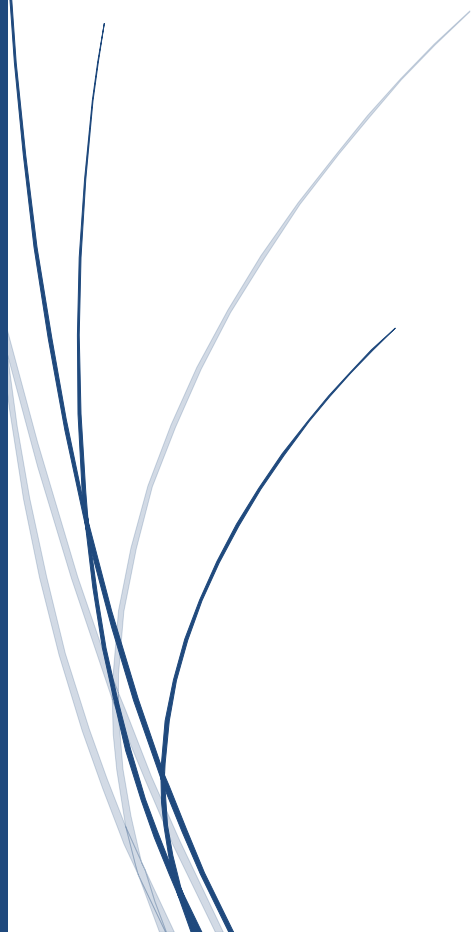




Autor: Krzysztof Małysa

Szkoła językowa

**Aplikacja łącząca bazę danych Oracle ze
środowiskiem Java oraz
funkcjonalnością z języka PL/SQL.**



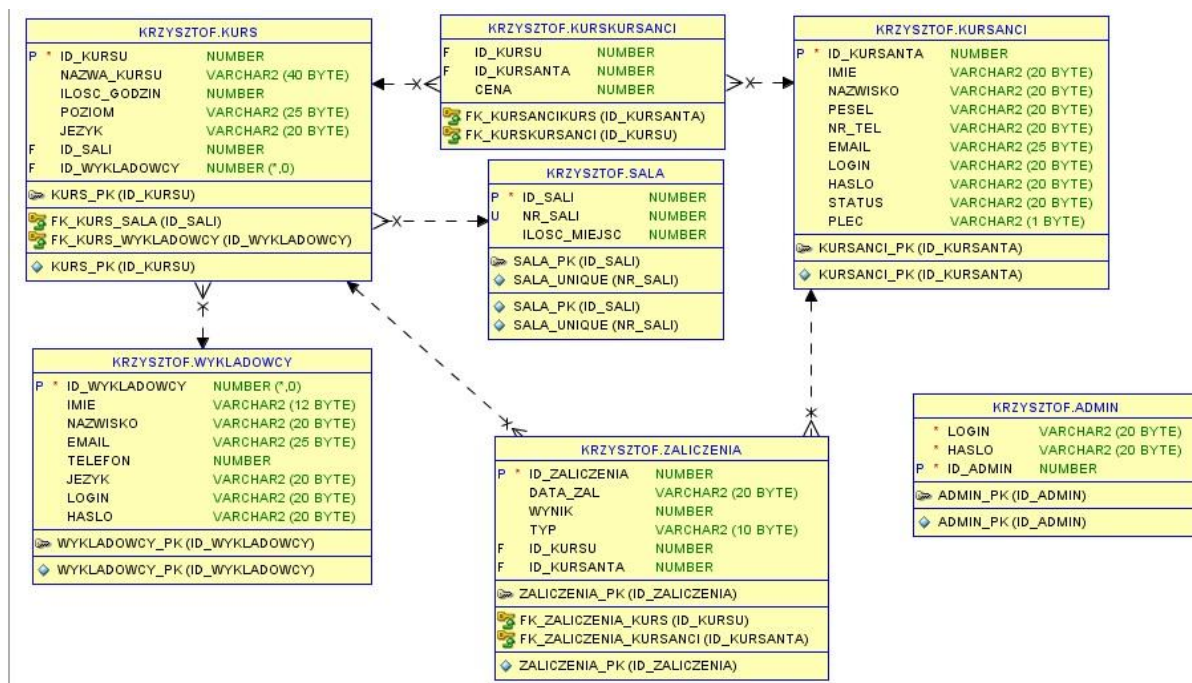
Użyte programy/pliki:

- NetBeans IDE 8.2,
- JavaFX Scene Builder Gluon,
- SQLDeveloper,
- Baza danych wykonana w Oracle,
- Sterownik do komunikacji z bazą oracle,
- Biblioteki do obsługi wysyłania E-maila,
- Biblioteka jfoenix i TrayNotification

Mój projekt dotyczy szkoły językowej. Aplikacja jest przeznaczona do użytku administratora, wykładowcy i kursanta. Jest także możliwa edycja wszystkich danych(z pewnymi wyjątkami) oraz podglądnięcie wybranych danych statystycznych.

Hasło do połączenia z bazą : 1potwierdzam@3.

Schemat mojej bazy:



```

create or replace procedure dodaj_kurs
(
    NAZWA_KURSU IN KURS.NAZWA_KURSU%TYPE,
    ILOSC_GODZIN IN KURS.ILOSC_GODZIN%TYPE,
    POZIOM IN KURS.POZIOM%TYPE,
    JEZYK IN KURS.JEZYK%TYPE,
    ID_SALI IN KURS.ID_SALI%TYPE,
    ID_WYKLADOWCY IN KURS.ID_WYKLADOWCY%TYPE
)
AS
begin
insert into KURS("NAZWA_KURSU","ILOSC_GODZIN","POZIOM","JEZYK","ID_SALI","ID_WYKLADOWCY")
values (NAZWA_KURSU,ILOSC_GODZIN,POZIOM,JEZYK,ID_SALI,ID_WYKLADOWCY);

commit;
end dodaj_kurs;

```

```

create or replace procedure usun_sale(d in sala.id_sali%type)
is
begin
delete sala where id_sali = d;
commit;
end;

