Bazy Danych I – dokumentacja projektu "Baza Hoteli"

Marta Dychała

I.Projekt koncepcji, założenia

1. Zdefiniowanie tematu projektu

Tematem projektu było zrealizowanie bazy danych, która symulowała by przeglądarkę hoteli, zarówno dla osoby, która chciałaby zarezerwować pokój w hotelu jak i dla managera, który zarządza hotelem lub kilkoma hotelami. Wynajmujący mógłby poprzez taką aplikację dokonywać rezerwacji lub kupować usługi w hotelach, w których ma zarezerwowane miejsce. Pracownik (lub zarządca) natomiast mógłby sprawdzić swoje wynagrodzenie oraz utworzyć swój własny hotel, w którym mógłby zatrudniać innych pracowników, zmieniać standardy pokoi oraz ceny w hotelu.

2. Analiza wymagań użytkownika

W projekcie użytkownik może posiadać jedną z dwóch ról: Wynajmujący (lub Gość) albo Pracownik. a) Uprawnienia Gościa to:

- logowanie i rejestracja
- rezerwacja pokoi w hotelu
- kupowanie usług w hotelach, w których ma zarezerwowane miejsce
- przeglądanie swoich rezerwacji, usuwanie ich.
- b) Jeśli pracownik nie jest przełożonym żadnego z hoteli jego uprawnienia to:
 - tworzenie hotelu
 - przeglądanie wynagrodzeń w hotelach w których jest zatrudniony
- c) W przypadku jeśli jest przełożonym któregokolwiek z hoteli, otrzymuje on następujące uprawnienia:
 - wyświetlanie własnych hoteli
 - zmienianie cen usług, standardów
 - dodawanie usług, standardów z bazy
 - zmienianie statusów rezerwacji, usuwanie rezerwacji
 - zmienianie standardów pokoi
 - dodawanie pokoi do hotelu
 - zarządzanie pracownikami zmiana wielkości wynagrodzenia, zatrudnianie i zwalnianie pracowników, zwiększanie im uprawnień (dodawanie do właścicieli hotelu)

Każdy z użytkowników serwisu niezależnie od roli może:

wyświetlać informacje o sobie

- przeglądać hotele ich standardy, usługi, pracowników, pokoje
- 3. Zaprojektowanie funkcji
- a) Przed dokonaniem rezerwacji sprawdzane jest czy Gość ma wystarczająco pieniędzy na jej opłacenie, czy pokój w terminie podanym przez gościa nie jest zajęty oraz czy Gość nie posiada w tym czasie innej rezerwacji. Także data rozpoczęcia rezerwacji musi być późniejsza od daty jej zakończenia
- Kiedy pokój dodawany jest do hotelu, przyjmowane jest założenie, że każde z pięter może mieć maksymalnie 5 pokoi, dlatego też kiedy np. pokój dodawany do hotelu jest szóstym, to znajduje się on na 2 piętrze
- c) Kiedy którykolwiek z pokoi, w których znajduje się Gość są zajęte, to nie można "dokonać jego remontu" (zmienić standardu)
- d) Kiedy zostaje utworzony nowy hotel, to za jego właściciela przyjmuje się osobę, która go utworzyła. Realizowane jest to przez odpowiednie wstawienie rekordu do tabeli pracownik_hotel. Sam hotel zaś musi posiadać unikalną nazwę
- e) Przed utworzeniem konta sprawdzane jest, czy imię i nazwisko składają się wyłącznie z małych lub wielkich liter, a numer telefonu składa się z 9 cyfr, natomiast adres email oraz hasło muszą składać się z od 5 do 30 znaków. Następnie dane po utworzeniu konta są normalizowane (imię i nazwisko pisane z wielkich liter, numer telefonu podzielony na 3 części). Sprawdzane jest także, czy adres email jest unikalny w bazie
- f) Kiedy pracownik jest jedynym właścicielem hotelu, nie może się zwolnić, czyli usunąć rekordu z tabeli pracownik_hotel związanego z jego zarządztwem hotelu. Analogicznie nie może znieść sobie uprawnień właściciela, kiedy jest jedynym z nich

