Test 1

**CS425 - Software Engineering**

**January 18, 2015**

**35 Points Total – Write all answers on the separate paper provided.**

1. **Use Case Description** (16pts)

Create a Use Case Description for the following new feature for our library example. For this use case show the basic success flow.

We want to allow library members to renew books that they have checked out. Library members can review the books they have checked out and select those that they want to renew. Books that have a hold on them by another library member cannot be renewed. Also, books that are already overdue cannot be renewed. The standard renewal time is 2 weeks. Books may only be renewed one time.

==============================================================

# Library Member renews a book

## Brief Description

## This use case allows a library member to renew books that they have checked out

## Actors

## Library Member

## Preconditions

Library Member is logged into the system and at the library member welcome page.

## Flow of Events

#### Basic Flow

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| 1. Library member selects the renew book form | 1. The system retrieves the books that this library member has checked out and displays them.   . |
| 2. Library member selects a book and selects renew. | 2.1The system checks that this book does not have a hold on it, is not currently overdue, and has not been renewed already.  2.2 If all those cases are met the system updates the due date to add an additional 2 weeks, updates the book to have been renewed, and gives a success message to the library member with the new due date.  Otherwise, the system returns a failure to renew message with the reason. |

#### 

## Post-Conditions

For the success case the book checkout is updated to be 2 weeks in the future and the book is marked as renewed. This information is saved to the database.

## Business Rules

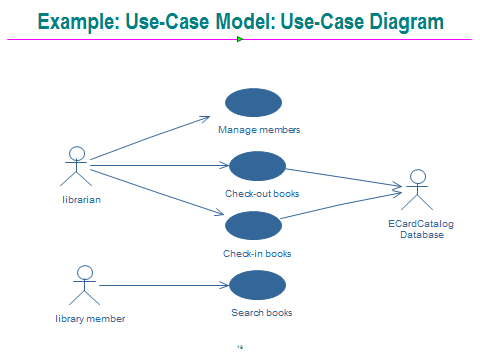
* 1. Books cannot be renewed more than once
  2. Overdue books cannot be renewed
  3. Books with a hold on them cannot be renewed

## Nonfunctional Requirements -- none

==============================================================

1. **(6pts) List and give a short example or definition (in one sentence) of 3 Architecture analysis outputs. For each of these outputs indicate if that output will change due to adding this new feature.**
   1. *Key abstractions -- this is our first high-level class diagram which shows key classes we need and obvious attributes. We should add new attributes to the book key abstraction class to determine if it has a hold, has already been renewed, and is overdue.*
   2. *Architecture mechanism – these are system wide solution for common design problems. For example ORM or web application. These should not change.*
   3. *System structure and layering. For example we may be using MVC for our system structure. This will not change.*
2. **(2 pts) Create a Use Case Diagram for our library example and indicate how it is changed with the addition of this new feature.**

*Below is the old use case diagram for the library. We need to add a use case for library member to renew books. Also, you may want to add that librarians can renew books for completeness.*

****

1. **(2pts)What is the purpose of boundary analysis classes? Explain how many boundary analysis classes we will have per use case.**

*Boundary classes are used to interface to the actors. We will have one boundary class for every actor for our use case.*

1. **(4 pts) What is the purpose of entity analysis classes? Give an example of three entity analysis classes that we will need for this library use case.**

*Entity classes are used for data we want to save/persist. Any 3 of the following are correct for entity classes:*

* 1. *library member*
  2. *book*
  3. *checkout-record*
  4. *renew-record*

1. **(2pts) What is the purpose of control analysis classes? Explain how many control analysis classes we usually have per use case. Explain why that number might vary.**

*The control class implements the business logic for the use case. We usually have one control class per use case. For simple use cases with no business rules, we may end up getting rid of the control class. For complex use cases with many business rules we may add helper control classes.*

1. (**3pts) Most faults are introduced in what phases of software engineering? Give a short description of two ways RUP attempts to address problems during this phase.**

*Most (approximately 2/3 of the system) faults are introduced during analysis and design.*

1. *Using use cases to create the SRS improves the quality of our system analysis and makes them easier to review with customers.*
2. *Iterative development finds problems sooner if there is a mismatch between our requirements and customer’s expectations.*