```

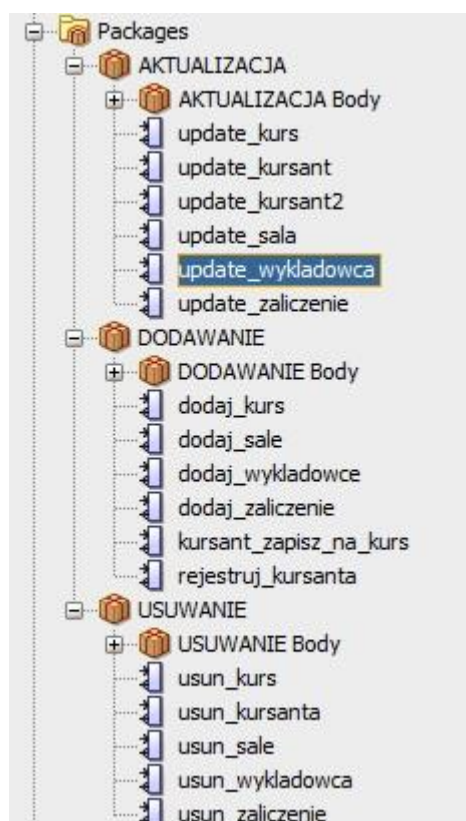
```

create or replace procedure update_kurs(
    idkursu in number,
    nazwakursu IN varchar2,
    iloscgodzin in number,
    poziomk IN varchar2,
    jezykk IN varchar2,
    idsali IN number,
    idwykladowcy IN number
)
is
begin
update KURS
SET
    nazwa_kursu = nazwakursu,
    ilosc_godzin = iloscgodzin,
    poziom = poziomk,
    jezyk = jezykk,
    id_sali = idsali,
    id_wykladowcy = idwykladowcy
WHERE id_kursu = idkursu;

END;

```

Wszystkie procedury do edycji, dodawania i usuwania danych wyglądają bardzo podobnie względem siebie. Różnią się tylko parametrami jakie trzeba podać. Dodatkowo zostały one umieszczone w pakietach co ułatwia dostęp.




Jest to okno startowe aplikacji.



Aby wywołać panel boczny wystarczy najechać kursorem myszy na lewą stronę okienka.

Z tego poziomu możemy odpowiednio przejść do interesujących nas rzeczy (logowanie, rejestracja, statystyki czy dzielenie się opinią). Zarejestrować możemy się tylko jako kursant, natomiast wykładowcy są rejestrowani przez administratora.



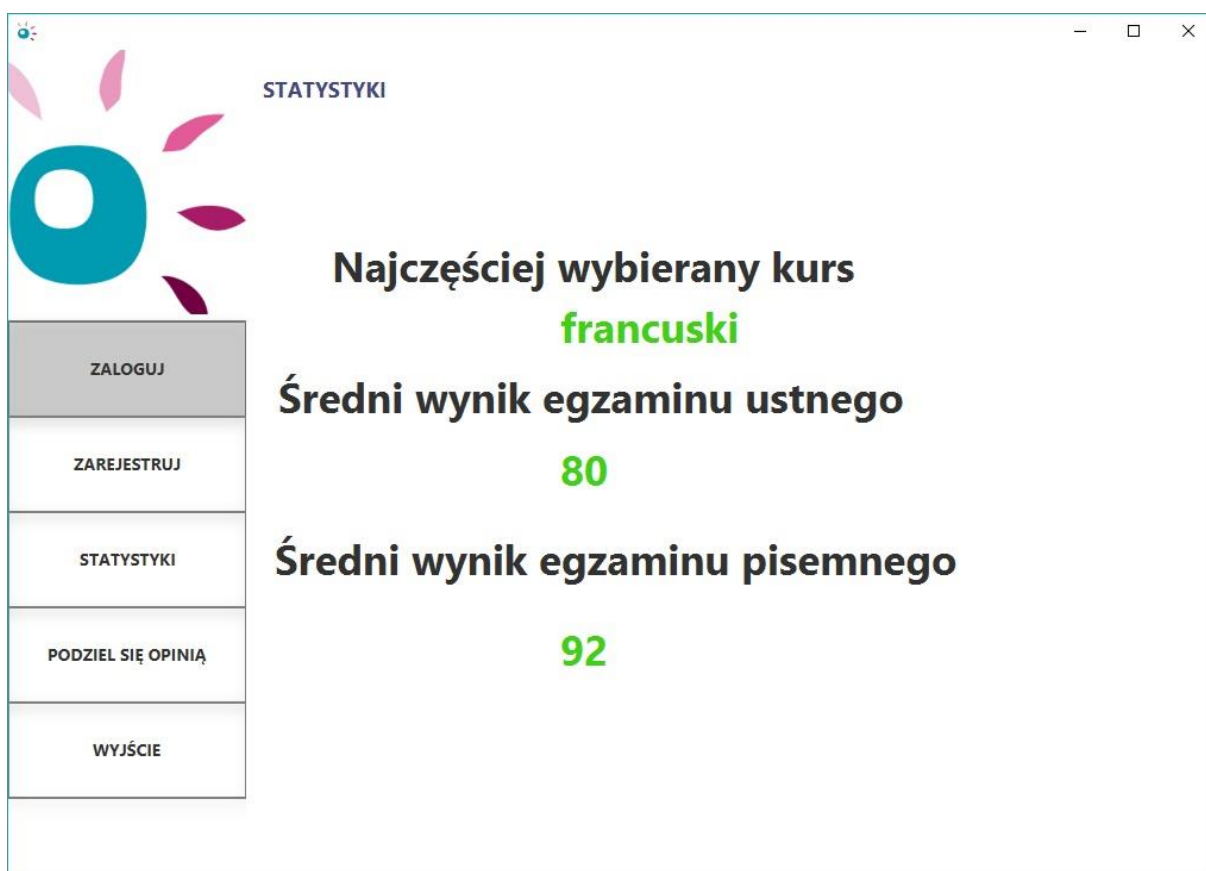
PANEL REJESTRACJI

[ZALOGUJ](#)
[ZAREJESTRUJ](#)
[STATYSTYKI](#)
[PODZIEL SIĘ OPINIĄ](#)
[WYJŚCIE](#)

