

Dokumentacja projektu wykonanego w ramach przedmiotu BAZY DANYCH I

Hotel

Wykonał Kamil Sudół

I. Projekt koncepcji, założenia

1. Zdefiniowanie tematu projektu:

Tematem projektu jest obsługa bazy danych dedykowanej sprawom hotelarskim. Dzięki napisanej aplikacji możliwe jest obsłużenie podstawowych działań związanych z branżą hotelarską, takich, jak możliwość dokonania rezerwacji danego pokoju na konkretny termin, możliwość opłaty bądź rezygnacji z rezerwacji.

2. Analiza wymagań użytkownika:

Użytkownik zalogowany jako klient powinien mieć możliwość:

- dokonania rezerwacji na pokój o określonej kategorii,
- dokonania rezerwacji na ten sam pokój w różnych terminach,
- wyboru dodatkowych usług do rezerwacji pokoju,
- podglądu informacji o pokojach podzielonych na kategorie,
- podglądu informacji o swoich rezerwacjach,
- opłacenia lub rezygnacji z rezerwacji.

Ponadto użytkownik zalogowany jako administrator hotelu poza możliwościami wymienionymi powyżej powinien mieć dodatkowo możliwość:

- podglądu wszystkich danych w bazie danych,
- zablokowania konkretnego użytkownika,
- dodania nowego pracownika do obsługi hotelu.

Dodatkowo:

- rezerwacja nie może zostać zaakceptowana, jeżeli termin zakwaterowania nastąpi szybciej, niż 7 dni od daty złożenia rezerwacji,
- rezerwacje przydzielane są na konkretny pokój,
- każdy gość może dokonywać wiele rezerwacji na wiele kategorii pokoiów,
- gościowi przydzielamy pokój nie kategorię,
- gość, który nie opłaci swojej rezerwacji do dnia zakwaterowania, zostaje automatycznie przeniesiony na czarną listę,
- gość, który dobrowolnie zrezygnuje z rezerwacji więcej, niż 5 razy, zostaje automatycznie przeniesiony na czarną listę.

3. Zaprojektowanie funkcji:

Użytkownik zalogowany jako Klient może dokonać rezerwacji na jeden konkretny pokój z wielu dostępnych o określonej kategorii, w konkretnym terminie (o ile wybrany pokój będzie dostępny w tymże terminie) na konkretną liczbę osób uwzględniając podział na dzieci i dorosłych, mając przy tym możliwość wyboru dodatkowych usług, takich, jak dostęp do basenu, kręgielni, bilarda czy siłowni. Ponadto klient ma możliwość podglądu swoich rezerwacji, wyświetlenia szczegółowych informacji każdej z nich oraz możliwość zdecydowania, czy daną rezerwację klient chce opłacić bądź zrezygnować z niej.

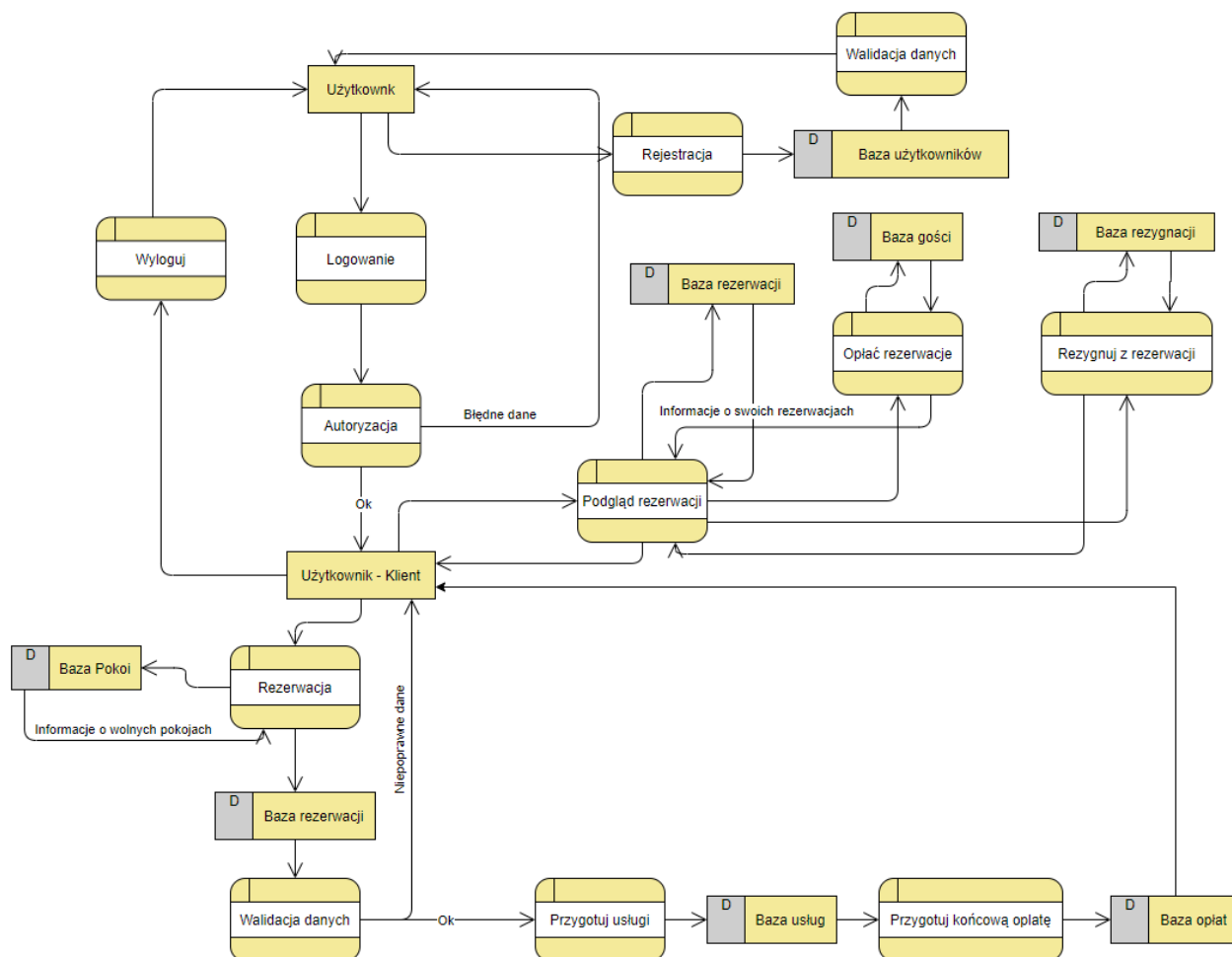
Użytkownik zalogowany jako HeadAdmin lub Pracownik poza możliwościami wymienionymi dla klientów ma udostępniony panel administratora. Na omawiany panel składa się możliwość dodania nowego pracownika do obsługi hotelu a także możliwość podglądu wszystkich raportów oraz tabel hotelowej bazy danych.

Dodatkowo baza danych ma funkcjonalność zapobiegania wprowadzania nieprawidłowych danych oraz możliwość zwalniania pokoi, jeżeli dany klient zrezygnował ze swojej rezerwacji.

II. Projekt diagramów (konceptualny)

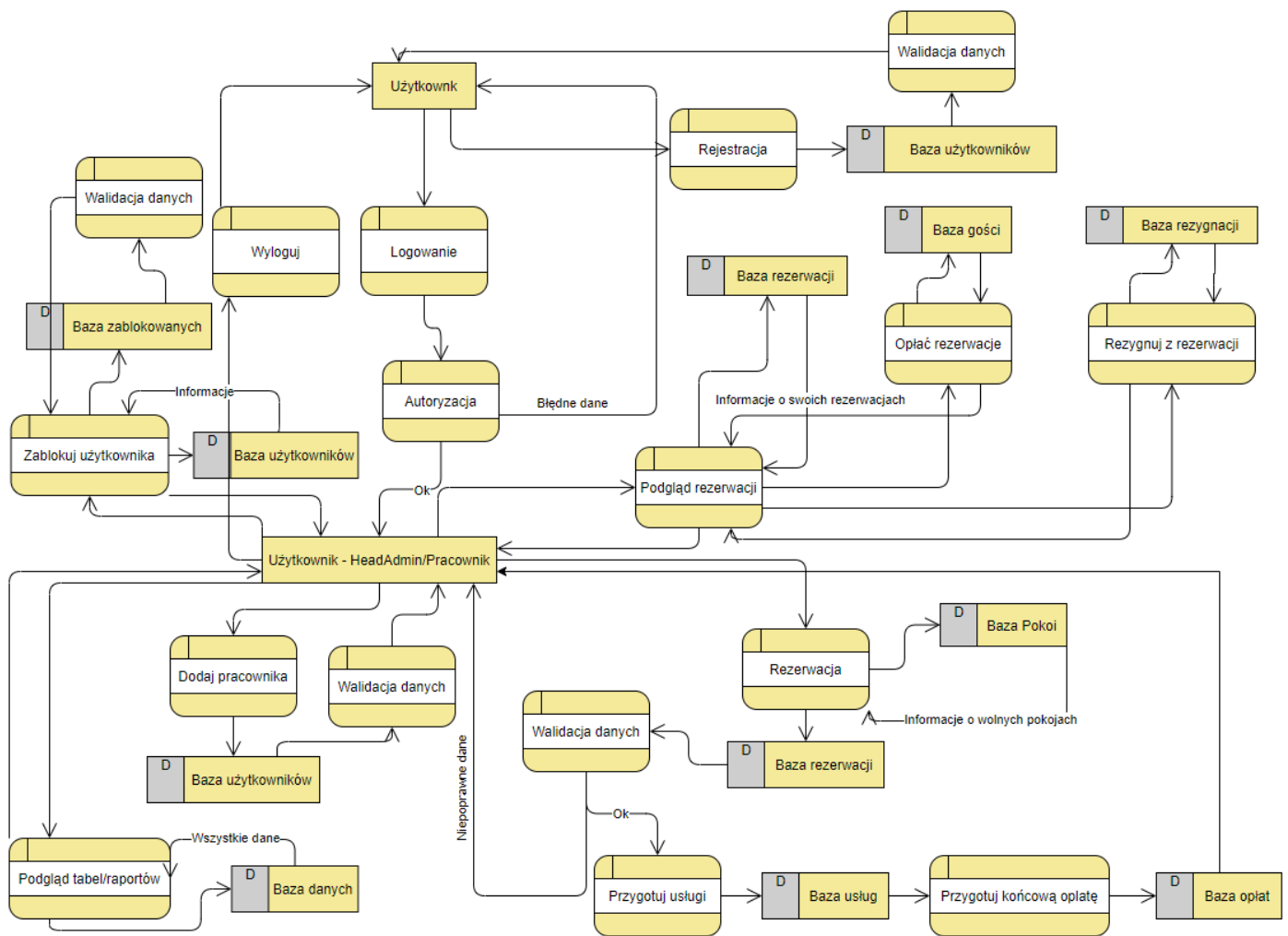
4. Budowa i analiza diagramu przepływu danych (DFD):

Diagram DFD dla użytkownika zalogowanego jako Klient:



Obrazek 1: Diagram DFD dla użytkownika zalogowanego jako Klient.

Diagram DFD dla użytkownika zalogowanego jako HeadAdmin lub Pracownik:



Obrazek 2: Diagram DFD dla użytkownika zalogowanego jako HeadAdmin lub Pracownik.

5. Zdefiniowanie encji (obiektów) oraz ich atrybutów:

Panel: *uzyt_id, login, haslo,*

Uzytkownik: *uzytkownik_id, imie, nazwisko, e_mail, numer, typ,*

Rezerwacje: *rezerwacja_id, uzytkownik_id, pokoj_id, data_rezerwacji, od_kiedy, do_kiedy, liczba_dzieci, liczba_doroslych,*

Dodatkowe_uslugi: *dodatkowe_uslugi_id, nazwa_uslugi, cena_od_osoby, rezerwacja_id,*

Oplata: *oplata_id, rezerwacja_id, status_czy_oplacone, kwota,*

Pokoj: *pokoj_id, numer_pokoju, pietro, liczba_miejsc, kategoria_id,*

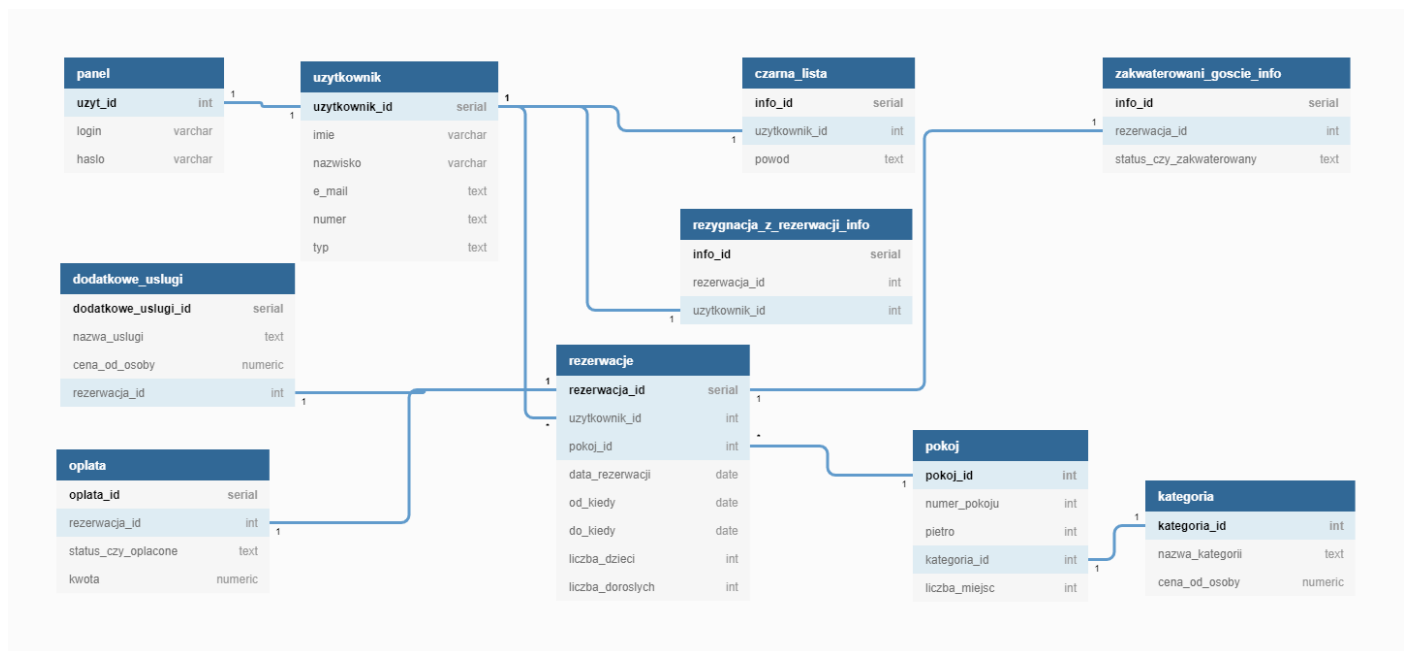
Kategoria: *kategoria_id, nazwa_kategorii, cena_od_osoby,*

Rezygnacja_z_rezerwacji_info: *info_id, rezerwacja_id, uzytkownik_id,*

Zakwaterowani_goscie_info: *info_id, rezerwacja_id, status_czy_zakwaterowany,*

Czarna_lista: *info_id, uzytkownik_id, powod.*

6. Zaprojektowanie relacji pomiędzy encjami:



Obrazek 3: Diagram ERD projektu.

