Dokumentacja projektu  
„Baza usług telekomunikacyjnych”



|  |  |
| --- | --- |
|  | Autor:  Paweł Trojnar |

1. Opis celu pracy

Zadaniem, jakie zostało postawione na etapie konsultacji było zaprojektowanie bazy danych sprowadzonej do co najmniej drugiej postaci normalnej. Praca projektowa obejmuje głównie cześć bazodanową, jednak do jej obsługi została wykonana również aplikacja ułatwiająca jej obsługę. Pozwala ona na takie operacje jak:

- wprowadzanie rekordów do bazy,

- usuwanie rekordów,

- wyświetlanie rekordów w sposób przyjazny dla użytkownika.

2. Technologie wykorzystane w projekcie

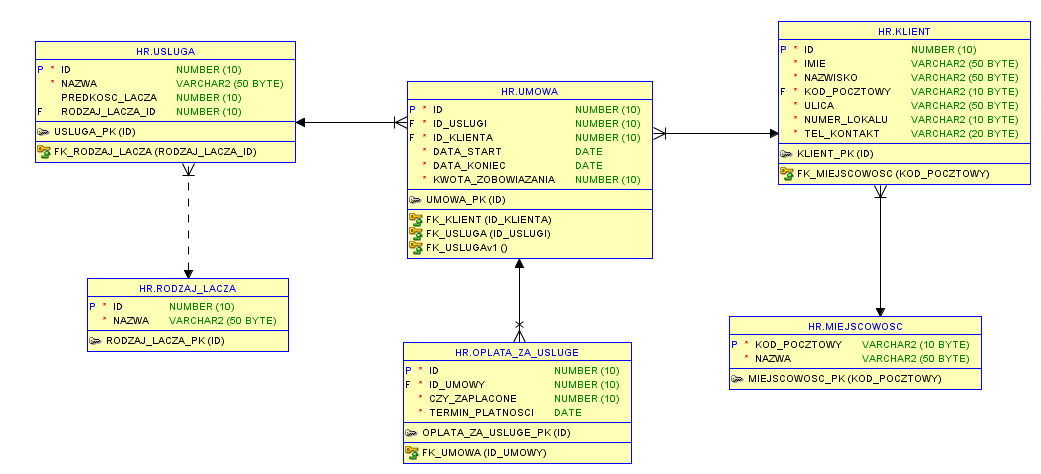
Aplikacja została wykonana w języku Java.

Interfejs graficzny został zaprojektowany przy pomocy oprogramowania SceneBuilder. Obsługa interfejsu jest zrealizowana za pomocą biblioteki JavaFx i plików fxml’owych. Baza danych to Oracle 11g2 Express Edition, jest ona dostępna za darmo nawet do użytku komercyjnego. Ponad to wersja ta ma ograniczone zużycie zasobów sprzętowych platformy na jakiej jest użytkowana co w naszym przypadku miało znaczenie, gdyż jednostka na której baza funkcjonowała nie posiada obszernych zasobów pamięci. Zarządzanie bazą odbywa się za pomocą sterownika OJDBC 6, wersja wybrana na podstawie deklaracji zgodności dostawcy sterownika.

3.Baza danych - opis

Baza danych posiada 5 tabel. Jedna z nich pełni funkcję tabeli łączącej, która posiada dodatkowe pola. W połączeniach występują związki obligatoryjne oraz nieobligatoryjne. Dane w tabelach przeszły proces normalizacji, tak aby nie występowały żadne anomalie np. redundancja danych(nadmiar informacji). Do dodawania oraz usuwania rekordów zostały użyte procedury pogrupowane według funkcjonalności do odpowiednich paczek. Każda z paczek usuń, oraz dodaj zwraca informację zwrotną o błędach w wykonywanych operacjach. Kontrola poprawności danych odbywa się z wykorzystaniem wyrażeń regularnych oraz predykatu „is not null”.

