**Kamil Filar**,

**Karol Bury**

*Informatyka II, lab 1*

27.05.2020r.

**Biblioteka**

***Bazy danych II***

**Oracle(PL/SQL), Java**

**Spis treści:**

Spis treści …………………………………………………………………………………………………….. 2

Diagram ERD …………………………………………………………………………………………………. 3

Opis i cel projektu …………………………………………………………………………………………. 4

Interfejs wraz z funkcjonalnościami ………………………………………………………………. 5

Połączenie z bazą danych ……………………………………………………………………..5

Logowanie ………………………………………………………………………………………….. 6

Główny panel ……………………………………………………………………………………… 7

Wyświetlanie danych w obiektach jTable ……………………………………………. 8

Baza książek ………………………………………………………………………………………… 9

Moje wypożyczenia ……………………………………………………………………………. 11

Panel rejestracyjny …………………………………………………………………………….. 12

Moje konto (użytkownik) …………………………………………………………………… 15

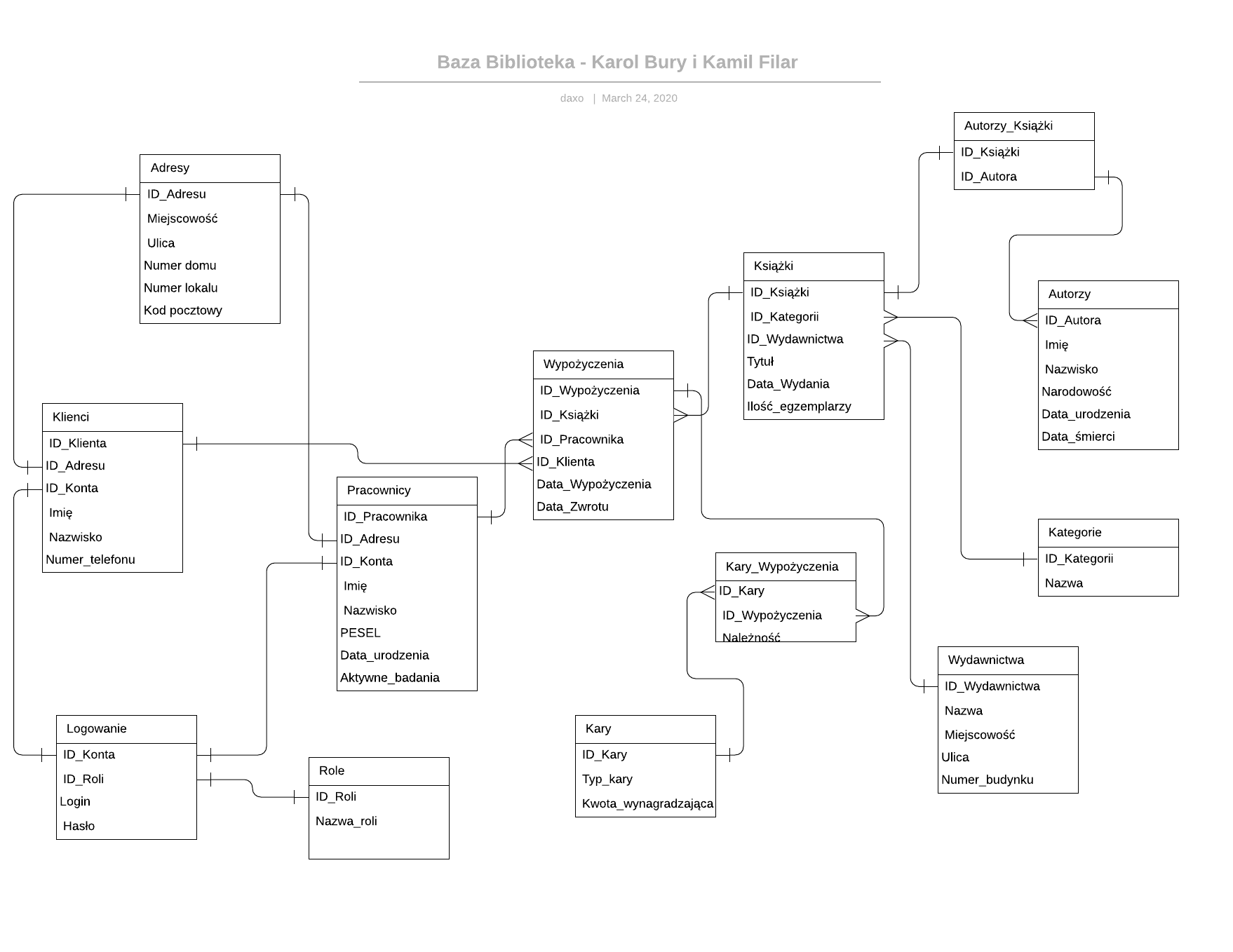
Moje konto (administrator) ……………………………………………………………….. 17

Panel administratora wraz z funkcjonalnościami ……………………………….. 18

Funkcja automatycznego nadawania kary za brak zwrotu książki ………. 24

Wymagania do korzystania z aplikacji ………………………………………………………….. 25

**Diagram ERD:**



**Opis i cel projektu:**

Projekt został stworzony w celu zarządzania i prowadzenia biblioteki. Program służy do administrowania bazy biblioteki przez pracownika. W projekcie został uwzględniony również interfejs dla klienta. Program jest oparty o system zarządzania relacyjną bazą danych Oracle, która zawiera trzynaście encji powiązanych ze sobą.

