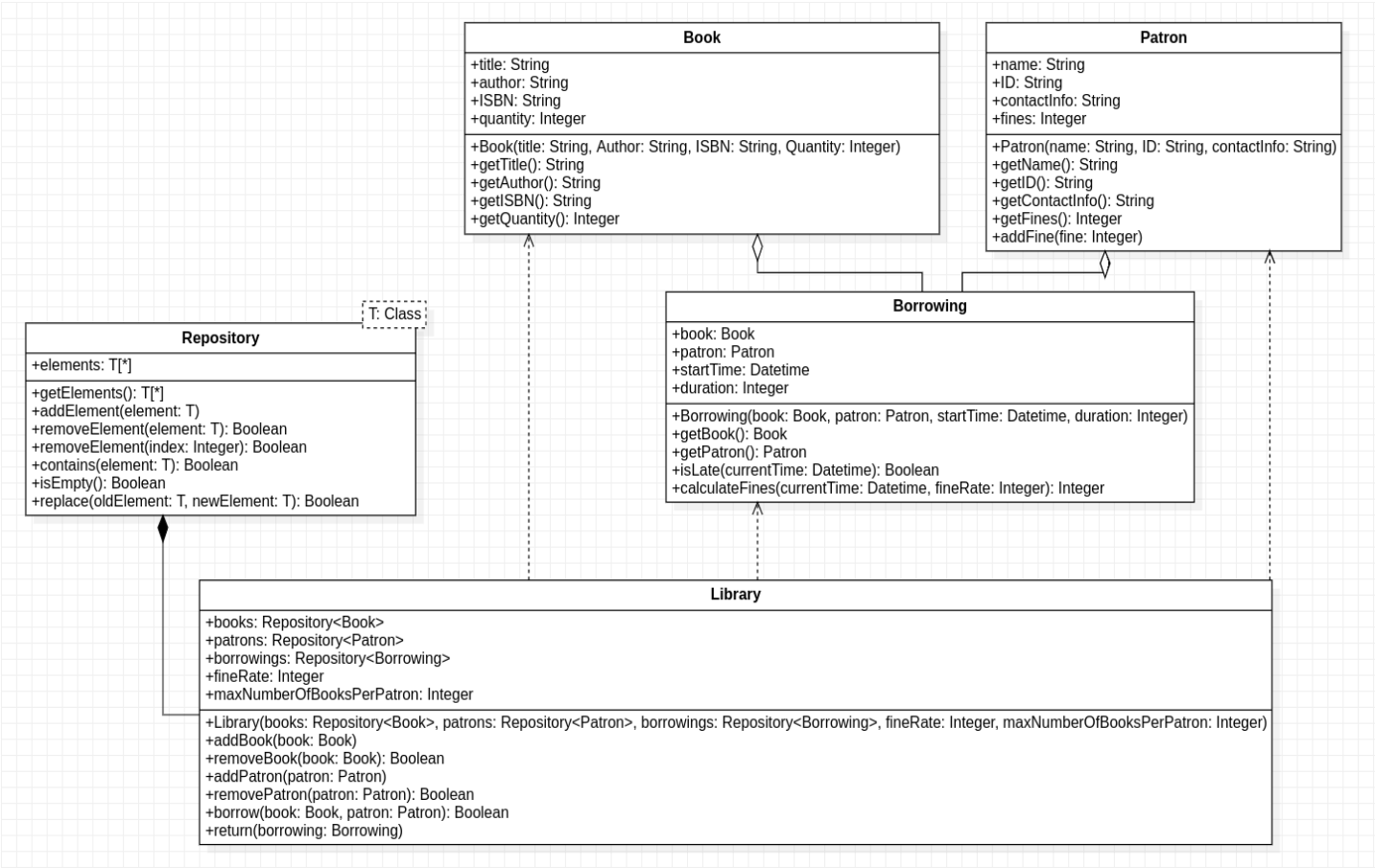
Problem 1 – Library Management System

1. Class diagram



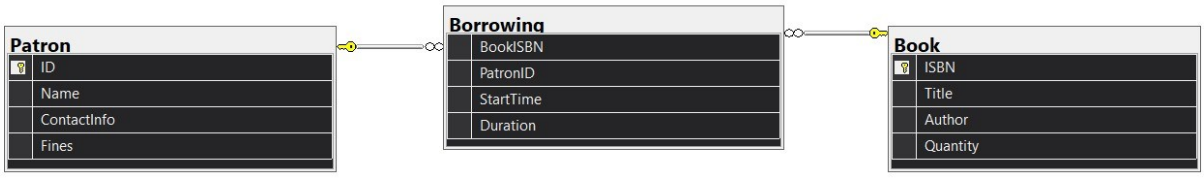
Book and Patron are the basic entities in this class diagram. Besides the given attributes, I added to the Patron class a field for keeping track of the total value of the fines.

Borrowing is made by the composition of the previous two, it references a book borrowed by a patron. Based on the startTime of the borrowing and the allowed duration it is decided if at the current moment the return would be late or not.

The Repository class is a generic one (or has a template parameter) which is used for holding and retrieving domain objects.

Finally, the **Library** knows about the books it has, the patrons it has and the borrowings so far. This class is responsible for borrowing a book to a patron (checks if there are books of the required type available and if the patron is allowed to borrow more books, if these requirements are met, then a new borrowing is added to the collection) and returning a book from a patron (fines are applied, if it is the case, and then the borrowing is removed).

2. Database diagram



In the database diagram, the entries in the tables **Patron** and **Book** are uniquely identified and referenced (primary keys) by the fields ID and ISBN, respectively. These two are joined by a third table, **Borrowing**, which has to foreign keys which reference the previous two primary keys.

Between the Patron and the Borrowing tables we have a **one-to-many relationship** (a patron can be involved in multiple borrowings), and between the Book and the Borrowing tables we have a **one-to-many relationship** also (a book can be borrowed multiple times).

In the end, this gives us a **many-to-many relationship** between the tables Book and Patron (multiple books can be borrowed by multiple patrons).