I. Projekt diagramów (konceptualny)

4. Zdefiniowanie encji (obiektów) oraz ich atrybutów

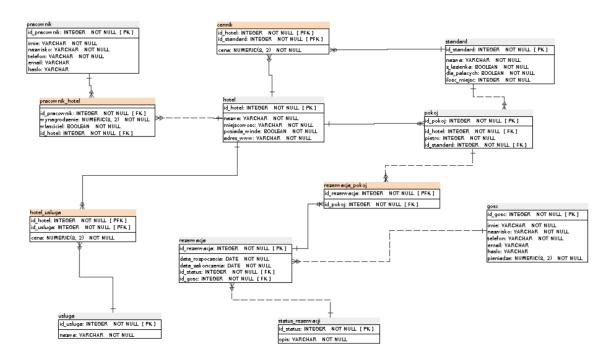
Schemat projekt

- tabela **pracownik** id_pracownik (klucz główny), imie, nazwisko, telefon, email, haslo
- tabela **pracownik_hotel** id_pracownik (klucz obcy do pracownik (id_pracownik)), wynagrodzenie, właściciel, id_hotel (klucz obcy do hotel (id_hotel)),
- tabela hotel id_hotel (klucz główny), nazwa, miejscowosc, posiada_winde, adres_www,
- tabela cennik id_hotel (klucz główny, klucz obcy do hotel (id_hotel)), id_standard (klucz_główny, klucz obcy do standard (id_standard)), cena
- tabela **standard id_standard** (klucz główny, obcy do standard (id_standard)), nazwa, z_lazienka, dla_palacych, ilosc_miejsc,
- tabela pokoj id_pokoj (klucz główny), id_hotel, (klucz obcy do hotel (id_hotel)), pietro, id_standard
- tabela **rezerwacja_pokoj id_rezerwacja** (klucz główny, obcy do rezerwacja (id_rezerwacja)), id_pokoj (klucz obcy do pokoj (id_pokoj))
- tabela **rezerwacja id_rezerwacja** (klucz główny), data_rozpoczecia, data_zakonczenia, id_status (klucz obcy do status_rezerwacji (id_status)), id_gosc
- tabela status_rezerwacji id_status (klucz główny), opis

- tabela gosc id_gosc (klucz główny), imie, nazwisko, telefon, email, haslo, pieniadze
- tabela usluga id_usluga (klucz główny), nazwa
- tabela **hotel_usluga id_hotel** (klucz główny, obcy do hotel (id_hotel)), **id_usluga** (klucz główny, obcy do usluga (id_usluga)), cena

5. Zaprojektowanie relacji pomiędzy encjami

Relacje pomiędzy encjami zostały przedstawione na poniższym diagramie ERD:



III Projekt logiczny

W każdej z tabel poza tabelą pracownik_hotel znajduje się co najmniej jeden klucz główny. Wynika to z faktu, że przyjęto założenie, że hotel może posiadać wielu pracowników oraz pracownik może pracować w wielu hotelach.

Także hotel_usluga jest encją asocjacyjną – jeden hotel może posiadać wiele usług oraz jedna usługa może być w wielu hotelach

Analogicznie jest w przypadku tabeli cennik – jest ona encją asocjacyjną między tabelą hotel a standard, ponieważ dany stardard może mieć różne ceny w innych hotelach, a hotel może posiadać wiele standardów

Tabela rezerwacja_pokoj znajduje się w bazie, ponieważ dany pokój może posiadać wiele rezerwacji w różnych terminach.