Aplikacja została stworzona w języku Java przy użyciu kompilatora NetBeans IDE 8.2 z wykorzystaniem technologii Java Swing oraz biblioteki ojdbc8.jar. **Projekt składa się z 46 różnych operacji wykonywanych w bazie danych za pomocą PL/SQL.** Oferuje: połączenie z bazą danych, wyświetlanie autorów, wydawnictw, regulaminu oraz bazy książek, także panel rejestracyjny (dla klientów) oraz panel logowania. Panel administratora zawiera podgląd na całą bazę danych i umożliwia interakcje z nią (poprzez interakcję mamy na myśli dodawanie książek, aktualizowanie ich, a także autorów i wydawnictw). Dla klienta został przygotowany osobny interfejs, klient może dokonać rezerwacji książki, sprawdzić stan swoich wypożyczeń, a także przeglądnąć regulamin i jakie wydawnictwa, autorzy figurują w bazie biblioteki.

Baza danych została wykonana przy użyciu Oracle Database Express Edition (XE) Release 18.4.0.0.0 (18c) . Zawiera trzynaście encji oraz wykorzystuje: funkcje, procedury, kursory, rekordy, sekwencje oraz tablice.

**Interfejs aplikacji wraz z funkcjonalnościami:**

Interfejs dzieli się na XXX formatek, które służą do: logowania, wyświetlenia autorów, bazy książek, wydawnictw oraz regulaminu i listy wypożyczeń, sterowania GUI, rejestracji (która składa się z trzech formatek), a także formatka do edycji konta jak i panel administratora, który służy do zarządzania biblioteką.

**Połączenie z bazą danych:**

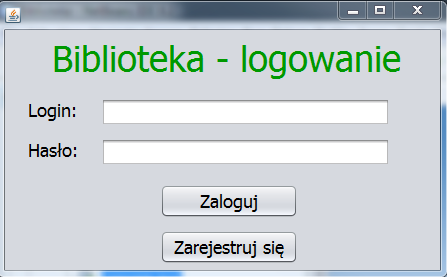
Do połączenia z bazą danych oracle wykorzystaliśmy klasę DbAccess.java, która wygląda następująco:



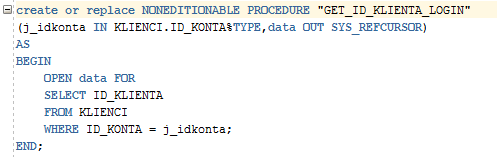
W formatkach wykorzystywaliśmy odwołanie do tej klasy, w celu łączenia się z bazą danych.

**Logowanie:**

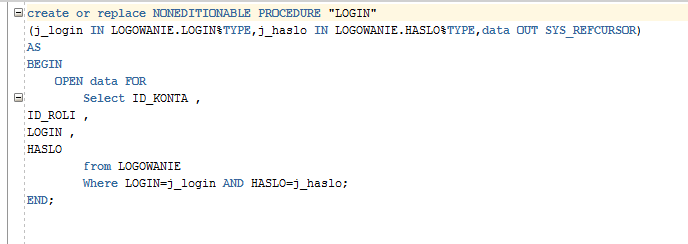
Po uruchomieniu aplikacji pojawi się okno logowania, w którym musimy podać nasze dane w dwóch polach do tego przeznaczonych, w momencie podania błędnych danych aplikacja poinformuje nas zwracając odpowiedni komunikat. W procesie logowania, dzięki pobraniu odpowiednich parametrów dzięki klasie User\_login aplikacja odróżni czy logujemy się jako klient, czy jako administrator dzięki czemu zostanie wyświetlony odpowiedni panel startowy.



Do logowania służy procedura LOGIN oraz GET\_ID\_KLIENTA\_LOGIN:



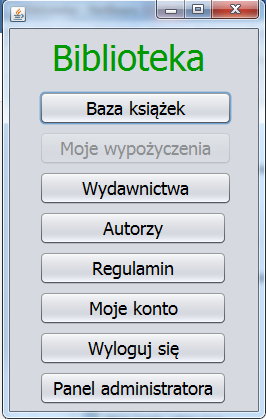
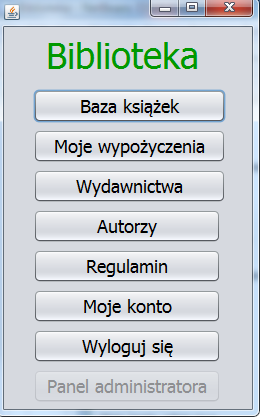
Procedura GET\_ID\_KLIENTA\_LOGIN



Procedura LOGIN

**Główny panel:**

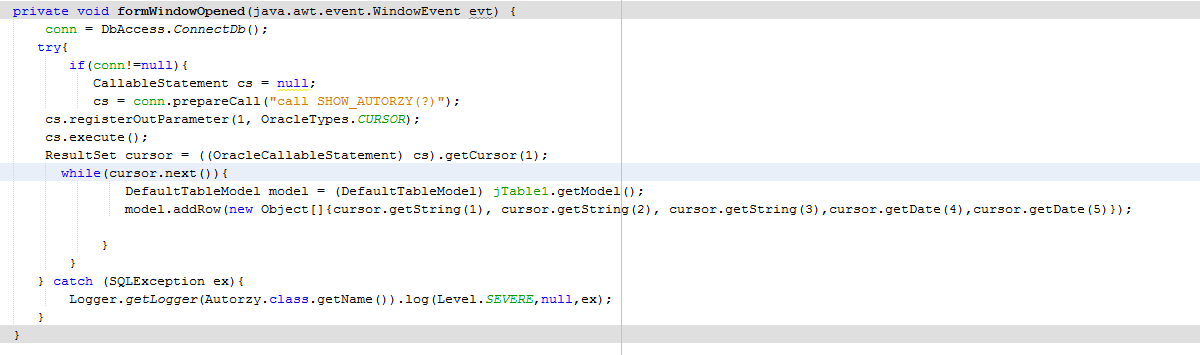
Po poprawnym zalogowaniu, zostajemy przekierowani do głównego panelu, który służy do nawigacji po aplikacji.

