

Gliwice, 19.06.2020

Programowanie obiektowe i graficzne

projekt zespołowy

"Rezerwacja pobytu hotelowego"

Skład zespołu projektowego:

Przemysław Chodór gr. lab. 3G

Miękina Michał gr. lab. 3G

Bartosz Hajduk gr. lab. 3G

Wykaz zadań i prac zrealizowanych przez zespół projektowy:

Imię Nazwisko	Odpowiedzialny za:	Zrealizował zadania:
Hajduk Bartosz	1. Zaprojektowanie bazy danych	1A. Zaprojektował Hajduk Bartosz
		1B. Wykonali: Hajduk Bartosz, Michał Miękina, Chodór Przemysław
		1C. Sprawdzili: Hajduk Bartosz, Michał Miękina, Chodór Przemysław
	2. Zaprojektowanie widoku aplikacji	2A. Zaprojektował Hajduk Bartosz
		2B. Poprawił Chodór Przemysław
Przemysław Chodór	3. Połączenie bazy danych z aplikacją	3A. Zaprojektował: Chodór Przemysław
		3B. Realizowali: Chodór Przemysław, Hajduk Bartosz, Miękina Michał
Michał Miękina	4. Testowanie oraz szukanie błędów po stronie bazy oraz aplikacji	4A. Realizował Michał Miękina
Przemysław Chodór	5. Tworzenie aplikacji WPF	5A. Realizowali: Chodór Przemysław, Miękina Michał, Hajduk Bartosz
		5B. Sprawdził: Miękina Michał

1. Opis projektu.

Projekt powinien pozwalać na przeprowadzanie procesu rezerwacji pobytu hotelowego dla bazy danych z wykorzystaniem technologii SQL, zawierając w sobie funkcjonalność pozwalającą na intuicyjne oraz logiczne przeprowadzenie rezerwacji w sieci hoteli przez użytkownika domyślnie nazwanego przez nas recepcjonistą, a to wszystko we wzorcu MVVM.

2. Wymagania

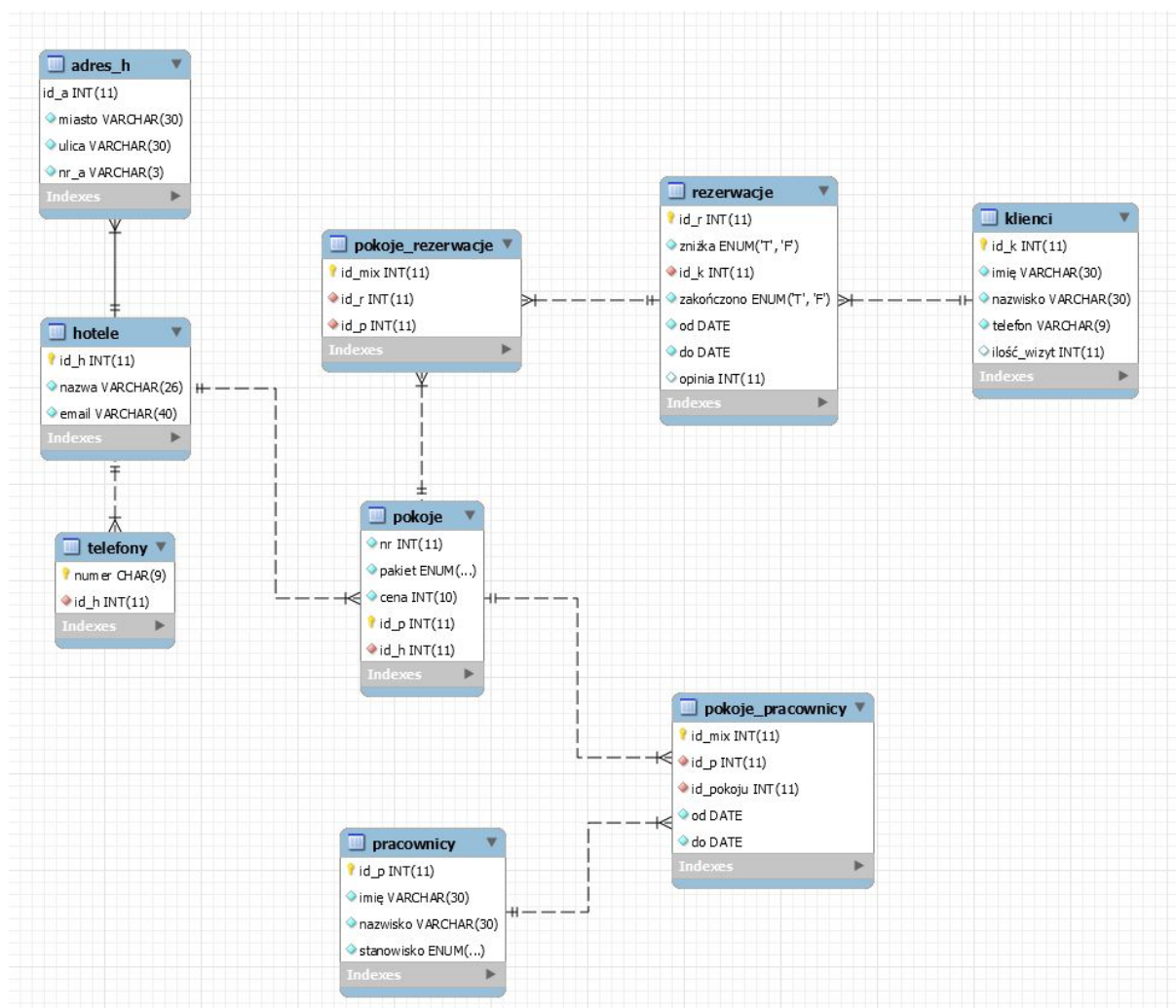
Wymagania:

1. Zastosowanie wzorca MVVM
2. Stosowanie się do zasad programowania obiektowego
3. Połączenie aplikacji z bazą danych SQL za pomocą warstwy DAL
4. Implementacja funkcjonalności pozwalającej na składanie rezerwacji z walidacją dat rozpoczęcia oraz zakończenia(brak kolizji dat rezerwacji)
5. Nadanie użytkownikowi uprawnień potrzebnych do użytkowania bazy danych, ale nie wykraczających poza potrzeby na rzecz aplikacji
6. Przedstawienie aplikacji w logiczny sposób, aby osoba intuicyjnie wiedziała jak z niej skorzystać
7. Możliwość oceny pobytu w hotelu (funkcjonalności nie zdążyliśmy zaimplementować)
8. Możliwość udzielenia zniżek na podstawie ilości wizyt przez danego użytkownika(funkcjonalności nie zdążyliśmy zaimplementować)

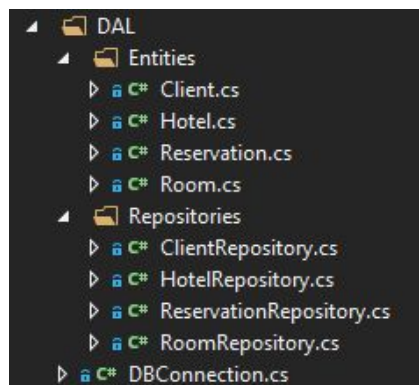
3. Przebieg realizacji

Rozpoczęliśmy od zaprojektowania funkcjonalnego widoku, następnie przeszliśmy do projektowania modelu naszej aplikacji.

Po ustaleniu na co dokładnie ma pozwalać aplikacja przeszliśmy do konsekwentnej realizacji implementacji warstwy DAL bazując na modelu relacyjnym naszej bazy danych, który prezentuje się następująco:



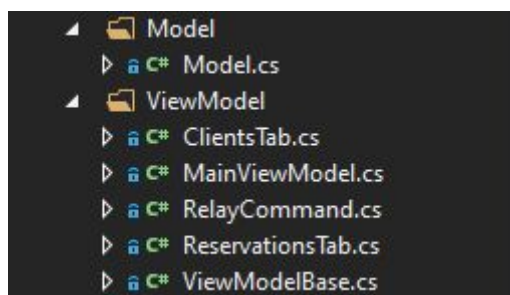
Stworzyliśmy bazowe klasy przechowujące informacje dotyczące encji(Entities), klasy odzwierciedlające działania po stronie bazy danych SQL(Repositories) oraz klasę pozwalającą na połączenie z bazą(DBConnection) za pomocą domyślnego użytkownika “repcjonisty”:



Użytkownik “repcjonista” posiada uprawnienia do: insert, select, update oraz delete dla tabel klienci, rezerwacje, pokoje_rezerwacje. Posiada także uprawnienia select dla tabel hotele oraz pokoje. Dane użytkownika “repcjonista” w aplikacji prezentują się następująco:

	Name	Type	Scope	Value
	server	string	Application	localhost
	database	string	Application	hotel
	userID	string	Application	repcjonista
	paswd	string	Application	haslo.123
	port	uint	Application	3306
*				

Po stworzeniu solidnej podstawy przeszliśmy do implementowania ViewModel potrzebnego do obsłużenia widoku:



Gdzie ClientsTab oraz ReservationTab są klasami reprezentujące poszczególne zakładki w aplikacji. MainViewModel jest klasą spajającą model z klasami zakładek. Ostatecznie mamy jeszcze klasy RelayCommand oraz ViewModelBase potrzebne do implementacji MVVM.