Tabela status_rezerwacji połączona jest z tabelą rezerwacja relacją 1:n, ponieważ status może być przypisany dla wielu rezerwacji

Gość może posiadać wiele rezerwacji – dlatego jest połączony z tabelą rezerwacja relacją 1:

6. Słowniki danych:

tabela pracownik

- id_pracownik (klucz główny) integer,
- o imie varchar,
- o nazwisko varchar,
- o telefon varchar z 9 znaków,
- email varchar unikatowy,
- o haslo varchar

- tabela pracownik_hotel

- o id_pracownik (klucz obcy do pracownik (id_pracownik)) integer,
- wynagrodzenie numeric o maksymalnie 8 cyfrach z dokładnością 2 miejsc po przecinku,
- właściciel boolean,
- o id_hotel (klucz obcy do hotel (id_hotel)) integer,

- tabela hotel

- o id_hotel (klucz główny) integer,
- nazwa varchar unikatowy,
- o miejscowość varchar,
- o posiada_winde boolean,
- adres_www varchar,

- tabela cennik

- o id_hotel (klucz główny, klucz obcy do hotel (id_hotel)) integer,
- o id_standard (klucz_główny, klucz obcy do standard (id_standard)) integer,
- o cena numeric o maksymalnie 8 cyfrach z dokładnością 2 miejsc po przecinku

- tabela standard

- o id_standard (klucz główny, obcy do standard (id_standard)) integer,
- o nazwa varchar,
- z_lazienka boolean,
- o dla palacych boolean,
- o ilosc_miejsc integer,

- tabela **pokoj**

- o id_pokoj (klucz główny) integer,
- id_hotel, (klucz obcy do hotel (id_hotel)) integer
- o pietro integer,
- o id standard integer

- tabela rezerwacja_pokoj

- o id_rezerwacja (klucz główny, obcy do rezerwacja (id_rezerwacja)) integer,
- o id_pokoj (klucz obcy do pokoj (id_pokoj)) integer

- tabela rezerwacja

- o id_rezerwacja (klucz główny) integer,
- o data_rozpoczecia date, nie większa od data_zakonczenia,
- o data_zakonczenia date nie mniejsza od data_rozpoczecia,
- o id_status (klucz obcy do status_rezerwacji (id_status)) integer,
- id_gosc integer

- tabela status_rezerwacji

- id_status (klucz główny) integer,
- o opis varchar

- tabela **gosc**

- o id_gosc (klucz główny) integer,
- o imie varchar,
- o nazwisko varchar,
- o telefon varchar składający się z 9 znaków,
- email varchar unikalny,
- o haslo varchar,
- o pieniadze numeric o maksymalnie 8 cyfrach z dokładnością 2 miejsc po przecinku

- tabela usluga

- o id_usluga (klucz główny) integer,
- o nazwa varchar

- tabela hotel_usluga

- id_hotel (klucz główny, obcy do hotel (id_hotel)) integer,
- o id_usluga (klucz główny, obcy do usluga (id_usluga)) integer,
- o cena numeric o maksymalnie 8 cyfrach z dokładnością 2 miejsc po przecinku

Wszystkie dane w tabelach nie mogą mieć wartości null

- 7. Zaprojektowanie operacji na danych
- Projektowane funkcje załączone są we folderze sql w pliku functions.sql, wyzwalacze w pliku triggers.sql, natomiast widoki we views.sql. Aby dodawanie usług i standardów oraz zmiana statusu rezerwacji działała poprawnie należy wykonać plik insert_data.sql, który wypełnia bazę przykładowymi danymi.

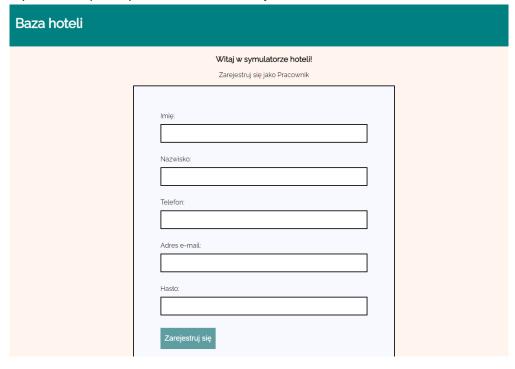
IV. Projekt funkcjonalny

- 8. Interfejsy do prezentacji, edycji i obsługi danych
- a) Formularz do logowania Gościa/Pracownika od strony bazy danych sprawdzane jest czy adres email istnieje w bazie oraz sprawdzana jest poprawna nazwa użytkownika lub hasło. Formularze wyglądają niemal identycznie, dlatego poniżej pokazany jest jedynie formularz gościa:

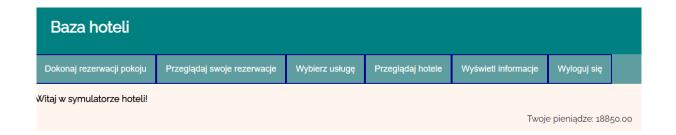
Baza hoteli	
	Witaj w symulatorze hoteli! Zaloguj się jako Gość
	E-mail: Hasto: Zaloguj się
Nie masz konta? Zarejestruj się!	
Rejestracja Powrót do menu główneg	go

Po zalogowaniu się użytkownik przekierowany jest na stronę główną, gdzie znajdują się przyciski "Panel pracownika" oraz "Panel wynajmującego". Jeśli użytkownik jest zalogowany, przyciski te przekierowują do strony pracownika lub gościa, w przeciwnym razie pojawia się strona z formularzem logowania.

a) formularz do rejestracji Gościa/Pracownika – z poziomu bazy danych sprawdzane jest czy użytkownik o podanym adresie email istnieje w bazie:



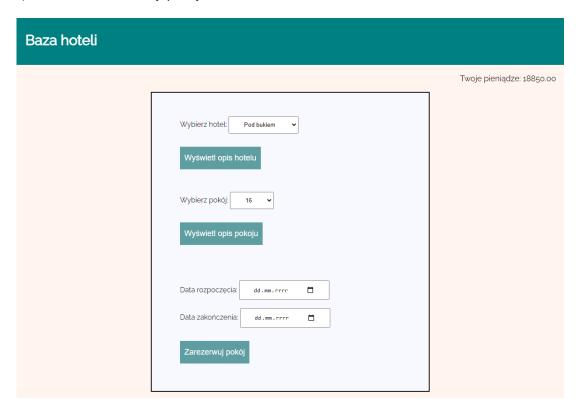
b) Pasek nawigacyjny panelu gościa:



c) Pasek nawigacyjny panelu pracownika:

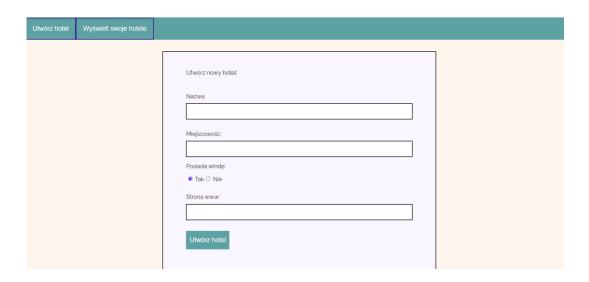


d) Formularz rezerwacji pokoju:

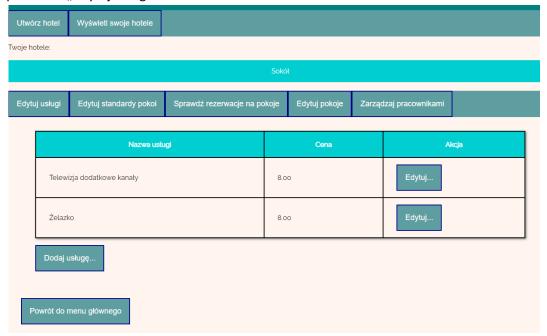


Dostępny jest z poziomu gościa. Z listy rozwijanej można wybrać hotel z bazy następnie za pomocą przycisków "Wyświetl opis hotelu" lub "Wyświetl opis pokoju" sprawdzić informacje o hotelu lub pokoju.

