CS5200 Semester Project Report

December 9, 2014

Team Name: BBQ -- Buy Book in Queue

Members: Hao Mao,
Wanting Jiang,
Zhuoli Liang

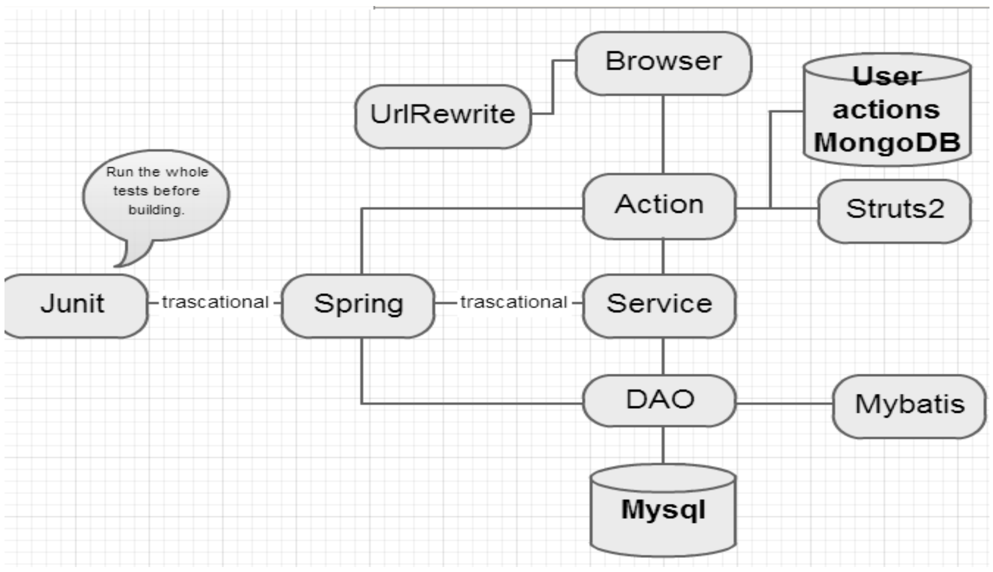
Problem statement:

The goal of this project is to design and build a online used book market that provides an efficient and convenient trading platform for sellers and users. By logging into the website, users can post books they want to sell, set the prices. Users can view the books on the website. If they find someone they want, they can add them in card and checkout, at the same time buyers can send message to the seller if have some questions. Administrators of the website can manage books and users information, send and receive messages through the website and view the daily logs.

Proposed solution

Language: Java,JSP
Database: Mysql, MongoDB
Tools: Eclipse, IntelliJ, SQLyog, Maven, GitHub, Jetty
Framework: Struts2, Spring, MyBatis, UrlRewrite, Jquery(Ajax)

Architecture:



When users enter URL address of our web site to the browser, the UrlRewrite first perform as a filter to rewrite URLs before the web application server get to the code. After rewriteing, it will parse the request to Action part. In this level, we use framework Struts2, to call action to handle the HttpServletRequest.
Next, the request will be parsed to the service level. This level contains business logics, it parse the action before get data from the database. All the actions in this level are transactional.
Services will call DAOs to get data from Mysql. Here we use Junit to test code in Action, service and Dao, and all the test are transactional.
We use MongoDB to record the daily logs of the website.

APIs:

For this project, the data of this project comes solely from the Seller and Buyer, so we currently do not use any API or outside data source.

Technology stack:

Language and Web Framworks: Java Spring Structs Mybatis
Web Server: Jetty
Database: Mysql and MongoDB

Use cases:

USE CASE – **Login**
Login [login]
-Description: User enter username and password, login to the website.
-Actors: All users
-Precondition: user have registered
-Step by step:
1. User - enters username and password
2. Website – check username and password, returns the login page to user if username and password is right, otherwise returns fault information

USE CASE – **Register to website**
Register to website [registerWebsite]
-Description: a new user who hasn’t register can create a new account to become a register buyer or seller.
-Actors: new register user
-Precondition: user who have not registered
-Step by step:
1. New user – enters username, password, email, gender, DOB and address.

2. Website – restores new user’s information, gives the new user an id.

USE CASE – **Update personal information**
Update personal information [updatePersonalInformation]
-Description: Users update personal information, change password, email or address.
-Actors: All registered users
-Precondition: user is login and can only update his own information
-Step by step:
1. Users – changes personal information
2. Website – update these changes to the database

USE CASE – **View Books**
View books [viewBooks]
-Description: User chooses a book or search a book, to view the detail information of this book.
-Actors: All users
-Precondition: books exist
-Step by step:
1. User - enters a good’s name, or click on a good’s image
2. Website – returns the good’s information to the user

USE CASE – **Add book to cart**
Add book to cart [addBookToCart]
-Description: Buyers can choose a book and select a quantity, then add this book to the shopping cart
-Actors: All buyers
-Precondition: user is login, book exists
-Step by step:
1. Buyers – click a book, choose quantity and add it to cart
2. Website – update the cart information

USE CASE – **Delete book from cart**
Delete book from cart [deleteBookFromCart]
-Description: Buyers can choose a book in his shopping cart, and delete it from the shopping cart
-Actors: All buyers
-Precondition: user is login, and book exists
-Step by step:
1. Buyers – click a book, delete it from the shopping cart
2. Website – update the cart information

