Implementing an API for Book Management System

Features:

1. Each user has an initial account number and includes a certain amount
2. The user can borrow books from the library (deduct a certain fee after each loan return), and cannot borrow books when there is no balance.
3. Query the account of a user and the details of the current borrowed book, The parameter is the user.
4. Query the current remaining number of each book, the total number of loans, and the current loan status between the users.

API interface that needs to be implemented:

1. Create a user interface, the requested parameters support setting the initial amount, returning the user ID
2. Create a borrowing transaction with parameters for the user's ID and the book's ID
3. Create a return transaction with parameters for the user's ID and the book's ID
4. Query the account status of a user, the parameter is the user ID, return the current balance, and borrow books.
5. Query the actual income of a book, the parameter is the ID and time range of the book, and return the transaction amount obtained by the book during this time.

Supplementary part:

1. You can choose any Ruby framework to complete this API
2. Please provide API documentation to let us know how to request your service.
3. Record the design ideas in README.md
4. To save time, the user module can be considered using a mature Gem implementation.
5. If time permits, it is best to deploy a demo address that can be called directly
6. At any time, the user's balance must be greater than or equal to 0.
7. Regarding the book income function, if there is no time, you can not do it.

Try to implement all the functions, we will evaluate according to the following points:

1. How many features are completed and the quality of the completion
2. The design, code quality, etc. of the entire application
3. Overall/partial solutions and design ideas
4. Test code coverage and quality

Submit:

After this test is complete, you can push the code onto GitHub and the email tells us the corresponding Repo address and the URL that can be called (if any).