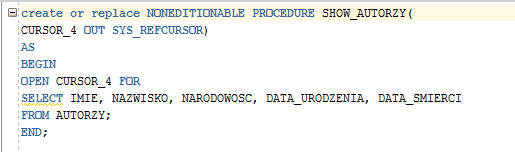
Widok dla administratora Widok dla użytkownika

**Wyświetlanie danych w obiektach jTable (na przykładzie formatki „Autorzy”):**

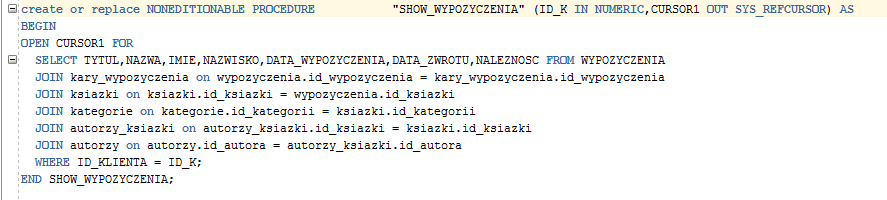
Do wyświetlenia danych w tabelkach użyliśmy następującej składni:



Dzięki zastosowaniu procedury SHOW\_AUTORZY za pomocą PL/SQL przekazaliśmy wszystkie wyniki do obiektu jTable. Wszystkie procedury rozpoczynające się od słowa SHOW (jest ich 10) służą do wypisania danych w tabelach. Przykłady:



Procedura SHOW\_AUTORZY

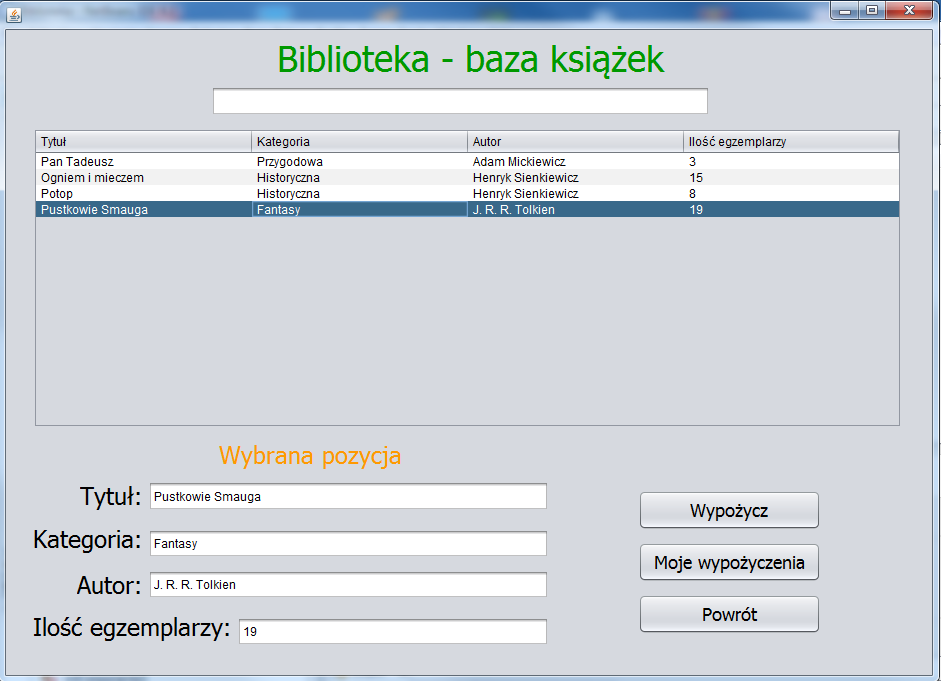


Procedura SHOW\_WYPOZYCZENIA

**Baza książek:**

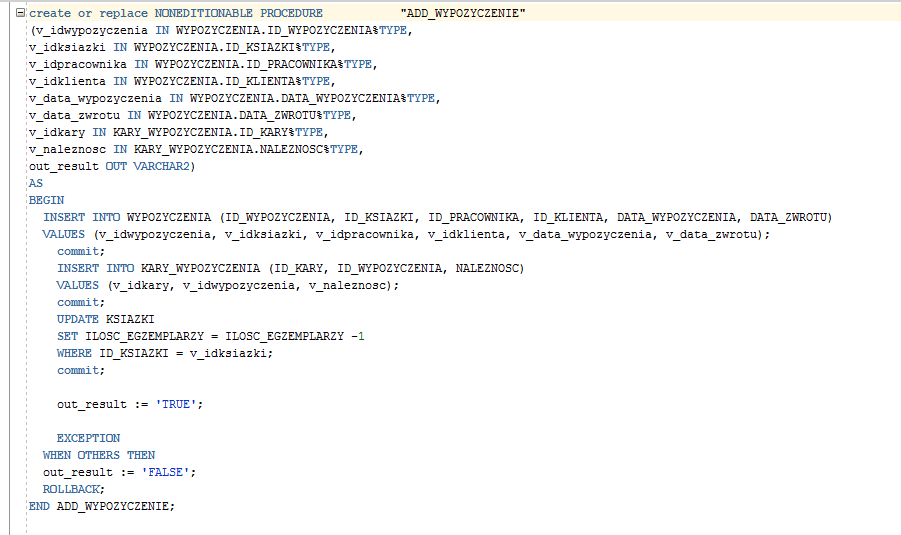
Jest to miejsce gdzie są wypisane wszystkie pozycje z biblioteki.