III. Projekt logiczny

7. Projektowanie tabel, kluczy, indeksów:

- tabela **panel**:

- `uzyt_id` INT PRIMARY KEY – jednoznacznie identyfikuje id użytkownika, połączenie z tabelą `uzytkownik(uzytkownik_id)` - relacja 1:1,

- tabela **uzytkownik**:

- `uzytkownik_id` SERIAL PRIMARY KEY – jednoznacznie identyfikuje id użytkownika,

- tabela **rezerwacje**:

- `rezerwacja_id` SERIAL PRIMARY KEY – jednoznacznie identyfikuje id rezerwacji,
- `uzytkownik_id` INT klucz obcy do `uzytkownik(uzytkownik_id)` - relacja N:1,
- `pokoj_id` INT klucz obcy do `pokoj(pokoj_id)` - relacja N:1,

- tabela **dodatkowe_uslugi**:

- `dodatkowe_uslugi_id` SERIAL PRIMARY KEY – jednoznacznie identyfikuje id dodatkowej usługi,
- `rezerwacja_id` INT klucz obcy do `rezerwacje(rezerwacja_id)` - relacja 1:1,

- tabela **opлата**:

- `opлата_id` SERIAL PRIMARY KEY – jednoznacznie identyfikuje id opłaty,
- `rezerwacja_id` INT klucz obcy do `rezerwacje(rezerwacja_id)` - relacja 1:1,

- tabela ***pokoje***:

- *pokoj_id* INT PRIMARY KEY – jednoznacznie identyfikuje id pokoju,
- *kategoria_id* INT klucz obcy do *kategoria(kategoria_id)* - relacja 1:1

- tabela ***kategoria***:

- *kategoria_id* INT PRIMARY KEY – jednoznacznie identyfikuje id kategorii,

- tabela ***rezygnacja_z_rezerwacji_info***:

- *info_id* SERIAL PRIMARY KEY – jednoznacznie identyfikuje id informacji,
- *uzytkownik_id* INT klucz obcy do *uzytkownik(uzytkownik_id)* - relacja 1:1,

- tabela ***zakwaterowani_goscie_info***:

- *info_id* SERIAL PRIMARY KEY – jednoznacznie identyfikuje id informacji,
- *rezerwacja_id* INT klucz obcy do *rezerwacje(rezerwacja_id)* - relacja 1:1,

- tabela ***czarna_lista***:

- *info_id* SERIAL PRIMARY KEY – jednoznacznie identyfikuje id informacji,
- *uzytkownik_id* INT klucz obcy do *uzytkownik(uzytkownik_id)* - relacja 1:1,

8. Słowniki danych:

Schemat **projekt**:

- tabela ***panel***:

- *uzyt_id* INT > 0,
- *login* VARCHAR długość > 5,
- *haslo* VARCHAR długość > 5

- tabela ***uzytkownik***:

- *uzytkownik_id* SERIAL > 0,
- *imie* VARCHAR,
- *nazwisko* VARCHAR,
- *e_mail* TEXT,
- *numer* TEXT,
- *typ* TEXT

- tabela **rezerwacje**:

- rezerwacja_id SERIAL > 0,
- uzytkownik_id INT > 0,
- pokoj_id INT > 0,
- data_rezerwacji DATE,
- od_kiedy DATE,
- do_kiedy DATE,
- liczba_dzieci INT >= 0,
- liczba_doroslych INT >= 0

- tabela **dodatkowe_uslugi**:

- dodatkowe_uslugi_id SERIAL > 0,
- nazwa_uslugi TEXT,
- cena_od_osoby NUMERIC >= 0,
- rezerwacja_id INT > 0,

- tabela **opлата**:

- oplata_id SERIAL > 0,
- rezerwacja_id INT > 0,
- status_czy_oplacone TEXT,
- kwota NUMERIC >= 0

- tabela **pokoje**:

- pokoj_id INT > 0,
- numer_pokoju INT > 0,
- pietro INT >= 0,
- liczba_miejsc INT > 0,
- kategoria_id INT > 0

- tabela **kategoria**:

- kategoria_id INT > 0,
- nazwa_kategorii TEXT,
- cena_od_osoby NUMERIC >= 0

- tabela **rezygnacja_z_rezerwacji_info**:

- info_id SERIAL > 0,
- rezerwacja_id INT > 0,
- uzytkownik_id INT > 0

- tabela ***zakwaterowani_goscie_info***:

- *info_id* SERIAL > 0,
- *rezerwacja_id* INT > 0,
- *status_czy_zakwaterowany* TEXT

- tabela ***czarna_lista***:

- *info_id* SERIAL > 0,
- *uzytkownik_id* INT > 0,
- *powod* TEXT

9. Analiza zależności funkcyjnych i normalizacja tabel:

Pierwsza postać normalna (1NF):

W każdej tabeli jest jednoznacznie definiującą encję klucz główny. Wszystkie dane są atomowe.

Druga postać normalna (2NF):

Każda tabela przechowuje dane dotyczące tylko konkretnej klasy obiektów. Każdy niekluczowy atrybut jest nieredukowalnie zależny od klucza głównego.

Trzecia postać normalna (3NF):

Wszystkie niekluczowe atrybuty (niewchodzące w skład klucza głównego) są wzajemnie niezależne oraz nieredukowalnie zależne od klucza głównego.

10. Zaprojektowanie operacji na danych:

Funkcje (dostępne w plikach SQL/first_funkcje.sql oraz SQL/last_funkcje.sql):

Numeric_check – pomocnicza funkcja wykorzystywana przy walidacji danych rejestracyjnych sprawdzająca czy podany numer telefonu składa się z samych cyfr.

ZalogujCheck – funkcja sprawdzająca, czy podany login oraz hasło się zgadzają. W przypadku powodzenia zwracany jest numer id użytkownika.

Get_pokoje – funkcja zwracająca z bazy listę dostępnych pokoi na podany termin i deklarowaną liczbę osób. Do tego celu został użyty cursor iterujący po tabeli *pokoj*, gdzie w pętli sprawdzany był warunek, czy wskazywany pokój na podany termin przypisany jest już do jakiejś rezerwacji – jeżeli nie, wówczas id pokoju było zapisywane do zwracanej tabeli.

Get_liczba_dni – pomocnicza funkcja wyznaczająca przedział czasowy pomiędzy podanymi datami.

Last_rezerwacja_id – funkcja zwracająca z bazy numer id ostatniej złożonej rezerwacji. W tym celu wykorzystano funkcję agregującą *MAX()*, która została wywołana na atrybucie *rezerwacja_id* z tabeli *rezerwacje*.

Last_uzytkownik_id – funkcja zwracająca z bazy numer id ostatniego zarejestrowanego użytkownika. W tym celu wykorzystano funkcję agregującą *MAX()*, która została wywołana na atrybucie *uzytkownik_id* z tabeli *uzytkownik*.

OplataZaplac – funkcja aktualizująca wartość atrybutu *status_czy_oplacone* na wartość 'Oplacone' krotki z tabeli *opлата* o podanym numerze id.

OplataRezygnuj – funkcja aktualizująca wartość atrybutu *status_czy_oplacone* na wartość 'Nieoplacone - rezygnacja' krotki z tabeli *opлата* o podanym numerze id.

ZbanowanyCheck – funkcja sprawdzająca, czy użytkownik o podanym id został z jakiegoś powodu zablokowany. W tym celu sprawdzane jest funkcja agregującą *COUNT()*, czy podany id sprawdzanego użytkownika znajduje się w tabeli *czarna_lista*.

Autoryzacja – funkcja sprawdzająca, czy logujący się użytkownik jest klientem, pracownikiem czy headadminem. W tym celu sprawdzany jest atrybut *typ* z tabeli *uzytkownik*.

StartUpdate – funkcja wywoływana na bazie danych przy każdym starcie aplikacji. Funkcja ta wykorzystuje kursor iterujący po widoku *OplataRezerwacje* sprawdzając między innymi:

- czy dzisiejszego dnia następuje jakieś zakwaterowanie,
- czy dzisiejszego dnia następuje jakieś wykwaterowanie,
- czy dzisiejszego dnia mija termin opłaty rezerwacji któregoś z klientów.

W przypadku zakwaterowania czy wykwaterowania modyfikowana jest krotka z tabeli *zakwaterowani_goscie_info* aktualizująca wartość atrybutu *status_czy_zakwaterowany*, zaś w przypadku braku uregulowania opłaty w terminie użytkownik ten automatycznie jest przenoszony na czarną listę, innymi słowy wykonywany jest *INSERT* do tabeli *czarna_lista* zawierająca id tegoż użytkownika oraz powód o treści *‘Nieopłacenie rezerwacji w terminie.’*, a rekordy związane z nieuregulowaną rezerwacją zostają usunięte w celu zwolnienia pokoi.

Wyzwalacze (dostępne w plikach *SQL/first_triggers.sql* oraz *SQL/last_triggers.sql*):

Register validator – wyzwalacz uruchamiany przed *INSERT’em* do tabeli *uzytkownik*. Wyzwalacz ten sprawdza poprawność wprowadzonych danych, czyli sprawdza, czy wszystkie pola są wypełnione, czy adres email jest poprawny bądź unikatowy oraz czy podany numer telefonu jest składa się z cyfr. W przypadku niepowodzenia do aplikacji wysyłany jest odpowiedni komunikat z bazy.

Panel validator – wyzwalacz uruchamiany przed *INSERT’em* do tabeli *panel*. Wyzwalacz ten sprawdza poprawność wprowadzonych danych, czyli sprawdza, czy wszystkie pola są wypełnione, czy login oraz hasło są dłuższe, niż 5 znaków oraz czy nazwa użytkownika jest unikatowa. W przypadku niepowodzenia do aplikacji wysyłany jest odpowiedni komunikat z bazy.

Platnosc rezerwacja – wyzwalacz uruchamiany po *INSERT’cie* do tabeli *rezerwacje*. Wyzwalacz ten oblicza sumaryczny koszt pobytu w zależności od ilości dni uwzględniając dzieci oraz dorosłych, wykonując *INSERT* do tabeli *opлата*.

Platnosc update – wyzwalacz uruchamiany po *INSERT’cie* do tabeli *dodatkowe_uslugi*. Wyzwalacz ten aktualizuje sumaryczny koszt pobytu w hotelu z tabeli *opлата* o koszt za dodatkowe usługi uwzględniając dzieci oraz dorosłych.

OplataStatus – wyzwalacz uruchamiany przed *UPDATE’em* do tabeli *opлата*. Wyzwalacz ten odpowiedzialny jest za aktualizacje statusu opłaty. W celu uniknięcia problemów z aktualizacjami wynikającymi z dwóch wcześniej opisanych wyzwalaczy sprawdzane jest najpierw, czy nowa kwota różni się od kwoty poprzedniej, wówczas wyzwalacze *Platnosc_rezerwacje* oraz *Platnosc_update* mogą spokojnie się wykonać. W przeciwnym wypadku sprawdzany jest warunek, czy stary status jest określony jako *‘Nieoplacone’*. Jeżeli tak, wówczas w przypadku nowego statusu *‘Opłacone’* wykonywany jest *INSERT* do tabeli *zakwaterowani_goscie_info* zawierający id rezerwacji wraz z informacją *‘Oczekiwanie na zakwaterowanie’*, zaś w przypadku nowego statusu *‘Nieoplacone - rezygnacja’* wykonywany jest *INSERT* do tabeli *rezygnacja_z_rezerwacji_info* zawierający id rezerwacji oraz id użytkownika. Gdy jednak żaden z wyżej wymienionych warunków nie jest spełniony, do aplikacji wysyłany jest komunikat informujący o braku możliwości modyfikacji wcześniej zmodyfikowanego statusu opłaty rezerwacji.

RezerwacjaNaTydzienPrzed – wyzwalacz uruchamiany przed *INSERT’em* do tabeli *rezerwacje*. Wyzwalacz ten sprawdza, czy wybrany termin zakwaterowania jest określony na więcej niż siedem dni od dnia złożenia rezerwacji. W przeciwnym wypadku do aplikacji wysyłany jest odpowiedni komunikat z bazy.

RezerwacjaOsobyValidator - wyzwalacz uruchamiany przed *INSERT'em* do tabeli *rezerwacje*. Wyzwalacz ten sprawdza, czy podana liczba osób w rezerwacji jest poprawna. Jeżeli sumaryczna liczba osób wynosi zero bądź przekracza pojemność wybranego pokoju, wówczas do aplikacji wysyłany jest odpowiedni komunikat z bazy.

