CMPE 275 Section 2 Fall 2016

Term Project

Last updated: 11/10/2016 (status: draft)

In this project you build a Library Management System (LMS) for a university. This must be a web app hosted in the cloud. The primary language you use for server implementation *must* be Java. You do not have to use Spring, but you need to exercise the principles, patterns, and methodologies that you learn in the class, such as DI, AOP, MVC, ORM, and transactions. You *must* use either a relational database, or Datastore if you choose AppEngine.

If any feature described below is unclear or ambiguous, and you fail to get a clear answer from the instructor or TA, you can use your best judgement to interpret and add the missing details, provided that you clearly document and explain your reasoning in the product report.

Functional Requirements

This app manages many aspects of a university library system, including cataloging, search, circulation, and waiting list. The interface must be web based, and the server needs to be hosted on the cloud, and accessible from anywhere with Internet connection.

Users and Authentication

There are two roles of users, librarian and patron.

1. A librarian manages cataloging, and can assist circulation as well.
2. A patron is a customer of the library. He can search for books, borrow and returns books.
3. For simplicity, we allow any user with any email address to be able to create his account using an email as the username, and password of his choice. The user also needs to provide a university ID of 6 digits.
4. Your app must send an email to the user with a verification code. The user needs to use that verification code to complete his account registration. A registered user cannot really use features in the system until his account is verified. A confirmation email must be sent to the user after completion of account verification.

5. For simplicity, we only and automatically register any user with an SJSU email account (@sjsu.edu) to be a librarian. A librarian cannot be a patron, which means he has to use a different email to register if he needs a patron account as well.
6. No two patrons can have the same university ID, neither can two librarians.

Cataloging

A librarian must be able to manage the catalog of the books.

7. A book item contains at least the following properties
    a. Author
    b. Title
    c. Call number
    d. Publisher
    e. Year of publication
    f. Location in the library;
    g. Number of copies
    h. Current status
    i. Keywords
    j. Coverage image (optional)
8. A librarian must be able to search, add, update, and delete books.
    a. Search capability needs to function as a superset of search features to be described below for patrons.
        i. The minimal feature to add for search is the capability to search for books created or updated by a particular librarian.
    b. Update and deletion must be able to work together with search, i.e., a librarian must have the capability to find a book through search, then decide to update/delete it.
    c. A book cannot be deleted if it's checked out by a patron.
    d. Deleting a book also removes the waiting list for it, if there is any.
9. [Bonus Feature] You can integrate with external APIs to simplify the input the book info based on things like ISBN.

Circulation

10. A patron must be able to check out up to 5 books in any day.
    a. Each check out transaction can handle up to 5 books.
    b. A book is due in 30 days from checking out.
    c. The checkout screen need to show the due date.

    d. A confirmation email is sent to the patron for each checkout transaction with the details, including the book info, the transaction date and time, and the due day.
11. The total number of books a user can keep at any given time cannot exceed 10.
12. If a book is due within 5 days, daily alerts will be sent to the patron; and the patron can renew the book for another 30 days before the book is overdue. A book can be renewed twice, which means a maximum of 90 days in total from checked out. If, however, there is a waiting list for the book, it can no longer be renewed.
13. A patron must be able to return up to 10 books in one transaction.
    a. Upon returning, an email confirmation should be sent to the patron with the detail of the transaction.
    b. If any book is overdue, a fine of $1 per day will be enforced.
        i. If the overdue period is not more than 24 hours, it is counted as one day
        ii. If it is more than 24 hours, but not more than 48, it is counted as two days
        iii. So on and so forth.

Waiting list

14. A patron can add himself to a waiting list if the book he is interested in is currently checked out by somebody else. There is no limit on how many people can be added to a waiting list, yet the order in the waiting is preserved. You cannot add the same person to the same list twice for the same book.
15. When a book becomes available, the first person on the waiting list is notified about its availability, and the person is removed from the waiting list. The book is reserved for this person for three days, during which only this person can check out this book. After these three days, the reservation is cancelled, and the next person in the waiting list (if there is any) is notified about the availability, and a three-day reservation is created for him as well. So on and so forth.

Testing assistance

16. For ease of testing and grading, you need to provide the capability for an librarian to set the current date and time for the app. This way it is easier to test features like due date and waiting lists. This interface for setting the date and time must be positioned at the top right corner of your main UI pages, and the resulted date and time must be clearly displayed along with the time setting interface as well.

Additional Bonus Features

TBA

Grouping

This project is group based, with group size up to four people. Once the project plan is submitted, group membership cannot be changed.

Source Code Management

You are recommended to use a Source Control Management (SCM) system to manage your team's source code. This can be a private Bitbucket repository or your local git. During the grading of the term project, you may be asked to provide commit history or any other document to help evaluate each team member's contribution.

Cheating Policy

Your app must be built by yourself, and cannot be based on the code base of any existing app. If you used any code not written by yourself, it must be clearly documented in your README.TXT file, unless it is part of publicly available libraries. *If your app is already used to serve the requirements of any other class, it will not be accepted by this class*. In the case any form of cheating is confirmed, your will get an F grade for this class.

Deliverables and Grading

The project is worth 25 points in total. The actual *due dates* of the deliverables will be specified in Canvas.

Project Plan (1 point)

The project plan needs to include

- The platform and technology choices (Spring/EJB, MySQL, etc)
- Architecture diagram
- Division of work with feature owner specified
- Milestones with expected dates

You are recommended not to exceed 3 pages. Must be submitted through Canvas as a PDF file.

Project Presentation and Demo (5 points)

To be presented in class.

- The presentation should cover introduction, high level design, and major features with screenshot. Time limit: 3 minutes.

- You must also do a live demo. The guideline for how to do demos is to be added. Time limit: 5 minutes.
- Grading will be based on successfulness of the demo, the content and clarity of the slides, and the delivery of the presentation

The presentation slides must be submitted through Canvas as a PDF file.

Project Report (4 points)

The report needs to cover the following topics.

1. Motivation and introduction of your app
2. High level and component level design
3. Technology choices
4. Description of features with final screenshots
5. Testing plan executed and results
6. Lessons learned and possible future work

You are recommended not to exceed 15 pages, but you will not be penalized just because the report is too long or too short, as long as the level of coverage for the required topics is reasonable and clear. The report must be submitted through Canvas as a PDF file.

Project App (15 regular points + 2 bonus points)

Note: the instruction for submission is still *subject to change*.

1. Your must submit all your source code / resource files through Canvas
2. Features correctness, stability, performance, choice of technology and implementation are worth 12 points
3. User interface and user experience are worth 3 points
4. The bonus features are worth 2 points
5. You need to keep your app live for at least a week before we finish grading
6. README.TXT, including
    a. The names, email IDs, and students IDs of the members
    b. The URL to access your app, and the admin account username/password, so that the user can test the provisioning features
    c. Any other instruction necessary for the TA to grade the app
    d. Build instructions