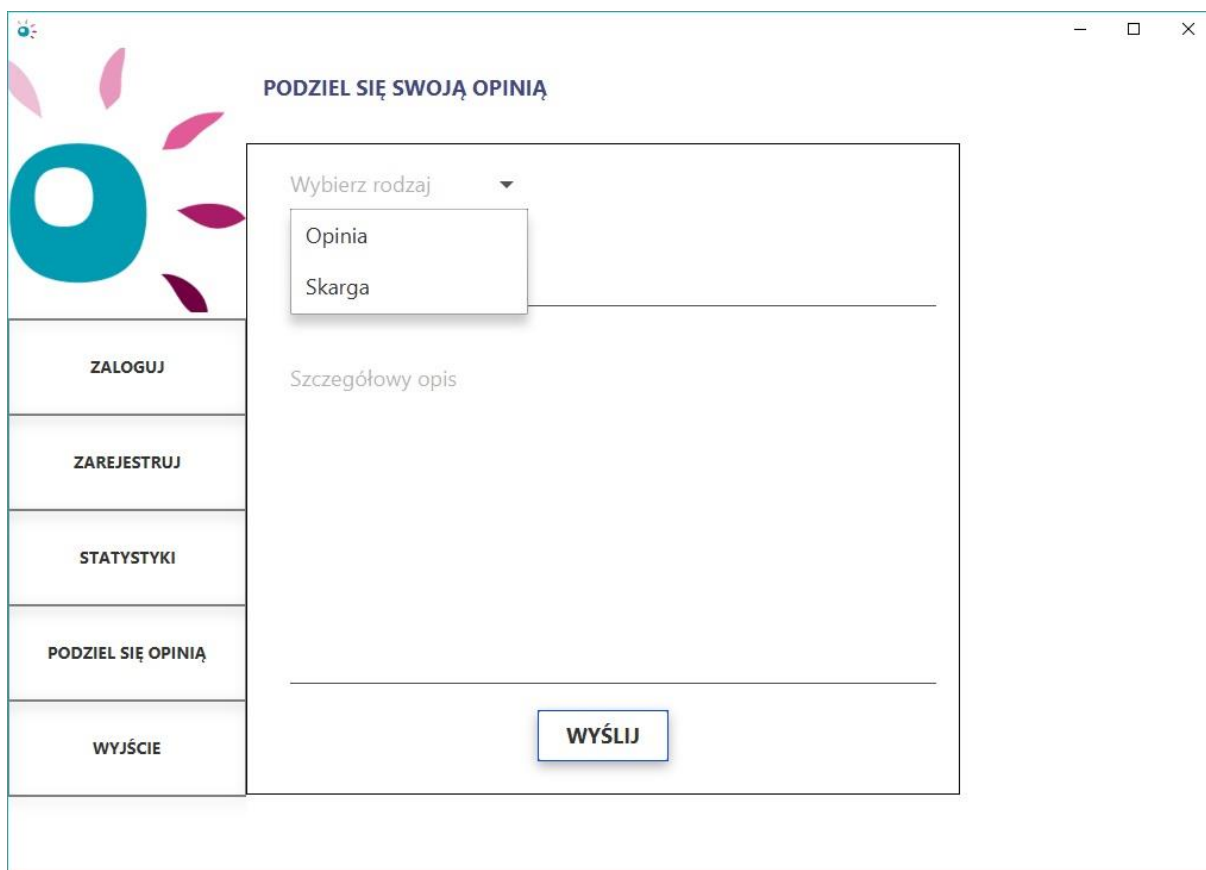
Login
 Hasło
 Wpisz ponownie swoje hasło
 Imię
 Nazwisko
 PESEL
 E-mail
 Numer telefonu
Wybierz płeć : ☐ Kobieta ☐ Mężczyzna
Wybierz swój status ▼

ZAREJESTRUJ

Powyższe okno przedstawia formularz rejestracji kursanta. Wszystkie pola do uzupełnienia są opisane. Ponadto login musi być unikalny.



Panel ze statystykami wyświetla 3 informacje, które mogą pozwolić ocenić nam jaki kurs jest najczęściej wybieranym czy też jakie są średnie wyniki efektów.



PODZIEL SIĘ SWOJĄ OPINIĄ

Wybierz rodzaj

- Opinia
- Skarga

Szczegółowy opis

WYŚLIJ

Jest to opcja „Podziel się opinią”. Wybieramy rodzaj, czy będzie to skarga czy opinia. Wszystko jest anonimowe. Następnie wiadomość trafia do administracji szkoły.

WYKŁADOWCA

[illegible]

Jest to panel widoczny po zalogowaniu się na wykładowcę i wybraniu opcji „Twoje kursy”. Wykładowca widzi w tabeli wszystkie kursy, do których jest przypisany.

Panel wykładowcy

TWOJE KURSY

TWOI KURSANCY

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

ZALICZENIA

Nazwa kursu	Kursant	Język	Poziom	Data	Typ	Wynik
Francuski użytkowy	Tadeusz Nowakowski	francuski	podstawowy	2017-05-31	pisemny	92
Francuski użytkowy	Tadeusz Nowakowski	francuski	podstawowy	2017-05-27	ustny	80

DODAJ ZALICZENIE

EDYTUJ ZALICZENIE

USUŃ ZALICZENIE

Na powyższym zrzucie widzimy zaliczenia wystawione przez zalogowanego wykładowcę. Może on je edytować, dodawać i usuwać. Aby aktywować przyciski edycji i usuwania należy wybrać jakiś rekord.