Przy usuwaniu wykorzystałem zliczanie wierszy z ograniczeniem do jednego rekordu tzn. jeśli w bazie istnieje rekord powiązany, to procedura nie pozwoli na jego skasowanie odsyłając stosowny komunikat do aplikacji. Z tej procedury wyłączona jest tabela umowa. Ponad to w bazie ustawione jest podejście standardowe do kasowania danych, co gwarantuje zabezpieczenie struktury bazy nawet jeśli coś nie jest przewidziane w napisanych procedurach. Wszystkie numeryczne klucze główne są inkrementowane przy pomocy sekwencji oraz triggerów. Funkcja opłaty\_dla\_umowy\_o\_id nie jest wykorzystywana w widoku aplikacji. Została stworzona w celu przedstawienia opanowania przeze mnie umiejętności korzystania z typów własnych, oraz typów tablicowych.



4. Opis składni PL/SQL

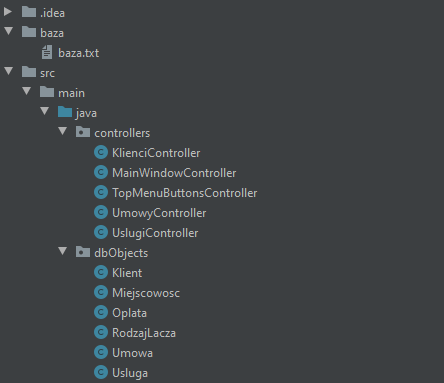
Ważną rolę w projekcie ma rozszerzenie zapytań poprzez składnię PL/SQL. W projekcie wykorzystałem sekwencje, wyzwalacze, pakiety, funkcje oraz procedury. Funkcjonalności te pozwoliły na zrezygnowanie z operowania na danych po stronie aplikacji. Użycie sql’a w aplikacji ogranicza się do wywołania procedury i odebrania rekordów w postaci resoult set’a. Podczas projektowania bazy oraz implementacji procedur warto nieco się nad tym zastanowić, ponieważ dobrze napisane procedury pozwolą na ich późniejszą edycję bez ingerencji w aplikację do obsługi bazy danych, pod warunkiem że nie będziemy chcieli zmieniać zwracanych typów, lub wyświetlanych wartości. Modułowość paczek pozwala na zachowanie przejrzystości w bazie.

|  |
| --- |
| create or replace TRIGGER DODAJ\_MIESIECZNE\_ZOBOWIAZANIA  AFTER INSERT ON UMOWA  FOR EACH ROW  BEGIN  DECLARE  record\_exists INTEGER;  newdate DATE :=:NEW.DATA\_START;  newdateend DATE := :NEW.data\_koniec;  newid NUMBER := :NEW.id;  BEGIN SELECT MONTHS\_BETWEEN (newdateend,newdate) INTO record\_exists   FROM DUAL;  IF record\_exists is not null THEN  FOR l\_counter IN 1..record\_exists  LOOP  INSERT INTO oplata\_za\_usluge(id\_umowy,termin\_platnosci)   VALUES(newid,ADD\_MONTHS(newdate, l\_counter));  END LOOP;    END IF;  END;  END; |

Wyzwalacz wykorzystany w projekcie

Powyżej prezentuję przykładowy kod prostego wyzwalacza. Triggery są blokami pl/sql’a które są wykonywane w trakcie określonych zdarzeń. Ten który został przedstawiony w tym przykładzie ma za zadanie wstawić do bazy unikalne rekordy odpowiadające płatności za usługi. Każda umowa, na podstawie daty startowej, oraz końcowej ma tworzone rekordy   
w tabeli opłata\_za\_usługę odpowiadającą ilości miesięcy pomiędzy podanymi okresami.

5.Opis aplikacji

******

Aplikacja została napisana w języku Java. Za interfejs graficzny odpowiada JavaFx.

