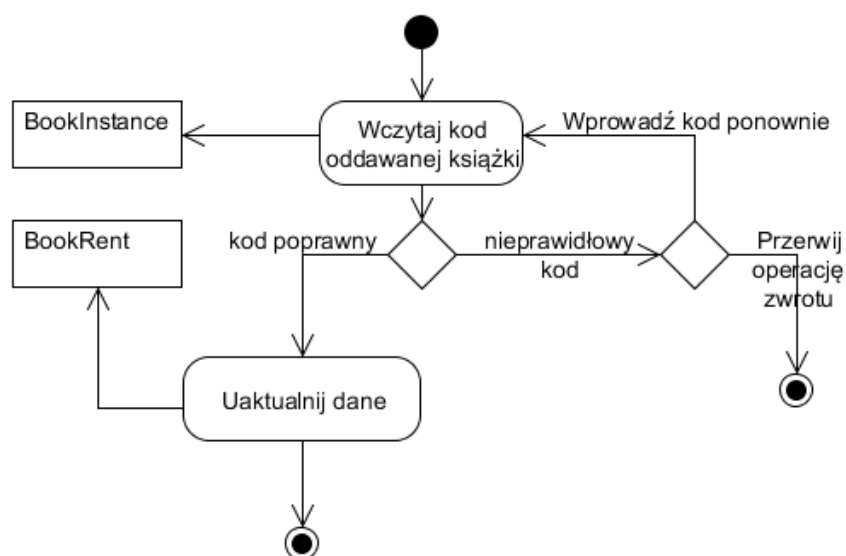


Modele UML
Projekt 'równodzień'
obsługa systemu bibliotecznego

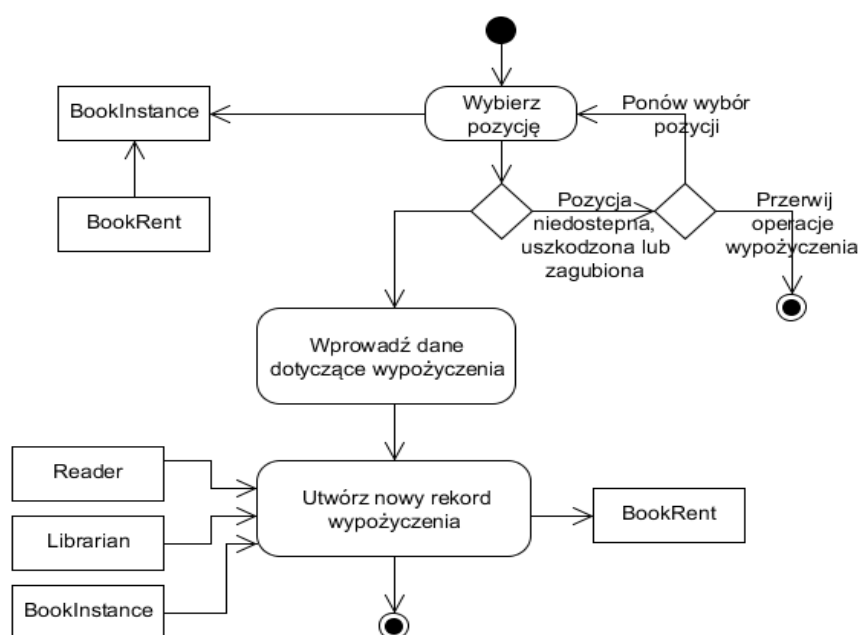
Diagram UML – diagram czynności dla „Dokonanie zwrotu”



Rysunek 1: Diagram czynności - dokonanie zwrotu`

W pierwszej fazie czynności "dokonywania zwrotu" użytkownik aplikacji (bibliotekarz) wczytuje kod oddawanej przez czytelnika książki. Po zatwierdzeniu sprawdzane jest, czy kod książki jest prawidłowy – czyli istnieje w bazie danych. Jeżeli kod jest niepoprawny możemy wprowadzić kod ponownie lub zakończyć daną czynność. Jeżeli kod jest poprawny, uaktualniamy dane, odwołując się do tabeli *BookRent* zawierającej dane o aktualnych wypożyczeniach. Po tych czynnościach książka jest dostępna do wypożyczenia przez innego czytelnika.

