Library Book Borrowing System

Our team has been hired to implement a Library Book Borrowing System. For first release of the software, we will implement a Rails application with the basic set of features: maintaining a book catalog, book loan management (borrowing), and book loan extension. The team has decided to divide the release into three sub-features and you have been selected to implement the book loan extension.

Since all the three sub-features access the same database, the team has designed the following schema. You first task is then, set up the database by writing the corresponding migrations.

The product manager has agreed with the customers on a sequence of wireframes that illustrate the underlying process. The first wireframe illustrates the situation where a user has borrowed two books (loan period: 9th to 16th of January).

The second wireframe illustrates the "happy path" for book loan extension: the user requests a loan extension for the book “The Ruby way” to which the system replies “Extension accepted” because the book is available for the requested period of time. Finally, the third wireframe illustrates the case where a loan extension cannot be accepted (e.g. another user has reserved the book in the requested period). As depicted in the wireframe, the system must notify “Extension rejected” accordingly.
A Gherkin user story capturing two scenarios (similar to the ones illustrated with the sequence of wireframes) can be found at: https://goo.gl/9olqhJ.

You are free to use HAML, ERB or AngularJS for implementing the views. In fact, you need to design only one view template.

For grading purposes, you are required to follow the BDD-TDD cycle and to record every step in the cycle in your Bitbucket repository. The task breakdown and corresponding marks are as follows:

- (BDD) Implementation of all cucumber steps 10 pts
  Cucumber steps must be specified with Capybara. The application must be updated accordingly.

- (TDD) Controller and model rspec tests 15 pts
  * Write a controller test to check that a loan extension is rejected when it conflicts with an existing book reservation.
  * Write a test and applicative code to implement of the following rules:
    - A book loan must not be extended more than 4 times.
    - Books are classified as type A, B or C. The classification serves as a guide for specifying the normal loan duration, which is set to 7, 14 and 21 days, respectively.

- Database related tasks (at least one database migration) 10 pts

- Evidence that you followed the “BDD-TDD cycle” approach 5 pts
  Commit every phase in the “BDD-TDD cycle” (one point will be deducted for each step that is found test and code updated in the same commit).