TEAM BEANS

Design Patterns of GreenBox Audiobook Lending System

Wenzheng Kang

Michael Plante

Sabbir Ahmed

Trevor Swann

Ian Wittler

Table of Contents

Design Pattern: Singleton Pattern

According to the problem statement, the customer reserves an available book at a specific kiosk location. The customer will go to one specific kiosk and retrieve audiobooks. The kiosk will be the single point of access for users to take action on requests and reservations. Hence, we will implement the Singleton design pattern to restrict the instantiation of a Kiosk class and ensure that only a single instance of it exists. Restricting the number of instances for Kiosk prevents stale or duplicate Request and Reservation objects. Without the need to instantiate the object of the Kiosk class, other classes can access the single Kiosk object directly.

Design Pattern: Observer Pattern

In the Greenbox system, requests for a title at a kiosk will be fulfilled when an audiobook is returned to the kiosk. The Kiosk needs to notify the requestQueue to service the first request for that title and the reservationQueue to automatically create a reservation for the customer for the book and will send an email to the customer as a notification of the reservation. Hence, the Kiosk object needs to notify the requestQueue and the reservationQueue and call their methods whenever the state changes. This can be implemented using the observer design pattern. It can prevent the object from asking the subject for status updates at regular time intervals, and it can save resources for the Greenbox system.

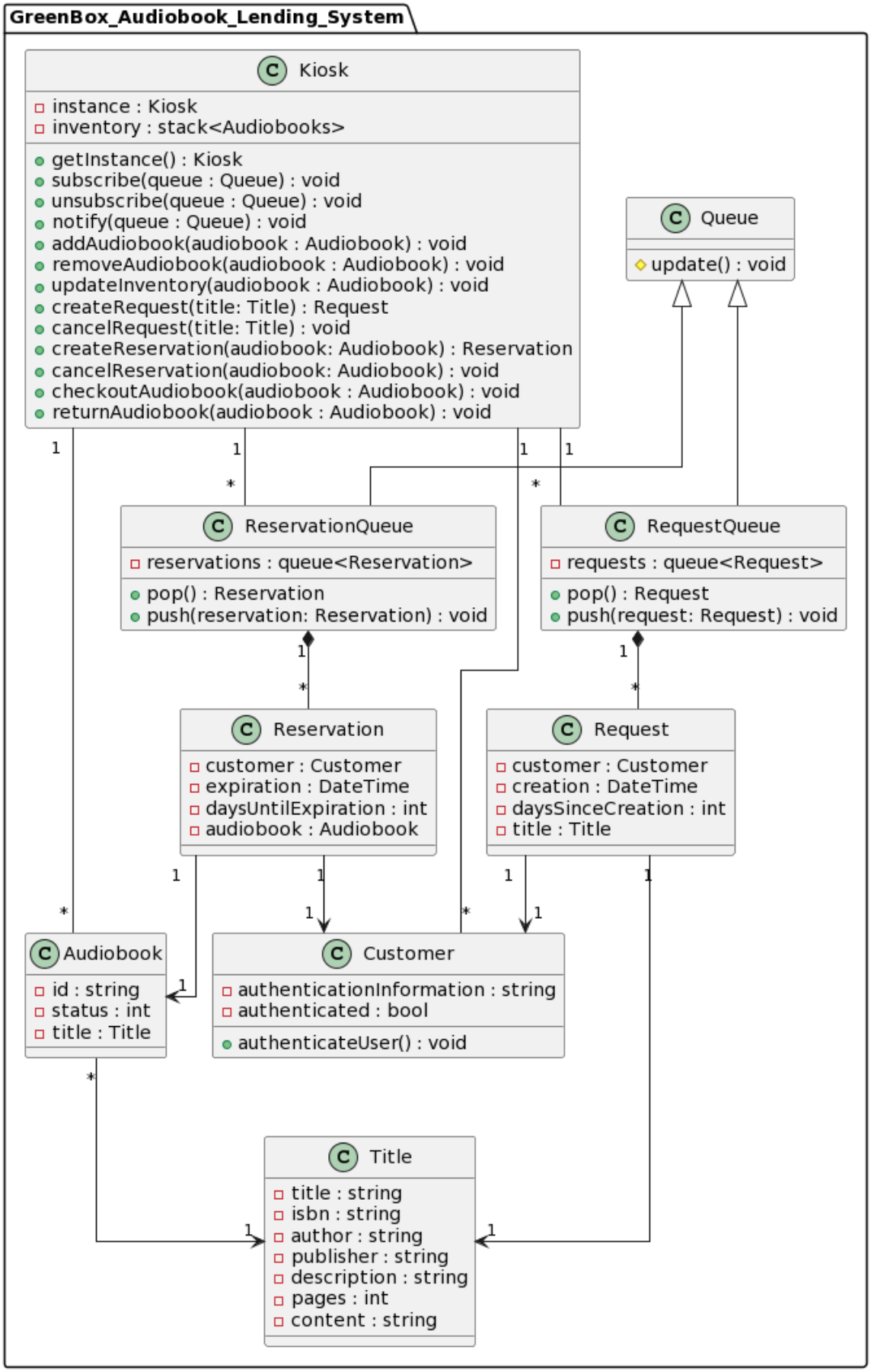

Figure 1: Class Diagram - GreenBox Audiobook Lending System Ver.2.0