Tutaj użytkownik może wypożyczyć książkę po kliknięciu na wybraną przez siebie pozycję w tabeli (może ją również wyszukać w górnym TextFieldzie).



Widok dla użytkownika

Do dodania wypożyczenia służy procedura ADD\_WYPOZYCZENIE (jest ona również używana przez administratora w momencie dodawania wypożyczenia ręcznie).



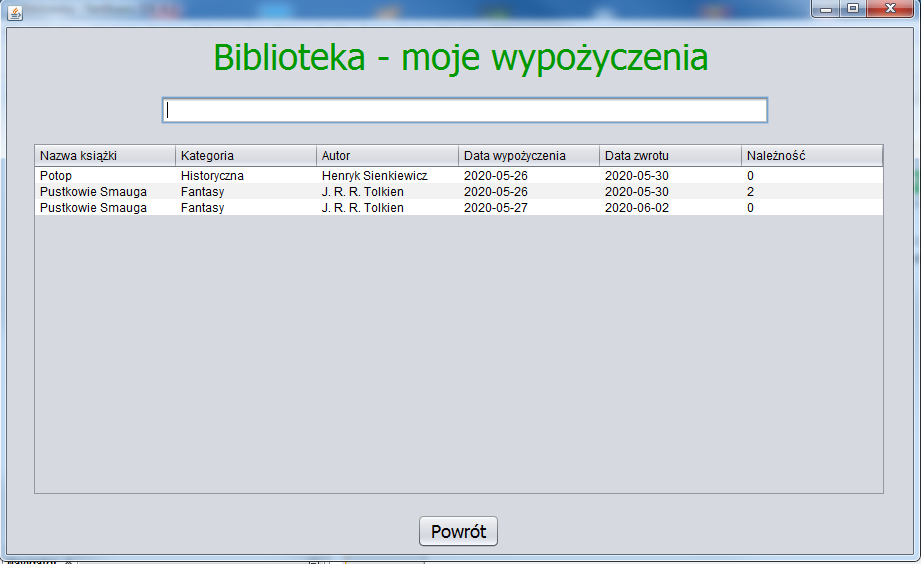
Procedura ADD\_WYPOZYCZENIE



Widok dla administratora

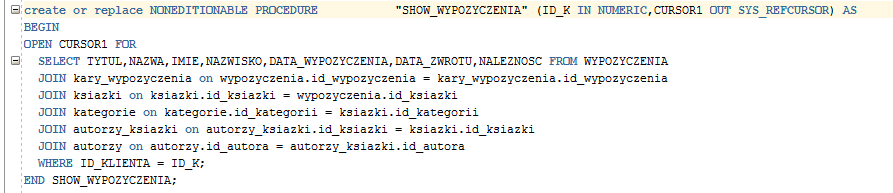
Pracownik aby wypożyczyć książkę musi posiadać również konto użytkownika.

**Moje wypożyczenia:**



Moje wypożyczenia

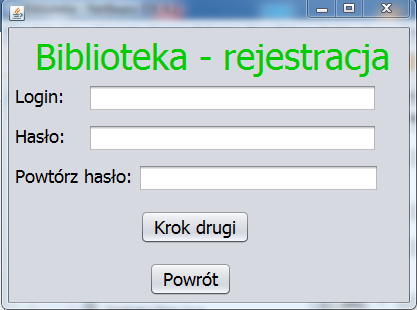
Do wypisania wypożyczeń użyliśmy procedury SHOW\_WYPOZYCZENIA, która wypisuje dane dla zalogowanego użytkownika (rozpoznaje jakie konto jest aktywne poprzez sprawdzenie ID\_KLIENTA



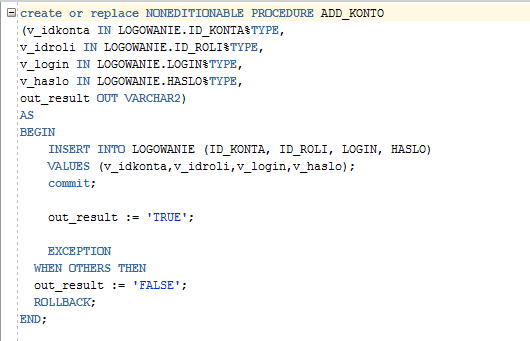
**Panel rejestracyjny:**

Plik Panel\_rejestracyjny.java

Zawiera pierwszy krok w celu utworzenia konta, w tym miejscu w aplikacji tworzymy nowe konto, gdzie użytkownik podaje swój login i hasło.

