**WYMAGANIA FUNKCJONALNE**

**SYSTEM WYSTAWIANIA / REZERWACJI APARTAMENTÓW**

SPIS TREŚCI

[1 Przedstawienie koncepcji systemu 3](#_Toc527288984)

[1.1 Główne funkcje systemu 3](#_Toc527288985)

[1.2 Charakterystyka użytkowników 4](#_Toc527288986)

[1.3 Kontekst funkcjonalny systemu 5](#_Toc527288987)

[2 Wymagania funkcjonalne 6](#_Toc527288988)

[2.1 Funkcje 6](#_Toc527288989)

[2.2 Diagram hierarchii funkcji 7](#_Toc527288990)

[2.3 Działanie systemu wystawiania/rezerwacji apartamentów 7](#_Toc527288991)

[2.3.1 Autoryzacja i autentykacja użytkownika 7](#_Toc527288992)

[2.3.2 Wyszukiwanie wolnych apartamentów 8](#_Toc527288993)

[2.3.3 Rezerwacja apartamentów 8](#_Toc527288994)

[2.3.4 Zarządzanie apartamentami 8](#_Toc527288995)

[2.3.5 Zarządzanie kontami 9](#_Toc527288996)

[2.4 Diagram przypadków użycia 9](#_Toc527288997)

[2.4.1 Nadzwyczajne ciągi zdarzeń 9](#_Toc527288998)

[2.5 Diagramy sekwencji 9](#_Toc527288999)

[2.5.1 Diagram sekwencji logowania do aplikacji 10](#_Toc527289000)

[2.5.2 Diagram sekwencji rejestrowania do aplikacji 11](#_Toc527289001)

[2.5.3 Diagram sekwencji wyszukiwania wolnych apartamentów 11](#_Toc527289002)

[2.5.4 Diagram sekwencji rezerwowania apartamentu 12](#_Toc527289003)

[2.5.5 Diagram sekwencji dodawania nowego hotelu i apartamentu 13](#_Toc527289004)

[3 Diagram ERD systemu 14](#_Toc527289005)

[3.1 Opis encji diagramu 14](#_Toc527289006)

[3.2 Opis związków encji 14](#_Toc527289007)

[4 Model architektury systemu 15](#_Toc527289008)

[4.1 Konsekwencje użycia 16](#_Toc527289009)

# **Przedstawienie koncepcji systemu**

System zarządzania apartamentami został wymyślony w ramach przedmioty projekt grupowy.

System ten ma na celu zapewnienie elastyczności i komfortu wynajmowania apartamentów oraz zarządzania hotelami. Aplikacja będzie miała możliwość wyszukiwania lokali zgodnie z określonymi kryteriami takimi jak miejscowość, zakres dat oraz nazwy hotelu. Użytkownicy po zdefiniowaniu swojej roli w procesie rejestracji otrzymają uprawnienia w zależności od wybranego profilu. Przewidywany jest system logowania dla klientów, właścicieli oraz administratorów, gdzie każda z grup będzie posiadała wyznaczone uprawnienia.

Klienci będą uprawnieni do wyszukiwania apartamentów po podaniu wybranych kryteriów, przeglądania swoich aktualne rezerwacje z opcją ich anulowania. Dodatkowo będą oni posiadali możliwość przeglądania swojej historii rezerwacji.

Właściciele będą w stanie zarządzać dodanymi przez siebie hotelami, oraz przeglądać historię posiadanego obiektu.

Administratorzy będą mieli uprawnienia do dodawania , modyfikacji i usuwania kont, jak i również zarządzania dostępnością hoteli.

## Główne funkcje systemu

Produkt udostępnia następujące funkcjonalności użytkownikowi:

* Możliwość rejestracji i logowania
* Uwierzytelnienie i autoryzacje użytkowników
* Zmiany hasła oraz danych użytkownika
* Wyszukiwanie apartamentów w zależności od podanych kryteriów wyszukiwania
* Rezerwację apartamentów po zalogowaniu lub rejestracji
* Anulować swoje aktualne rezerwacje
* Wystawiania ocen dla apartamentów
* Historię rezerwacji
* Przeglądania aktualnych rezerwacji użytkownika
* Dodawania nowych hoteli i apartamentów jako właściciel
* Modyfikację i usuwanie istniejących hoteli i apartamentów jako właściciel
* Przeglądania historię hoteli i apartamentów
* Aplikacja zapewni profil (administratora) zarządzający kontami użytkowników

