

2.4. Specyfikacja architektury systemu

1. Założenia i cele architekuralne

W części Backend systemu zastosowana została architektura REST-API, oparta o framework SpringBoot,, spełniająca podstawowe warunki:

- Uniform Interface (jednolity interfejs)
- Stateless (bezstanowość)
- Cacheable (możliwość cache’owania danych)
- Client-Server (model klient-serwer)
- Layered System (warstwowość)
- Code on Demand (kod na żądanie)

Użyte wzorce to m.in.

- Dependency Injection (Wstrzykiwanie zależności)
- Factory pattern
- Proxy pattern
- Singleton

Komunikacja z bazą danych została oparta o strukturę JPA Repository/Hibernate.

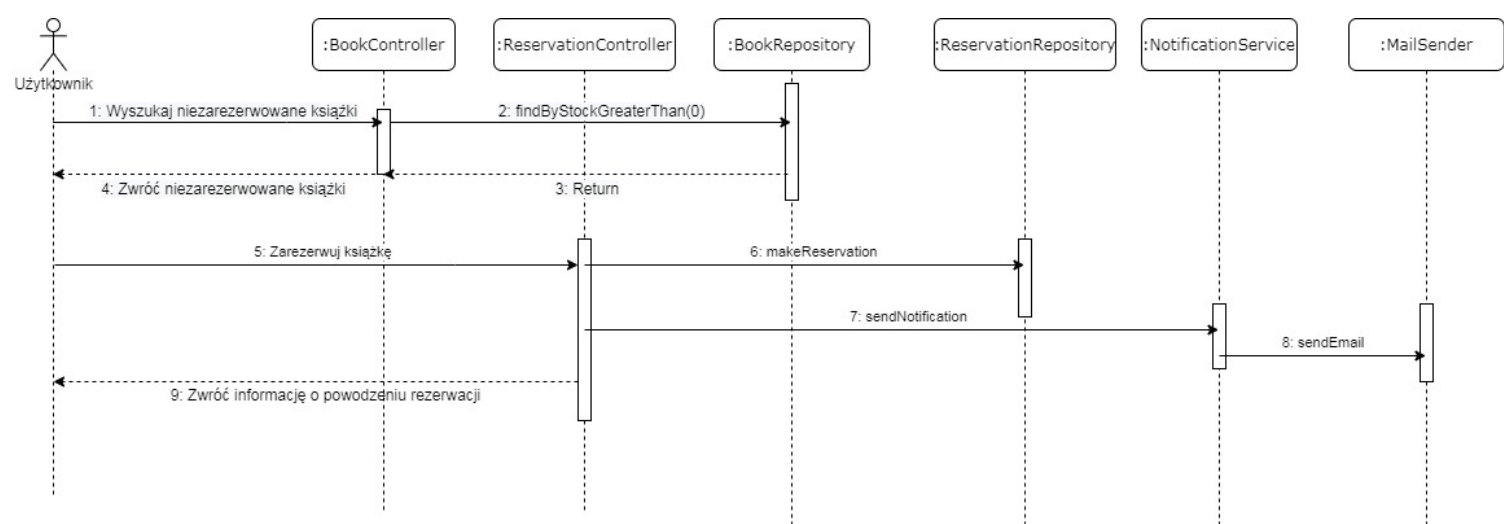
W części Frontend została zastosowana architektura MVC (trójwarstwowa), oparta na frameworku Vue.JS. Wykorzystane wzorce projektowe to m.in.:

- Dependency Injection (Wstrzykiwanie zależności)
- Composition (kompozycja)
- Observer (obserwator)

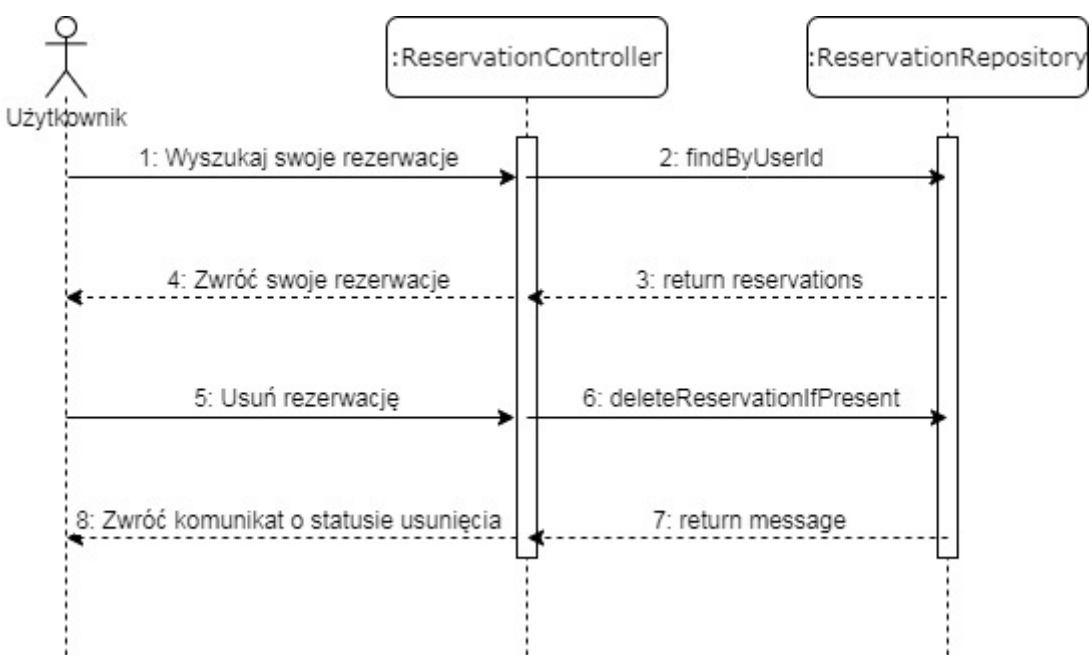
2. Model logiczny

Diagramy sekwencji:

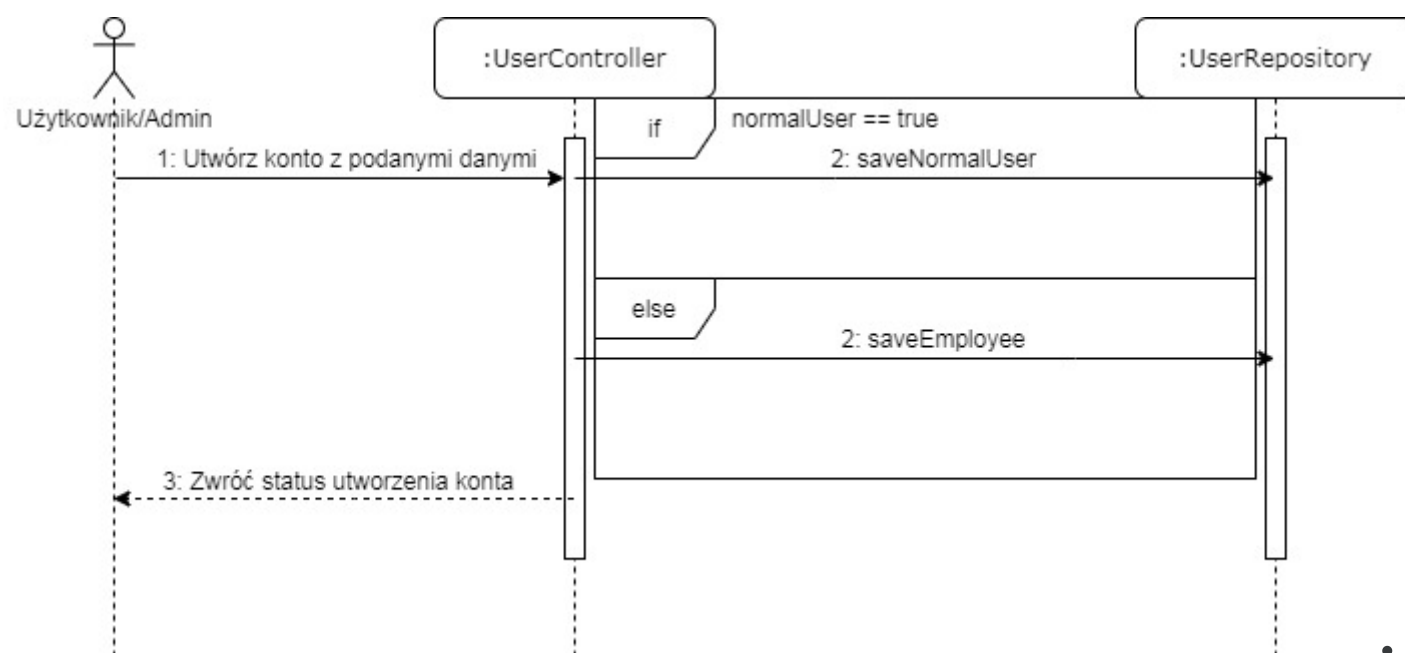
- Diagram sekwencji dodawania rezerwacji przez użytkownika:



- Diagram sekwencji usuwania rezerwacji przez użytkownika:

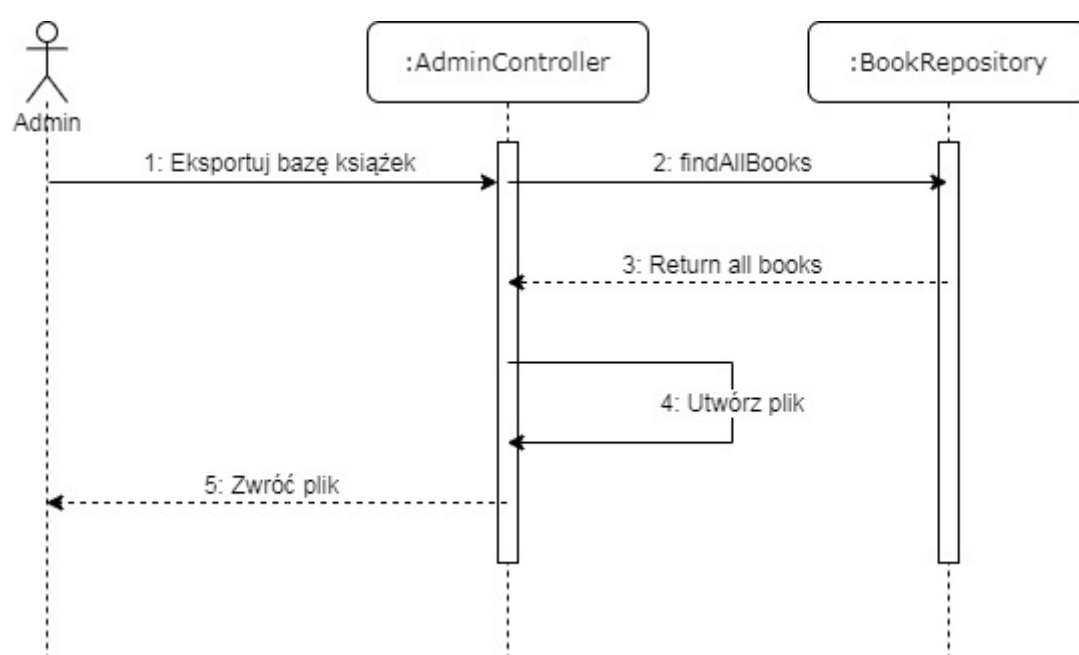


- Diagram sekwencji dodawania konta przez użytkownika/admina:

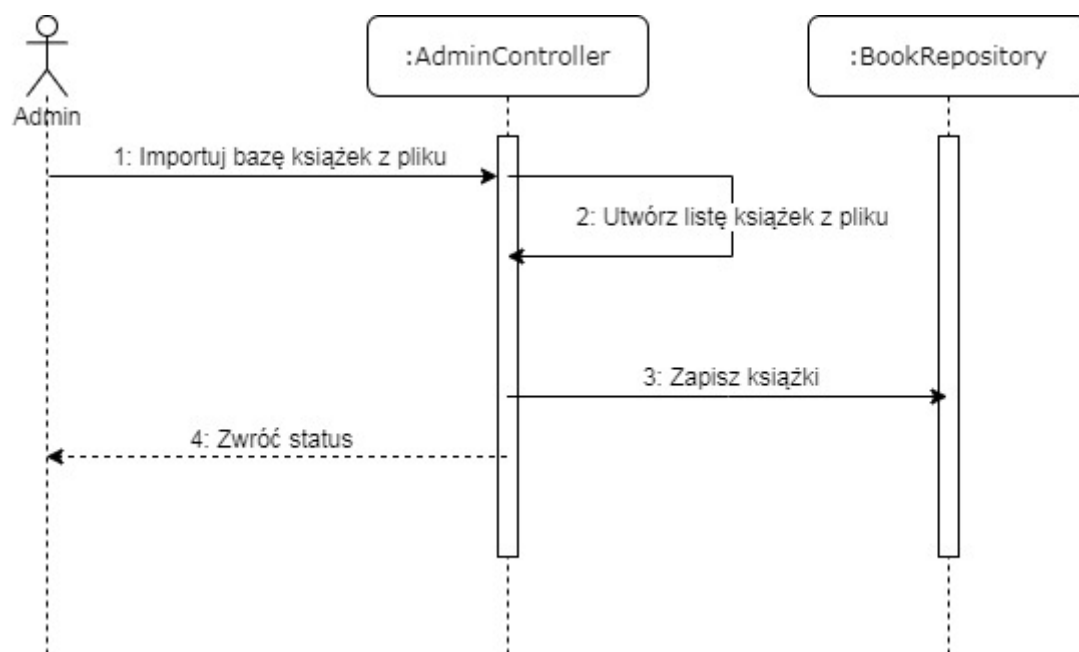


przez admina:

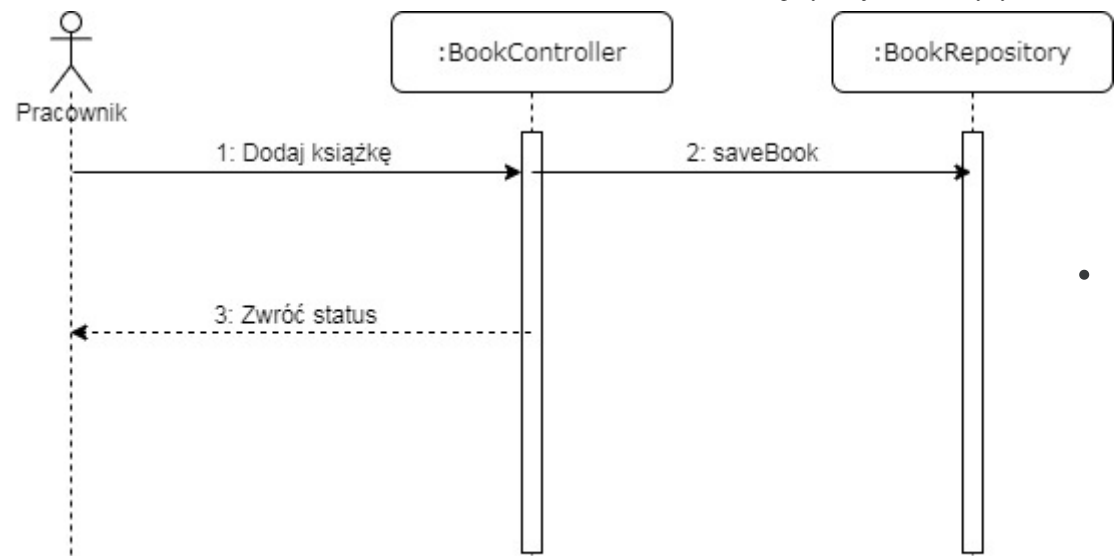
- Diagram sekwencji eksportowania danych



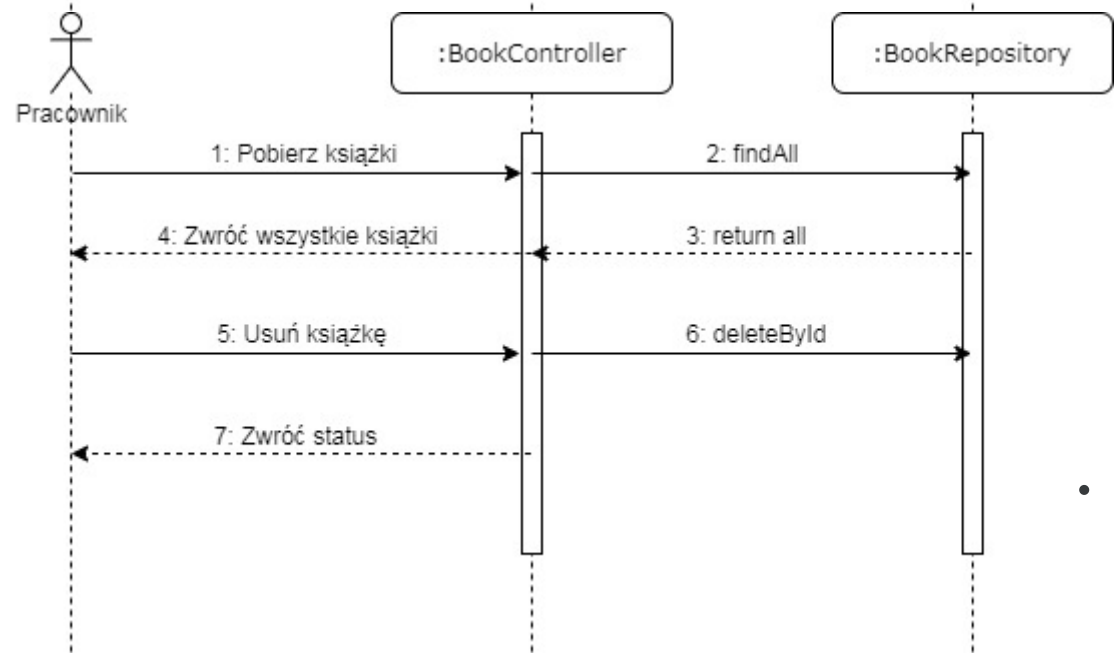
- Diagram sekwencji importowania danych przez admina:



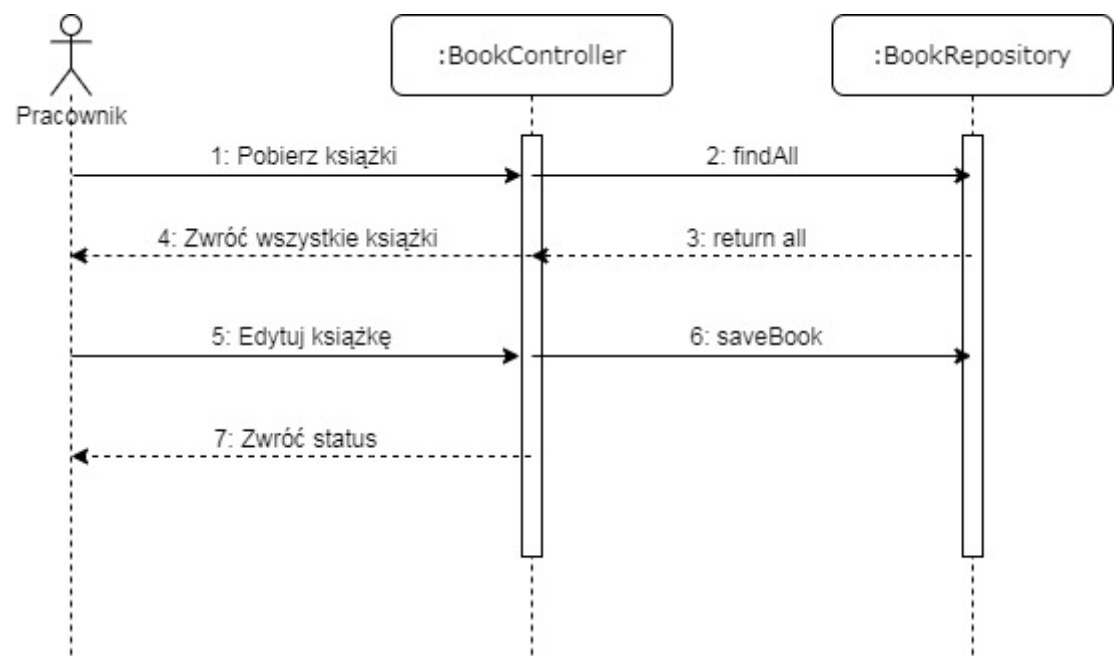
- Diagram sekwencji dodawania książek przez pracownika:



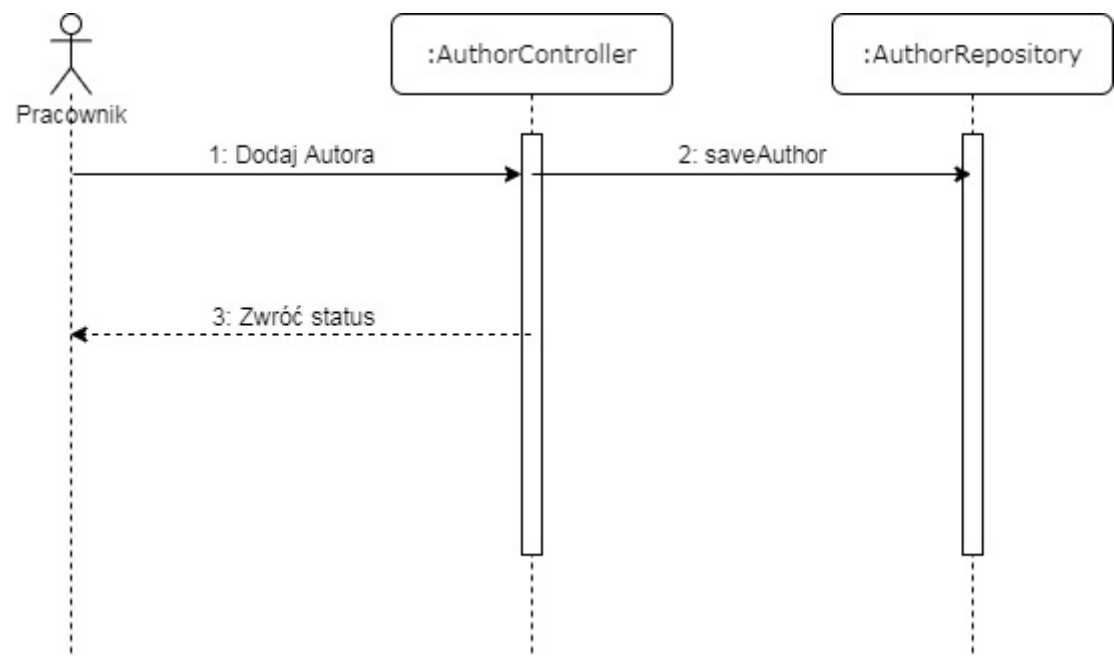
- Diagram sekwencji usuwania książek przez pracownika:



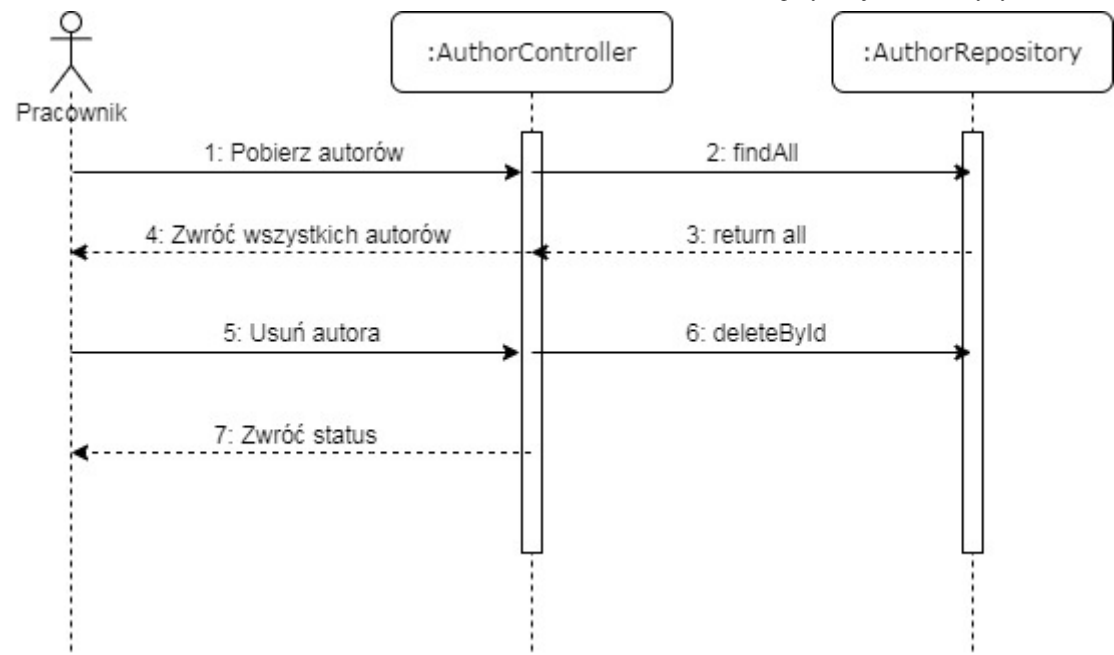
- Diagram sekwencji edytowania książek przez pracownika:



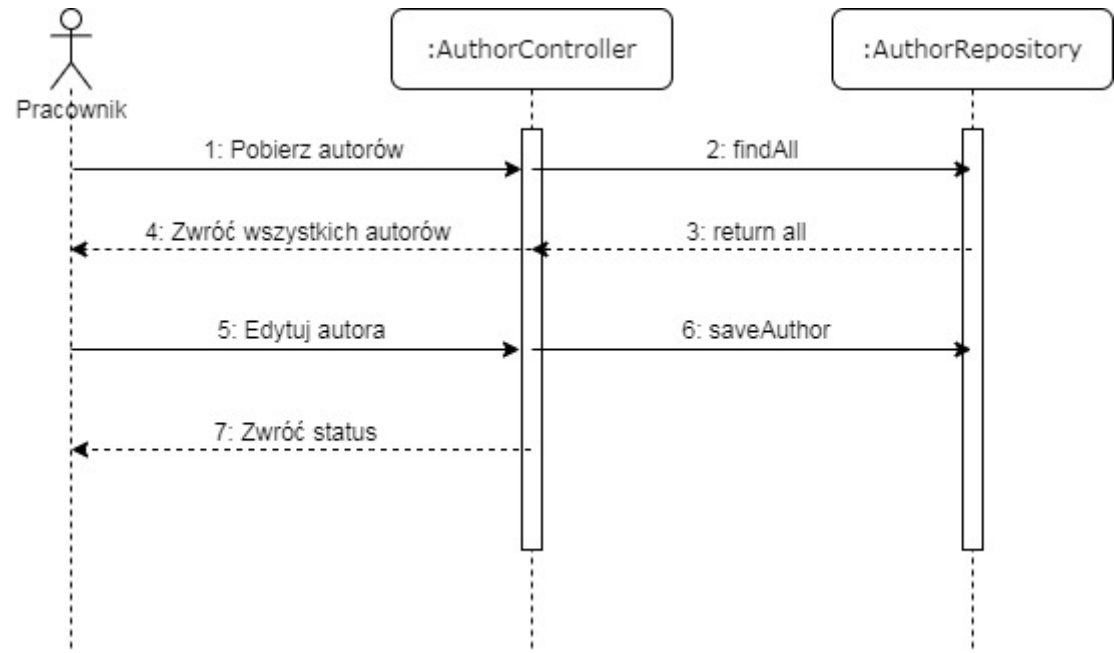
- Diagram sekwencji dodawania autorów przez pracownika:



- Diagram sekwencji usuwania autorów przez pracownika:

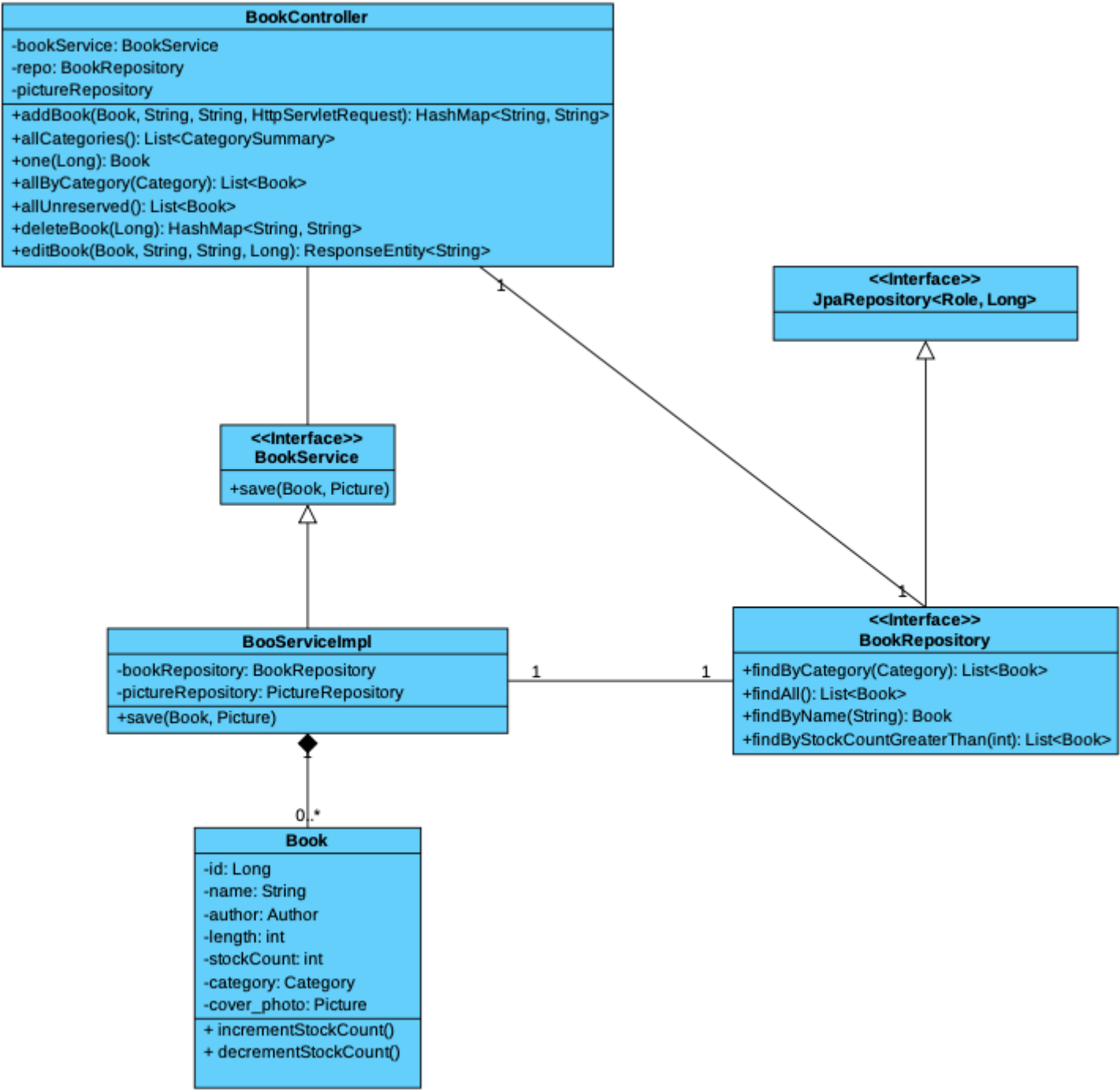


- Diagram sekwencji edytowania autorów przez pracownika:

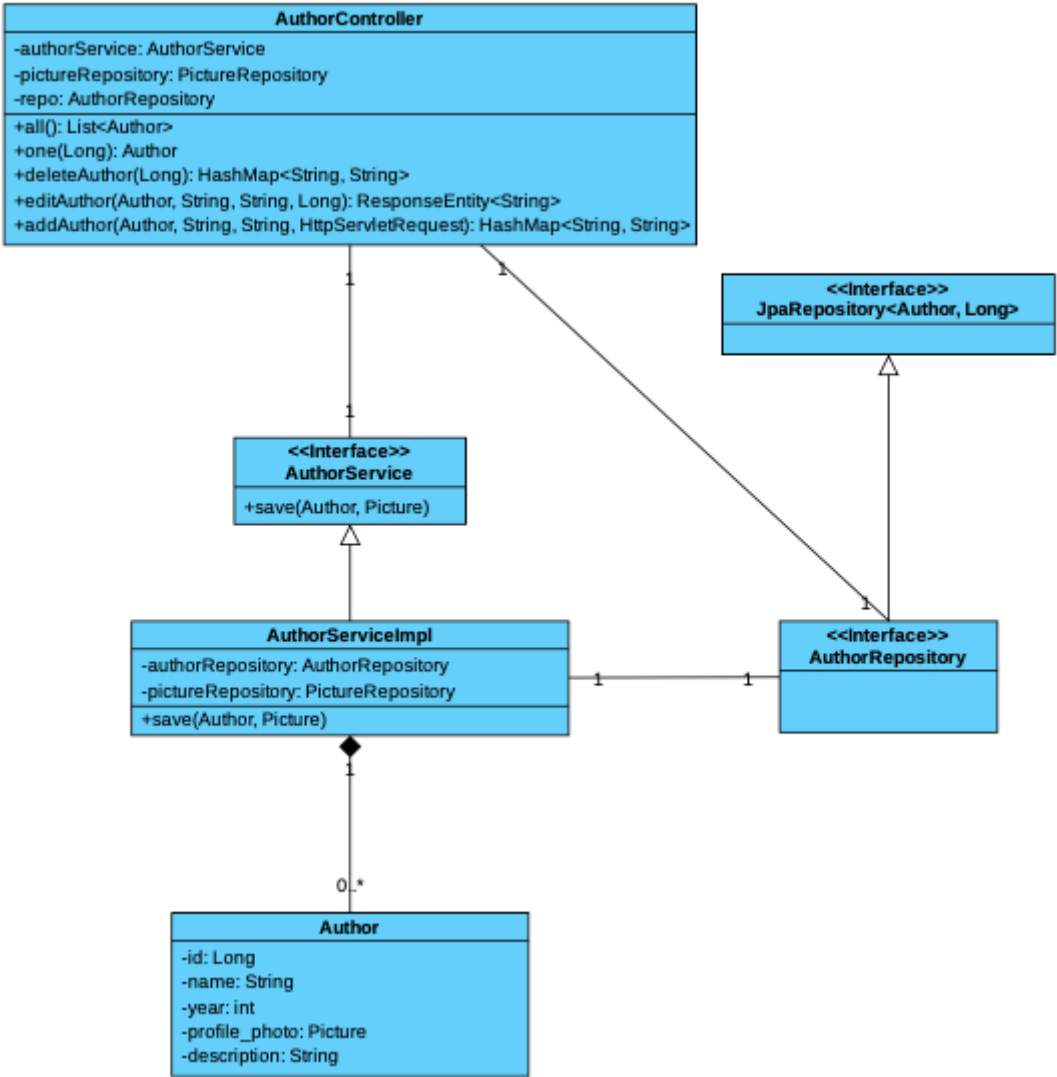


Diagramy klas:

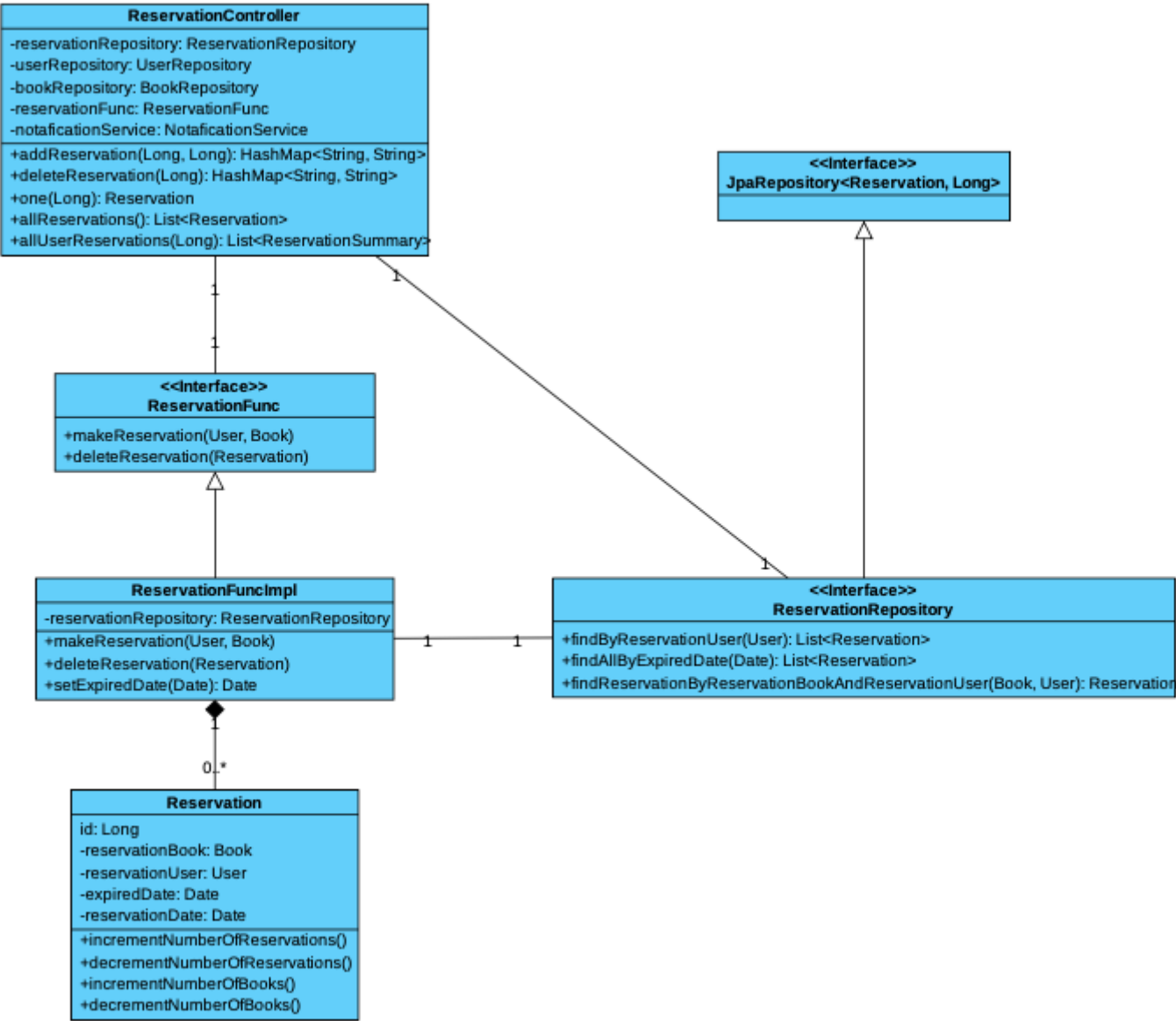
- Struktura oraz zależności pomiędzy klasami związanymi z klasą Book:



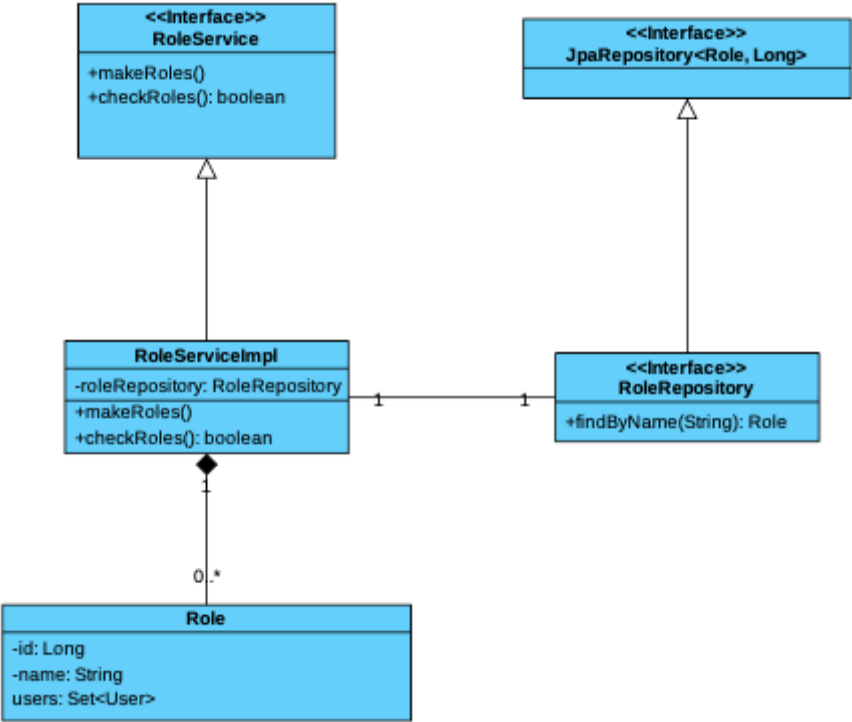
- Struktura oraz zależności pomiędzy klasami związanymi z klasą Author:



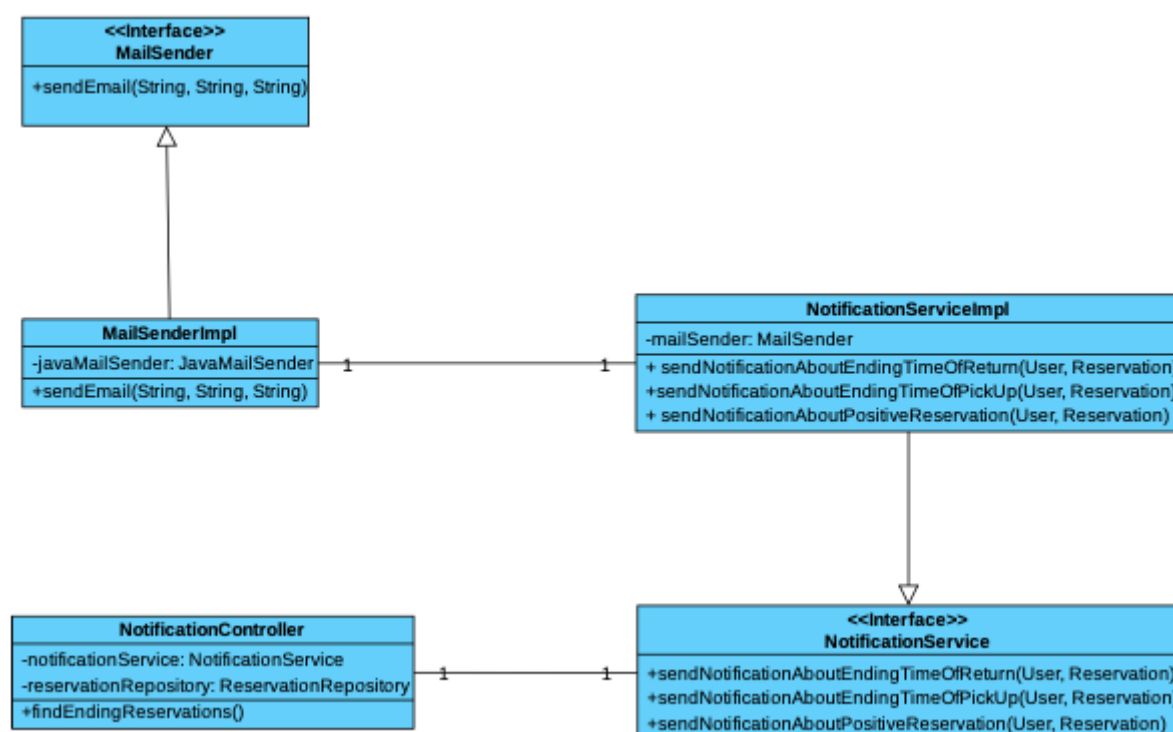
- Struktura oraz zależności między klasami związanymi z klasą Reservation - zawiera informacje kto i na jaką książkę zrobił rezerwację, oraz jaki jest czas jej wygaśnięcia:



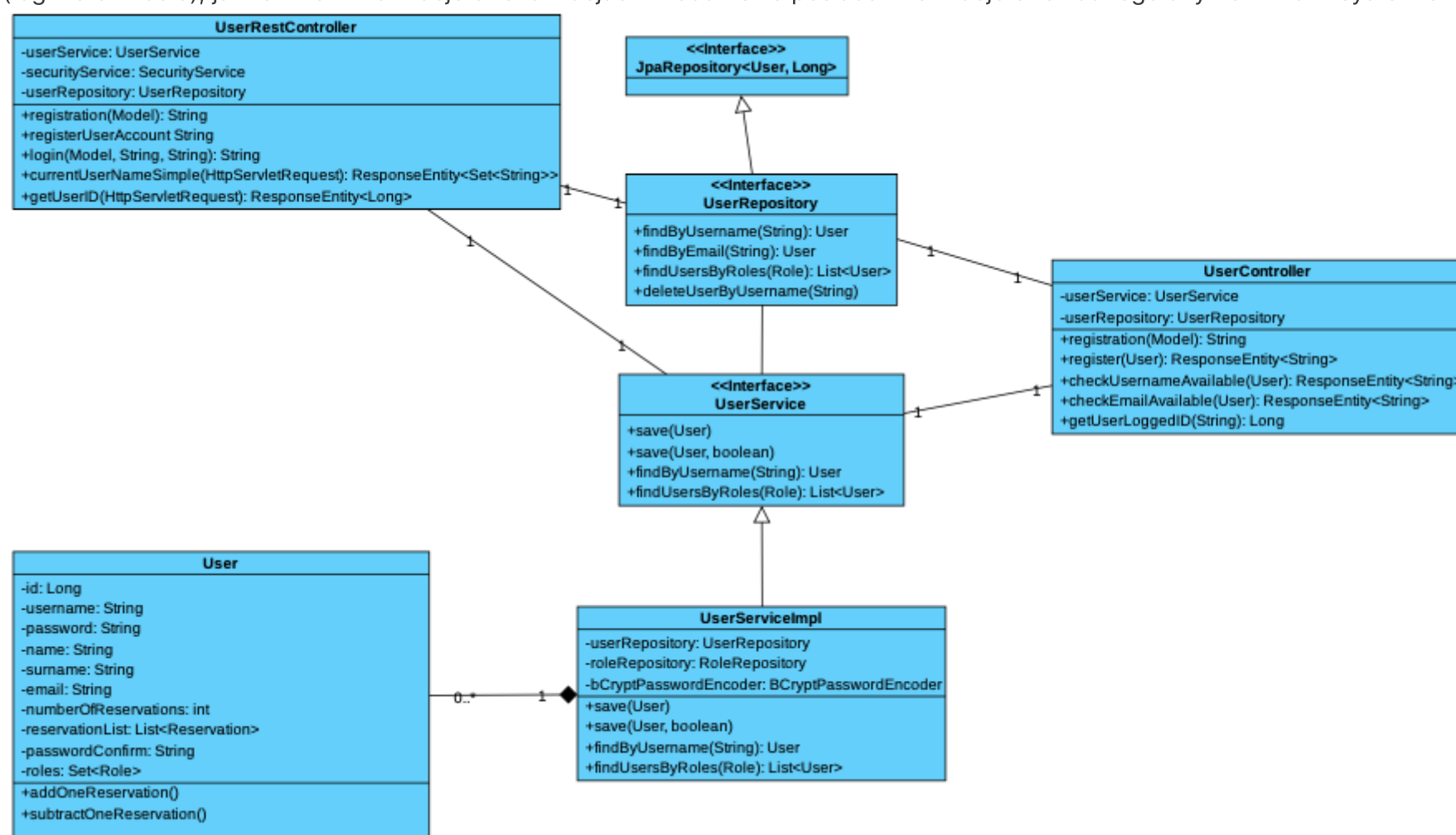
- Struktura oraz zależności pomiędzy klasami związanymi z klasą Role - określa jaką rolę w systemie pełni dany użytkownik (administrator, pracownik, zwykły użytkownik):