RezygnacjaChecker – wyzwalacz uruchamiany po *INSERT'cie* do tabeli *rezygnacja_z_rezerwacji_info*. Wyzwalacz ten usuwa rekordy związane z podaną rezerwacją w celu zwolnienia pokoi oraz sprawdza liczbę rekordów w tejże tabeli dla danego użytkownika. Jeżeli liczba rezygnacji przekroczy 5 rekordów, wówczas użytkownik zostaje dodany na czarną listę, czyli wykonywany jest *INSERT* do tabeli *czarna_lista* zawierający id użytkownika oraz powód o treści '*Ciagle rezygnowanie ze skladanych rezerwacji.*'.

CzarnaListaChecker - wyzwalacz uruchamiany przed *INSERT'em* do tabeli *czarna_lista*. Wyzwalacz ten jest głównie używany w czasie manualnego blokowania użytkownika przez administratora – sprawdzane jest, czy powód blokady jest uzupełniony oraz czy blokowany użytkownik nie jest headadminem lub pracownikiem – w przeciwnym wypadku do aplikacji zostanie wysłany odpowiedni komunikat z bazy.

TransakcjaRezerwacjaChecker – wyzwalacz uruchamiany przed *INSERT'em* do tabeli *dodatkowe_uslugi*. Wyzwalacz ten jest dosyć prostym i prymitywnym zabezpieczeniem przed wymieszaniem się danych w momencie, kiedy dwie osoby w tym samym czasie próbują dokonać rezerwacji na ten sam pokój w pokrywających się terminach. Jeżeli warunki te zostaną spełnione, do aplikacji zostanie wysłany odpowiedni komunikat z bazy. Dlaczego wyzwalacz ten jest podpięty pod tabelę *dodatkowe_uslugi*? Otóż proces rezerwacji został zaimplementowany jako procedura transakcji, gdzie *INSERT* do tabeli *dodatkowe_uslugi* wykonuje się zaraz po *INSERT'cie* do tabeli *rezerwacje* i *INSERT* ten jest ściśle związany z poprzednim poprzez wartość atrybutu *rezerwacja_id*. Jest to mało prawdopodobny scenariusz, jednakże, jeżeli w tym samym momencie dwie osoby przy wyżej wymienionych warunkach będą chciały dokonać rezerwacji, wówczas bez tego zabezpieczenia może nastąpić przypisanie informacji o dodatkowych usługach do zupełnie innej rezerwacji.

TransakcjaRejestracjiChecker – wyzwalacz uruchamiany przed *INSERT'em* do tabeli *panel*. Wyzwalacz ten spełnia podobną rolę, co wyzwalacz opisany powyżej. Sprawdzane jest, czy nie następuje dopisanie informacji do istniejącego już użytkownika, a w przypadku spełnienia warunku do aplikacji zostanie wysłany odpowiedni komunikat z bazy.

DeleteRezerwacje – wyzwalacz uruchamiany przed *DELETE'm* z tabeli *rezerwacje*. Wyzwalacz ten usuwa wszystkie rekordy z pozostałych tabel związane z usuwanym rekordem.

Widoki (dostępne w pliku SQL/widoki.sql):

Uzytkownicy – widok będący złączeniem tabel *uzytkownik* oraz *panel* po atrybutach odpowiednio *uzytkownik_id* oraz *uzyt_id*.

PokojeView - widok będący złączeniem tabel *pokoj* oraz *kategoria* po atrybucie *kategoria_id*.

PokojeRezerwacjeView – pomocniczy widok będący złączeniem tabel *pokoj* oraz *rezerwacje* po atrybucie *pokoj_id*.

UslugiPokojeRezerwacje – pomocniczy widok będący złączeniem tabeli *dodatkowe_uslugi* oraz widoku *PokojeRezerwacjeView* po atrybucie *rezerwacja_id*.

RezerwacjeInfoView – pomocniczy widok będący złączeniem tabeli *opлата* oraz widoku *UslugiPokojeRezerwacje* po atrybucie *rezerwacja_id*.

RezerwacjeAllInfoView – widok będący złączeniem tabeli *uzytkownik* oraz widoku *RezerwacjeInfoView* po atrybucie *uzytkownik_id*.

OplataRezerwacje – pomocniczy widok będący złączeniem tabel *opлата* oraz *rezerwacje* po atrybucie *rezerwacja_id*.

RezerwPokojeView – pomocniczy widok będący złączeniem tabel *pokoj* oraz *rezerwacje* po atrybucie *rezerwacja_id*.

ZakwaterowaniView –widok będący złączeniem tabel *uzytkownik* i *zakwaterowani_goscie_info* oraz widoku *RezerwPokojView*.

ZablokowaniView – widok będący złączeniem tabel *czarna_lista* oraz *uzytkownik* po atrybucie *uzytkownik_id*.

UzytRezerwView – pomocniczy widok będący złączeniem tabel *rezerwacje* oraz *uzytkownik* po atrybucie *uzytkownik_id*.

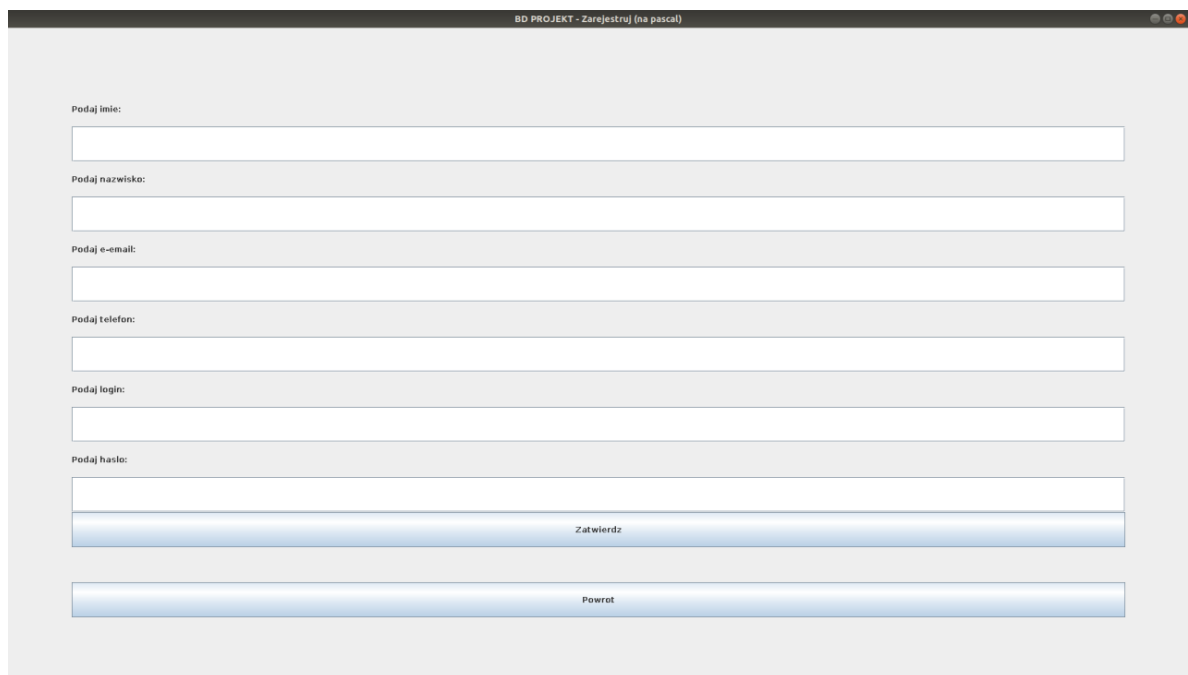
RezygnacjaView – widok będący złączeniem tabeli *rezygnacja_z_rezerwacji_info* oraz widoku *UzytRezerwView* po atrybucie *rezerwacja_id*.

IV. Projekt funkcjonalny

11. Interfejsy do prezentacji, edycji i obsługi danych:

Obsługa oraz edycja danych jest dostępna zarówno z poziomu administratora, jak i klienta. Poniżej przedstawiono dostępne formularze:

- formularz rejestracji nowego użytkownika – nowy użytkownik ma możliwość wprowadzenia swoich danych personalnych, po kliknięciu przycisku 'Zatwierdź' rozpoczyna się procedura transakcji, dane wysyłane są do bazy danych, a następnie odpowiednie wyzwalacze sprawdzają poprawność wprowadzonych danych (czy wszystkie pola są wypełnione, czy użytkownik o podanym e-mailu już istnieje, czy podany login jest unikatowy itp.),



Obrazek 4: Formularz rejestracji użytkownika.

W przypadku powodzenia bądź też niepowodzenia wyświetlane są odpowiednie komunikaty wysyłane z bazy. Poniżej przedstawiono przykład takiego komunikatu:

BD PROJEKT - Zarejestruj (na pascali)

Podaj imię:

Jan

Podaj nazwisko:

Kowalski

Podaj e-mail:

jan@kowalski.com

Podaj telefon:

asdasd

Podaj login:

Podaj hasło:

Zatwierdź

Błąd przy dodawaniu nowego użytkownika! Proszę podać poprawny numer telefonu!

Powrót

Obrazek 5: Formularz rejestracji użytkownika wyświetlający błąd.

- formularz logowania – każdy z użytkowników ma możliwość zalogowania się na swoje konto. Po wciśnięciu guzika 'Zaloguj' do bazy wysyłane jest zapytanie, czy dane logowania są poprawne, a następnie w zależności od rezultatu wyświetlany jest odpowiedni komunikat:

BD PROJEKT - Zaloguj (na pascali)

Zaloguj się

Zarejestruj się

Obrazek 6: Formularz logowania.

BD PROJEKT - Zaloguj (na pascal)

Bledne dane logowania!

nieistniejace

nieistniejace

Zaloguj sie

Zarejestruj sie

Obrazek 7: Formularz logowania wyświetlający błąd.

- formularz dokonywania rezerwacji – każdy użytkownik ma możliwość dokonania rezerwacji w hotelu. Użytkownik ma możliwość wyboru daty zakwaterowania oraz wykwaterowania, wyboru pokoju czy dodatkowych usług. Po wciśnięciu guzika 'Potwierdź rezerwację' rozpoczynana jest procedura transakcji, podczas której uruchamiane są odpowiednie wyzwalacze sprawdzające poprawność wprowadzonych danych. W zależności od rezultatu operacji wyświetlany jest odpowiedni komunikat z bazy:

BD PROJEKT - Dokonaj rezerwacji (na pascal)

Proszę wybrać termin zakwaterowania (jednak nie wcześniej, niż 7 dni od dnia dzisiejszego):

Proszę wybrać rok

Proszę wybrać miesiąc

Proszę wybrać dzień

Proszę wybrać termin wykwaterowania:

Proszę wybrać rok

Proszę wybrać miesiąc

Proszę wybrać dzień

Proszę podać liczbę dorosłych (100% ceny od osoby):

Maksymalna dostępna liczba osób na pokój: 7

Proszę podać liczbę dzieci (50% ceny od osoby):

Pozostała dostępna liczba osób na pokój: 7

Proszę wybrać jeden z dostępnych pokoi (w przypadku wyboru pokoju z większą liczbą miejsc, niż zadeklarowana, wówczas pobrana będzie opłata wysokości 25% kwoty za pokój od każdego nadmiarowego miejsca):

Proszę wybrać dodatkowe usługi (opcjonalnie):

☐ Dostęp do siłowni - koszt 5 zł od osoby (w przypadku dzieci obowiązuje ulga 50%)

☐ Dostęp do bilardu - koszt 10 zł od osoby (w przypadku dzieci obowiązuje ulga 50%)

☐ Dostęp do basenu - koszt 15 zł od osoby (w przypadku dzieci obowiązuje ulga 50%)

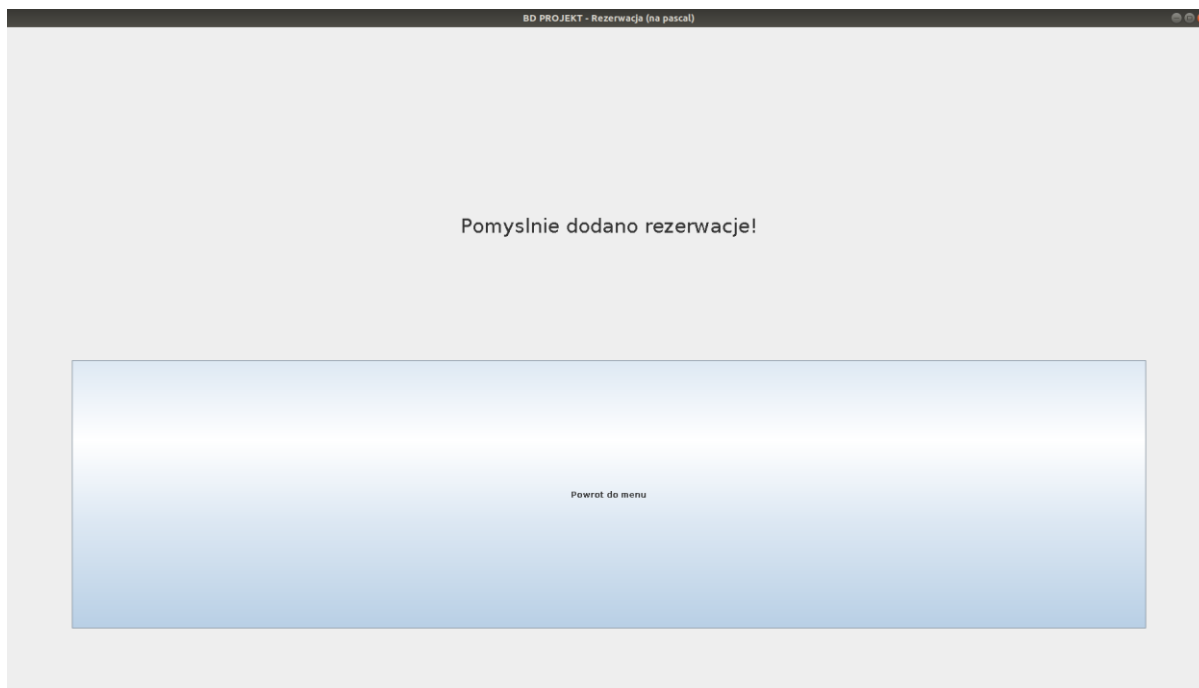
☐ Dostęp do kregielni - koszt 25 zł od osoby (w przypadku dzieci obowiązuje ulga 50%)

