Projekt PROO

Zespół:
Mikołaj Bańkowski 310408
Rafał Zan 311214

Hotelooo app

1.Cel projektu:

a) Przeznaczenie:

Na projekt składa się aplikacja z interfejsem użytkownika pozwalająca na rezerwację pokoju hotelowego w danym mieście oraz hotelu.

b) Sposób działania i jego efekty:

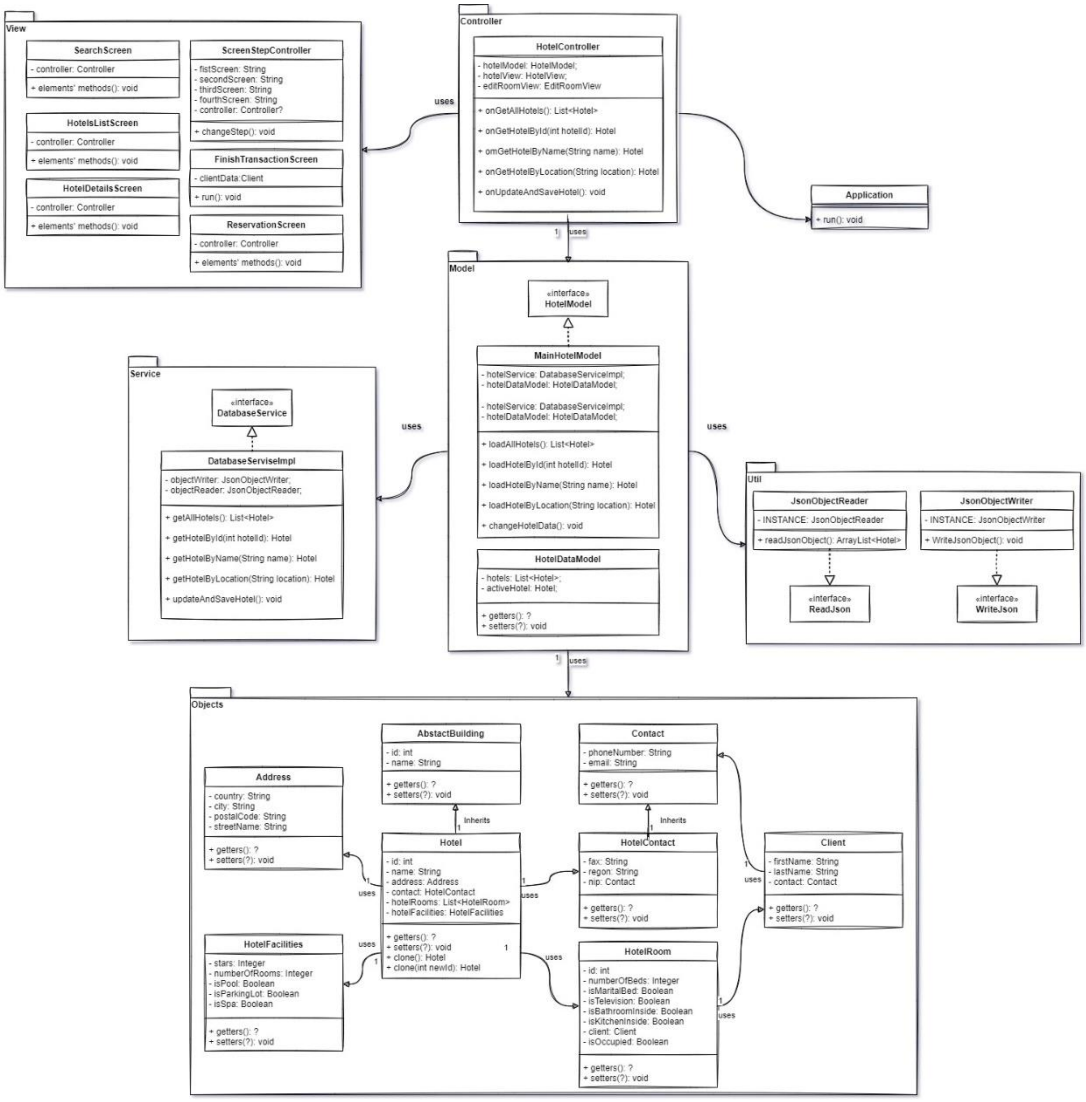
Klient dostanie możliwość wyszukania wszystkich hoteli w danej miejscowości (jeśli czas pozwoli będzie możliwość wyszukiwania także według innych kryteriów, takich jak nazwa hotelu, udogodnienia itp.). Wybrania z listy interesującego go obiektu a następnie zarezerwowania na swoje (na chwilę obecną) podstawowe dane osobowe danego pokoju.

c) Potrzebne pliki:

Jedynym potrzebnym plikiem jaki będzie wymagany do działania aplikacji to baza danych wszystkich dostępnych hoteli wraz ze zdjęciami tychże hoteli.