W projekcie możemy zaobserwować następujące paczki:

* .controllers – formatki fxml,
* .dbObjects – encje bazy danych przełożone na klasy Javy,
* .dbUtils – narzędzia do wymiany informacji pomiędzy aplikacją, a bazą danych, rzutuje zwrócone wiersze z bazy danych na wcześniej przygotowane obiekty w Javie,
* .dialogUtils –komunikaty ostrzeżeń wyświetlane w aplikacji,
* Main – główna funkcja programu
* JDBCSettings.properties – plik przechowujący konfigurację bazy danych

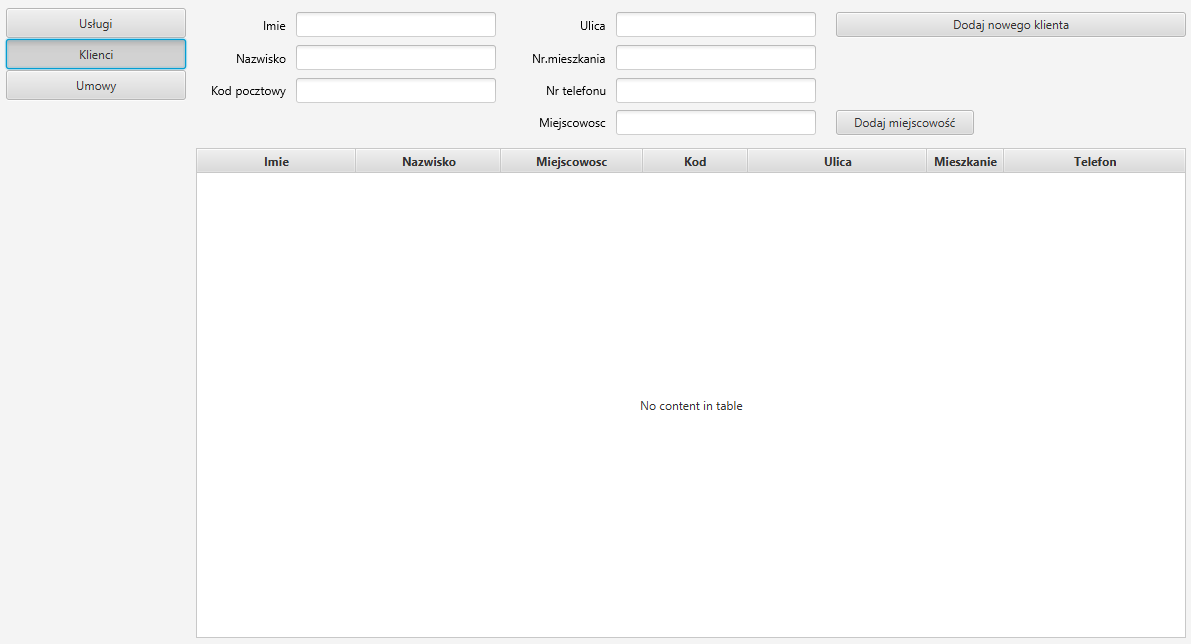
Ekran powitalny



Menu posiada 3 opcji do wyboru, natomiast na środku ekranu wypisany jest dowolny ustalony przez programistę tekst.