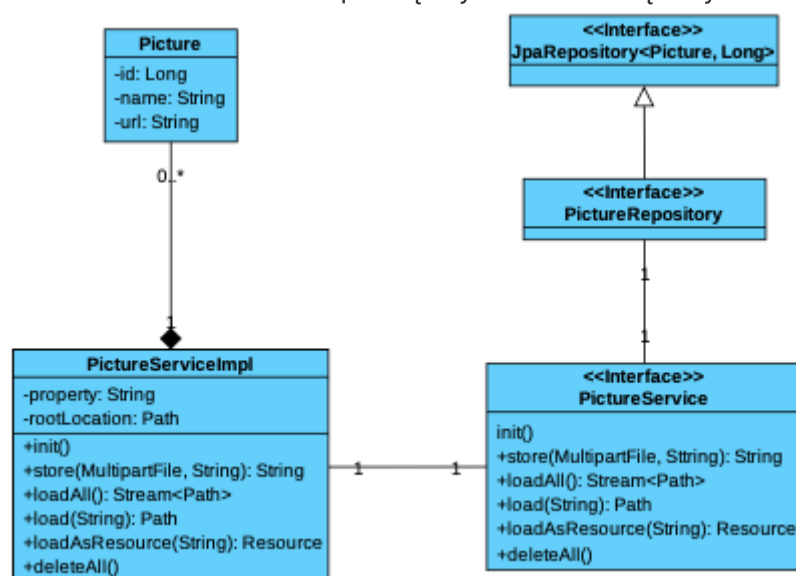
- Struktura oraz zależności pomiędzy klasami odpowiedzialnymi za powiadomienia wysyłane drogą mailową:
 - koniec czasu na odbiór książki,
 - koniec czasu na zwrot książki,
 - pozytywne rozpatrzenie rezerwacji.



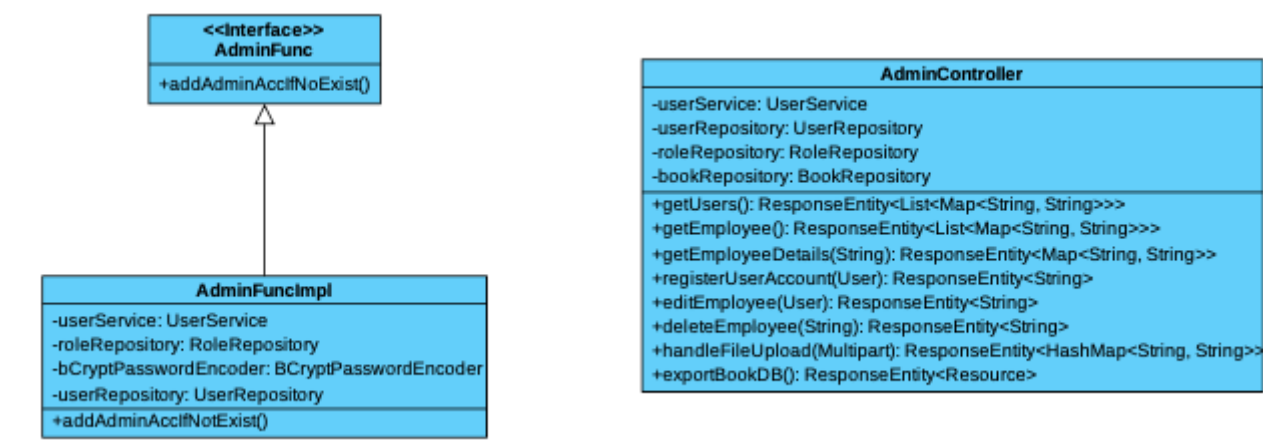
- Struktura oraz zależności pomiędzy klasami związanymi z klasą User, która zawiera najważniejsze informacje o użytkowniku - jego dane do logowania (login oraz hasło), jak również informacje o rezerwacjach. Dodatkowo posiada informacje o roli danego użytkownika w systemie.



- Struktura oraz zależności pomiędzy klasami związanymi z klasą Picture - wykorzystywana przez klasę Author oraz Book do przechowywania zdjęć.

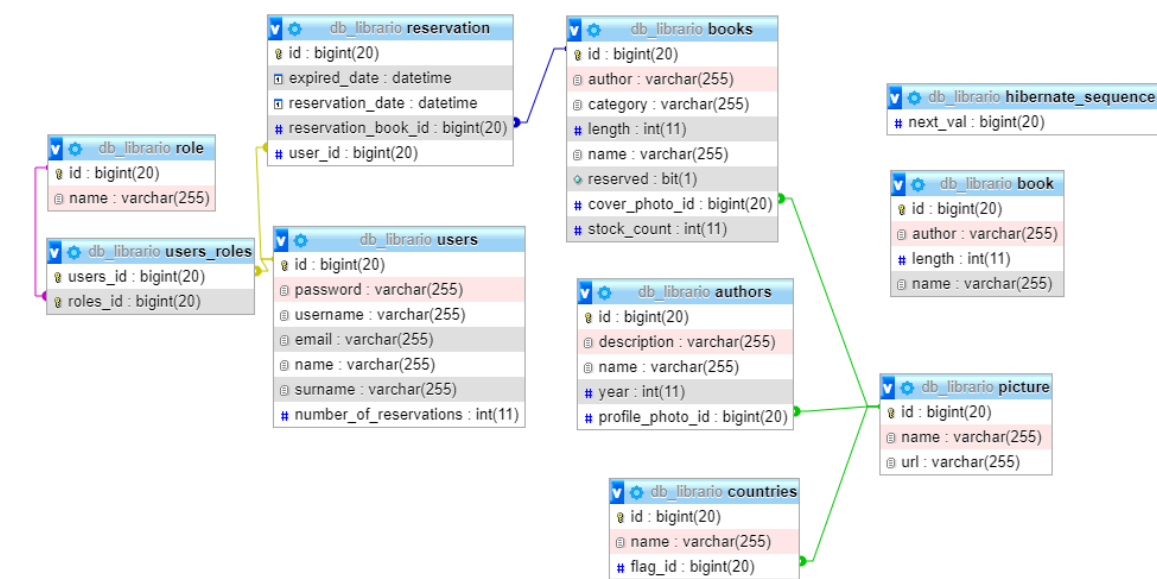


Struktura oraz zależności pomiędzy klasami obejmującymi funkcje administratora systemu:



3. Model danych

Aktualna struktura bazy danych:



Baza danych została oparta o schemat relacyjny. Wyróżniamy następujące tabele:

- **role** - w której przechowywane są role użytkowników. (admin, pracownik, użytkownik)
- **users** - w której przechowywane są dane o użytkownikach.
- **users_roles** - na podstawie której następuje identyfikacja ról dla danego użytkownika na podstawie user_id oraz id roli (użytkownik może mieć przypisanych wiele ról)
- **reservation** - w której przechowywane są dane o złożonej rezerwacji na książkę. kluczem obczym jest user_id - identyfikator użytkownika który złożył rezerwację oraz book_id - id książki której dotyczy rezerwacja.
- **books** - miejsce na dane o każdej książce dostępnej w bibliotece. Relacje definiują się na podstawie author_id (id autora) oraz picture_id (id zdjęcia, które jest okładką książki)
- **picture** - przechowujemy tam nazwę zdjęcia oraz ścieżkę do niego.
- **countries** (niezaimplementowane) - tabela z krajem pochodzenia autora
- **hibernate_sequence** - tabela pomocnicza frameworka hibernate (tworzona automatycznie przez niego).

Nie wykluczamy jej drobnych modyfikacji w kolejnej iteracji w fazie construction.