Diagram UML – diagram czynności dla „Dokonanie wypożyczenia”



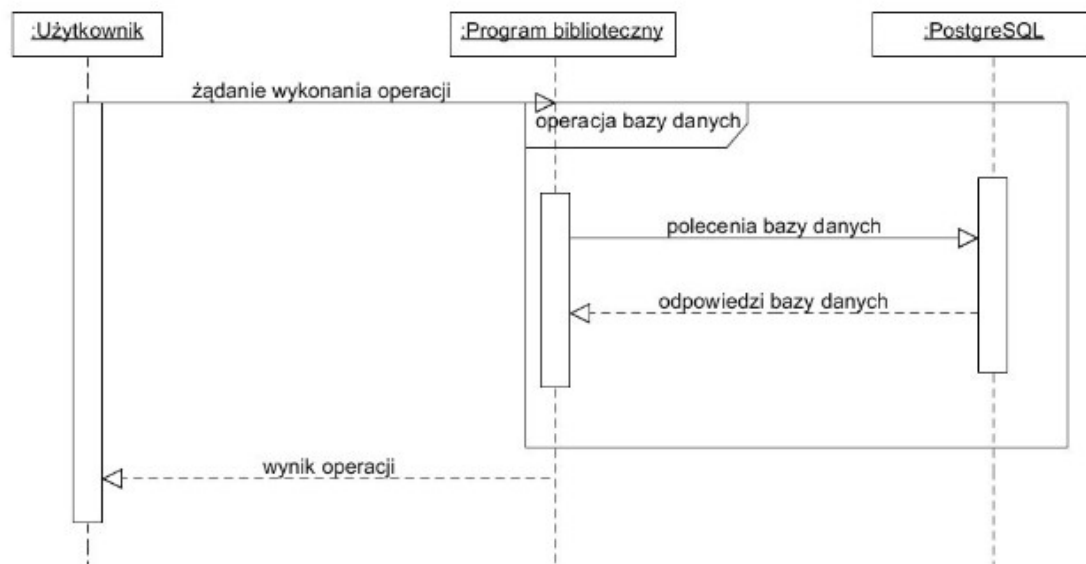
Rysunek 2: Diagram czynności - dokonanie wypożyczenia

Rysunek 2. przedstawia diagram czynności ukazujący sposób wypożyczania egzemplarzy książek przez użytkownika.

W pierwszej fazie wybierana jest pozycja z tabeli *BookInstance* oraz sprawdzane jest, czy dana książka jest możliwa do wypożyczenia (czyli nie jest już wypożyczona, uszkodzona lub zagubiona). Jeżeli wszystkie te warunki zostaną spełnione, przechodzimy do wprowadzania danych dotyczących wypożyczenia książki – kto, komu i w jaki dzień wypożycza dany egzemplarz. Na końcu tworzymy nowy rekord wypożyczenia i wprowadzamy go do tabeli *BookRent*.