Całkowity koszt: 0.0 zł

Potwierdź rezerwację

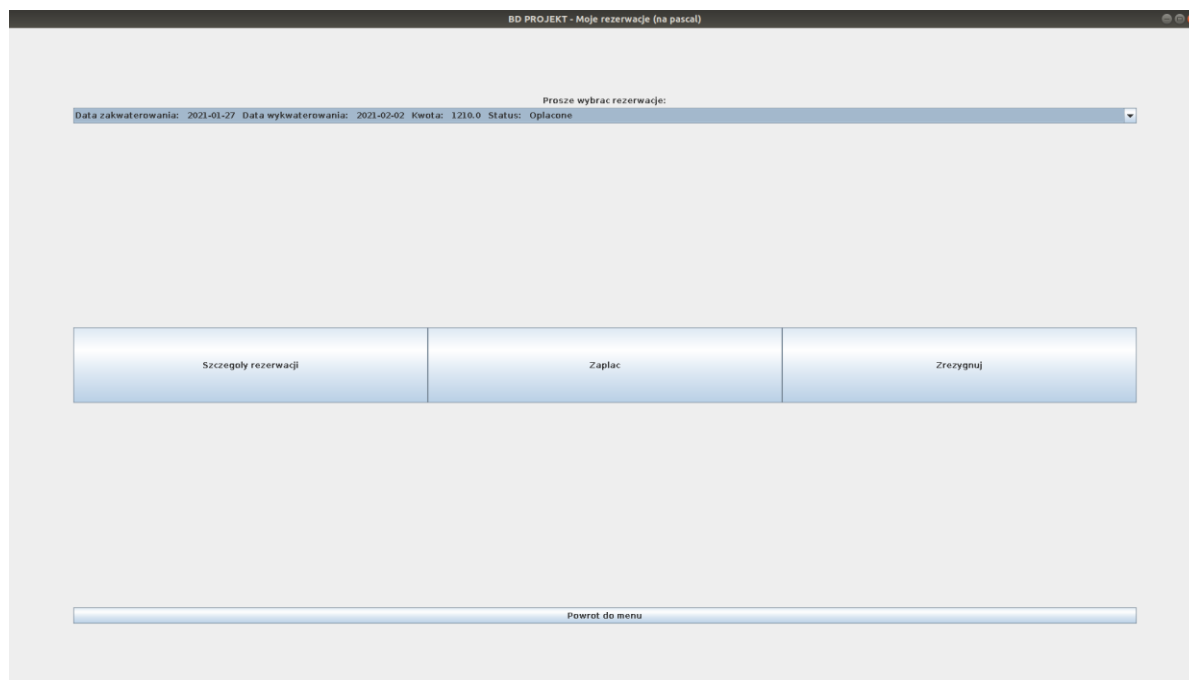
Powrót do menu

Obrazek 8: Formularz dodawania rezerwacji.

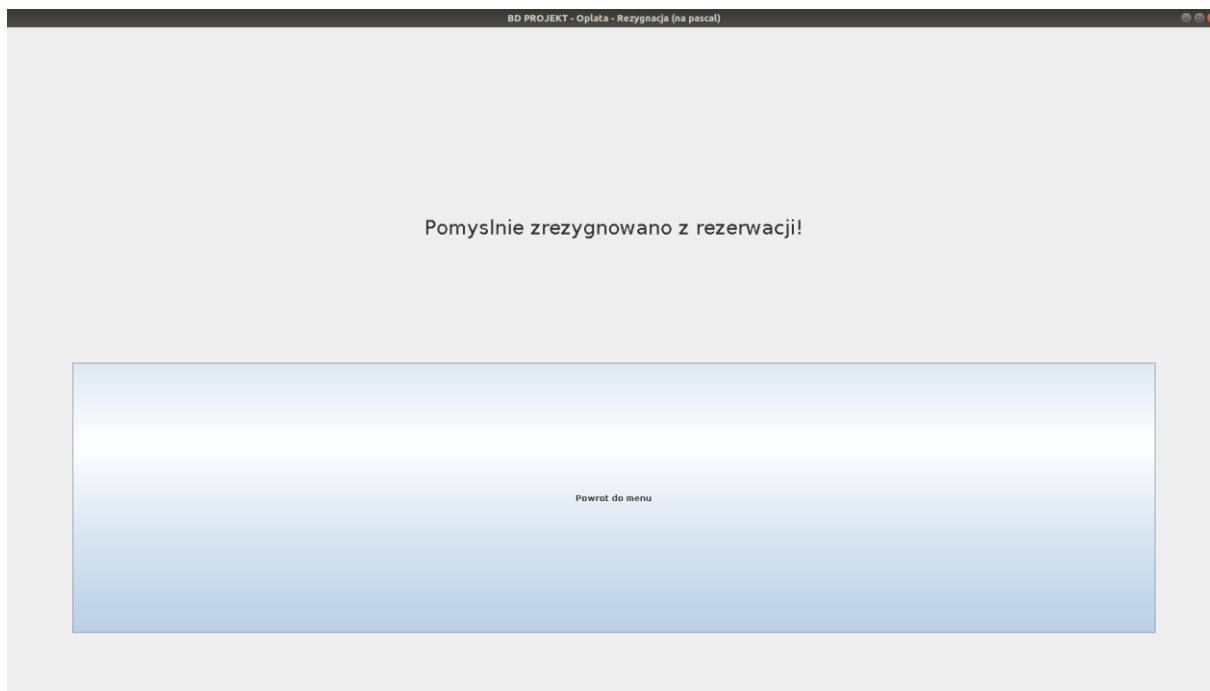


Obrazek 9: Formularz dodawania rezerwacji wyświetlający komunikat.

- formularz opłaty bądź rezygnacji z rezerwacji – każdy użytkownik ma możliwość zdecydowania, czy dana rezerwacja ma być opłacona czy też nie. Po wciśnięciu odpowiedniego guzika uruchamiany jest odpowiedni wyzwalacz walidujący, a na koniec z bazy wysyłany jest odpowiedni komunikat:



Obrazek 10: Formularz opłaty lub rezygnacji z rezerwacji.



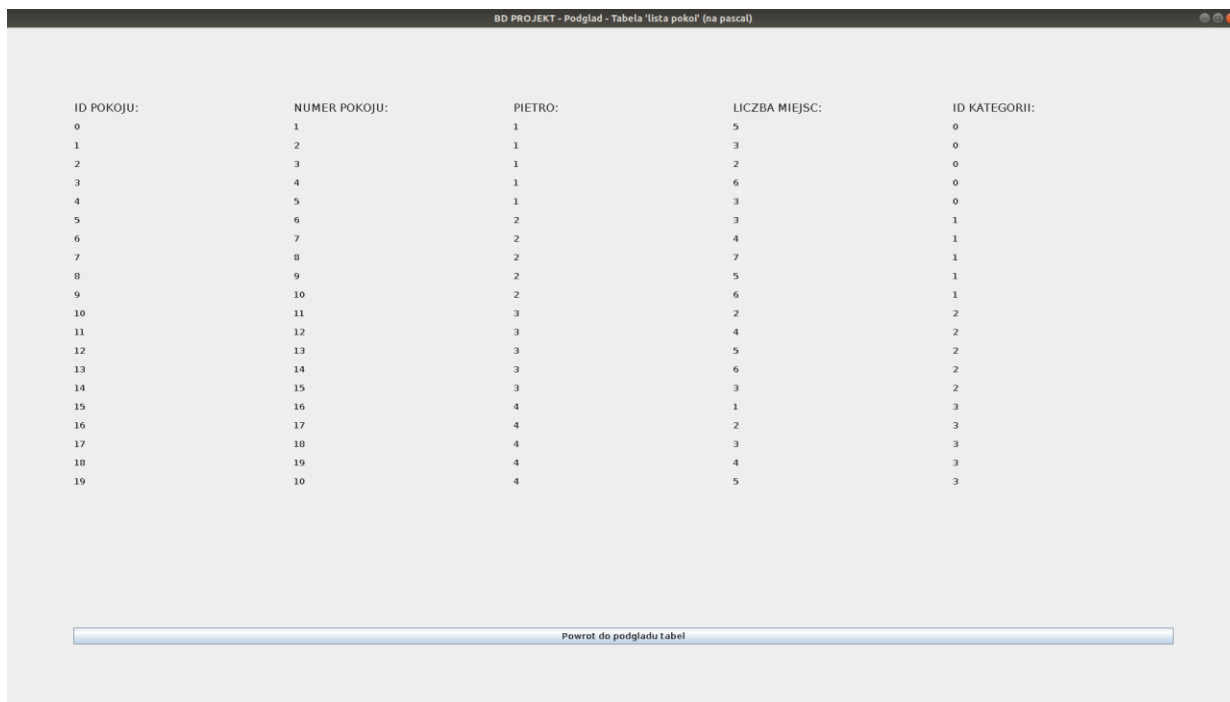
Obrazek 11: Formularz opłaty lub rezygnacji z rezerwacji wyświetlający odpowiedni komunikat.

- formularz rejestracji nowego pracownika – opcja ta dostępna jest tylko dla użytkowników zalogowanych jako HeadAdmin lub Pracownik. Procedura ta jest niemalże taka sama jak w przypadku rejestracji nowego użytkownika – wpisy do bazy różnią się jedynie typem konta. Poniżej przedstawiono wygląd tegoż formularza:

Obrazek 12: Formularz rejestracji pracownika.

12. Wizualizacja danych:

Wszelkie dane z hotelowej bazy danych zostały przedstawione w postaci interaktywnych raportów oraz bezpośredniego podglądu wszystkich tabel. Każda pojedyncza tabela z omawianego projektu ma swój własny podgląd, który zawiera nazwy wszystkich atrybutów oraz odpowiadające im wartości. Poniżej przedstawiono jeden z takich podglądów:



| ID POKOJU: | NUMER POKOJU: | PIETRO: | LICZBA MIEJSC: | ID KATEGORII: |
|------------|---------------|---------|----------------|---------------|
| 0 | 1 | 1 | 5 | 0 |
| 1 | 2 | 1 | 3 | 0 |
| 2 | 3 | 1 | 2 | 0 |
| 3 | 4 | 1 | 6 | 0 |
| 4 | 5 | 1 | 3 | 0 |
| 5 | 6 | 2 | 3 | 1 |
| 6 | 7 | 2 | 4 | 1 |
| 7 | 8 | 2 | 7 | 1 |
| 8 | 9 | 2 | 5 | 1 |
| 9 | 10 | 2 | 6 | 1 |
| 10 | 11 | 3 | 2 | 2 |
| 11 | 12 | 3 | 4 | 2 |
| 12 | 13 | 3 | 5 | 2 |
| 13 | 14 | 3 | 6 | 2 |
| 14 | 15 | 3 | 3 | 2 |
| 15 | 16 | 4 | 1 | 3 |
| 16 | 17 | 4 | 2 | 3 |
| 17 | 18 | 4 | 3 | 3 |
| 18 | 19 | 4 | 4 | 3 |
| 19 | 10 | 4 | 5 | 3 |

Powrot do podgladu tabel

Obrazek 13: Podgląd tabeli *pokoj*.

W przypadku raportów wykorzystywane są odpowiednie widoki z bazy danych. Dane w raportach wyświetlane są w ten sam sposób, jak w przypadku podglądu tabel, jednakże posiadają one dodatkowo możliwość operowania na danych – wyszukiwanie rekordów odpowiadających podanej wartości oraz wyświetlanie danych wykorzystując funkcje agregujące, klauzule GROUP BY, HAVING oraz ORDER BY. Stworzono dostęp do sześciu takich raportów:

- Raport o użytkownikach – wykorzystano tutaj widok *uzytkownicy*,
- Raport o rezerwacjach – wykorzystano tutaj widok *RezerwacjeAllInfoView*,
- Raport o pokojach – wykorzystano tutaj widok *pokojeView*,
- Raport o zakwaterowanych gościach – wykorzystano tutaj widok *zakwaterowaniView*,
- Raport o rezygnacjach z rezerwacji – wykorzystano tutaj widok *rezygnacjaView*,
- Raport o zablokowanych użytkownikach – wykorzystano tutaj widok *zablokowaniView*.