Implementacja funkcjonalności przebiegała następująco:

- Implementacja listy hoteli pozwalającej na wybór hotelu, na którym będziemy operować
- Szukanie klienta na podstawie jego nazwiska, w razie pozostawienia pustego pola program wyszukuje wszystkich użytkowników w bazie danych
- Dodanie statycznej listy pakietów
- Dodanie możliwości dodawania, edycji oraz usuwania klientów
- Dodanie wyświetlania oraz wyboru pokoju na podstawie wybranego pakietu
- Dodanie wyświetlania listy zrobionych rezerwacji po wybraniu pokoju
- Dodanie tworzenia i usuwania rezerwacji
- Naprawienie tworzenia rezerwacji - weryfikacja wolnego pokoju w wybranym przedziale czasowym
- Dodanie możliwości zmienienia statusu rezerwacji na zakończony i na odwrót(brak reakcji widoku na zmianę)
- Poprawienie nazewnictwa w aplikacji, oraz zoptymalizowanie kodu pod kątem wydajności
- Naprawienie zmiany statusu rezerwacji po stronie widoku
- Poprawa błędów nazewnictwa oraz kolizji oznaczeń

4. Instrukcja użytkownika

Aplikacją operujemy za pomocą:

The screenshot shows a web application window titled "Rezerwacje". It has two tabs: "Rezerwacje" (1) and "Klienci" (2). The "Rezerwacje" tab is active. On the left, there are several input fields: "Hotel:" (3) with a dropdown menu showing "Mokoszewski"; "Szukaj klienta:" (4) with a text input and a "Szukaj" button; a list of search results (5) including "Karol Niedźwiedź 945455454", "Karolina Ryś 111111111", "Natalia Pruch 294245678", and "Karolina Niewaszka 124214241"; "Pakiet:" (6) with a dropdown menu showing "standard"; "Pokój:" (7) with a list of room options: "Numer: 4 Cena: 100zł" and "Numer: 5 Cena: 130zł"; "Od:" (8) with a date input showing "09.07.2020"; and "Do:" (9) with a date input showing "24.07.2020". On the right, there is a table (10) with columns: "Imię", "Nazwisko", "Zniżka", "Zakończono", "Od", and "Do". The table contains three rows of reservation data. At the bottom, there are three buttons: "Dodaj" (11), "Zakończ" (12), and "Usuń" (13).

Imię	Nazwisko	Zniżka	Zakończono	Od	Do
Karol	Niedźwiedź	F	F	08.07.2020	08.07.2020
Karolina	Ryś	F	T	08.07.2020	08.07.2020
Karolina	Niewaszka	F	F	08.07.2020	08.07.2020

1. Rezerwacje - zakładka rezerwacje będąca podstawą aplikacji
2. Klienci - zakładka pozwalająca na zarządzanie klientami
3. Hotel - lista pozwalająca na wybór hotelu na którym chcemy operować
4. Szukaj klienta - pole tekstowe pozwalające na wpisanie nazwiska szukanego klienta(w przypadku pustego pola program wyszukuje wszystkich klientów)
5. Lista klientów - lista zawierająca klientów wyszukanych przez szukanie klienta
6. Pakiet - wybór pakietu według którego następnie są wyszukiwane pokoje o danym pakiecie w wybranym hotelu
7. Pokój - lista pokoi odpowiadająca danemu hotelowi oraz pakietowi
8. Od - data początku rezerwacji
9. Do - data końca rezerwacji
10. Lista rezerwacji - lista przedstawiająca wszystkie rezerwacje dla danego pokoju
11. Dodaj - przycisk pozwalający na dodanie rezerwacji po wybraniu hotelu, pokoju, klienta oraz dat początku i końca nie kolidujących z istniejącymi rezerwacjami
12. Zakończ - przycisk zmieniający status Zakończono rezerwacji z 'F' na 'T' i na odwrot
13. Usuń - przycisk pozwalający na usunięcie rekordu z listy rezerwacji