Formuła realizacji projektu:

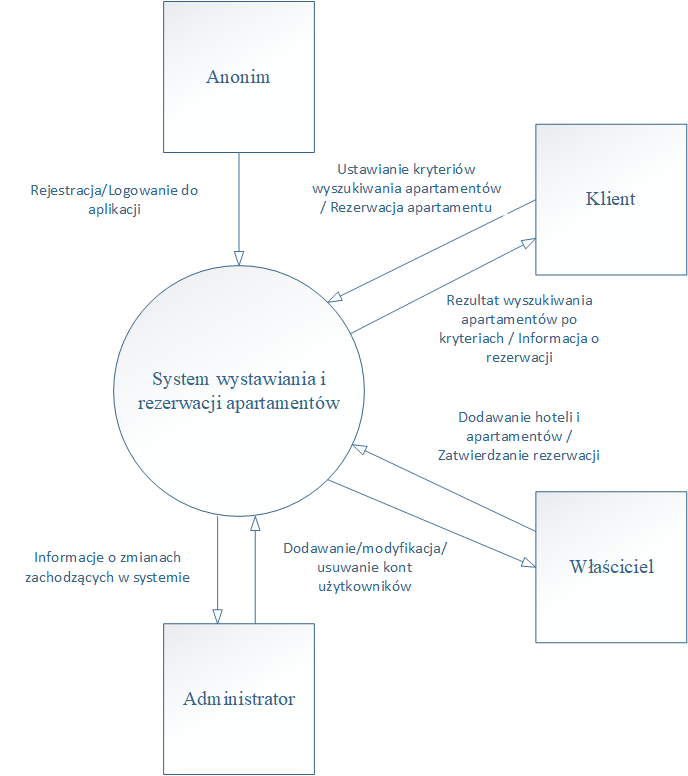
Formuła zostanie napisana w oparciu o współpracę:

* Agnieszki Szajdeckiej
* Michała Kopaczewskiego
* Rafała Urbaniaka

## Charakterystyka użytkowników

* Anonim – odnosi się do kontekstu procesu, jest to osoba, która nie jest zidentyfikowana przez system,
* Klient – odnosi się do kontekstu procesu, jest to osoba zalogowana w systemie, która przegląda i dokonuje rezerwacji apartamentów,
* Właściciel – odnosi się do kontekstu procesu, jest to osoba zalogowana w systemie, która dodaje nowe hotele i apartamenty,
* Administrator – odnosi się do kontekstu procesu, jest to osoba zalogowana w systemie, która zarządza kontami użytkowników.

## Kontekst funkcjonalny systemu



|  |  |  |
| --- | --- | --- |
| Nazwa elementu | Typ elementu | Opis |
| Administrator | Terminator | Zarządza całym system |
| Właściciel | Terminator | Zarządza swoimi apartamentami oraz rezerwacjami |
| Klient | Terminator | Rezerwuje apartamenty oraz zarządza swoimi rezerwacjami |
| Anonim | Terminator | Wyszukuje apartamenty, ma możliwość rejestracji lub logowania |
| System wystawiania i rezerwacji apartamentów | Proces | Zarządzanie rezerwacjami i wystawianiem apartamentów |
| Informacje o zmianach zachodzących w systemie | Przepływ | Przepływ danych zawierający aktualizację zawartości systemu |
| Dodawanie/modyfikacja usuwanie kont użytkowników | Przepływ | Przepływ danych zawierający informacje o dodaniu/usunięciu kont |
| Dodawanie hoteli i apartamentów/ Zatwierdzanie rezerwacji | Przepływ | Przepływ danych zawierający informacje o dodaniu lub usunięciu apartamentów lub hoteli / przepływ danych zawierający potwierdzenie rezerwacji |
| Rezultat wyszukiwania apartamentów po kryteriach / Informacja o rezerwacjach | Przepływ | Przepływ informujący o rezultacie wyszukiwania/ Informacje o aktualnych wolnych apartamentach |
| Ustawianie kryteriów wyszukiwania apartamentów / Rezerwacja apartamentu | Przepływ | Przepływ informacji o wybranych kryteriach w wyszukiwaniu/ Przepływ z żądaniem rezerwacji |
| Rejestracja / Logowanie do aplikacji | Przepływ | Przepływ z danymi nowego klienta /przepływ z danymi do logowania |

# **Wymagania funkcjonalne**

Profile aplikacyjne oferują unikatowe funkcje użytkownikom, które są dobierane  
w zależności od przeznaczenia konta.