Panel wykładowcy

TWOJE KURSY

TWOI KURSANCY

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

ZALICZENIA

Francuski użytkowy

Tadeusz Nowakowski

27.05.2017

ustny

80

ZAPISZ ZMIANY

ANULUJ

Po kliknięciu edytuj, wyświetla nam się taki formularz z uzupełnionymi danymi. Możemy dokonać modyfikacji i zapisać zmiany.

Panel wykładowcy

ZALICZENIA

TWOJE KURSY

TWOI KURSANTI

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

Wybierz kurs

Wybierz kursanta

Data zaliczenia

Typ zaliczenia

Ilość punktów (od 0 do 100)

DODAJ ZALICZENIE

ANULUJ

Tak jak wyżej, jednak tutaj mamy do czynienia z dodawaniem nowego zaliczenia. Aby wybrać datę należy kliknąć na zieloną ikonkę kalendarza.

Panel kursanta

ZAPISZ SIĘ NA KURS

TWOJE ZALICZENIA

TWOJE DANE

TWOJE KURSY

WYLOGUJ SIĘ


WYJŚCIE

ZAPISZ SIĘ NA KURS

Nazwa kursu	Język	Poziom	Godziny	Otrzymany rabat	Cena	Wolne miejsca
Angielski na codzień	angielski	podstawowy	50	100.0	400.0	40
Angielski zaawansowany	angielski	zaawansowany	70	280.0	1120.0	55
Francuski na poziomie	francuski	zaawansowany	90	360.0	1440.0	40

ZAPISZ NA KURS

Jest to okno widoczne po zalogowaniu na kursanta. Na liście są dostępne kursy, na które są wolne miejsca oraz na których nie jesteśmy zapisani. Po wybraniu kursu wystarczy kliknąć przycisk „Zapisz na kurs”.



ZAPISZ SIĘ NA KURS

TWOJE ZALICZENIA

TWOJE DANE

TWOJE KURSY

WYLOGUJ SIĘ

WYJŚCIE


TWOJE ZALICZENIA

Nazwa kursu	Prowadzący	Język	Poziom	Data	Typ	Wynik
Francuski użytkowy	Tomasz Nowakowski	francuski	podstawowy		pisemny	92
Francuski użytkowy	Tomasz Nowakowski	francuski	podstawowy		ustny	80

Rezultat :

pozytywny

Po wybraniu „Twoje zaliczenia” kursant może podglądać zaliczenia, które go dotyczą. Aby zobaczyć, czy kurs jest zaliczony czy nie, wystarczy kliknąć na zaliczenie ustne lub pisemne.



ZAPISZ SIĘ NA KURS

TWOJE ZALICZENIA

TWOJE DANE

TWOJE KURSY

WYLOGUJ SIĘ

WYJŚCIE

TWOJE DANE

mkowalska

...

...

Martyna

Kowalska

90101009893

mkowalska@email.com

777666555


Wybierz płeć : ☒ Kobieta ☐ Mężczyzna

student

ZAPISZ

Po wybraniu z menu opcji „Twoje dane” możemy edytować swoje dane osobowe. Nie można jednak edytować loginu, płci czy też statusu.

Panel kursanta



ZAPISZ SIĘ NA KURS

TWOJE ZALICZENIA

TWOJE DANE

TWOJE KURSY

WYLOGUJ SIĘ

WYJŚCIE

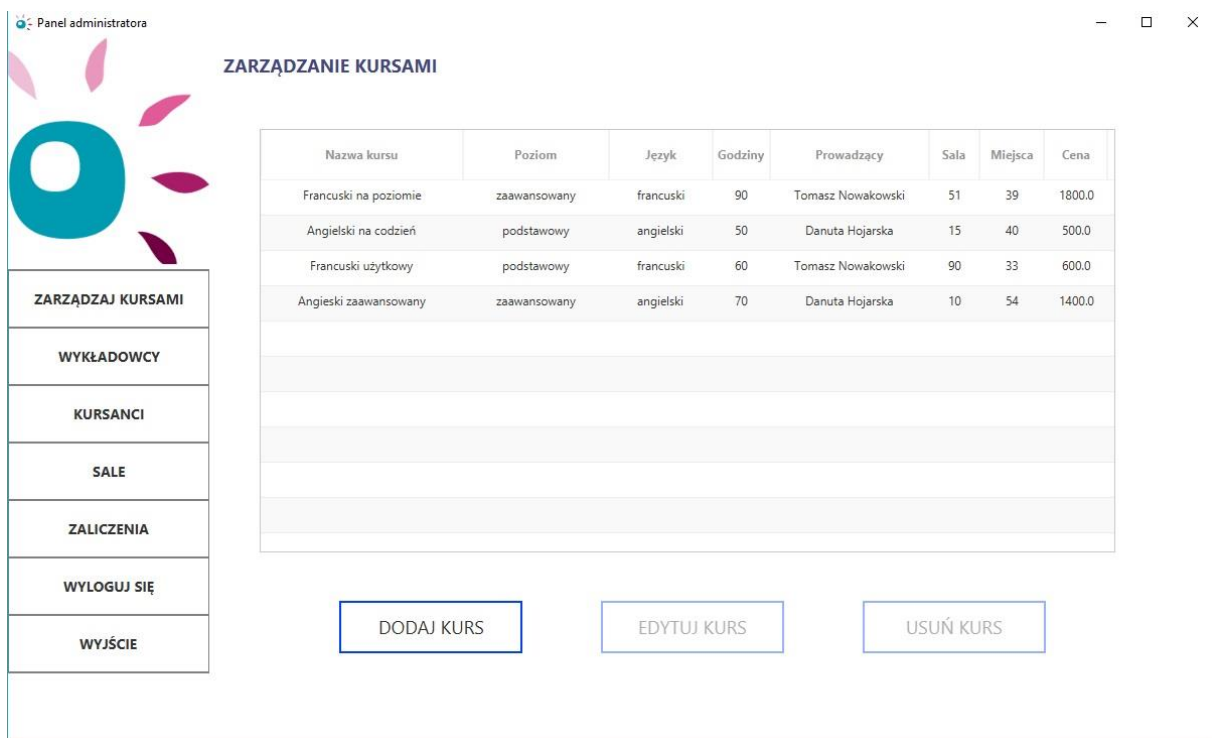
TWOJE KURSY

Nazwa kursu	Prowadzący	Język	Poziom	Godziny	Cena
Francuski użytkowy	Tomasz Nowakowski	francuski	podstawowy	60	510.0
Francuski na poziomie	Tomasz Nowakowski	francuski	zaawansowany	90	1440.0
Angielski zaawansowany	Danuta Hojarska	angielski	zaawansowany	70	1120.0