Interaktywność w każdym raporcie poza raportem o rezerwacjach jest taka sama, ta wcześniej wspomniana różni się tylko większym wyborem funkcji agregujących. Poniżej przedstawiono wygląd przykładowych raportów:

BD PROJEKT - Podgląd raportów - Raport o pokojach (na pascal)

☐ Zlicz elementy po:
☐ po konkretnej wartości:
☒ Sortuj po: ☐ malejaco

Wyszukaj rekord dla:
o wartości:
☐ Sortuj po: ☐ malejaco

| pokoj_id | numer_pokoju | pietro | liczba_miejsc | kategoria_id | nazwa_kategorii | cena_od_osoby |
|----------|--------------|--------|---------------|--------------|----------------------------|---------------|
| 0 | 1 | 1 | 5 | 0 | Dla osób specjalnej troski | 50 |
| 1 | 2 | 1 | 3 | 0 | Dla osób specjalnej troski | 50 |
| 2 | 3 | 1 | 2 | 0 | Dla osób specjalnej troski | 50 |
| 3 | 4 | 1 | 6 | 0 | Dla osób specjalnej troski | 50 |
| 4 | 5 | 1 | 3 | 0 | Dla osób specjalnej troski | 50 |
| 5 | 6 | 2 | 3 | 1 | Pakiet podstawowy | 40 |
| 6 | 7 | 2 | 4 | 1 | Pakiet podstawowy | 40 |
| 7 | 8 | 2 | 7 | 1 | Pakiet podstawowy | 40 |
| 8 | 9 | 2 | 5 | 1 | Pakiet podstawowy | 40 |
| 9 | 10 | 2 | 6 | 1 | Pakiet podstawowy | 40 |
| 10 | 11 | 3 | 2 | 2 | Pakiet premium | 100 |
| 11 | 12 | 3 | 4 | 2 | Pakiet premium | 100 |
| 12 | 13 | 3 | 5 | 2 | Pakiet premium | 100 |
| 13 | 14 | 3 | 6 | 2 | Pakiet premium | 100 |
| 14 | 15 | 3 | 3 | 2 | Pakiet premium | 100 |
| 15 | 16 | 4 | 1 | 3 | Pakiet ekstra-premium | 150 |
| 16 | 17 | 4 | 2 | 3 | Pakiet ekstra-premium | 150 |
| 17 | 18 | 4 | 3 | 3 | Pakiet ekstra-premium | 150 |
| 18 | 19 | 4 | 4 | 3 | Pakiet ekstra-premium | 150 |
| 19 | 10 | 4 | 5 | 3 | Pakiet ekstra-premium | 150 |

Obrazek 14: Podgląd raportu o pokojach.

BD PROJEKT - Podgląd raportów - Raport o pokojach (na pascal)

☒ Zlicz elementy po:
☐ po konkretnej wartości:
☒ Sortuj po: ☐ malejaco

Wyszukaj rekord dla:
o wartości:
☐ Sortuj po: ☐ malejaco

| NAZWA_KATEGORII | ZLICZ |
|----------------------------|-------|
| Dla osób specjalnej troski | 5 |
| Pakiet ekstra-premium | 5 |
| Pakiet podstawowy | 5 |
| Pakiet premium | 5 |

Obrazek 15: Podgląd raportu o pokojach wyświetlający dane z użyciem funkcji agregującej *COUNT()*.

BD PROJEKT - Podgląd raportów - Raport o pokojach (na pascal)

☐ Zlicz elementy po: ☐ po konkretnej wartości: Sortuj po: ☐ malejąco

Wyszukaj rekord dla: o wartości: Sortuj po: ☐ malejąco

| pokoj_id | numer_pokoju | pietro | liczba_miejsc | kategoria_id | nazwa_kategorii | cena_od_osoby |
|----------|--------------|--------|---------------|--------------|-----------------------|---------------|
| 18 | 19 | 4 | 4 | 3 | Pakiet ekstra-premium | 150 |

Obrazek 16: Podgląd raportu o pokojach pokazujący wynik poszukiwania rekordu.

BD PROJEKT - Podgląd raportów - Raport o rezerwacjach (na pascal)

☐ po konkretnej wartości: Sortuj po: ☐ malejąco

Wyszukaj rekord dla: o wartości: Sortuj po: ☐ malejąco

| rezerwacja_id | imie | nazwisko | data_rezerwacji | od_kiedy | do_kiedy | numer_pokoju | nazwa_kategorii | nazwa_uslugi | kwota | status_czy_opla... |
|---------------|--------|------------|-----------------|------------|------------|--------------|-----------------------|--------------|-----------|--------------------|
| 1 | Sylwia | Abacki | 2021-01-13 | 2021-06-05 | 2021-11-10 | 13 | Pakiet premium | Basen | 34352.50 | Nieopłacone |
| 4 | Jan | Okon | 2021-01-13 | 2020-04-07 | 2021-06-18 | 10 | Pakiet ekstra-pre... | Basen | 187162.50 | Nieopłacone |
| 11 | Kamil | Wisniewski | 2021-01-13 | 2020-06-10 | 2020-12-28 | 9 | Pakiet podstawowy | Brak | 22220.00 | Nieopłacone |
| 14 | Terasa | Okon | 2021-01-13 | 2021-06-08 | 2021-11-23 | 9 | Pakiet podstawowy | Brak | 12960.00 | Nieopłacone |
| 20 | Tomasz | Kowalski | 2021-01-13 | 2020-06-23 | 2020-12-27 | 8 | Pakiet podstawowy | Brak | 15240.00 | Nieopłacone |
| 22 | Sylwia | Abacki | 2021-01-13 | 2020-10-25 | 2021-10-25 | 13 | Pakiet premium | Silownia | 118965.00 | Nieopłacone |
| 24 | Tomasz | Kutyla | 2021-01-13 | 2020-10-04 | 2021-12-29 | 4 | Dla osob specjalne... | Silownia | 56510.00 | Nieopłacone |
| 25 | Kamil | Leszcz | 2021-01-13 | 2020-07-23 | 2020-11-23 | 12 | Pakiet premium | Kregielnia | 24837.50 | Nieopłacone |
| 28 | Anna | Kowalski | 2021-01-13 | 2021-05-04 | 2021-06-05 | 4 | Dla osob specjalne... | Billard | 16480.00 | Nieopłacone |
| 29 | Jan | Kowalski | 2021-01-13 | 2021-07-24 | 2022-07-28 | 18 | Pakiet ekstra-pre... | Billard | 55505.00 | Nieopłacone |
| 6 | Terasa | Wisniewski | 2021-01-13 | 2021-05-06 | 2021-07-30 | 1 | Dla osob specjalne... | Basen | 15102.50 | Oplacone |

Obrazek 17: Podgląd raportu o rezerwacjach.

BD PROJEKT - Podgląd raportów - Raport o rezerwacjach (na pascal)

☐ po konkretnej wartości: ☐ Sortuj po: ☐ malejąco

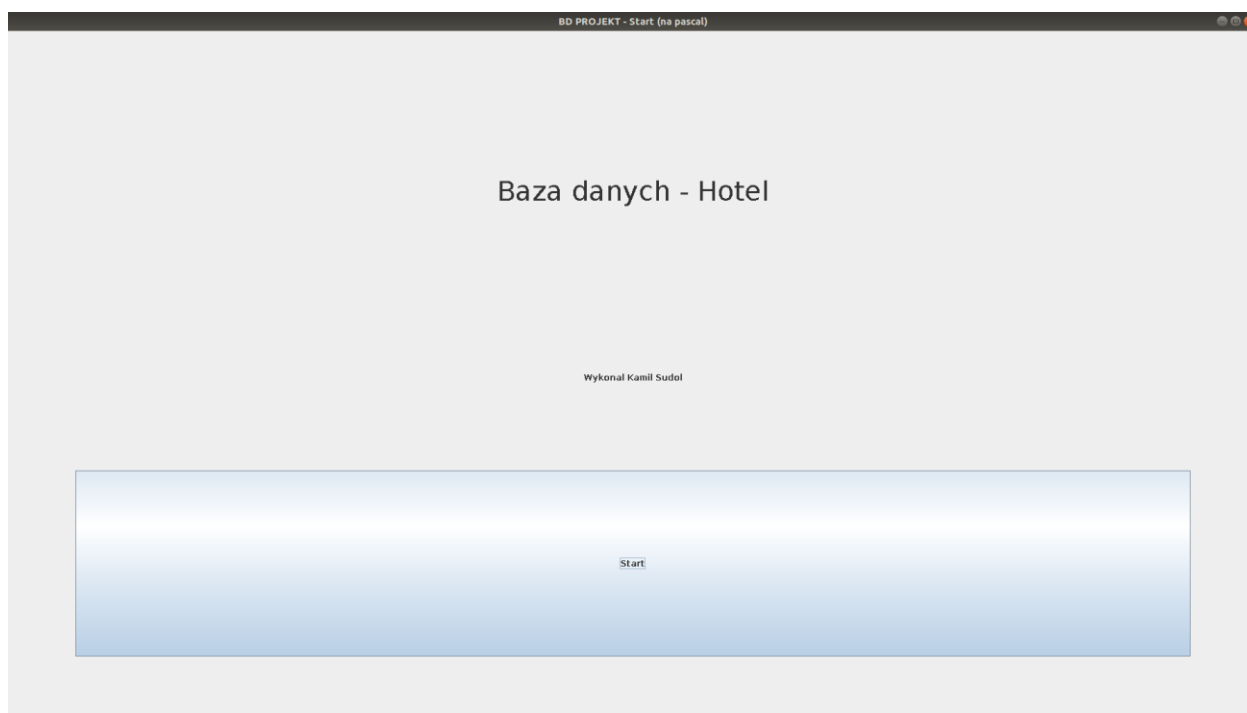
Wyszukaj rekord dla: o wartości: ☐ Sortuj po: ☐ malejąco

| NAZWISKO | MIN |
|------------|----------|
| Kutyla | 56510.00 |
| Oken | 12960.00 |
| Wisniewski | 15102.50 |
| Abacki | 34352.50 |
| Kowalski | 15240.00 |
| Leszcz | 24837.50 |

Obrazek 18: Podgląd raportu o rezerwacjach wyświetlający dane z użyciem funkcji agregującej *MIN()*.

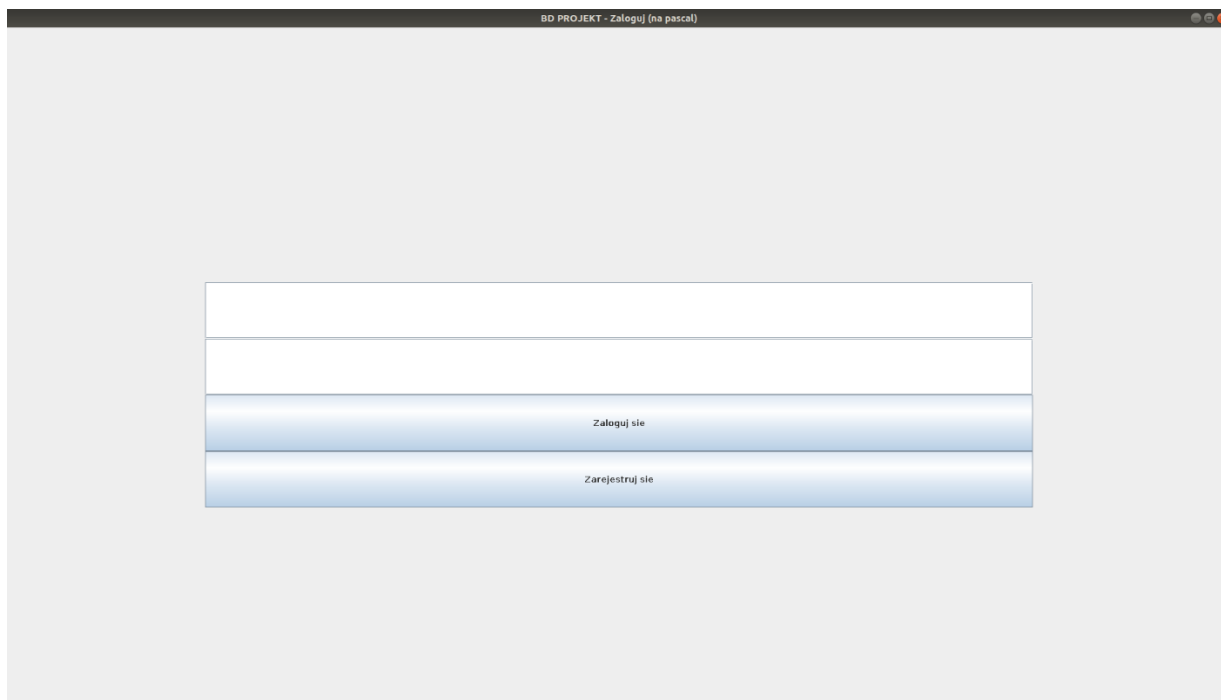
13.Zdefiniowanie panelu sterowania aplikacji:

Uruchamiając aplikację pojawia się banner witający posiadający przycisk przenoszący użytkownika do obszaru logowania:



Obrazek 19: Banner witający.

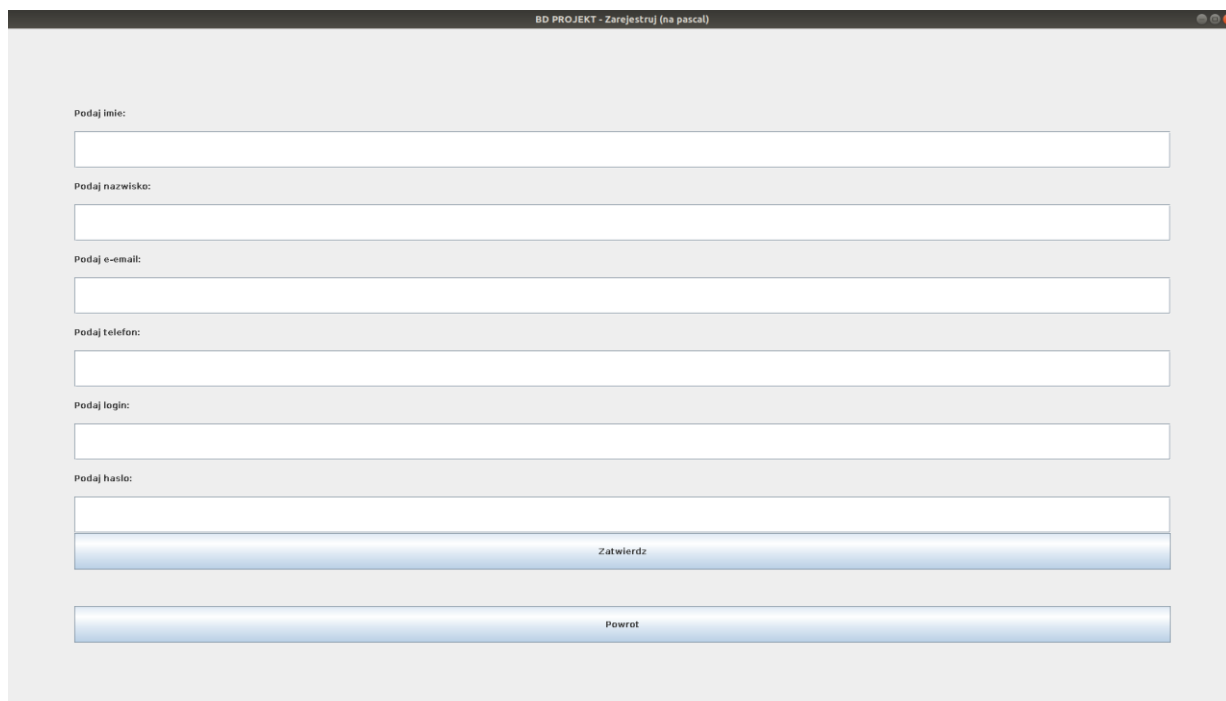
Użytkownik w tym momencie ma dwie możliwości – albo wprowadzić dane logowania i się zalogować do serwisu albo wybrać przycisk ‘Zarejestruj się’, który to przenosi użytkownika do obszaru rejestracji:

A screenshot of a web application window titled "BD PROJEKT - Zaloguj (na pascal)". The window has a light gray background. In the center, there is a white rectangular box containing two input fields for login and password. Below these fields are two blue buttons with white text. The top button is labeled "Zaloguj się" and the bottom button is labeled "Zarejestruj się".

| |
|-----------------|
| |
| |
| Zaloguj się |
| Zarejestruj się |

Obrazek 20: Panel logowania.