Opis wymagań funkcjonalnych, został podzielony na grupy według funkcji systemu.

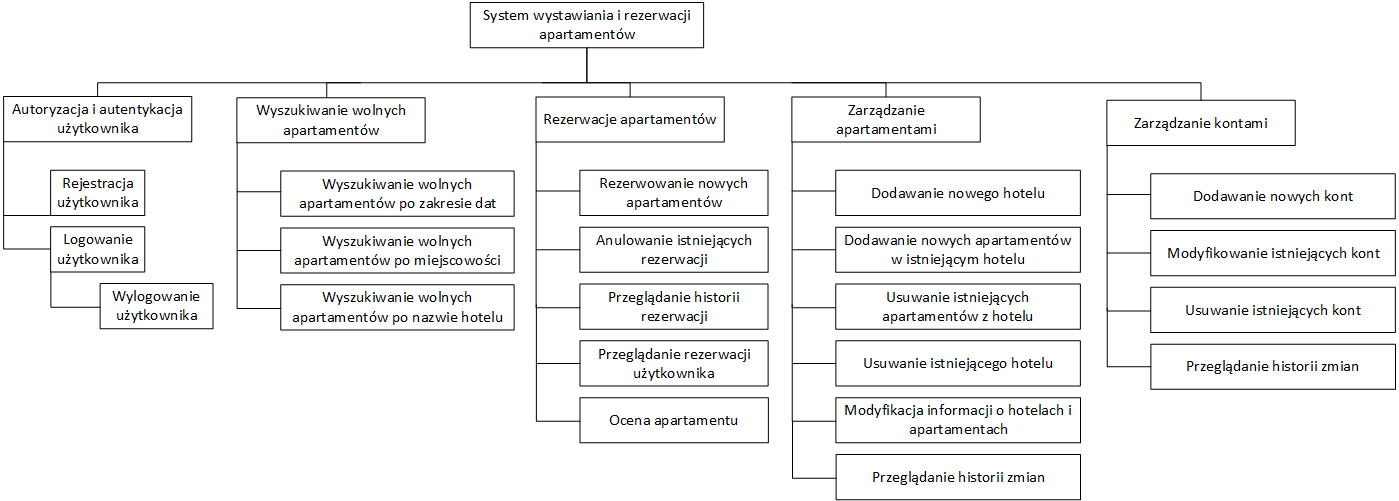
## Funkcje

Podrozdział przedstawiający listę funkcji realizowanych przez aplikacje. Najistotniejsze funkcje zostały szczegółowo opisane w dalszych częściach dokumentu.

**Lista funkcji:**

1. Autoryzacja i autentykacja użytkownika
   1. Rejestracja użytkownika
   2. Logowanie użytkownika
      1. Wylogowanie użytkownika
2. Wyszukiwanie wolnych apartamentów
   1. Wyszukiwanie wolnych apartamentów po zakresie dat
   2. Wyszukiwanie wolnych apartamentów po miejscowości
   3. Wyszukiwanie wolnych apartamentów po nazwie hotelu
3. Rezerwacje apartamentów
   1. Rezerwowanie nowych apartamentów
   2. Anulowanie istniejących rezerwacji
   3. Przeglądanie historii rezerwacji
   4. Przeglądanie rezerwacji użytkownika
   5. Ocena apartamentu
4. Zarządzanie apartamentami
   1. Dodawanie nowego hotelu
   2. Dodawanie nowych apartamentów w istniejącym hotelu
   3. Usuwanie istniejących apartamentów z hotelu
   4. Usuwanie istniejącego hotelu
   5. Modyfikacja informacji o hotelach i apartamentach
   6. Przeglądanie historii zmian
5. Zarządzanie kontami
   1. Dodawanie nowych kont
   2. Modyfikowanie istniejących kont
   3. Usuwanie istniejących kont
   4. Przeglądanie historii zmian

## Diagram hierarchii funkcji



## Działanie systemu wystawiania/rezerwacji apartamentów

### **Autoryzacja i autentykacja użytkownika**

W celu pełnego wykorzystania systemu, użytkownik będzie musiał założyć konto w procesie rejestracji. Podane w procesie rejestracji email i hasło będą używane do procesu uwierzytelniania użytkownika. Po zalogowaniu się do systemu email i hasło może zostać zmienione.

