

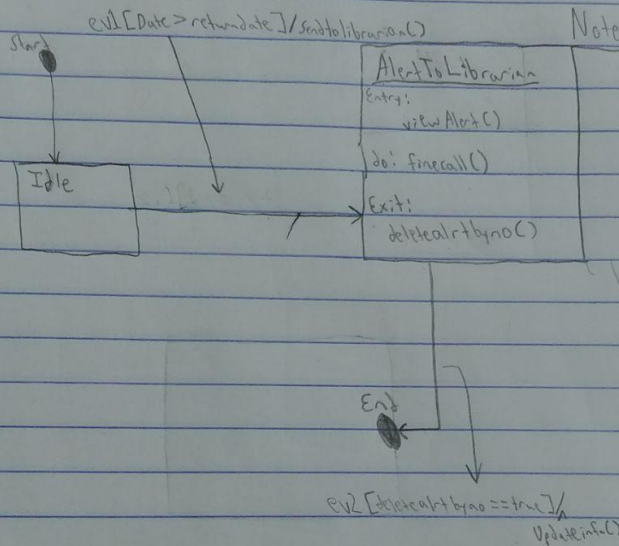
Create a simple single-use case using this Class Diagram. (30 mins)

In other words, as a student, I want to be able to become a member, then speak to a librarian and search the catalog for a specific book, which the library gets and returns to me to be rented out for two weeks. This use case represents the vast majority of the class diagram illustrated on the quiz. In the question below, I will be creating a state diagram to represent a portion of this use case: an alert being sent to a librarian about an overdue book, and the librarian viewing the alert, sending out a fine, and deleting the alert.

Once you have a Use Case, construct a single State machine/State Chart associated with the Use Case and Class. (30 mins)

The state diagram is below. When there is still time for the book to be returned, to when the book is overdue, is a change in state. That is represented in the diagram. It utilizes the Alert class and the Librarian class.

Quiz 2 Question 2



Notes: Behavior model

- Don't need and or nested states
- if overdue, send an alert to librarian.
- librarian checks alert, does finecall(), then deletes the alert.
- This is a state machine for if there's an overdue book that is not returned.
- I'm assuming that finecall() does more than just return the amount of the fine for a book. finecall() should use the user's information to bill them if it is called.
- The Updateinfo() updates the availability of the book in the catalog.

Transcript for my notes:

- Behavior Model.
- Don't need "and" or "nested" states.
- If overdue, send an alert to the librarian.
- Librarian checks alert, does `finecall()`, then deletes the alert.
- This is a state machine for if there's an overdue book that is not returned.
- I'm assuming that `finecall()` does more than just return the amount of the fine for a book. `finecall()` should use the user's information to bill them if it is called.
- The `Updateinfo()` updates the availability of the book in the catalog.