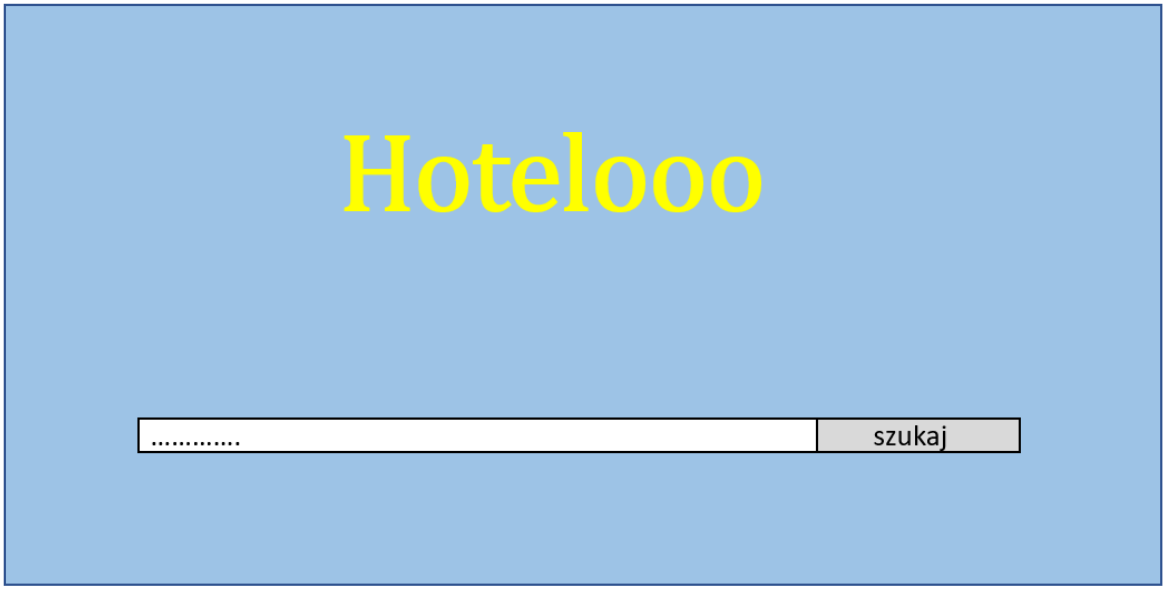
2.Propozycja logo:

3.Diagram UML architektury klas:

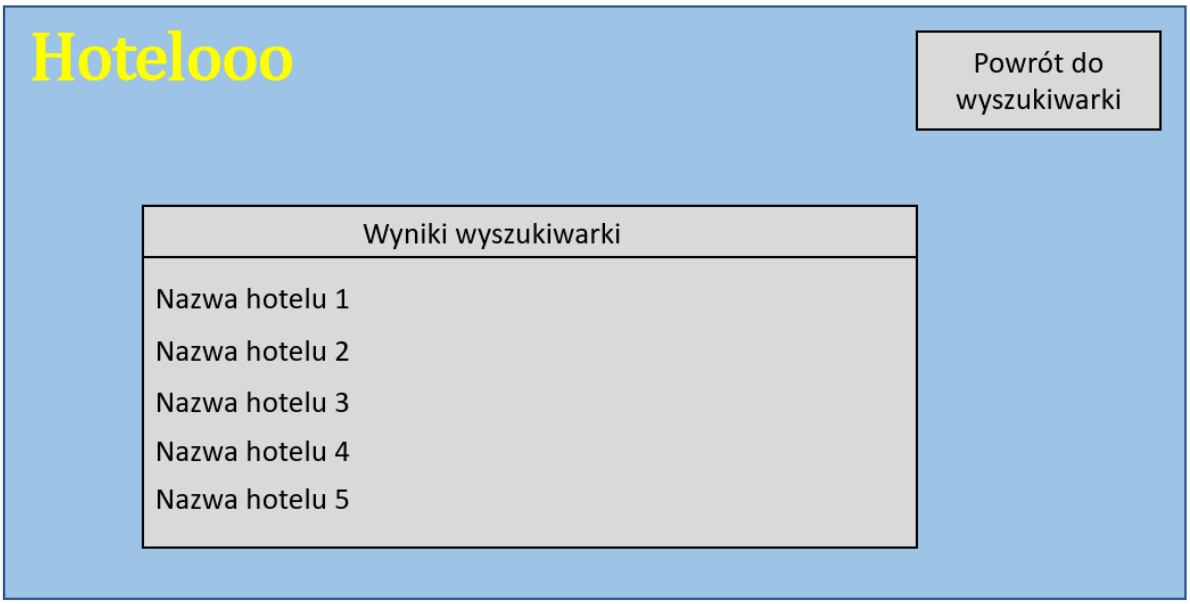


4.Makiety ekranów interfejsu użytkownika:

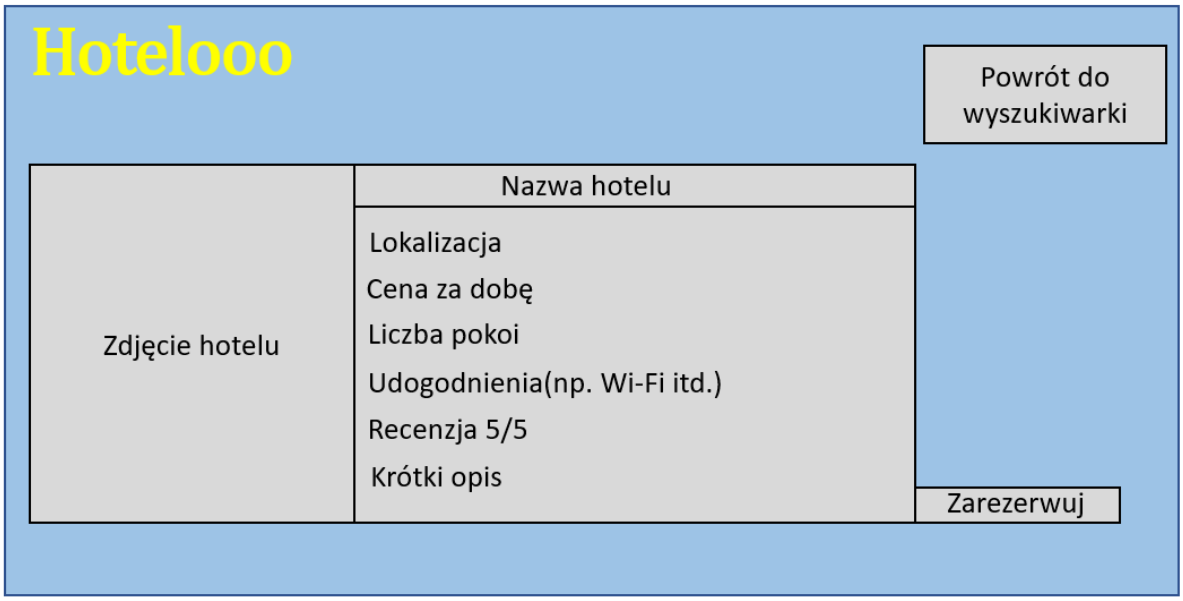
Ekran wyszukiwania:



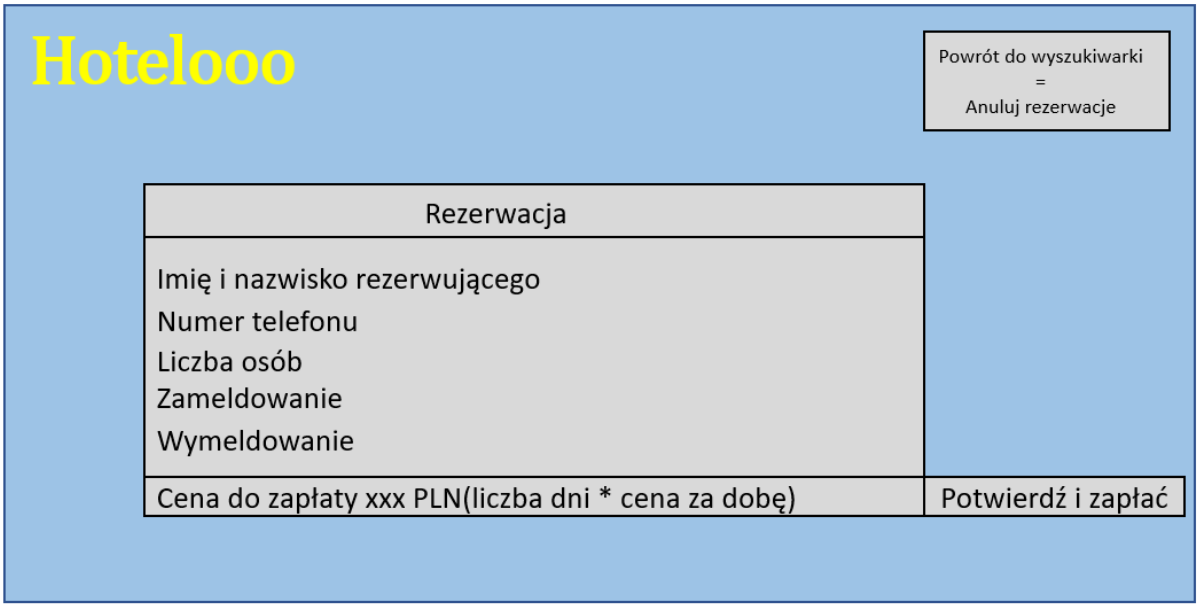
Lista dostępnych obiektów:



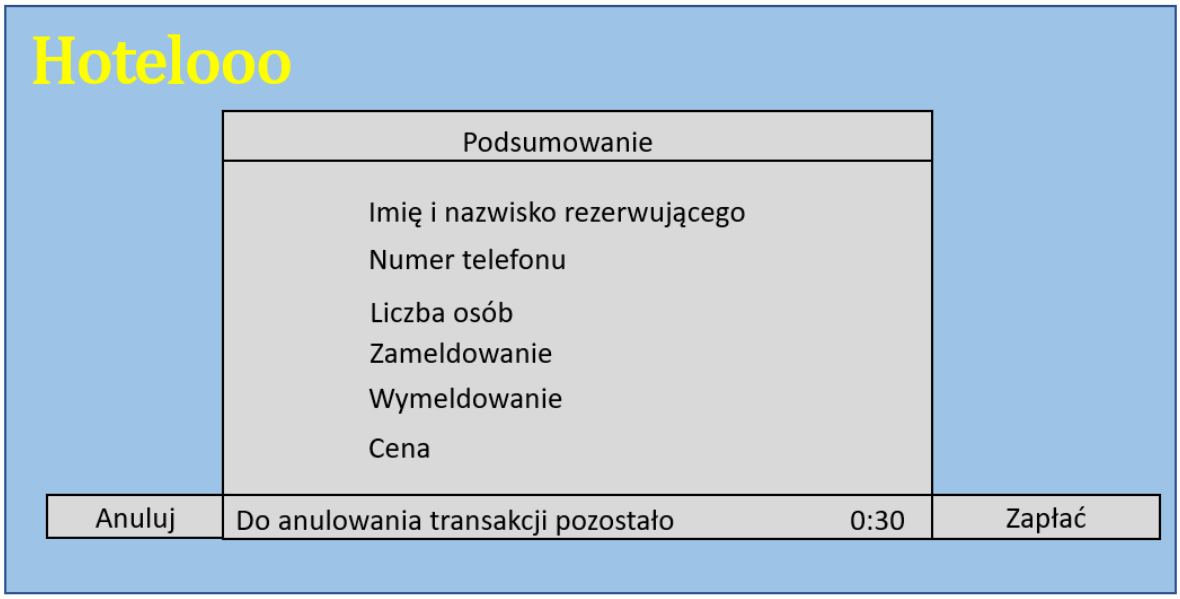
Dane wybranego hotelu:



Ekran rezerwacji:



Ekran finalizacji procesu rezerwacji pokoju:



5.Opis zależności i zastosowań poszczególnych modułów:

Moduł Service:

- znajdą się w nim klasy, których zadanie będzie polegało na wystawieniu podstawowych operacji na bazie dla reszty programu w tym przypadku modułu model, o którym zaraz. Moduł ten będzie w sobie zawierał klasy DatabaseService wystawiające gotowe, odczytane z bazy obiekty, za pomocą spersonalizowanych getterów.

Moduł Util:

- moduł ten będzie zawierać klasy ObjectReader oraz ObjectWriter, których zadanie będzie polegało na odczycie i zapisie obiektów do pliku. Będzie on zawierał także odpowiednie interfejsy do opisanych w poprzednim zdaniu klas.

- *Jeśli czas pozwoli w module tym pojawią się także klasy odpowiedzialne za walidację wprowadzonych i wyświetlanych danych, oraz error handlery.

Moduł Model:
- moduł ten odpowiedzialny będzie za całą logikę biznesową aplikacji. Zawierać będzie on klasy odpowiedzialne za wystawianie danych, do kontrolera, najważniejszymi klasami będą HotelModel, MainHotelModel oraz interfejsy. Będzie tu także ekran wątku przypominającego o dokończeniu rezerwacji hotelu.

Moduł Objects:
- zadaniem tego podmodułu będzie przechowywanie wszystkich obiektów z których korzysta aplikacja, czyli m.in. Hotel, Client, Address, Contact itp.

Moduł View:

- moduł ten odpowiedzialny będzie za przechowywanie całej logiki ekranów(czyli interfejsu użytkownika), wyświetlanie, aktualizacja ekranów, mechanizm przełączania między nimi.

Moduł Controller:

- moduł ten będzie szyną mediacyjną pomiędzy backendem a frontendem czyli modułami View oraz Model. Będzie on najprawdopodobniej zawierać góra 1,2 klasy odpowiedzialne właśnie za ten fakt.

*Ostateczna architektura oraz forma projektu może ulec zmianie w zależności od ilości dostępnego czasu (oraz ilości kawy w szafkach kuchennych deweloperów oraz wytrzymałości ich mięśni sercowych)