Proces rejestracji będzie nadawał użytkownikom odpowiednie uprawnienia w zależności od wybranego profilu użytkownika. Przewidziane są trzy profile użytkowników aplikacji: Klient, Właściciel i Administrator. Profil kliencki będzie umożliwiał rezerwacje apartamentów w hotelach. Profil właściciela będzie pozwalał na zarządzanie hotelami i apartamentami. Dodatkowo przewidziano utworzenie profilu administratora umożliwiającego zarządzanie kontami użytkowników aplikacji. Profil administracyjny będzie mógł założyć tylko inny administrator.

W trakcie logowania do aplikacji system będzie potwierdzał tożsamość użytkownika i weryfikował jego uprawnienia. Nieuwierzytelniony w aplikacji użytkownik będzie uprawniony do wyszukiwania i przeglądania informacji o wolnych apartamentach w hotelu.

Użytkownik będzie miał możliwość wylogowania się z aplikacji, jeżeli nie przekroczy czasu trwania sesji użytkownika, po której nastąpi automatyczne wylogowanie z aplikacji.

### **Wyszukiwanie wolnych apartamentów**

Wyszukiwanie i przeglądanie informacji o oferowanych apartamentach w hotelach będzie obsługiwane za pomocą wyszukiwarki wolnych apartamentów. Przewidziane są trzy kryteria wyszukiwania: po zakresie dat, po miejscowości i po nazwie hotelu. Proces wyszukiwania będzie wymagał przynajmniej jednego kryterium wyszukiwania apartamentów.

### **Rezerwacja apartamentów**

Zalogowany użytkownik o profilu klienckim będzie miał możliwość rezerwacji apartamentu w wybranym hotelu jak i anulowania istniejącej rezerwacji. Aplikacja będzie pozwalała na sprawdzanie wszystkich aktualnych rezerwacji oraz przeglądanie historii zakończonych i anulowanych rezerwacji. System będzie pozwalał na ocenę poszczególnych apartamentów.

### **Zarządzanie apartamentami**

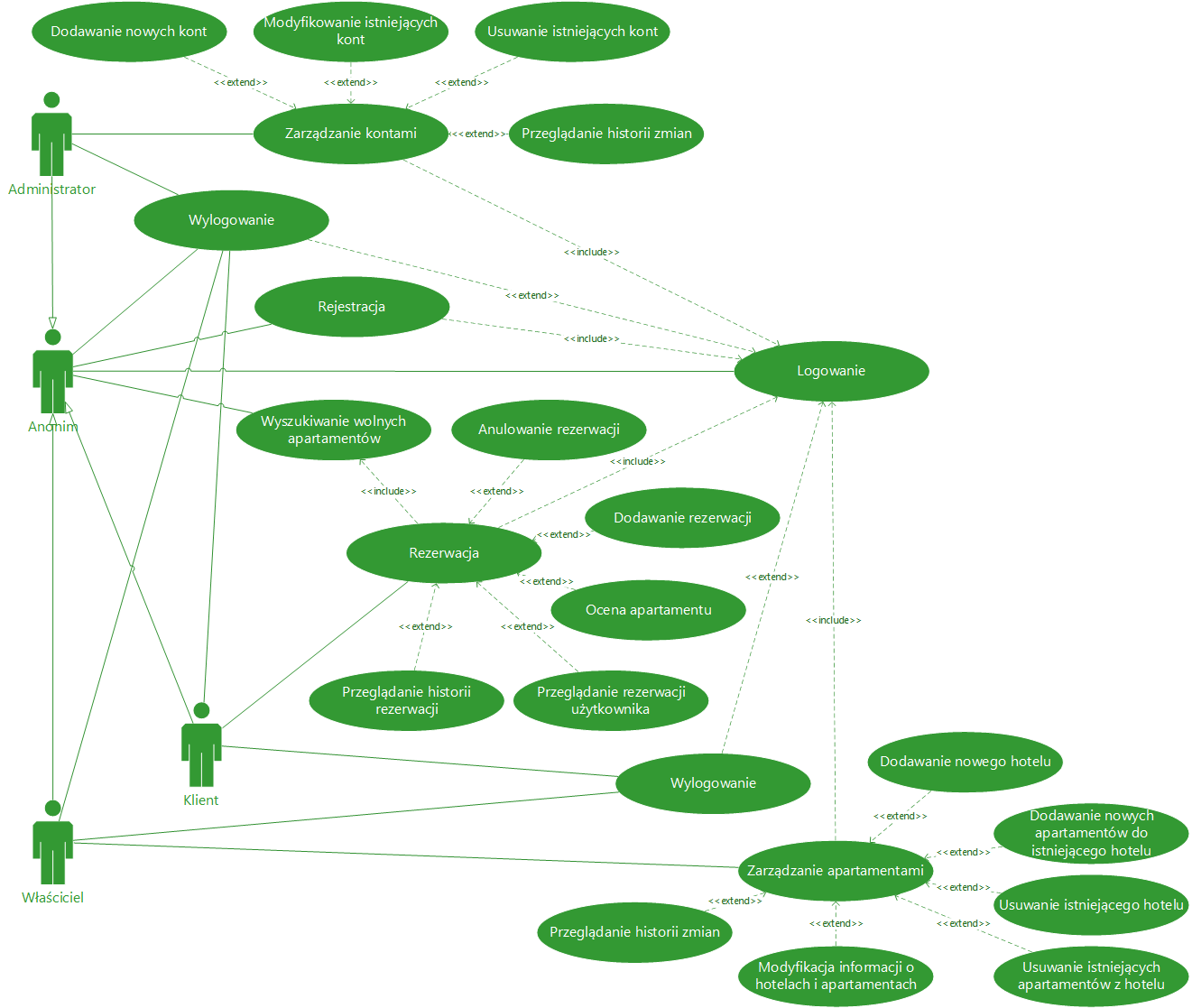
Uwierzytelniony użytkownik posiadający profil Właściciel będzie mógł wystawić apartamenty do rezerwacji. Aplikacja będzie pozwalała na dodawanie i usunięcie hoteli użytkownikom posiadającym uprawnienia właściciela. Właściciel będzie miał uprawnienia do dodawania nowych apartamentów do posiadanego hotelu, jak i usuwania istniejących apartamentów z niego. Dodatkowo system będzie pozwalał na modyfikacje informacji o hotelach i apartamentach oraz przeglądanie historii zmian.