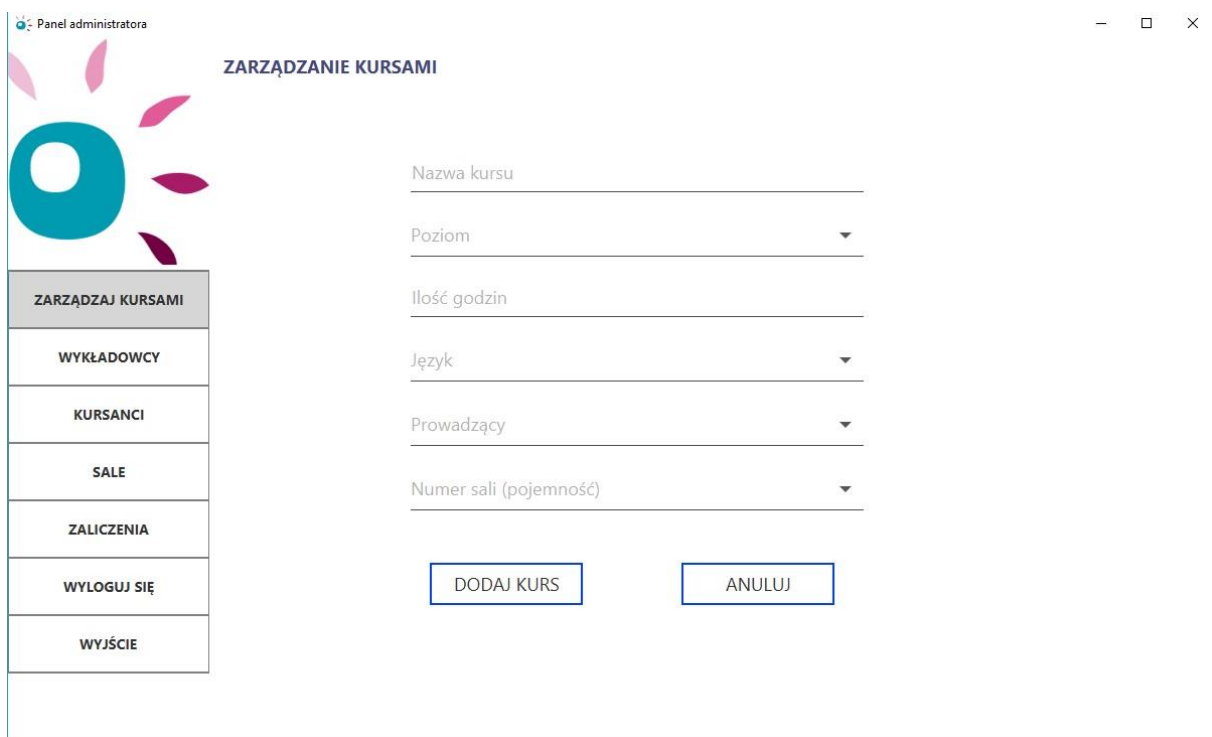
Okno „Twoje kursy” zawiera listę kursów, na które zostaliśmy zapisani, wraz z opłatą obliczoną dla nas.

ADMINISTRATOR


admin/admin



Z poziomu administratora możemy na wszelki sposób zarządzać wszystkimi danymi. Kolejne opcje wyglądają bardzo podobnie do siebie.



Powyższy zrzut przedstawia okno widoczne przy dodawaniu nowego kursu.



ZARZĄDZAJ KURSAMI

WYKŁADOWCY

KURSANTI

SALE

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

Panel administratora


ZARZĄDZANIE WYKŁADOWCAMI

Imię	Nazwisko	E-mail	Numer kontaktowy	Język	Login	Zdawalność %
Tomasz	Nowakowski	t_nowak@gmail.com	700920398	francuski	tnowak	67
Danuta	Hojarska	dhojarska@email.com	799000111	angielski	dhojarska	0

DODAJ WYKŁADOWCĘ

EDYTUJ WYKŁADOWCĘ

USUŃ WYKŁADOWCĘ



ZARZĄDZAJ KURSAMI

WYKŁADOWCY

KURSANTI

SALE

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

Panel administratora

ZARZĄDZANIE WYKŁADOWCAMI

Login

Hasło

Wpisz ponownie swoje hasło

Imię

Nazwisko

Numer telefonu

E-mail


Język

ZAREJESTRUJ

„Zarządzanie wykładowcami” umożliwia dodawanie, usuwanie i edytowanie wykładowców.