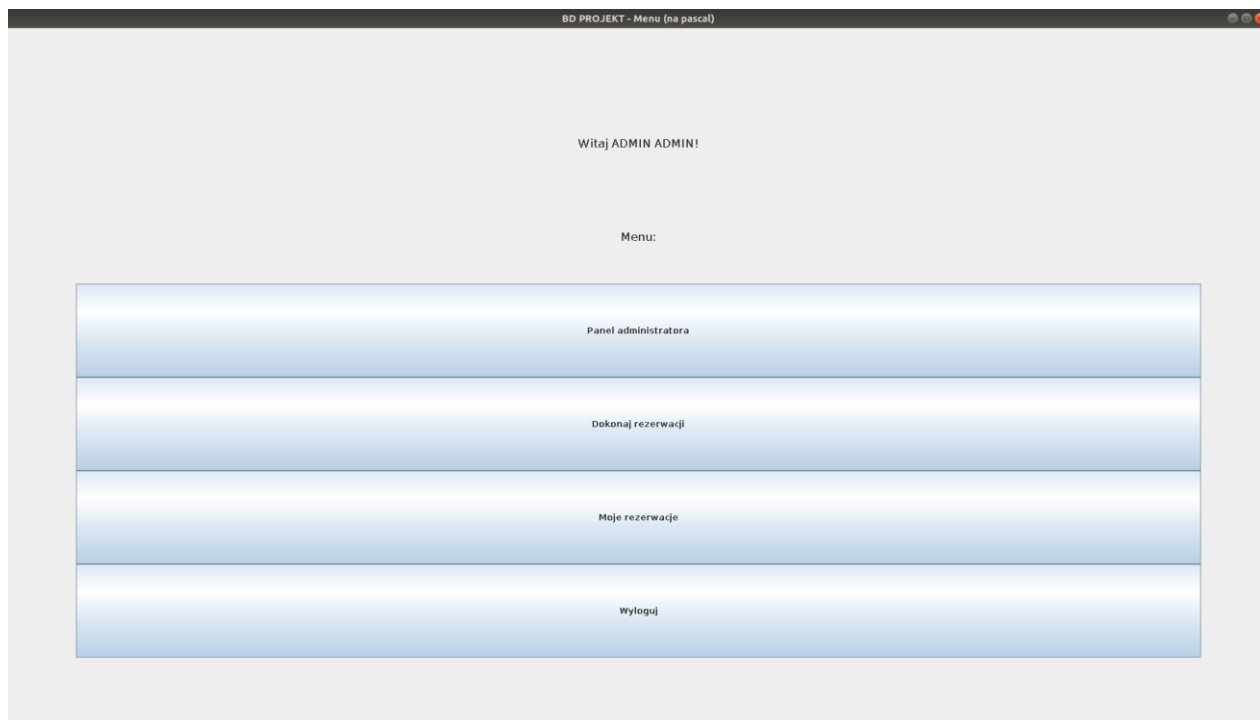
Poniżej przedstawiono wygląd obszaru rejestracji:

A screenshot of a web application window titled "BD PROJEKT - Zarejestruj (na pascal)". The window has a light gray background. It contains a registration form with several input fields and two buttons. The fields are labeled: "Podaj imię:", "Podaj nazwisko:", "Podaj e-mail:", "Podaj telefon:", "Podaj login:", and "Podaj hasło:". Below the "Podaj hasło:" field is a blue button labeled "Zatwierdź". At the bottom of the form is another blue button labeled "Powrót".

| | |
|-----------------|-----------|
| Podaj imię: | |
| Podaj nazwisko: | |
| Podaj e-mail: | |
| Podaj telefon: | |
| Podaj login: | |
| Podaj hasło: | |
| | Zatwierdź |
| | Powrót |

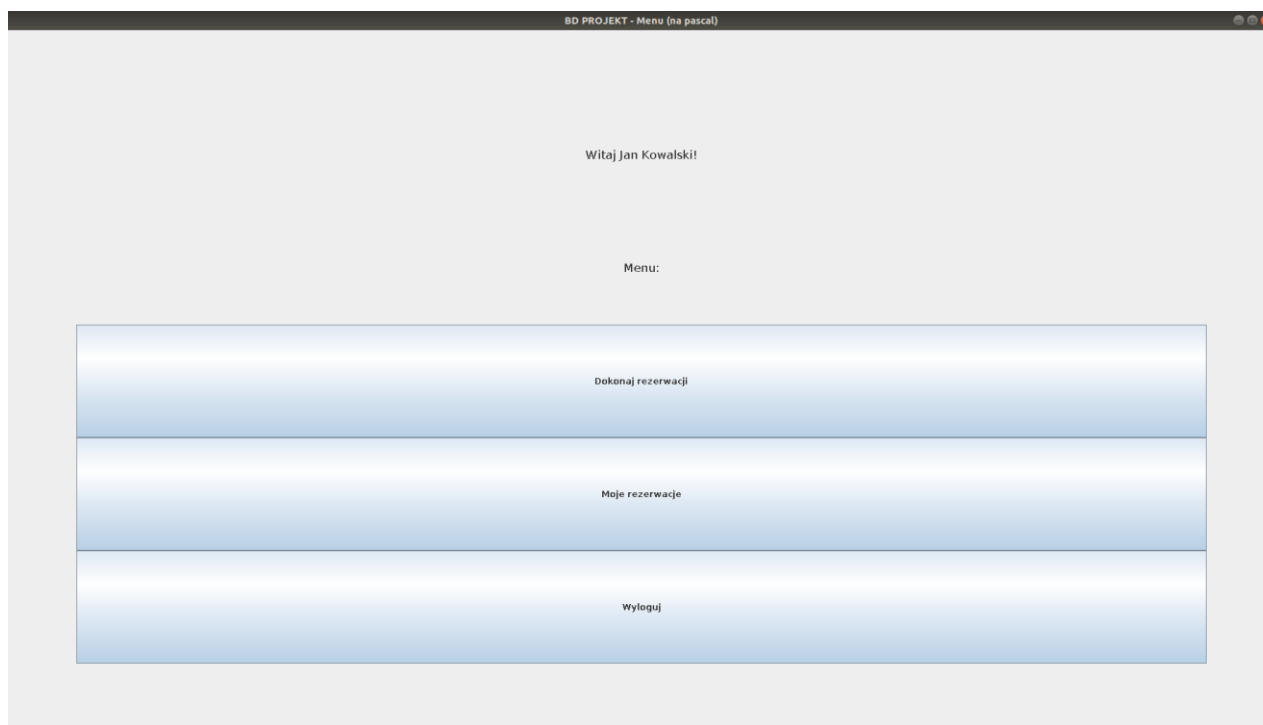
Obrazek 21: Panel rejestracji.

Po zalogowaniu użytkownikowi pojawia się na ekranie menu wszystkich możliwości. W zależności od typu konta bądź też statusu na czarnej liście wyświetlane są różne konfiguracje menu. Poniżej przedstawiono menu wyświetlane dla HeadAdmina lub Pracownika hotelu:



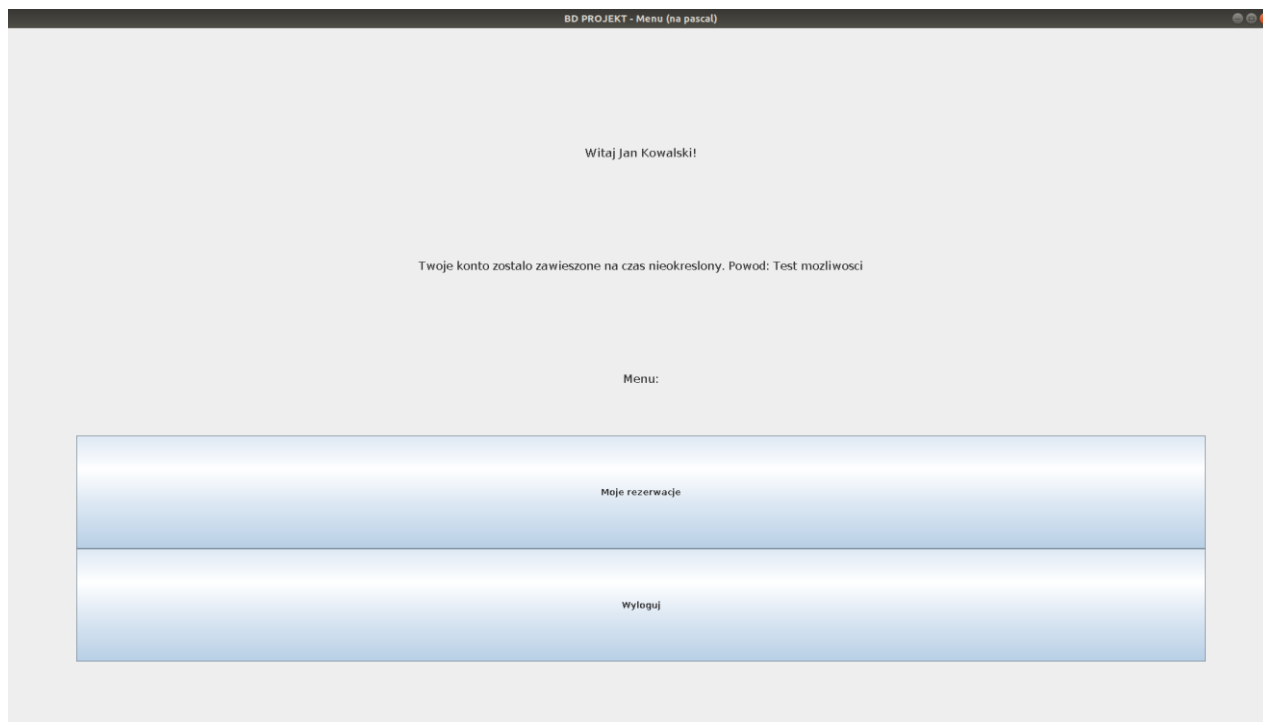
Obrazek 22: Wygląd menu dla HeadAdmina lub Pracowników.

Poniżej przedstawiono menu wyświetlane dla zwykłego klienta serwisu:



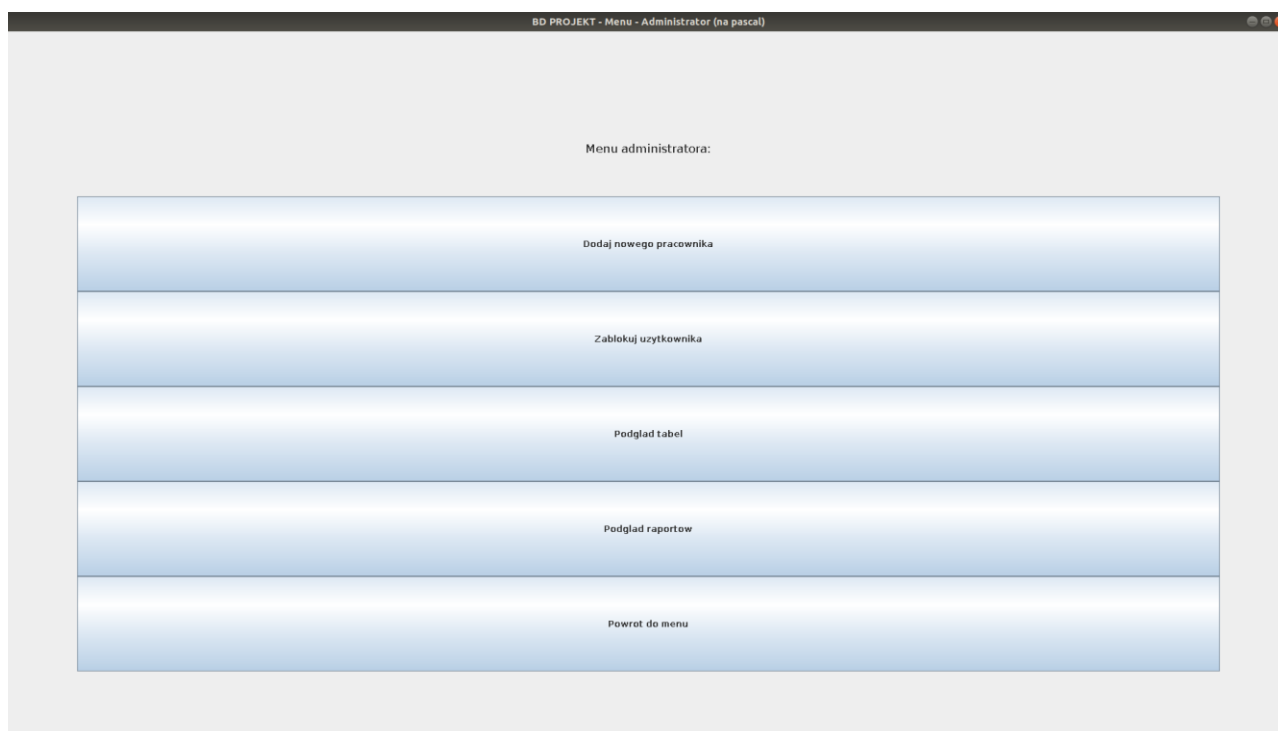
Obrazek 23: Wygląd menu dla Klientów.