### **Zarządzanie kontami**

Profil posiadający uprawnienia administracyjne będzie upoważniony do zarządzania kontami użytkowników aplikacji. Administrator będzie miał uprawniania do dodawania nowych kont, modyfikacji i usuwania istniejących kont użytkownikom. Dodatkowo administrator będzie miał możliwość przeglądania historii zmian.

## Diagram przypadków użycia

Diagram przypadków użycia przedstawia ogólne funkcjonowanie systemu z punktu widzenia użytkowników. Skupiono się na ukazywaniu głównych funkcji aplikacji jakimi są rezerwacje apartamentów, zarządzanie apartamentami oraz zarządzanie kontami.



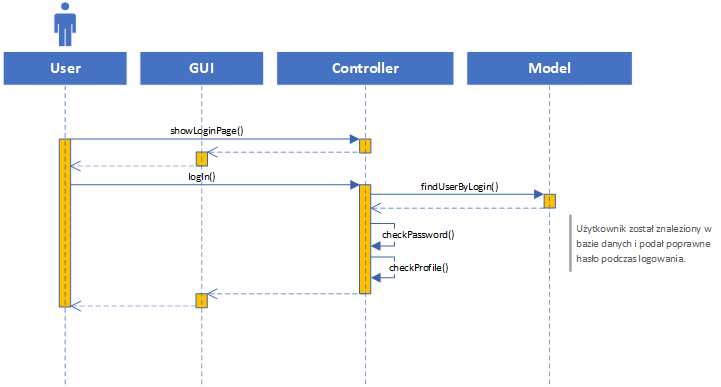
### **Nadzwyczajne ciągi zdarzeń**

Na tym etapie nie zostały zdefiniowane.

## Diagramy sekwencji

Diagramy sekwencji dla systemu wystawiania i rezerwacji apartamentów wyglądają następująco:

### **Diagram sekwencji logowania do aplikacji**

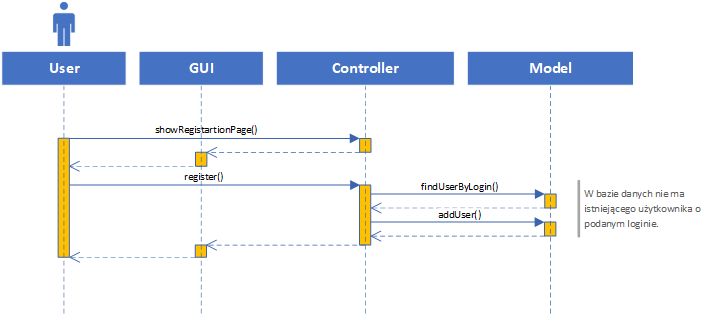


**Przebieg zdarzeń**

Główny ciąg zdarzeń, podczas logowania użytkownika do aplikacji:

* Przypadek użycia zaczyna się gdy użytkownik wybierze opcje [Zaloguj się] w menu głównym,
* System wyświetla formularz „Zaloguj się”,
* Użytkownik wpisuje login (email) i hasło,
* Po wciśnięciu przycisku [Zaloguj się], użytkownik zostaje poprawnie zalogowany do systemu i przekierowany na odpowiednią stronę w zależności od posiadanego profilu (np. Użytkownik z profilem klienckim przekierowywany jest na stronę z wyszukiwarką apartamentów).

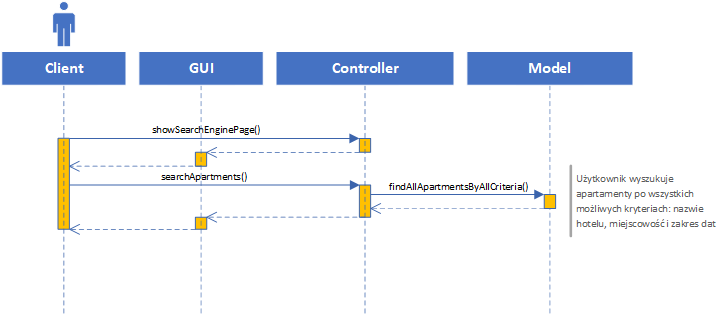
### **Diagram sekwencji rejestrowania do aplikacji**



Główny ciąg zdarzeń, podczas rejestracji użytkownika w aplikacji:

* Przypadek użycia zaczyna się gdy użytkownik wybierze opcje [Zarejestruj się] w menu głównym,
* System wyświetla formularz „Zarejestruj się”,
* Użytkownik wpisuje login (email), hasło oraz dane osobowe,
* Po wciśnięciu przycisku [Zarejestruj się], użytkownikowi zostaje założone konto w systemie i jest on przekierowany na odpowiednią stronę w zależności od posiadanego profilu (np. Użytkownik z profilem klienckim przekierowywany jest na stronę z wyszukiwarką apartamentów).

### **Diagram sekwencji wyszukiwania wolnych apartamentów**

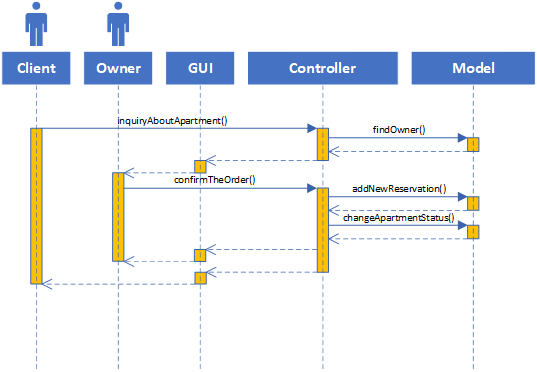


**Przebieg zdarzeń**

Główny ciąg zdarzeń, podczas wyszukiwania wolnych apartamentów w hotelu:

* Przypadek użycia zaczyna się gdy klient wybierze opcje [Wyszukiwarka apartamentu] w menu głównym,
* System wyświetla formularz „Wyszukiwarka apartamentów”,
* Użytkownik wpisuje w kryteriach wyszukiwania nazwę hotelu, miejscowość i zakres dat.
* Po wciśnięciu przycisku [Szukaj], użytkownikowi wyświetla się lista wolnych apartamentów w danym hotelu w określonym przedziale czasowym.