USE CASE – **Make an order**
Make an order [makeOrder]

-Description: Buyers can select several books, choose a deliver address and submit this order.
-Actors: All buyers
-Precondition: user is login, shopping cart has books
-Step by step:
1. Buyers – submit an order
2. Website – record this order

USE CASE – **View Order**
View order [viewOrder]
-Description: User can select an order and view the detail information of this order
-Actors: All users
-Precondition: user is login, order exists and user is authenticated
-Step by step:
1. User – click on an order
2. Website – returns the order's information to the user

USE CASE – **Add a book**
Add book [addBook]
-Description: Sellers can add a new book to his shop, enter book's name, author, publisher, year, quantity and price.
-Actors: All sellers
-Precondition: user is login, the new book doesn't exist
-Step by step:
1. Buyers – input book name, price, quantity etc.
2. Website – record this book

USE CASE – **Update book information**
Update book information [updateBookInformation]
-Description: Sellers can select a book in his own shop and changes book's quantity, price and picture.
-Actors: All sellers
-Precondition: user is login, book exists
-Step by step:
1. Buyers – update a book's price, quantity and so on.
2. Website – update book's information to the database.

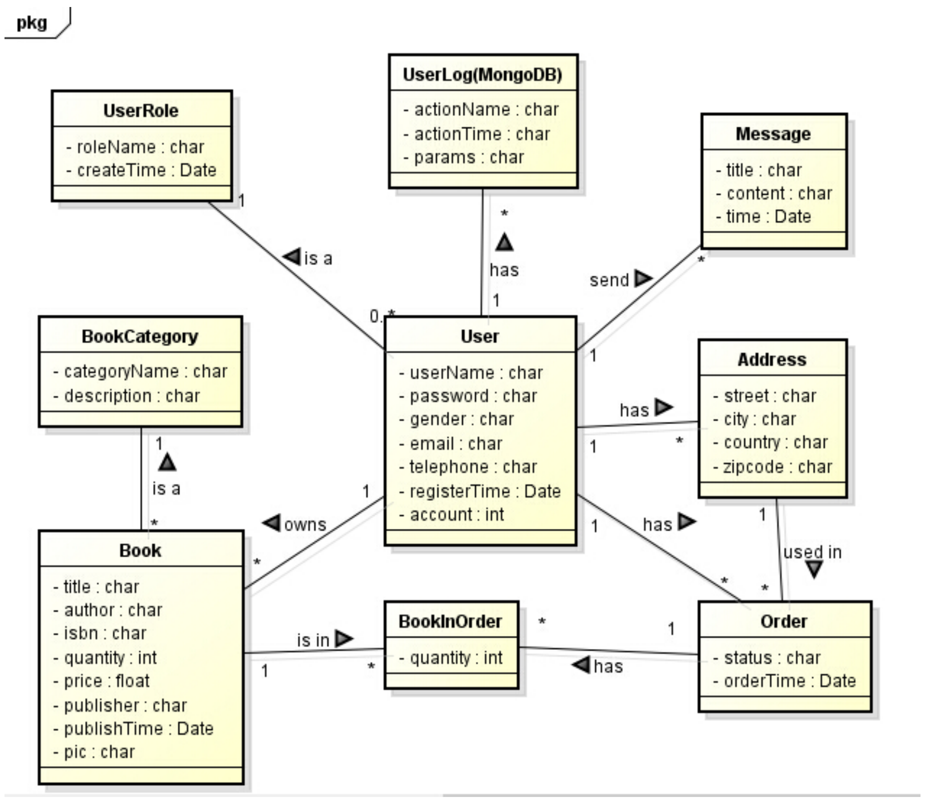
USE CASE – **View message**
View message [viewMessage]
-Description: User view a message
-Actors: all registered users
-Precondition: user is login, message exists
-Step by step:
1. User – click a message

2. Website – returns information of this message to user
3. User – view the message

USE CASE – **Send message**
Send message [sendMessage]
-Description: User sends message to other users
- Actors: all registered users
-Precondition: duser is login, message is not blank
-Step by step:
1. User – write a message and send it to someone
2. Website – record this message

USE CASE – **View Logs**
View logs [viewLogs]
-Description: Admin can view the daily logs of the website
- Actors: admin
-Precondition: admin is login
-Step by step:
1. User – enter the log pages
2. Website – show all the logs in database

Data model:



Usage Count:

As requested, following is the statistics of our project

Tables: 9

Fields: 33
Foreign Keys: 9

DAOs: 8

Create Methods: 10

Find One Methods: 12

Find All Methods: 9

Find Other Methods: 8

Delete Methods: 9
Update Methods: 10

Services: 8
Actions: 25

Pages: 16

Input Fields : 27

Links: 20

Buttons: 15

Data Tables: 6

Data Lists: 5

Javascript Libraries: 5
CSS Libraries: 5

Conclusion

Our web service works as expect, we implemented all the scheme tables and used the knowledge we learned from CS5200.

Github repository: https://github.com/Zhuoli/CS5200FinalProject