Diagram UML – diagram sekwencji

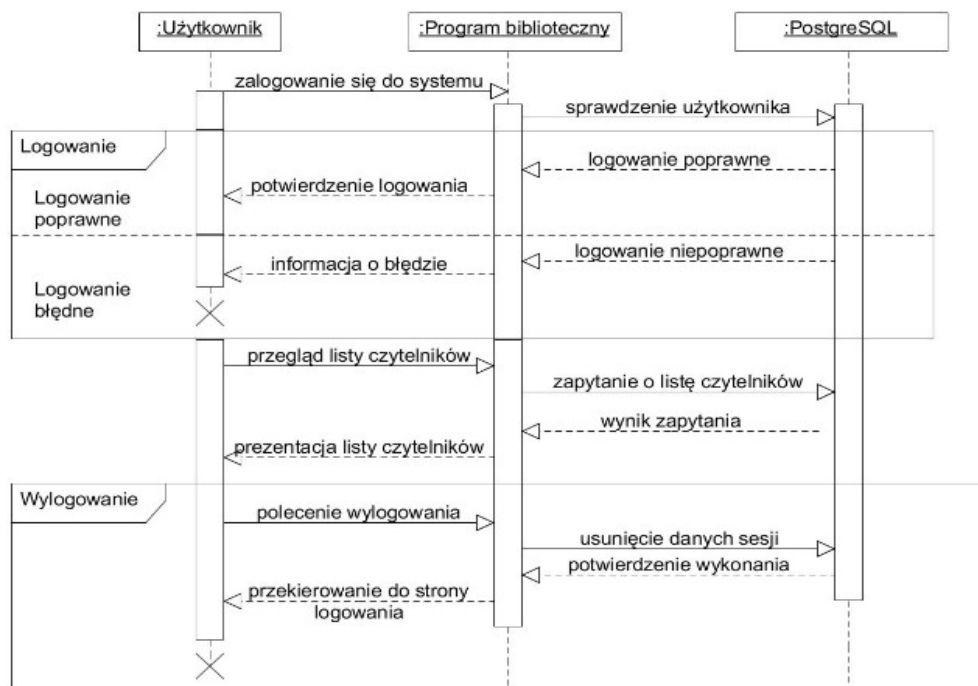
Diagram sekwencji, ze względu na fakt że projektowany system biblioteczny jest z inżynierskiego punktu widzenia elementarnie prosty, wymagał właściwie jednego diagramu – opisującego ogólne działania jego funkcji. Wspólnym mianownikiem funkcji systemu jest konieczność odwołania się użytkownika do serwera za pomocą żądania HTTP i odwołania się systemu do backendubazy danych. Ten właśnie wspólny mianownik zaprezentowano na poniższym diagramie.



Rysunek 3: Diagram sekwencji

W ten sposób zachowują się prawie wszystkie zwołania bazy danych – oprócz tych, które nie wymagają pobierania bazy danych, jednak mimo to, dla uproszczenia, program biblioteczny będzie nawiązywał połączenie z bazą.

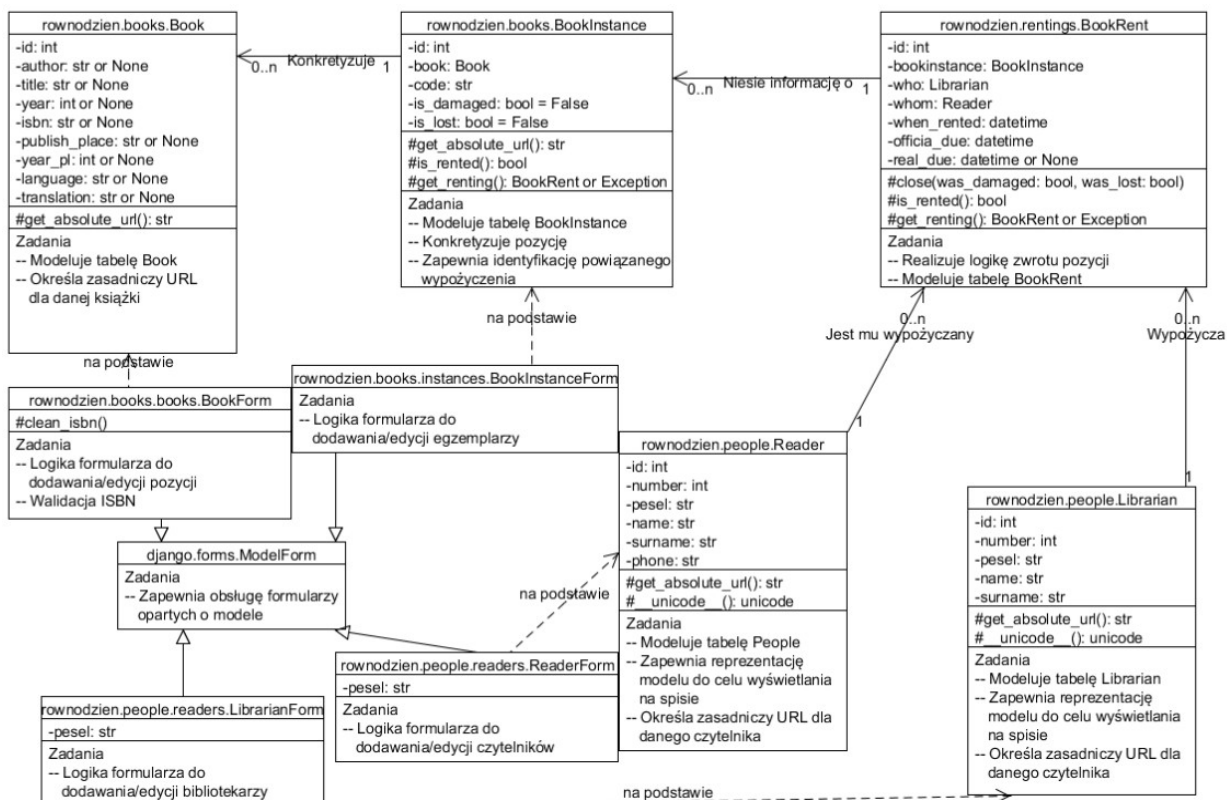
Można poczynić analogiczny schemat, rozpatrując interakcję użytkownika holistycznie – jako całą sesję. Tak więc sesja użytkownika jest czasem (wliczając) od zalogowania się do wylogowania się. Taką sesję – biorąc za przykład przeglądnięcie listy czytelników – przedstawia Rysunek 4.