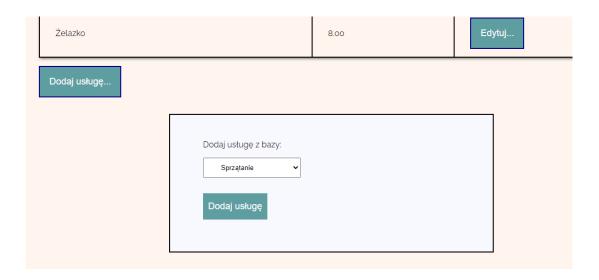
e) Formularz tworzenia nowego hotelu – po naciśnięciu "Zarządzaj hotelami" -> "Utwórz hotel" sprawdzane jest czy hotel o podanej nazwie istnieje w bazie



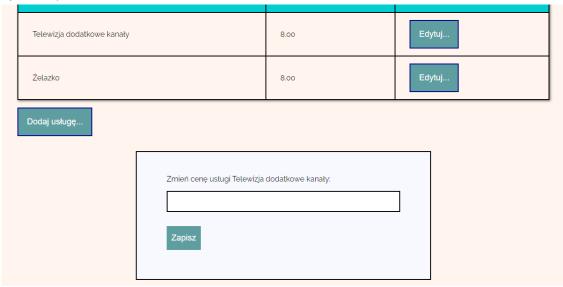
f) Zarządzanie hotelem – po naciśnięciu "Wyświetl swoje hotele" pojawia się podmenu, w którym można zarządzać posiadanymi przez pracownika hotelami. Można m.in. dodawać oraz zmieniać ceny usług lub standardów. Przykładowo zostało to przedstawione dla podmenu "Edytuj usługi":



Po naciśnięciu "Dodaj usługę" pojawia się formularz z listą rozwijaną standardów znajdujących się w bazie, ale nie w danym hotelu:



Natomiast po naciśnięciu "Edytuj" pojawia się formularz z możliwością zmiany ceny usługi. Dane wprowadzane są przez pole tekstowe na którym sprawdza się czy wprowadzone dane są numeryczne:



g) Formularz zarządzania pracownikiem – po przejściu na zakładkę "Zarządzaj pracownikami" pojawia się tabela z listą pracowników w hotelu. Po naciśnięciu "Zarządzaj" w odpowiednim wierszu pojawia się formularz, w którym można zwolnić pracownika, uczynić go właścicielem/zdegradować z roli właściciela oraz zmienić wynagrodzenie pracownikowi



9. Wizualizacja danych

Do wizualizacji danych wykorzystano tabele zawierające wyszukane rekordy. Większość raportów znajduje się w zakładce "Przeglądaj hotele". Przykładowo poniżej znajduje się raport listujący usługi w jednym z hoteli wraz z cenami usług:



V. Dokumentacja

12. Wprowadzanie danych

Dane początkowe, na których testowano aplikację zostały wprowadzone ręcznie (sql/insert_data.sql). Większość danych wprowadzanych w aplikacji odbywa się za pomocą

formularzy. Najwięcej uprawnień posiada właściciel, który poprzez formularze może m.in. tworzyć hotele, dodawać do nich pokoje, zmieniać ceny w hotelu. Gość może jedynie tworzyć i usuwać rezerwacje oraz zamawiać usługi.

12. Dokumentacja użytkownika

Użytkownik może wybrać odpowiedni panel ze strony startowej a następnie zalogować się na niego. Po zalogowaniu w zależności od poziomu dostępu może wykonywać określone dla niego operacje, tj. wyświetlanie oraz modyfikowanie danych. Możliwe operacje znajdują się w większości przypadków na pasku nawigacyjnym aplikacji

16. Opracowanie dokumentacji technicznej

Projekt został wykonany przy użyciu Node.js. Aby uruchomić projekt należy wpisać komendę **node server.js**. Projekt powinien się uruchomić na porcie 34343.