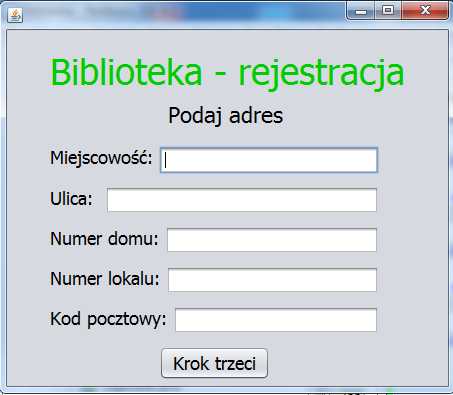


Do dodania danych do tabeli Logowanie użyliśmy procedury ADD\_KONTO:

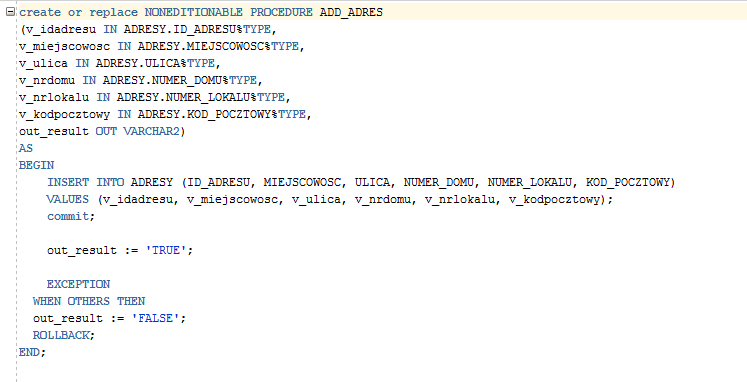


Plik Panel\_rejestracyjny2.java

Zawiera drugi krok w celu utworzenia konta, w tym miejscu w aplikacji podajemy adres użytkownika (miejscowość, ulica, numer domu, numer lokalu i kod pocztowy).

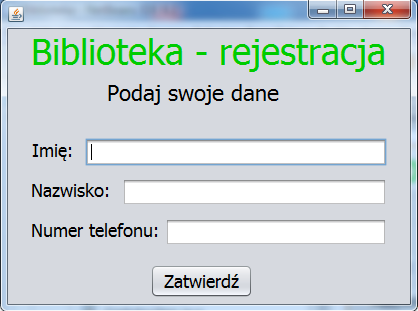


Do dodania danych do tabeli Logowanie użyliśmy funkcji ADD\_ADRES:

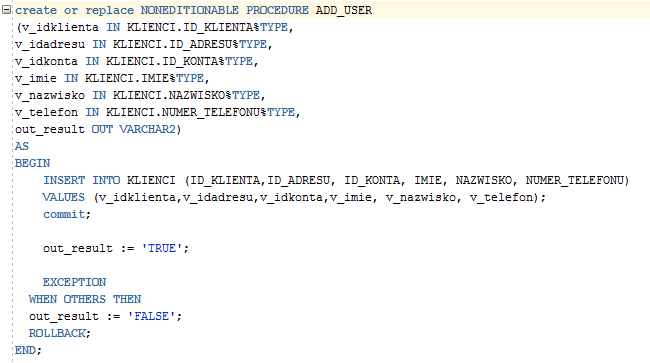


Plik Panel\_rejestracyjny3.java

Zawiera drugi trzeci, ostatni krok w celu utworzenia konta, w tym miejscu w aplikacji podajemy dane użytkownika (imię, nazwisko oraz numer telefonu).



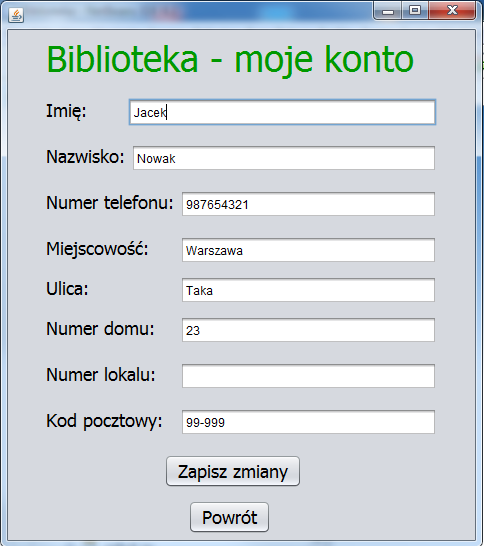
Do dodania danych do tabeli Logowanie użyliśmy funkcji ADD\_USER:



Do automatycznej inkrementacji numeru ID konta, klienta i adresu użyliśmy sekwencji IDKONTA, IDKKLIENTA, IDADRESU:

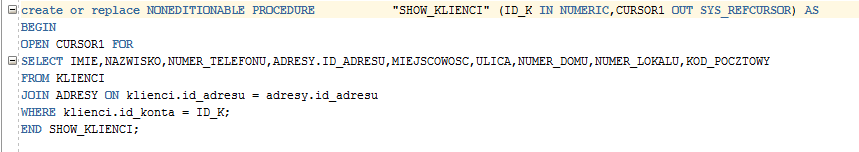
https://media.discordapp.net/attachments/689800346004095003/712938134144286801/unknown.png

**Moje konto (użytkownik):**

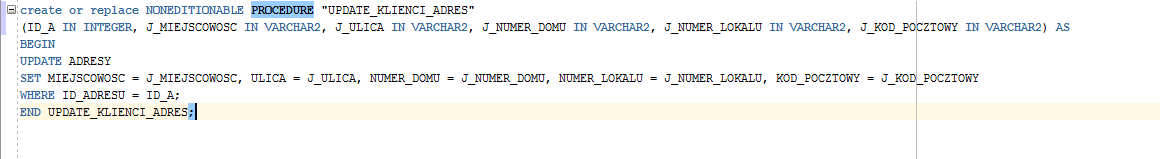


Widok dla użytkownika

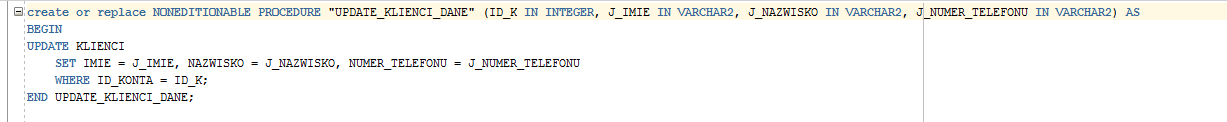
W tej formatce jest podgląd na konto użytkownika, użytkownik może dokonać zmian na swoim koncie. Do wyświetlenia danych użyliśmy procedury SHOW\_KLIENCI:



Do aktualizacji danych służy procedura UPDATE\_KLIENCI\_ADRES, UPDATE\_KLIECI\_DANE:

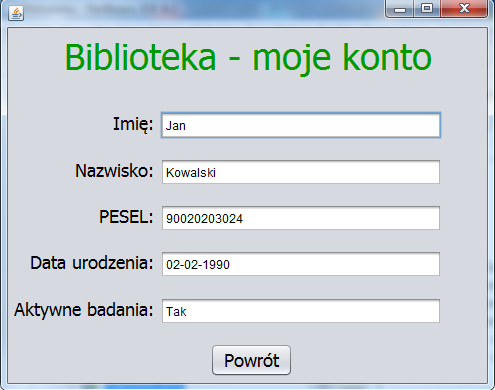


UPDATE\_KLIENCI\_ADRES



UPDATE\_KLIENCI\_DANE

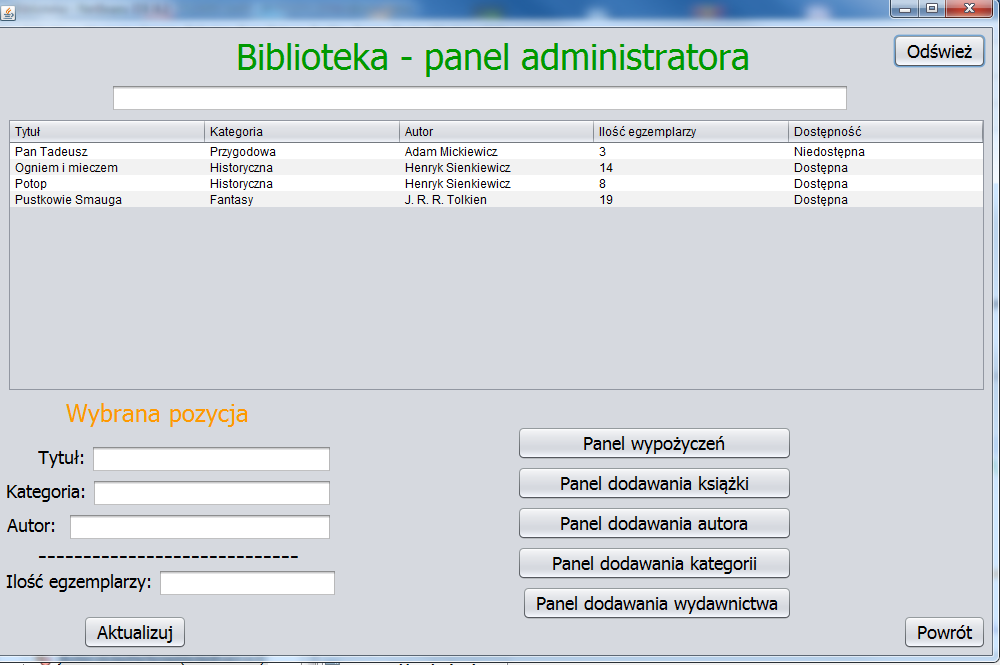
**Moje konto (administrator):**



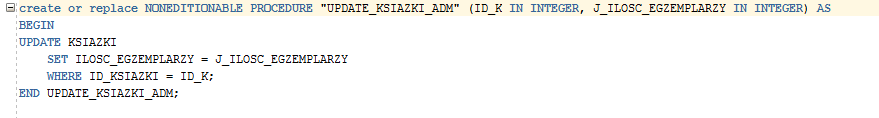
Widok dla administratora

W tej formatce jest podgląd na konto administratora.

**Panel administratora:**

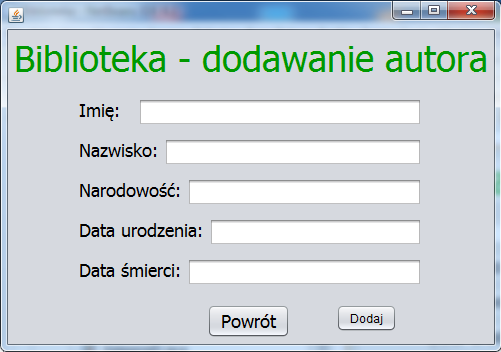


Panel administratora służy do aktualizacji ilości egzemplarzy książek i nawigacji do odpowiednich panelów. Do aktualizacji ilości egzemplarzy wykorzystywana jest procedura UPDATE\_KSIAZKI\_ADM:

