**Project #1: Library Management System (SQL application)**

**Project purpose –**

The main purpose of this project is to understand how to deal with databases with web/ desktop application and how to design relational model based on requirements. In this project, I created an online web application to search, check-in and check-out library books. This application is mainly for librarians who work at library. With the help of this system they can do following actions –

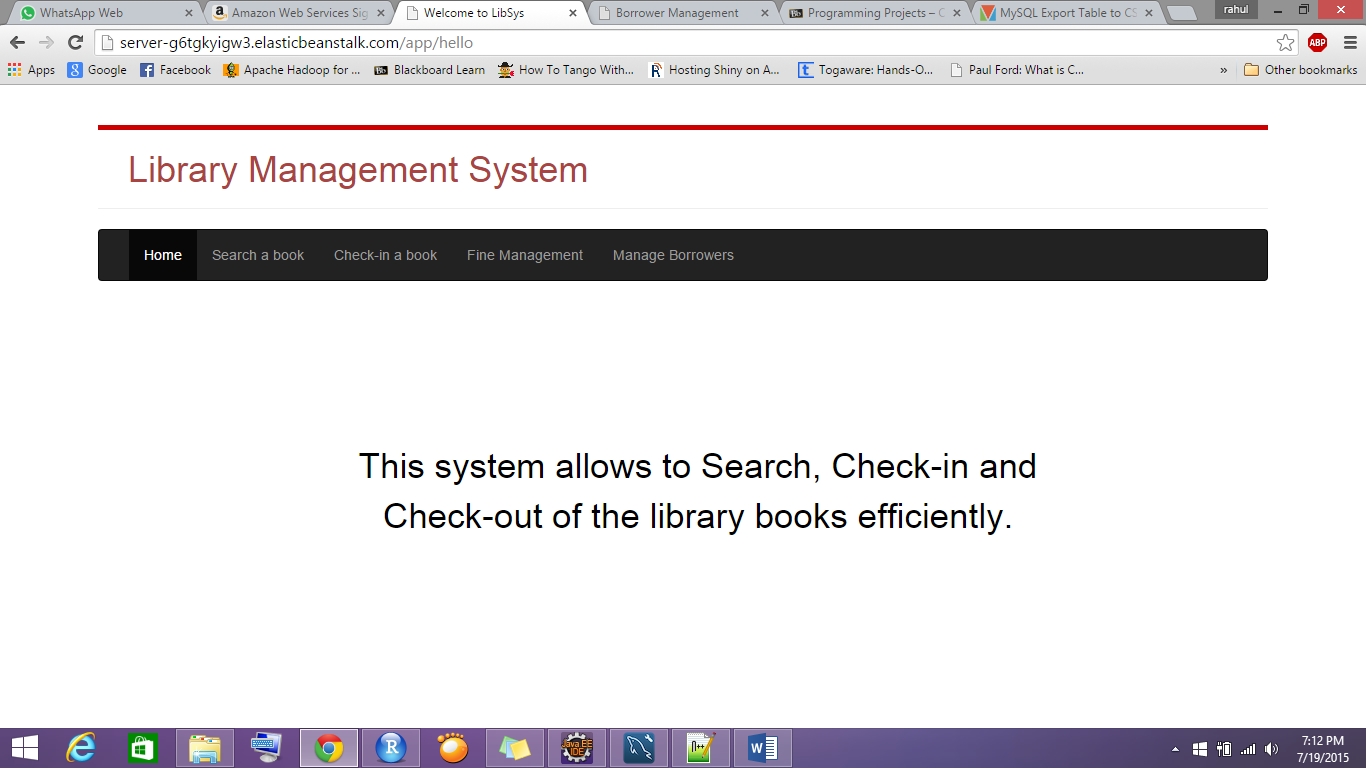
1. Create users/ Borrowers
2. Search a book
3. Check-in book for Borrower
4. Check-out a book for Borrower
5. Collect late fees (fines)

To use this web application librarians can use below link (This is hosted on Amazon Beanstalk) –