* Usługi
* Klienci
* Umowy

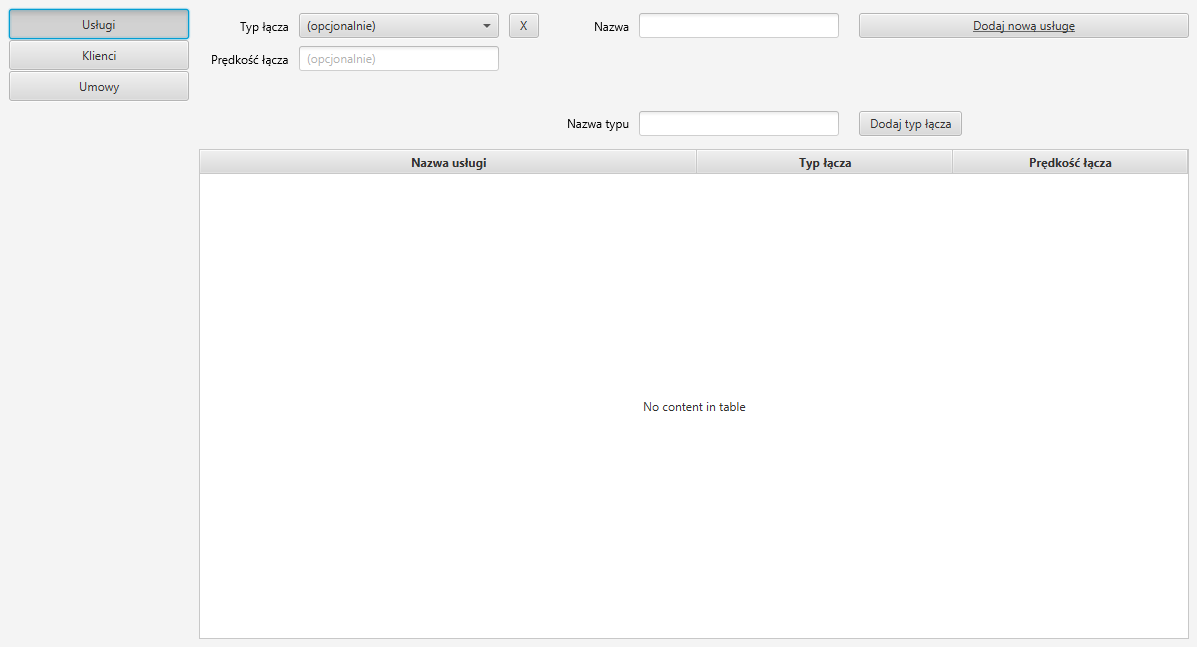
Klienci

******

Zakładka ta służy do gromadzenia informacji o klientach. W widoku znajduje się formularz dodawania klienta oraz tabela wyświetlająca informację w sposób czytelny dla operatora.

Formularz, który znajduje się w górnej części aplikacji pozwala na wprowadzanie klientów do bazy danych, brak danych w jednym z pól uniemożliwi wprowadzenie rekordu do bazy. Przycisk „dodaj miejscowość.” służy do dodania miejscowości do bazy danych, jeśli jeszcze jej nie posiadamy. W przypadku kiedy posiadamy miejscowość w bazie wystarczy wpisać kod miejscowości, a zostanie ona automatycznie przyporządkowana.