Panel administratora

ZARZĄDZANIE KURSANTAMI



ZARZĄDZAJ KURSAMI

WYKŁADOWCY

KURSANCI

SALE

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

Login

Hasło

Wpisz ponownie swoje hasło

Imię

Nazwisko

PESEL

E-mail

Numer telefonu


Wybierz płeć : ☐ Kobieta ☐ Mężczyzna

Wybierz swój status

ZAREJESTRUJ

Panel administratora

ZARZĄDZANIE KURSANTAMI



ZARZĄDZAJ KURSAMI

WYKŁADOWCY

KURSANCI

SALE

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

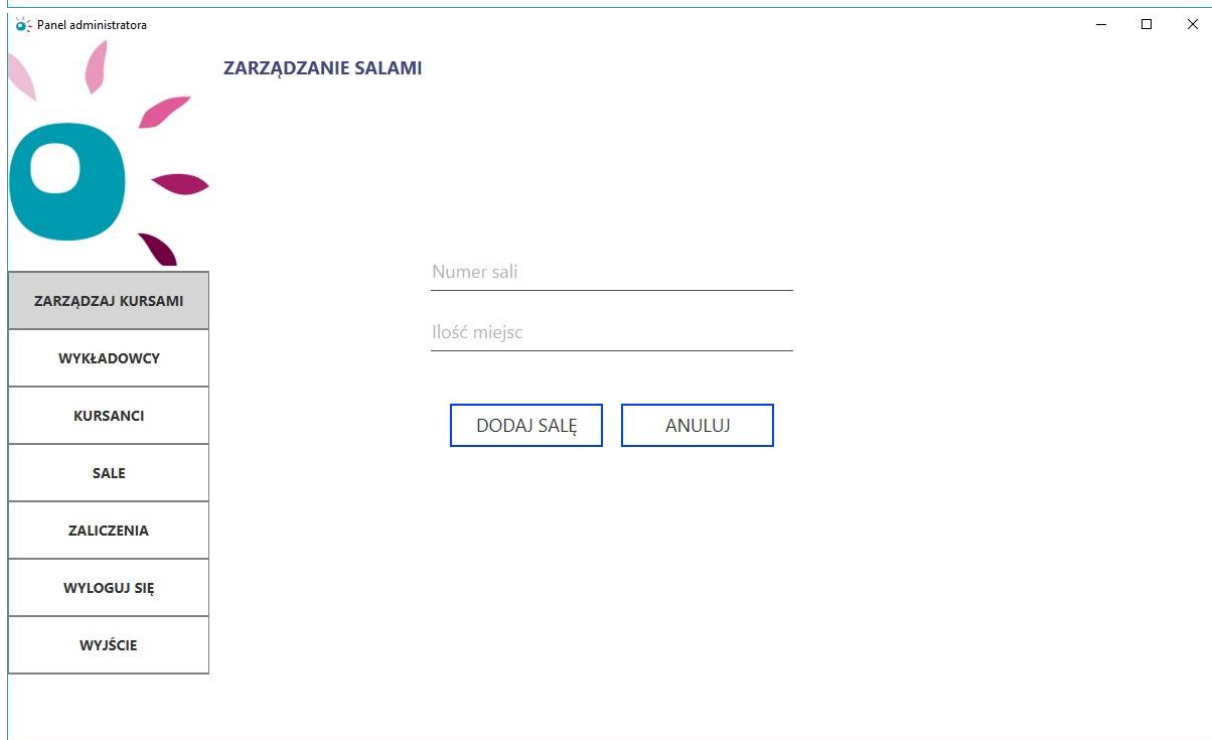
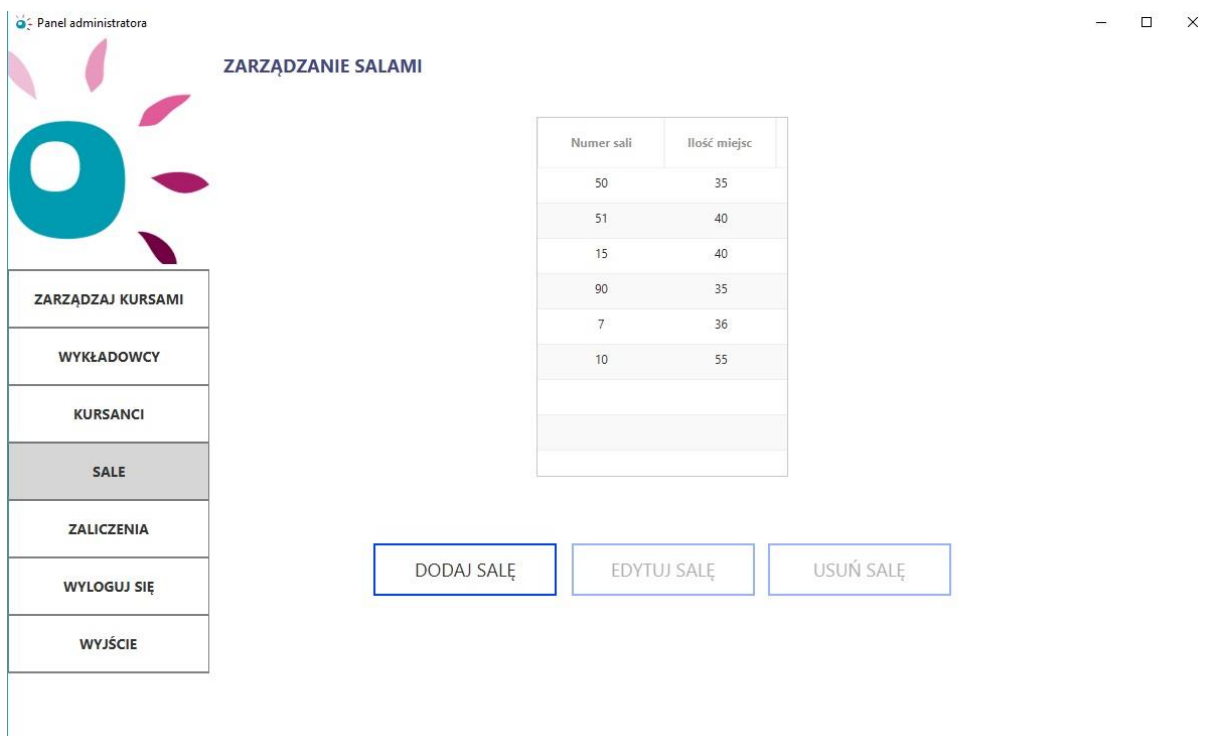
Imię	Nazwisko	PESEL	Telefon	E-mail	Login	Płeć	Status
Martyna	Kowalska	90101009893	777666555	mkowalska@email.com	mkowalska	K	student
Tadeusz	Nowakowski	90010293382	660009018	nowakowski@onet.pl	tnowak1	M	inny

DODAJ KURSANTA

EDYTUJ KURSANTA

USUŃ KURSANTA

„Zarządzanie kursantami” jest to kolejna bliźniacza opcja do wcześniejszych. Wszystko wygląda bardzo podobnie.