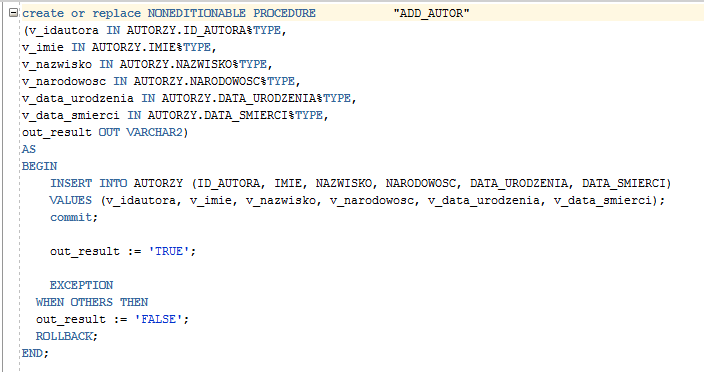


**Panel administratora (dodawanie autora):**

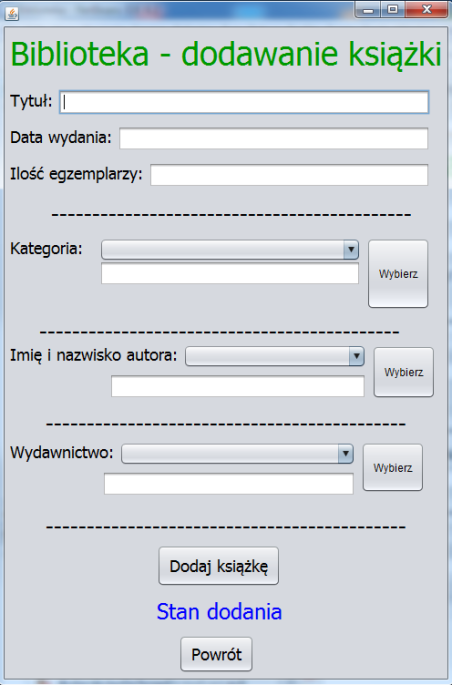
Panele z dodawaniem autora, kategorii oraz wydawnictwa służy do dodawania rekordów do bazy danych. Przykład działania na podstawie „Panel dodawania autorów”:



W TextField-ach administrator podaje odpowiednie dane. Do dodania danych używamy procedury ADD\_AUTOR, która wygląda następująco:

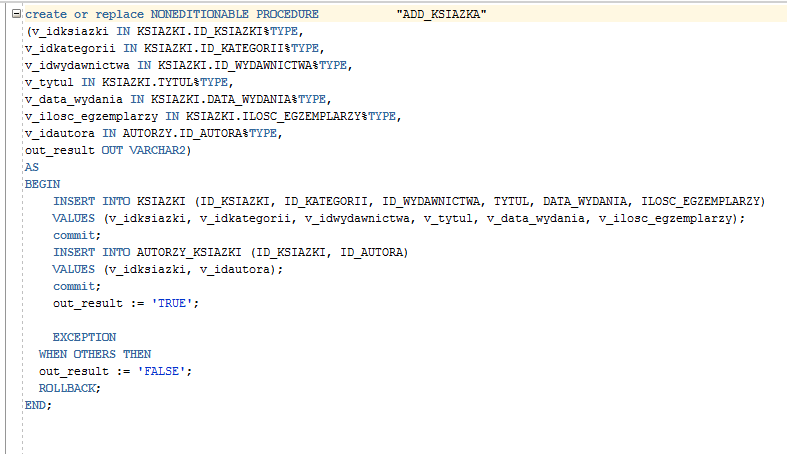


**Panel administratora (dodawanie książki):**

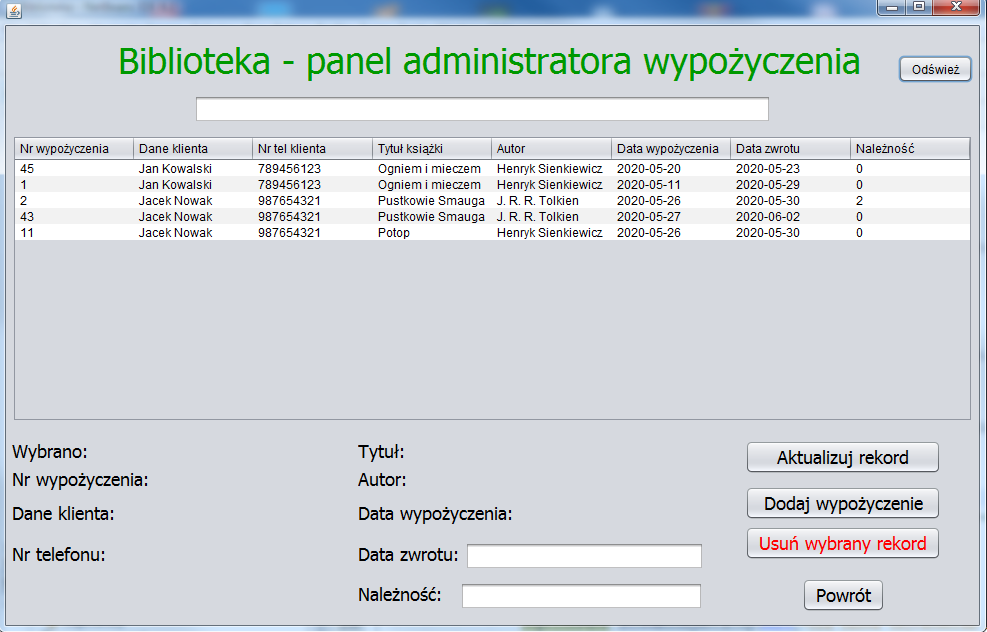


Widok panelu dodawania książki

Do dodania rekordu używamy bardziej intuicyjnej formatki, w której administrator wybiera odpowiednie dane. Do dodania do bazy danych książki użyliśmy procedury ADD\_KSIAZKA

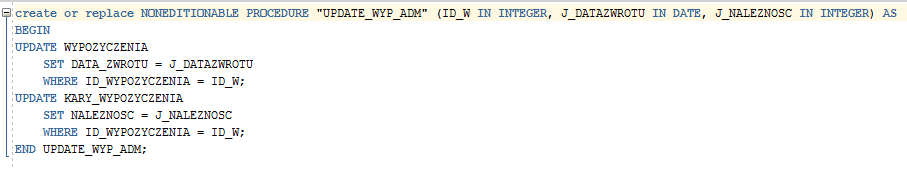


**Panel wypożyczeń:**

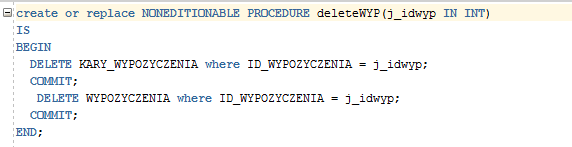


W panelu wypożyczeń administrator po kliknięciu na wybrany rekord ma możliwość aktualizacji lub usunięcia wybranego wiersza (np. przedłużenia terminu, lub wpisania kwoty w momencie, kiedy coś stało się z książką).