### **Diagram sekwencji rezerwowania apartamentu**

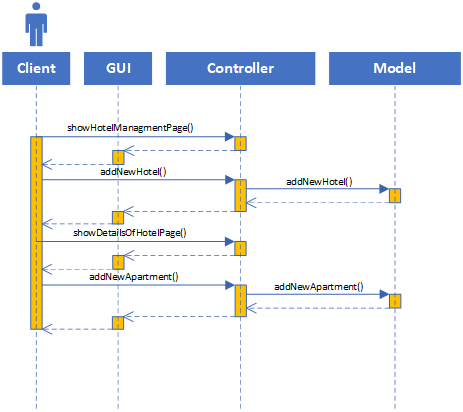


**Przebieg zdarzeń**

Główny ciąg zdarzeń, podczas rezerwowania wolnego apartamentu w hotelu:

* Klient wybiera interesujący go apartament na ekranie wyszukiwarki i wciska przycisk [Rezerwuj apartament],
* Właściciel dostaje informacje o chęci zarezerwowania konkretnego apartamentu przez Klienta,
* Właściciel wciska przycisk [Akceptuj rezerwacje],
* System dodaje wybrany apartament do listy rezerwacji Klienta i wyświetla formularz „Moje rezerwacje”,
* Właścicielowi znika komunikat informacyjny i wyświetla się formularz „Klienci”.

### **Diagram sekwencji dodawania nowego hotelu i apartamentu**

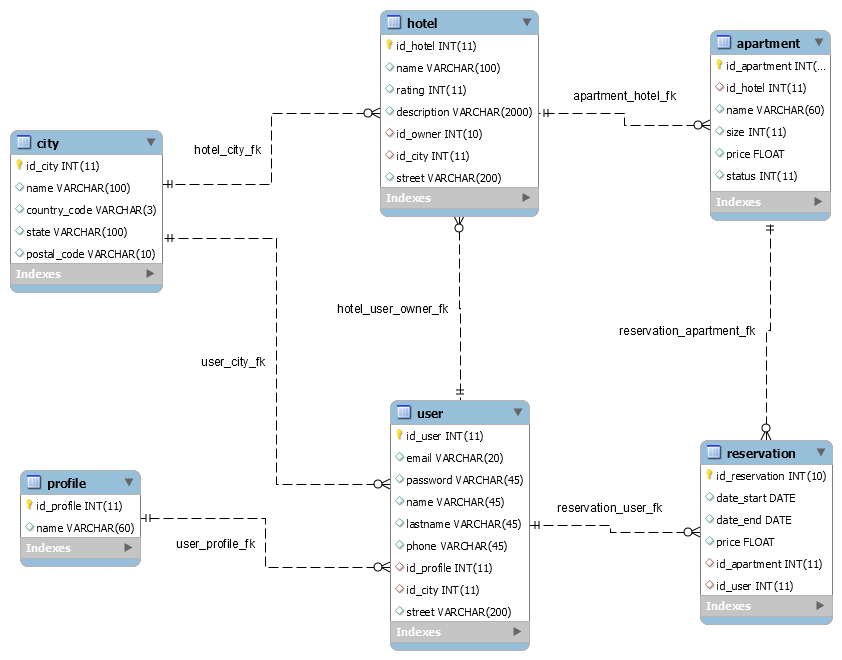


**Przebieg zdarzeń**

Główny ciąg zdarzeń, podczas dodawania nowego hotelu i apartamentu przez Właściciela:

* Przypadek użycia zaczyna się gdy Właściciel wybierze opcje [Zarządzaj hotelami] w menu głównym,
* System wyświetla formularz „Zarządzaj hotelami”,
* Właściciel wpisuje wszystkie niezbędne dane dotyczące hotelu,
* Po wciśnięciu przycisku [Dodaj], pojawia się nowy wiersz w tabeli „Moje hotele” z nazwą nowo dodanego hotelu,
* Właściciel wciska przycisk [Zarządzaj apartamentami] przy dodanym hotelu,
* System wyświetla formularz „Zarządzaj apartamentami”
* Właściciel wypełnia wszystkie niezbędne dane dotyczące apartamentu,
* Wciśniecie przez właściciela przycisku [Dodaj], powoduję przypisanie hotelowi nowego apartamentu.

# **Diagram ERD systemu**



## Opis encji diagramu

|  |  |  |
| --- | --- | --- |
| Nr | Nazwa encji | Opis encji |
| 1 | Profile | Encja, w której zostaną zapisane profile użytkowników w systemie potrzebne do ograniczania dostępu do funkcji programu. |
| 2 | User | Encja, w której zostaną zapisane informację o użytkowniku systemu |
| 3 | Reservation | Encja, w której zostaną zapisane informacje o rezerwacjach apartamentów przez użytkowników systemu |
| 4 | Apartment | Encja, w której zostaną zapisane informacje o apartamentach w hotelu |
| 5 | Hotel | Encja, w której zostaną zapisane informacje o hotelach |
| 6 | City | Encja, w której zostaną zapisane informacje o miastach zamieszkania użytkowników systemu oraz o miastach pod jakimi znajdują się hotele |