Kolejną możliwością u administratora jest zarządzanie salami. Również wygląda ono i działa bardzo podobnie jak opcje wyżej.

ZARZĄDZAJ KURSAMI

WYKŁADOWCY

KURSANCY

SALE

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

Panel administratora

ZARZĄDZANIE ZALICZENIAMI

Nazwa Kursu	Język	Poziom	Prowadzący	Kursant	Data	Typ	Wynik
Francuski użytkowy	francuski	podstawowy	Tomasz Nowakowski	Martyna Kowalska	2017-06-18	ustny	70
Francuski użytkowy	francuski	podstawowy	Tomasz Nowakowski	Tadeusz Nowakowski	2017-05-27	ustny	80
Francuski na poziomie	francuski	zaawansowany	Tomasz Nowakowski	Martyna Kowalska	2017-06-03	pisemny	89
Francuski użytkowy	francuski	podstawowy	Tomasz Nowakowski	Tadeusz Nowakowski	2017-05-31	pisemny	92
Francuski użytkowy	francuski	podstawowy	Tomasz Nowakowski	Martyna Kowalska	2017-06-01	pisemny	20
Francuski na poziomie	francuski	zaawansowany	Tomasz Nowakowski	Martyna Kowalska	2017-06-01	ustny	70

DODAJ ZALICZENIE

EDYTUJ ZALICZENIE

USUŃ ZALICZENIE

ZARZĄDZAJ KURSAMI

WYKŁADOWCY

KURSANCY

SALE

ZALICZENIA

WYLOGUJ SIĘ

WYJŚCIE

Panel administratora

ZARZĄDZANIE ZALICZENIAMI

Wybierz kurs

Wybierz kursanta

Data zaliczenia

Typ zaliczenia

Ilość punktów (od 0 do 100)

DODAJ ZALICZENIE

ANULUJ

„Zarządzanie zaliczeniami” – na liście widoczne są wszystkie wystawione zaliczenia ze szczegółami.

Teraz opiszę kilka wybranych fragmentów kodu.

Poniżej pokazana jest klasa `Polaczenie`, która łączy aplikację `javafx` z bazą danych.

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package szkolajezykowa.polaczenie;
7
8   import java.sql.Connection;
9   import java.sql.DriverManager;
10  import java.sql.SQLException;
11
12  /**
13   *
14   * @author BlackHawk
15   */
16  public class Polaczenie {
17
18      Connection con = null;
19
20      public static Connection ConnectDB() throws SQLException {
21          try {
22              Class.forName("oracle.jdbc.OracleDriver");
23              Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "krzysztof", "lpotwierdzam@3");
24              return con;
25          } catch (ClassNotFoundException | SQLException e) {
26              e.printStackTrace();
27              return null;
28          }
29      }
30  }

```

Kolejny fragment dotyczy edycji wykładowcy. Wywoływana jest procedura do edycji danych wykładowcy z odpowiednimi argumentami.

```

92  @FXML
93  private void aktualizujWykladowce(ActionEvent event) throws SQLException, IOException {
94      Connection con = Polaczenie.ConnectDB();
95      boolean rejestruj = true;
96      if (!fHaslo.getText().equals(fHaslo2.getText())) {
97          //rejestruj = false;
98      }
99      if (rejestruj) {
100         String query = "(CALL KRZYSZTOF.update_wykladowca(?, ?, ?, ?, ?, ?, ?, ?))";
101         //Connection con = Polaczenie.ConnectDB();
102         CallableStatement cst = con.prepareCall(query);
103         cst.setInt(1, WykladowcyViewController.idWykladowcy);
104         cst.setString(2, fImie.getText());
105         cst.setString(3, fNazwisko.getText());
106         cst.setString(4, fEmail.getText());
107         cst.setString(5, fTelefon.getText());
108         cst.setString(6, fJezyk.getText());
109         cst.setString(7, fLogin.getText());
110         cst.setString(8, fHaslo.getText());
111         cst.execute();
112         cst.close();
113         con.close();
114         AudioClip powiadomMnie = new AudioClip((SzkolaJezykowa.class.getResource("audio/Notify.wav")).toString());
115         powiadomMnie.play();
116         TrayNotification tray = new TrayNotification();
117         tray.setTitle("Sukces!");
118         tray.setMessage("Wykładowca został dodany");
119         tray.setNotificationType(NotificationType.SUCCESS);
120         tray.showAndDismiss(Duration.millis(1500));
121     }
122 }

```

Fragment przedstawiony poniżej dotyczy wywołania funkcji usuwania wykładowcy, parametr jaki musimy podać to jego id.

```

134 @FXML
135 private void usunWykladowce(ActionEvent event) throws SQLException {
136     String query = "{CALL KRZYSZTOF.usun_wykladowca(?)}";
137     Connection con = Polaczenie.ConnectDB();
138     CallableStatement cst = con.prepareCall(query);
139     cst.setInt(1, tableWykladowcy.getSelectionModel().getSelectedItem().getId_wykladowcy());
140
141     cst.execute();
142     cst.close();
143     con.close();
144
145     AudioClip powiadomMnie = new AudioClip((SzkolaJezykowa.class.getResource("audio/Notify.wav")).toString());
146     powiadomMnie.play();
147
148     TrayNotification tray = new TrayNotification();
149     tray.setTitle("Sukces!");
150     tray.setMessage("Wykladowca został usunięty");
151     tray.setNotificationType(NotificationType.SUCCESS);
152     tray.showAndDismiss(Duration.millis(1500));
153
154 }

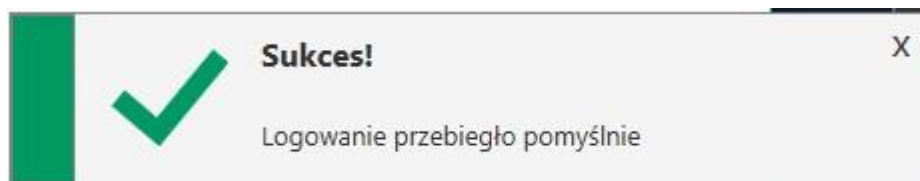
```

Procedura obejmująca zapisywanie kursantów na kursy jest wywoływana jak na obrazku poniżej.