Poniżej przedstawiono menu wyświetlane dla zwykłego klienta serwisu, który z jakiegoś powodu został zablokowany:



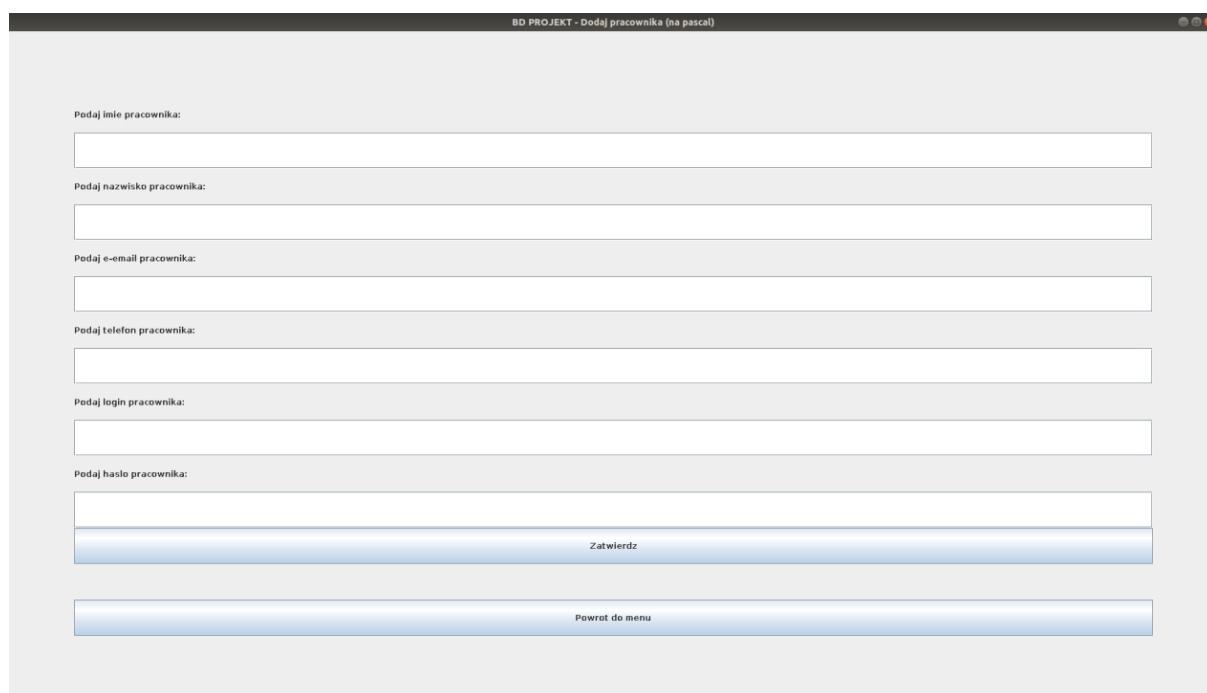
Obrazek 24: Wygląd menu dla Klientów, którzy zostali zablokowani.

Wracając do menu wyświetlanego dla użytkownika zalogowanego jako HeadAdmin lub Pracownik, po wciśnięciu guzika 'Panel administratora' pojawiają się dodatkowe opcje widoczne tylko dla ludzi administrujących hotelem. Poniżej przedstawiono omawiane menu:



Obrazek 25: Wygląd dodatkowego menu dostępnego tylko dla HeadAdmina lub Pracowników.

Po wciśnięciu guzika 'Dodaj nowego pracownika' użytkownik przenoszony jest do obszaru bliźniaczo podobnego do rejestracji nowego użytkownika, który pokazano poniżej:



BD PROJEKT - Dodaj pracownika (na pascal)

Podaj imię pracownika:

Podaj nazwisko pracownika:

Podaj e-mail pracownika:

Podaj telefon pracownika:

Podaj login pracownika:

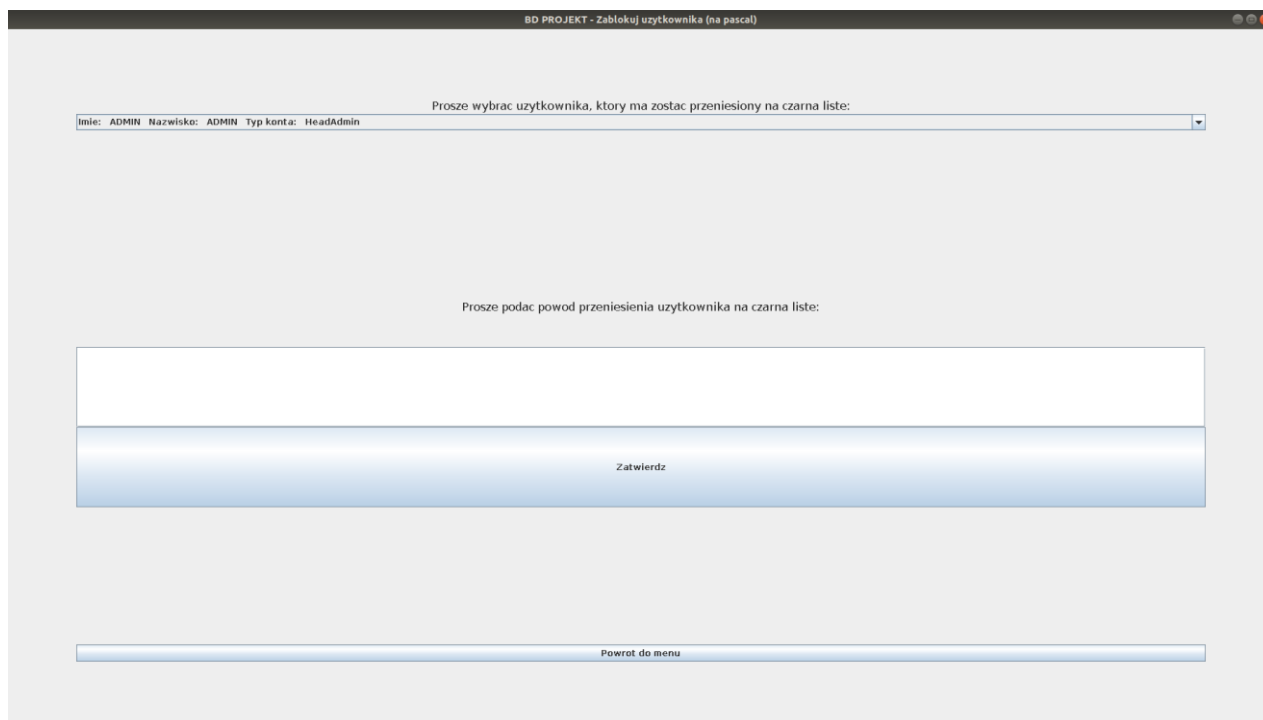
Podaj hasło pracownika:

Zatwierdź

Powrot do menu

Obrazek 26: Wygląd panelu rejestracji nowego pracownika.

Po wciśnięciu guzika 'Zablokuj użytkownika' użytkownik przenoszony jest do obszaru umożliwiającego zablokowanie dowolnego użytkownika, jednakże przed wciśnięciem guzika 'Zatwierdź' konieczne jest podanie przyczyny blokady, ponieważ w przeciwnym wypadku baza danych poinformuje nas o niepowodzeniu operacji. Omawiany obszar pokazano poniżej:



BD PROJEKT - Zablokuj użytkownika (na pascal)

Proszę wybrać użytkownika, który ma zostać przeniesiony na czarna listę:

Imię: ADMIN Nazwisko: ADMIN Typ konta: HeadAdmin

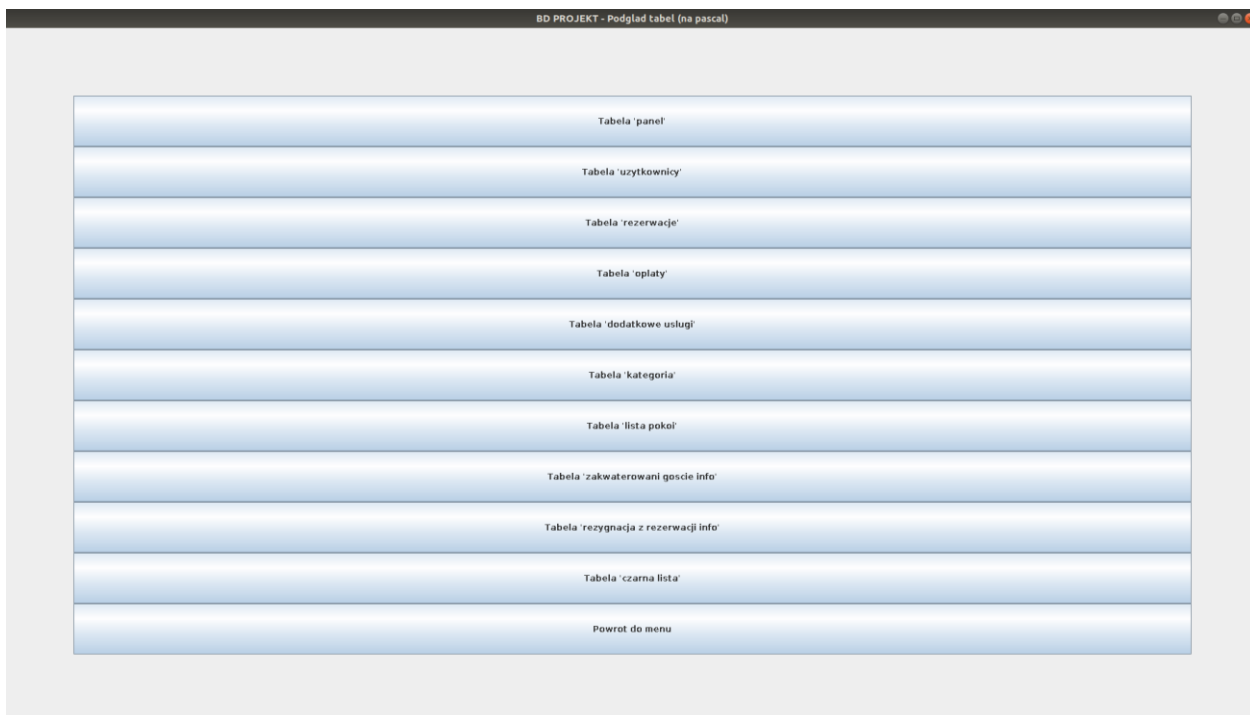
Proszę podać powód przeniesienia użytkownika na czarna listę:

Zatwierdź

Powrot do menu

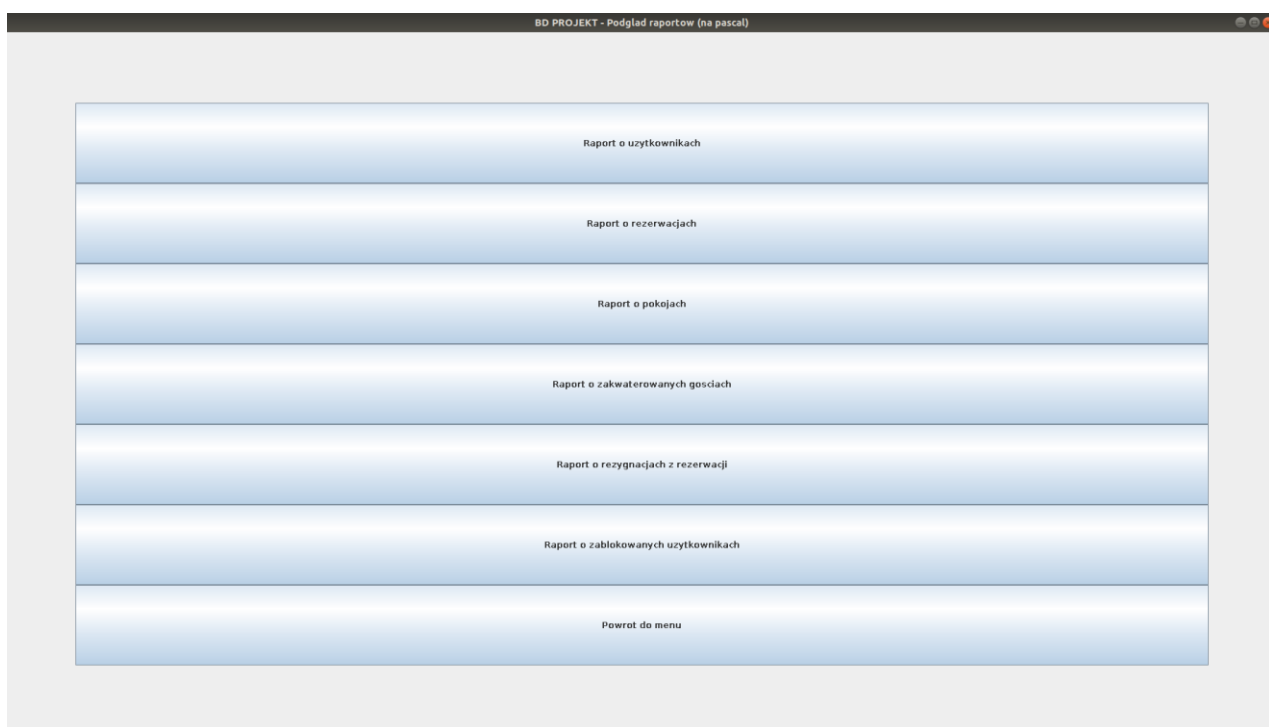
Obrazek 27: Wygląd panelu blokowania użytkownika.