Rezerwacje

Rezerwacje Klienci

Imię: **14**

Nazwisko:

Telefon:

Imię	Nazwisko	Telefon
Karol	Niedźwiedź	945455454
Karolina	Ryś	111111111
Natalia	Pruch	294245678
Karolina	Niewaszka	124214241

15

16 **17** **18**

Dodaj Edytuj Usuń

14. Dodawanie klienta - pozwala na dodanie nowych klientów do bazy

15. Lista klientów - pozwala na zarządzanie klientami

16. Dodaj - pozwala na dodanie nowego użytkownika na podstawie wprowadzonych danych w okienku 14

17. Edytuj - pozwala na zmianę danych zaznaczonego użytkownika na podstawie wprowadzonych danych w okienku 14

18. Usuń - pozwala na usunięcie zaznaczonego klienta z bazy

5. Podsumowanie i wnioski.

Zrealizowaliśmy większość początkowych założeń. Nie udało się zaimplementować oceniania rezerwacji na zakończenie rezerwacji oraz systemu zniżek, który miał naliczać się na podstawie ilości wizyt klienta w hotelu. Problemy wynikały z nieprzemysłanej budowy bazy danych oraz z powodu indywidualnych różnic w podejściu do pewnych aspektów projektu. Ostatecznie projekt zawiera funkcjonalność wystarczającą, którą można rozbudować w znacznym stopniu w przyszłości.

6. Dodatek - udokumentowanie wykorzystania systemu kontroli wersji.

350c4c62	bartosz	08.07.2020 17:16:39	Dodanie usuwania rezerwacji
af902a24	Przemysław Chodór	08.07.2020 18:06:05	Dodano usuwanie i dodawanie
482bbcb3	bartosz	08.07.2020 17:16:39	Dodanie usuwania rezerwacji
04058bab	Przemysław Chodór	08.07.2020 17:14:07	Dodawanie rezerwacji.
105671a3	Przemysław Chodór	08.07.2020 13:43:12	Wyświetlanie rezerwacji po wybraniu pokoju.
823e009a	Przemysław Chodór	08.07.2020 11:50:17	Dodano wybór pokoju
ae7244db	Przemysław Chodór	07.07.2020 16:29:13	Zmieniono układ, dodano edycję i usuwanie klientów.
c1fd630a	bartosz	06.07.2020 23:49:47	Dodanie dodawania klientów do bazy danych
ea3094d9	bartosz	06.07.2020 22:12:33	Dodano szukanie klientów po nazwisku.
ca4f6f48	Przemysław Chodór	06.07.2020 20:47:41	Pojedyncze wczytywanie bazy hoteli
535147e3	Przemysław Chodór	06.07.2020 20:42:03	Zmiana listy hoteli
5767e2fd	Przemysław Chodór	06.07.2020 16:17:55	Zmiana na model jak w prezentacji
4568b396	Przemysław Chodór	06.07.2020 15:04:36	Zmieniono organizację warstw
97e5a870	bartosz	05.07.2020 23:39:04	Dodano listę hoteli, listę pakietów oraz listę klientów.
062c359b	bartosz	04.07.2020 17:08:58	Próba
6d466a42	bartosz	04.07.2020 16:59:49	Add .gitignore and .gitattributes.

63bb7434	Przemysław Chodór	08.07.2020 21:56:08	Poprawa błędów i nazw
ecac49a7	bartosz	08.07.2020 21:48:24	Naprawione zakończenie rezerwacji
597b5069	Przemysław Chodór	08.07.2020 21:54:03	Poprawa nazw i błędów
ee83b509	Przemysław Chodór	08.07.2020 21:14:54	Dodano zakończenie rezerwacji
471821c4	Przemysław Chodór	08.07.2020 21:11:11	Merge pull request #1 from PrzChodor/Zfuzjuj_to_Przemek
c4fe532a	bartosz	08.07.2020 21:03:59	Zfuzjowanie
e8feaf2f	bartosz	08.07.2020 20:54:19	Dodanie (brak widoku) zakończenia rezerwacji
098f60bb	Przemysław Chodór	08.07.2020 21:08:36	Dodano sprawdzanie czy data jest poprawna
a929971a	Przemysław Chodór	08.07.2020 20:52:01	Dodano sprawdzenie czy pokój jest wolny
f6609a29	Przemysław Chodór	08.07.2020 19:33:09	Naprawiono dodawanie