```

122 @FXML
123 private void zapiszNaKurs(ActionEvent event) throws IOException, SQLException {
124
125     Integer idKursanta = Integer.parseInt(LoginViewController.);
126     Integer idKursu = tableZapiszNaKurs.getSelectionModel().getSelectedItem().getId_kursu();
127     cenaZaKurs = tableZapiszNaKurs.getSelectionModel().getSelectedItem().getCena();
128     System.out.println(cenaZaKurs);
129     Connection con = Polaczenie.ConnectDB();
130     String query = "{CALL KRZYSZTOF.kursant_zapisz_na_kurs(?,?,?)}";
131     CallableStatement cst = con.prepareCall(query);
132     cst.setInt(1, idKursu);
133     cst.setInt(2, idKursanta);
134     cst.setDouble(3, cenaZaKurs);
135     cst.execute();
136     cst.close();
137     con.close();
138
139     AudioClip powiadomMnie = new AudioClip((SzkolaJezykowa.class.getResource("audio/Notify.wav")).toString());
140     powiadomMnie.play();
141
142     TrayNotification tray = new TrayNotification();
143     tray.setTitle("Sukces!");
144     tray.setMessage("Zostałeś zapisany na kurs.");
145     tray.setNotificationType(NotificationType.SUCCESS);
146     tray.showAndDismiss(Duration.millis(1500));
147     odswiez();
148
149 }

```



Powyższy obrazek przedstawia przykładowe powiadomienie z programu.

Teraz opiszę kilka wybranych funkcji.

```

create or replace function zdawalnosc(idwykladowcy in number) return number is
cos varchar(255);
CURSOR kursor is SELECT czy_zdane(kursanci.id_kursanta,kurs.id_kursu) as wynik from kursanci,kurs,kurskursanci WHERE
kursanci.id_kursanta=kurskursanci.id_kursanta AND kurskursanci.id_kursu=kurs.id_kursu AND kurs.id_wykladowcy=idwykladowcy;

wynik Number(3,0);
zdane number default 0;
niezdane number default 0;
zmienna varchar(15) ;
zmiennaxd number default 0;
begin
for zmienna in kursor loop
if (zmienna.wynik='negatywny') then
niezdane:=niezdane+1;
else
zdane:=zdane+1 ;
end if;
end loop;
zmiennaxd:=zdane+niedzane;
if (zmiennaxd=0) then wynik:=0;
else
wynik:=(zdane/(zdane+niedzane))*100;
end if;
return wynik;
end;

```

Powyższa funkcja oblicza zdawalność dla wybranego wykładowcy. Jako parametr przy wywołaniu funkcji musimy podać jedynie id_wykladowcy. Zdawalność jest podawana w procentach, a wyliczana przy użyciu kursora.

```

create or replace FUNCTION "OBLICZ_CENE_ZNIZKI" (idkursanta IN NUMBER, idkursu IN NUMBER) return number is
statusKursanta varchar(20);
znizka number;
cena number;
iloscKursow number;
BEGIN
znizka:=1;
SELECT status INTO statusKursanta FROM Kursanci WHERE Kursanci.id_kursanta=idkursanta;
CASE statusKursanta
WHEN 'uczeń' THEN znizka:=0.9;
WHEN 'student' THEN znizka:=0.85;
WHEN 'emeryt/rencista' THEN znizka:=0.75;
else cena:=oblicz_cene(idkursu);
END CASE;
select count(id_kursu) into iloscKursow from kurskursanci where id_kursanta=idkursanta;
if (iloscKursow >= 3) THEN
znizka:=znizka-0.1;
elsif (iloscKursow >= 1) THEN
znizka:=znizka-0.05;
END if;
cena:=oblicz_cene(idkursu)*znizka;
RETURN cena;
end;

```

Powyższa funkcja oblicza cenę po zniżce dla danego kursu i kursanta.

Cena brutto liczona jest w innej funkcji, która jest tutaj wywoływana. Zniżka wynosi kolejno 10% jeżeli kursant jest uczniem, 15% jeżeli jest studentem, lub 25% jeżeli jest emerytem/rencistą. Dodatkowa zniżka przysługuje jeżeli kursant uczęszczał już na jakieś kursy. Jeżeli skorzystał z jednego kursu, przysługuje mu dodatkowa 5% zniżka, jeżeli z 3 lub więcej to dodatkowe 10%.

```

create or replace FUNCTION          POLICZ_WOLNE_MIEJSCA2 return table_dwie_zmienne as wynik table_dwie_zmienne;
cursor moj_kursor is select nazwa_kursu,id_kursu,policz_wolne_miejsca(id_kursu)as miejsca from kurs;
zmiana_kursora NUMBER;
nazwa_kursu varchar(255);
ilosc_miejsc number(8,2);
BEGIN
wynik:=table_dwie_zmienne();
for zmiana_kursora in moj_kursor loop
nazwa_kursu :=zmiana_kursora.nazwa_kursu;
ilosc_miejsc :=zmiana_kursora.miejsca;

wynik.extend;
wynik(wynik.count):=obiekt_dwie_zmienne(ilosc_miejsc,nazwa_kursu);
end loop;

RETURN wynik;
end;

create or replace type obiekt_dwie_zmienne as object(
tabela_1 number(8,2),
tabela_2 varchar (255)
);

create or replace type table_dwie_zmienne as table of obiekt_dwie_zmienne;

```

Powyższa funkcja zwraca kilka rekordów, są to nazwy kursów oraz ilość wolnych miejsc na nie. Efekt :

```
select * from table(policz_wolne_miejsca2)
```

	TABELA_1	TABELA_2
1	33	Francuski użytkowy
2	54	Angielski zaawansowany
3	39	Francuski na poziomie
4	40	Angielski na codzień