Do aktualizacji służy procedura UPDATE\_WYP\_ADM:



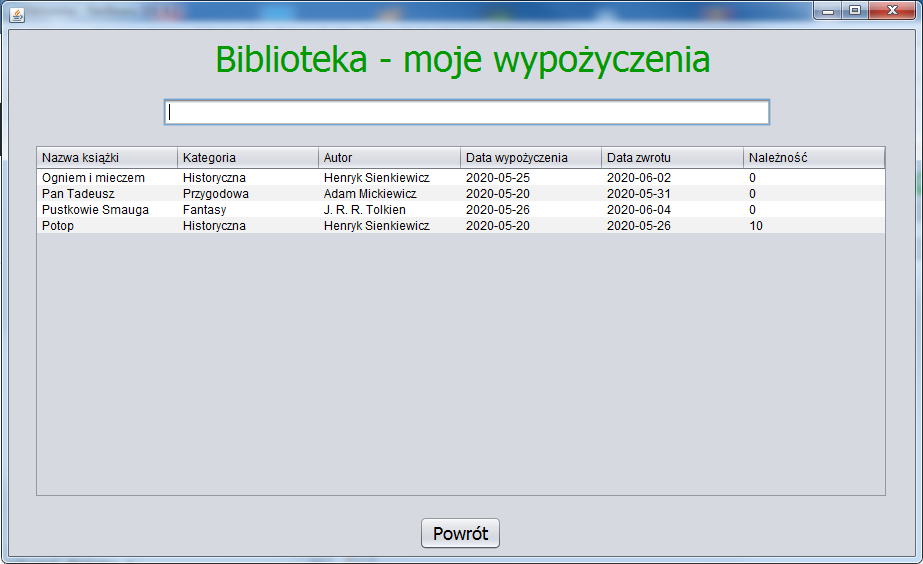
Do usuwania służy procedura deleteWYP:



Operacja dodawania wypożyczenia odbywa się za pomocą tej samej procedury co w przypadku zwykłego użytkownika (opisana na str. X). Widok dla dodawania wypożyczenia przez administratora:

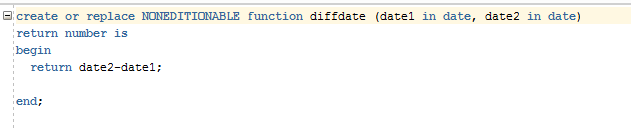


**Moje wypożyczenia (kolumna „należność”):**

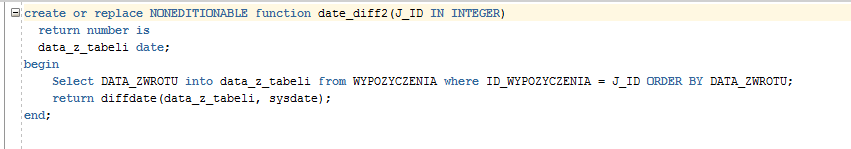


Do wyświetlania należności wykorzystaliśmy jednocześnie trzy funkcje: DATA\_DIFF2, DIFFDATE, UPDATE\_NALEZNOSC3:

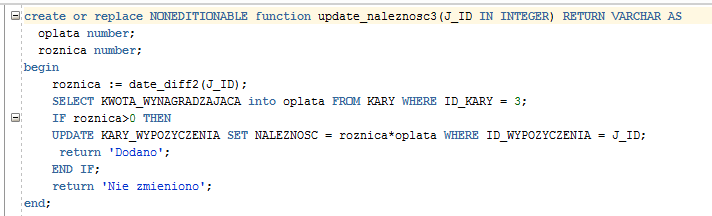
DIFFDATE – funkcja zwracająca różnicę dni pomiędzy datą wypożyczenia, a datą zwrotu



DATE\_DIFF2 – funkcja sprawdza czy data zwrotu nie jest większa od daty dzisiejszej



UPDATE\_NALEZNOSC3 – funkcja która uaktualnia dane w bazie dla rekordów, w których termin zwrotu jest przekroczony



**Wymagania do korzystania z aplikacji:**

Aby uruchomić aplikację na samym początku należy włączyć formatkę „Logowanie” (powinna się sama uruchomić).

**Dane logowania** (konto administratora):

Login: **admin**

Hasło: **admin**

**Dane logowania** (przykładowe konto użytkownika):

Login: **student**

Hasło: **student**

**Załączniki:**

Folder **docs:**

* **Diagram ERD.png** (zawiera schemat ERD bazy danych)
* **Funkcjonalność\_bazy\_biblioteka.docx** (zawiera opis funkcjonalnośći)

Folder **gui:**

* Biblioteka (folder zawiera aplikację)

Folder **pass**:

* **pass.txt** (zawiera hasła do kont używanych w aplikacji)

Folder **sql:**

* **biblioteka.sql** (plik z bazą danych)