Usługi

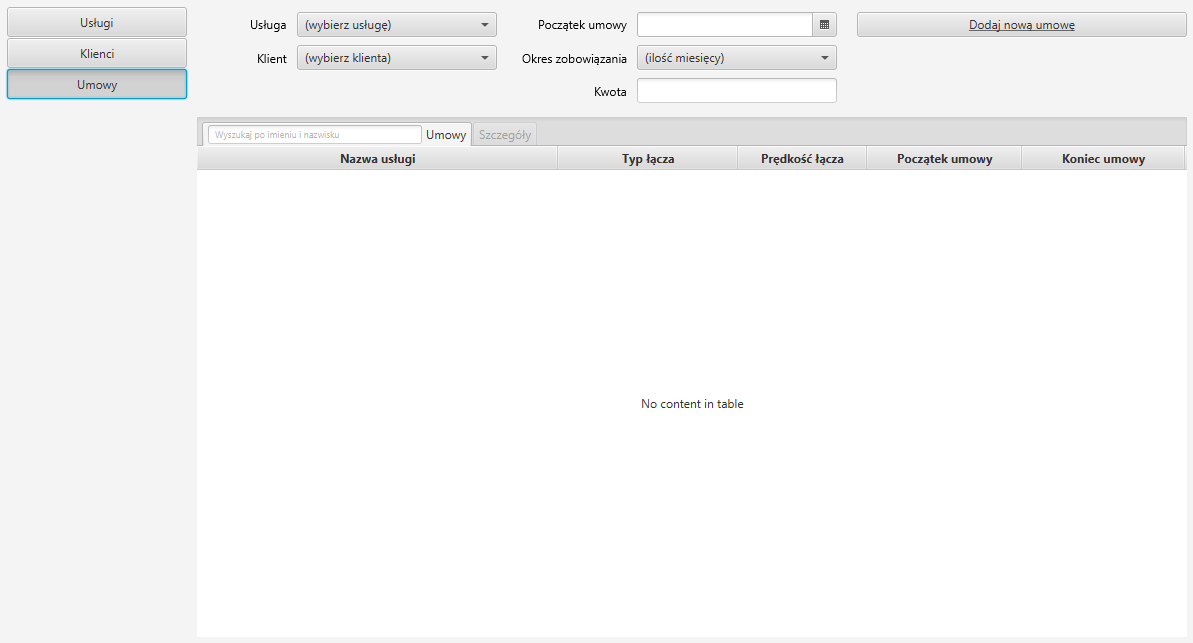


Ta zakładka służy do administracji dostarczanymi przez firmę usługami telekomunikacyjnymi. Pozwala ona na wprowadzenie nazw pakietów oferowanych przez dostawcę. Ponad to opcjonalnie przyjmuje typ oraz prędkość łącza. Jeśli chcemy podać typ lub prędkość łącza musimy wypełnić obie informacje.

Klikając prawym przyciskiem na wybraną usługę możemy dokonać jej usunięcia tylko   
w przypadku jeśli nie narusza ona integralności bazy.

Przycisk „X” pozwala na usunięcie typu łącza w przypadku gdy nie jest związany z żadną usługą.