Po wciśnięciu guzika 'Podgląd tabel' użytkownik zostaje przeniesiony do obszaru podglądu wszystkich możliwych tabel z bazy danych. Przykładowy wygląd takiego podglądu został pokazany w poprzednim paragrafie, a poniżej przedstawiono obszar możliwych wyborów:



Obrazek 28: Wygląd panelu podglądu wszystkich tabel.

Po wciśnięciu guzika 'Podgląd raportów' użytkownik zostaje przeniesiony do obszaru podglądu stworzonych raportów. Przykładowy wygląd takiego raportu został pokazany w poprzednim paragrafie, a poniżej przedstawiono obszar możliwych wyborów:



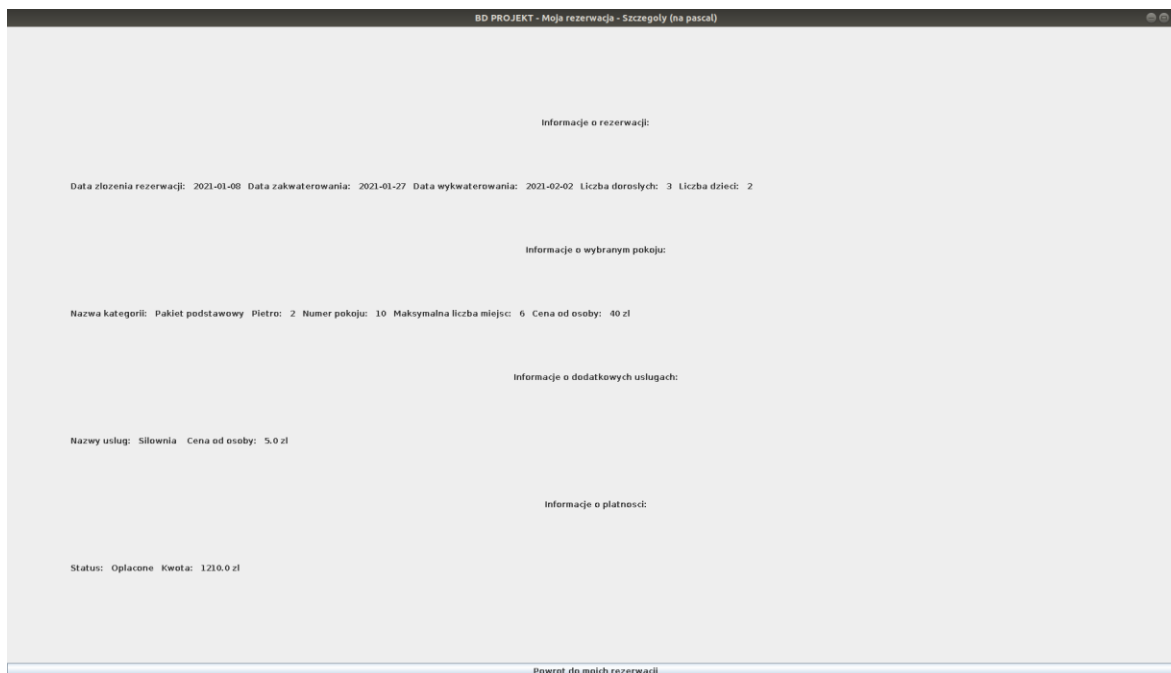
Obrazek 29: Wygląd panelu podglądu wszystkich raportów.

Wracając do głównego menu dostępnego dla wszystkich użytkowników bez względu na typ konta, następnym przyciskiem jest ‘Dokonaj rezerwacji’, po wciśnięciu którego użytkownik zostaje przeniesiony do formularza rezerwacji. Tutaj użytkownik może wybrać datę rezerwacji (która nie może być wcześniej, niż 7 dni od dnia złożenia), datę wykwaterowania, liczbę dorosłych, liczbę dzieci, dodatkowe usługi, żeby to następnie móc wybrać interesujący klienta pokój. Warto wspomnieć, że możliwość wyboru pokoju odblokowuje się dopiero po wypełnieniu wszystkich wcześniejszych pól. Omawiany obszar przedstawiono poniżej:

Obrazek 30: Wygląd panelu składania nowej rezerwacji.

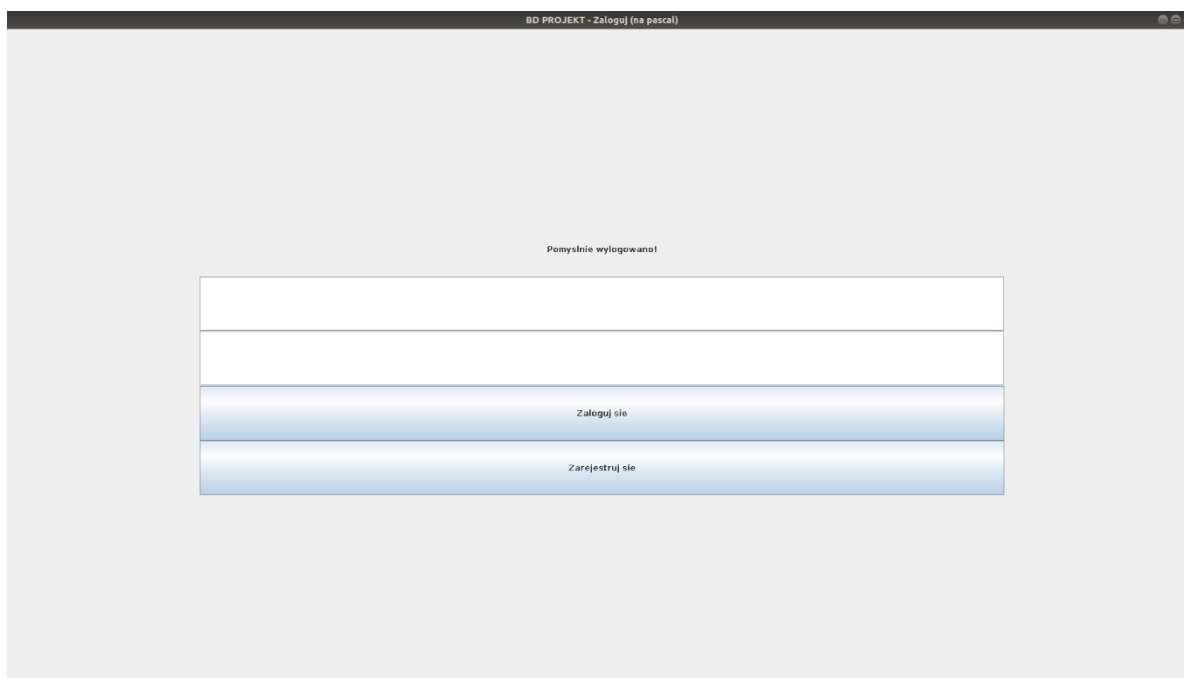
Po wciśnięciu guzika ‘Moje rezerwacje’ użytkownik zostaje przeniesiony do obszaru, w którym może wybrać jedną ze swoich rezerwacji, wyświetlić jej wszystkie szczegóły (przykład został przedstawiony na obrazku 32) lub zdecydować, czy dana rezerwacja ma zostać opłacona, czy też nie. Omawiany obszar przedstawiono poniżej:

Obrazek 31: Wygląd panelu podglądu rezerwacji danego użytkownika.



Obrazek 32: Wygląd panelu pokazującego przykładowe szczegóły dokonanej rezerwacji przez danego użytkownika.

Ostatnim dostępnym guzikiem w menu użytkownika jest przycisk 'Wyloguj', po kliknięciu którego użytkownik zostaje przeniesiony do panelu logowania, a na ekranie zostaje wypisany komunikat informujący o pomyślnym wylogowaniu, co pokazano poniżej:



Obrazek 33: Wygląd panelu logowania po wylogowaniu.

14. Makropolecenia:

Dla ułatwienia obsługi wprowadzania danych zostały zaimplementowane liczne droplisty oraz checkboxy w następujących obszarach:

- formularz dokonania rezerwacji:
 - droplisty wyboru roku, miesiąca oraz dnia zarówno dla daty zakwaterowania, jak i dla daty wykwaterowania,
 - droplisty wyboru liczebności osób pełnoletnich, jak i niepełnoletnich,
 - droplista wyboru pokoju,
 - checkboxy wyboru dodatkowych usług.
- formularz opłaty bądź rezygnacji z rezerwacji:
 - droplista wyboru rezerwacji.
- formularz blokady użytkownika:
 - droplista wyboru użytkownika.
- raporty:
 - droplista wyboru funkcji agregującej,
 - droplista wyboru atrybutu, na którym ma zostać uruchomiona funkcja agregująca,
 - droplista wyboru atrybutu, po którym mają zostać posortowane dane,
 - droplista wyboru atrybutu, według którego ma zostać wyszukany rekord,
 - checkboxy uwzględniające sortowanie do pokazania danych,
 - checkbox uwzględniający konkretna wartość do operacji `'GROUP BY <atrybut> HAVING '%<podana_wartość>%''`.

V. Dokumentacja

15. Wprowadzanie danych:

Dane do bazy mogą zostać wprowadzone w większości jedynie poprzez interfejs użytkownika. Dane można wprowadzać do niemalże wszystkich tabel w sposób manualny tj. wypełnianie odpowiednich formularzy. Wyjątkiem są tabele *pokoj* oraz *kategoria*, do których niezbędne dane do poprawnego działania projektu wprowadzane są z pliku `'SQL/inserty.sql'`. Ponadto baza została wypełniona przykładowymi danymi, które zostały wygenerowane przy pomocy autorskiego generatora rekordów, znajdującego się w lokalizacji `'SQL/InsertGenerator.java'`. Wygenerowane rekordy znajdują się w plikach `'SQL/uzytkownicy.sql'` oraz `'SQL/rezerwacje.sql'`. Dodatkowo do niektórych tabel dane wprowadzane są automatycznie tj. dane o opłacie do tabeli *opлата*, dane o użytkowniku do tabeli *czarna_lista*, jeżeli ten na czas nie opłaci rezerwacji albo zrezygnuje za dużą ilość razy, dane o użytkowniku rezygnującym z rezerwacji do tabeli *rezygnacje_z_rezerwacje_info*, dane o użytkowniku, który opłacił rezerwację do tabeli *zakwaterowani_goscie_info*.

16. Dokumentacja użytkownika:

Aplikacja w całości została napisana w języku Java i dostępna jest wyłącznie na wydziałowym serwerze Pascal ze względu na znajdującą się na nim bazę danych. Uruchomienie projektu zaleca na systemach typu Linux (na systemach typu Windows aplikacja nie zawsze automatycznie otwiera się na pełnym ekranie, dlatego należy manualnie kliknąć przycisk 'Maksymalizuj' w prawym górnym rogu okna). Aby uruchomić program (który został już wcześniej skompilowany), należy zalogować się do sieci `pascal.fis.agh.edu.pl` i z dowolnego miejsca na serwerze wywołać w konsoli polecenie `'bash ~8sudo/BD_PROJEKT/./run.sh'`. Użytkownik może bezproblemowo założyć swoje konto, jednakże, aby zalogować się jako HeadAdmin, należy posłużyć się (login: `admin` hasło: `admin`). Po zalogowaniu użytkownik ma dostępne wszystkie możliwości opisane w paragrafie 13.

17. Opracowanie dokumentacji technicznej:

Aby skompilować program, należy przejść do katalogu `'~8sudo/BD_PROJEKT/SRC'`, w którym to należy wywołać w konsoli polecenie `'javac Main.java'`, a następnie w celu uruchomienia programu wywołać polecenie `'java -cp ../postgresql-9.4.1212.jre7.jar Main'` lub z dowolnego miejsca na serwerze wywołać w konsoli polecenie `'bash ~8sudo/BD_PROJEKT/./compile_and_run.sh'`. Dokumentacja klas oraz metod znajduje się w kodzie źródłowym projektu. Gdyby zaistniała nagła potrzeba 'postawienia od zera na nogi' całej bazy, należy z dowolnego miejsca na serwerze wywołać w konsoli skrypt `'bash ~8sudo/BD_PROJEKT/./rebuild_base.sh'`, który wywołuje plik `'przebuduj.sql'` wprowadzający dane do bazy w następującej kolejności:

- `tabele.sql` – zawiera utworzenie schematu wymaganego do poprawnego działania oraz definicje wszystkich encji,
- `widoki.sql` – zawiera wszystkie widoki potrzebne do poprawnego działania projektu,
- `inserty.sql` – zawiera NIEZBĘDNE rekordy wymagane do poprawnego działania projektu,
- `first_funkcje.sql` – zawiera część funkcji niezbędnych do poprawnego działania projektu,
- `first_triggery.sql` – zawiera część triggerów niezbędnych do poprawnego działania projektu,
- (opcjonalnie) `uzytkownicy.sql` – zawiera wygenerowane rekordy użytkowników,
- (opcjonalnie) `rezerwacje.sql` – zawiera wygenerowane rekordy rezerwacji,
- `last_funkcje.sql` – zawiera pozostałe funkcje potrzebne do poprawnego działania projektu,
- `last_triggery.sql` – zawiera pozostałe triggery potrzebne do poprawnego działania projektu.

Wszelkie funkcje autoryzujące, walidujące wprowadzane dane czy wprowadzające rekordy w sposób automatyczny zostały zdefiniowane bezpośrednio w bazie danych projektu, graficzny interfejs użytkownika wysyła jedynie odpowiednie zapytania do bazy.

18. Wykaz literatury.

- <https://dbdiagram.io/>,
- <https://www.visual-paradigm.com/>,
- https://newton.fis.agh.edu.pl/~antek/index.php?sub=db_wykl,
- <https://www.postgresql.org/docs/>,
- <https://docs.oracle.com/javase/8/docs/>,
- <https://stackoverflow.com/>