Rysunek 4: Diagram sekwencji – diagram ten przedstawia przykład operacji przeglądnięcia listy czytelników w szerszym kontekście sesji użytkownika – zaczynając się zalogowaniem, a kończąc się wylogowaniem.

Diagram UML – diagram klas

Projekt, ze względu na to że pisany jest w obiektowym języku programowania wykorzystuje szereg klas. Zostały one zaprezentowane na poniższych diagramach UML. Prezentuje ona zarówno klasy-modele jak i klasy-formularze.

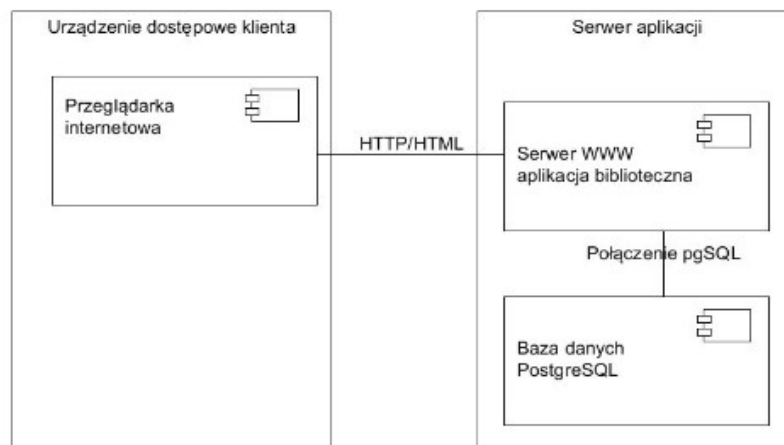


Rysunek 5: Diagram klas

Za pomocą notacji implementacji interfejsów przedstawiono tutaj wykorzystanie klas modeli przez klasy formularzy. Nie jest to *de facto* implementacja, ze względu na fakt że Django przeprowadza tutaj bardziej dynamiczną introspekcję klasy, jednak ze względu na ograniczenia UML-a w tej kwestii zdecydowano się na właśnie taką notację.

Diagram UML – diagram wdrożenia

Jak już zaznaczono wcześniej, projektowany i wykonywany system wsparcia biblioteki nie jest systemem złożonym. W zasadzie jego część wykonawcza składa się z dwóch części – aplikacji internetowej wspieranej przez system WWW i części bazodanowej. Klientem dostępowym klienta – bo i o nim nie wolno zapomnieć – jest urządzenie dostępowe (tablet, komputer PC, smartfon) wyposażone w przeglądarkę internetową zgodną z HTML5.



Rysunek 6: Diagram wdrożenia

Oczywiście możliwym wdrożeniem jest separacja serwera WWW od bazy danych, jednak w skali projektu takie rozwiązanie nie będzie celowym ani wytłumaczalnym kosztowo.