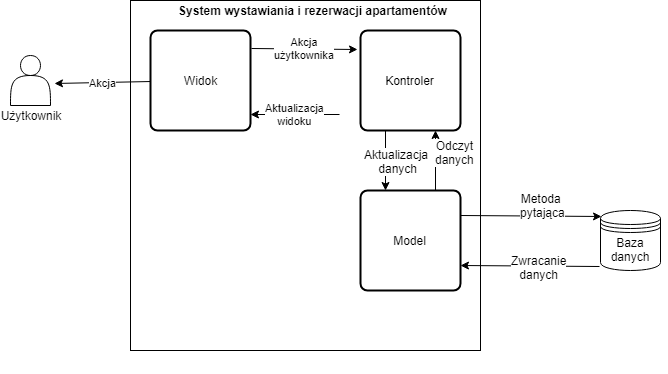
## Opis związków encji

|  |  |  |
| --- | --- | --- |
| Nr | Nazwa związku | Opis związku |
| 1 | Apartment\_hotel\_fk | Do hotelu należy zero lub wiele apartamentów |
| 2 | Reservation\_apratment\_fk | Jeden apartament może być zarezerwowany wiele razy lub wcale |
| 3 | Reservation\_user\_fk | Użytkownik może mieć wiele rezerwacji ale może też nic nie rezerwować. |
| 4 | User\_profile\_fk | Jeden profil może mieć przypisanych wielu użytkowników lub żadnego. |
| 5 | Hotel\_user\_owner\_fk | Jeden użytkownik może być właścicielem wielu hoteli, ale może też nie był właścicielem. |
| 6 | Hotel\_city\_fk | Hotel musi mieć określony tylko jedno miasto |
| 7 | User\_city\_fk | Użytkownik musi mieć określony tylko jedno miasto |

# **Model architektury systemu**

System zostanie napisany w Intelij IDEA w technologii Java EE + Spring.

System będzie pracował w oparciu o relacje z baz danych. Jako system zarządzający baza danych został zastosowany MySQL.



Główną koncepcją MVC jest wymuszenie podziału aplikacji na 3 niezależne warstwy reprezentujące kolejno:

* (Model) Model danych - opis struktur danych i powiązań pomiędzy nimi
* (Widok) Interfejs, czyli to co widzi użytkownik
* (Kontroler) Logika działania - powiązania między zdarzeniami zachodzącymi w systemie

Podział na warstwy służy uporządkowaniu architektury systemu. Dzięki temu, że każda logiczna część jest od siebie oddzielona, zmiana w jednym miejscu, nie powoduje konieczności wykonywania lawinowej ilości zmian w innych miejscach systemu.

Mimo, że istnieje wiele frameworków wspierających MVC w innych technologiach, [J2EE](http://vavatech.pl/j2ee) wiedzie zdecydowany prym jeżeli chodzi o ilość sprawdzonych narzędzi zapewniających implementację oprogramowania zgodnego z wzorcem MVC.

## Konsekwencje użycia

Zalety:

* Niezależność modelu - model nie jest zależny od widoku i aplikacja może posiadać wiele niezależnych widoków dla tego samego modelu
* Duża elastyczność widoków - ze względu na oddzielenie widoku od modelu, widoki mogą być modyfikowane częściej i niższym kosztem. Jest to szczególnie istotne, ponieważ w życiu systemu interfejs i warstwa prezentacji zmieniają się częściej niż logika biznesowa aplikacji
* Podział na moduły porządkujące kod aplikacji
* Oddzielenie logiki biznesowej od widoku
* Brak zależności modelu od widoku
* Ułatwia odnalezienie konkretnej części kodu
* Łatwiejsza rozbudowa poprzez modułową budowę
* Zapobiega tworzeniu się bałaganu w kodzie
* Łatwiejsze przeprowadzanie testów
* Ułatwia prace zespołową

Wady:

* Złożoność - architektura MVC nanosi dodatkową warstwę abstrakcji i nowe sposoby interakcji, co prowadzi do wzrostu jej złożoności. Zależność widoków od modeli i dodatkowa logika widoków czyni je szczególnie skomplikowane w testowaniu.
* Mała elastyczność modelu - modyfikacja modelu może w konsekwencji wymagać modyfikacji wielu widoków operujących na tym modelu (ponieważ model jest niezależny od widoku, ale widok zawsze opiera się na modelu)