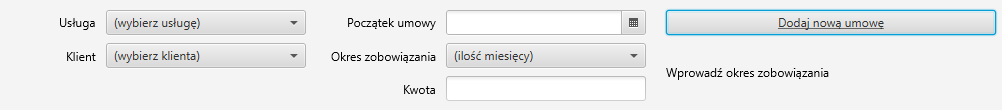
Umowy

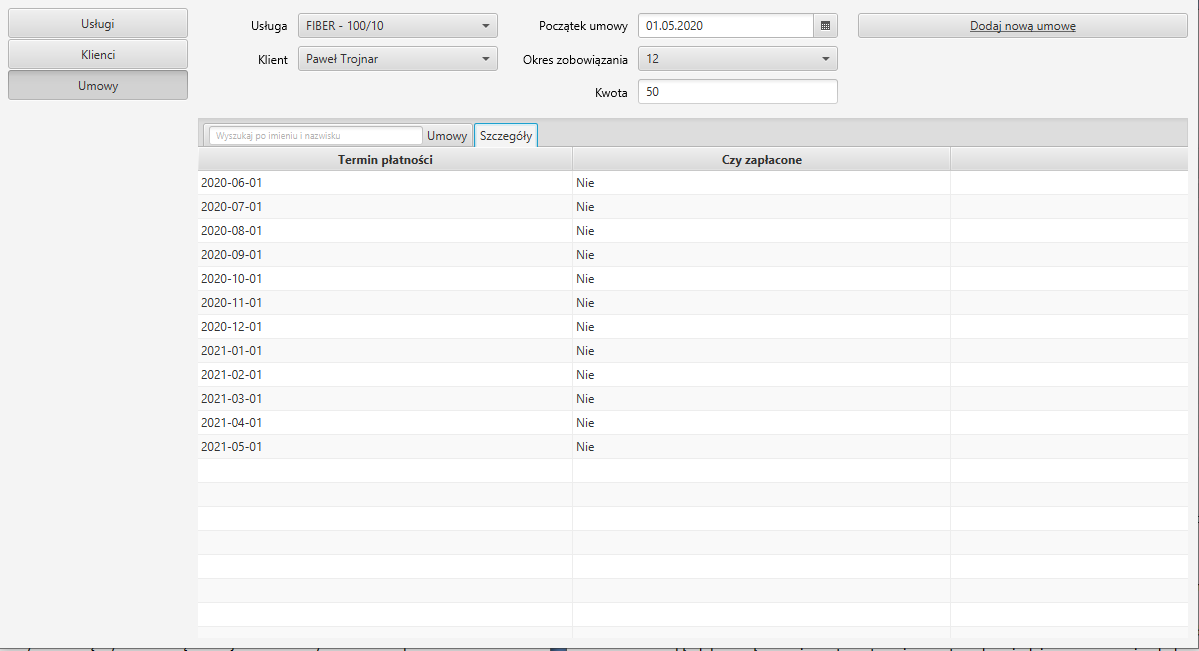


Jest to ostatnia, a zarazem najbardziej rozbudowana zakładka w aplikacji. Dane podobnie jak w poprzednich oknach są reprezentowane przez widok tabelaryczny. Menu kontekstowe pozwala na usunięcie umowy w dowolnym momencie. Nawet jeśli jej okres nie upłyną. Jest to spowodowane zastosowanym podejściem odnośnie usuwania rekordów z tej tabeli.

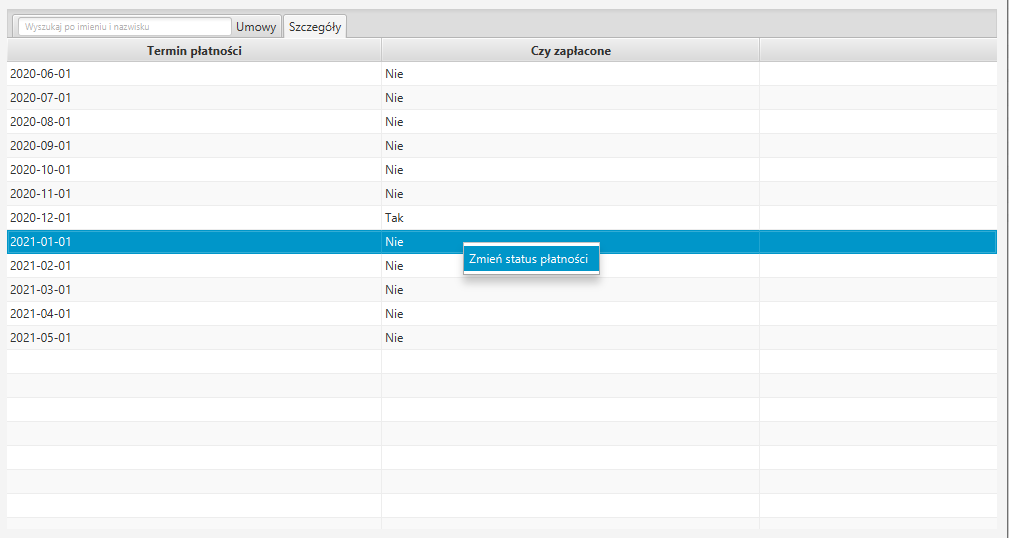
Zakładka Umowy posiada dynamiczną wyszukiwarkę. Filtruje ona umowy w bazie w oparciu o imię oraz nazwisko klienta. Dzięki temu operator programu może znaleźć umowy przyporządkowane do danego klienta z pośród wszystkich innych.

Formularz w górnej części aplikacji pozwala na dodanie nowej umowy. Brak jakiejkolwiek z wartości skutkuje zwróceniem komunikatu o błędzie.





Po wybraniu umowy poprzez kliknięcie odblokuje się zakładka szczegóły. Znajdują się w niej informacje o płatnościach za daną usługę. Prawy przycisk myszy wywołuje menu kontekstowe, które pozwala na zmianę statusu płatności.



5. Jak uruchomić aplikację?

Aplikacja została napisana przy pomocy środowiska InteliJ IDEA. Najprostszą formą uruchomienia jest zaimportowanie projektu do ww. aplikacji. Warto zaznaczyć że aplikacja została stworzona w oparciu o projekt Maven’a, a więc powinna bez problemu uruchomić się również w innych środowiskach programistycznych.

Dane do bazy danych:

Login: hr  
Hasło: hr

Przed pierwszym uruchomieniem programu należy wykonać skrypt init.sql dołączony do projektu np. przy pomocy narzędzia sqlDeveloper.