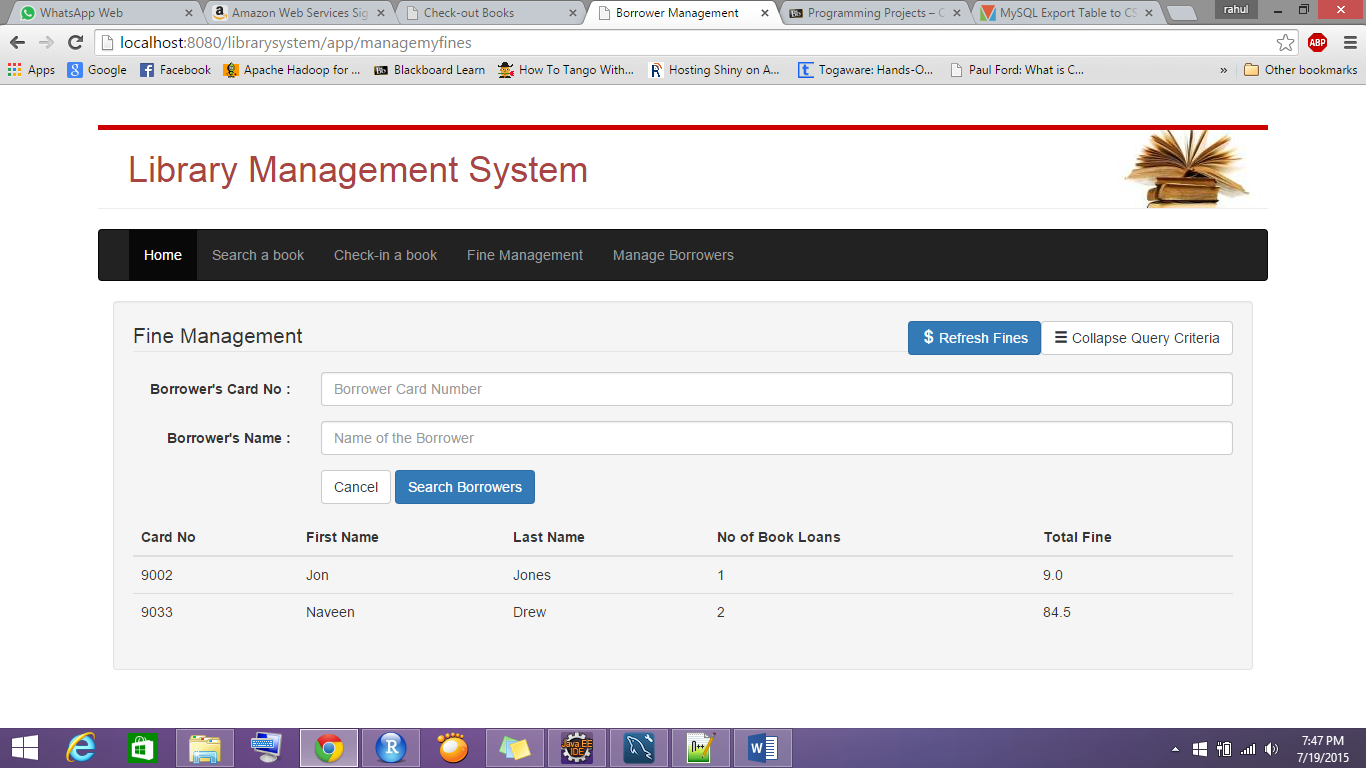
<http://server-g6tgkyigw3.elasticbeanstalk.com/app/hello>

**Instructions for Librarians –**

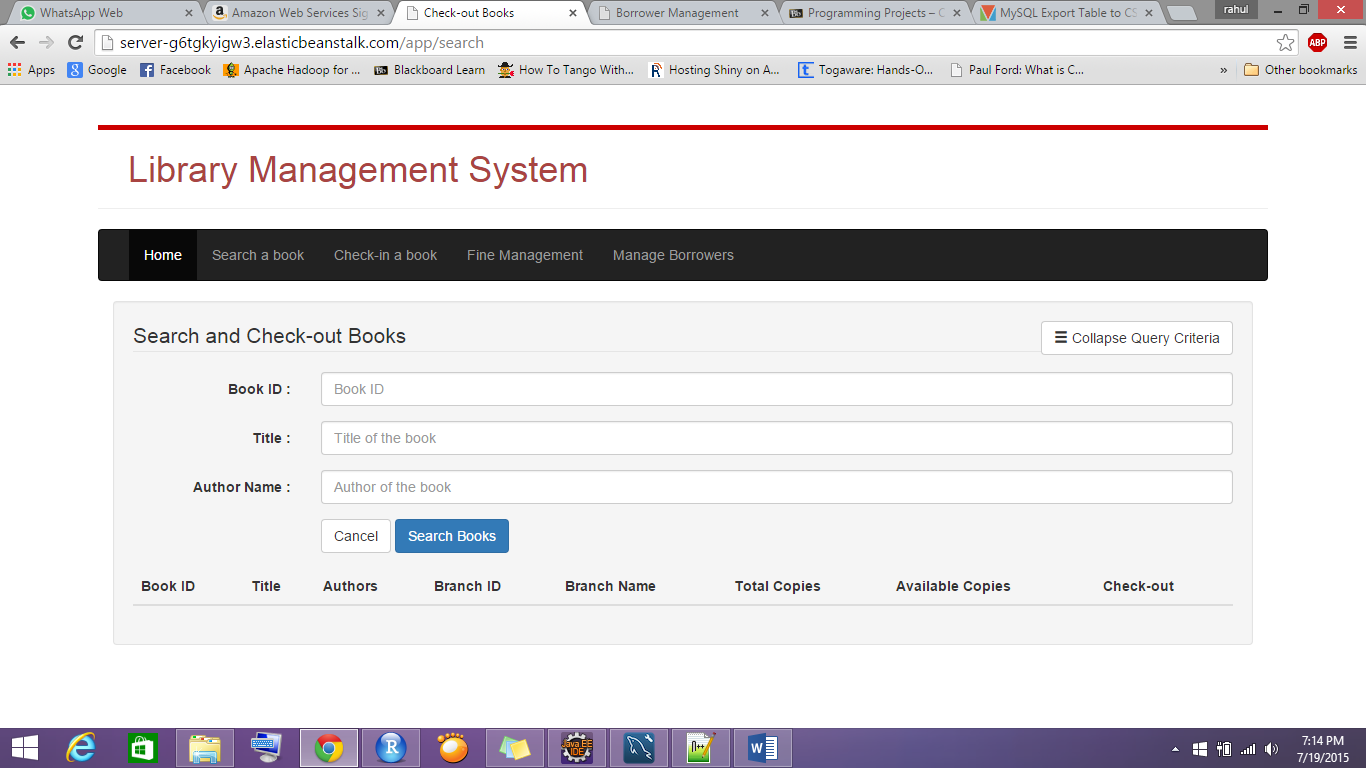
1. **Welcome Page –**



1. Navigation Bar at the top allows to navigate through different action items.



1. Menu –
   1. Search a book

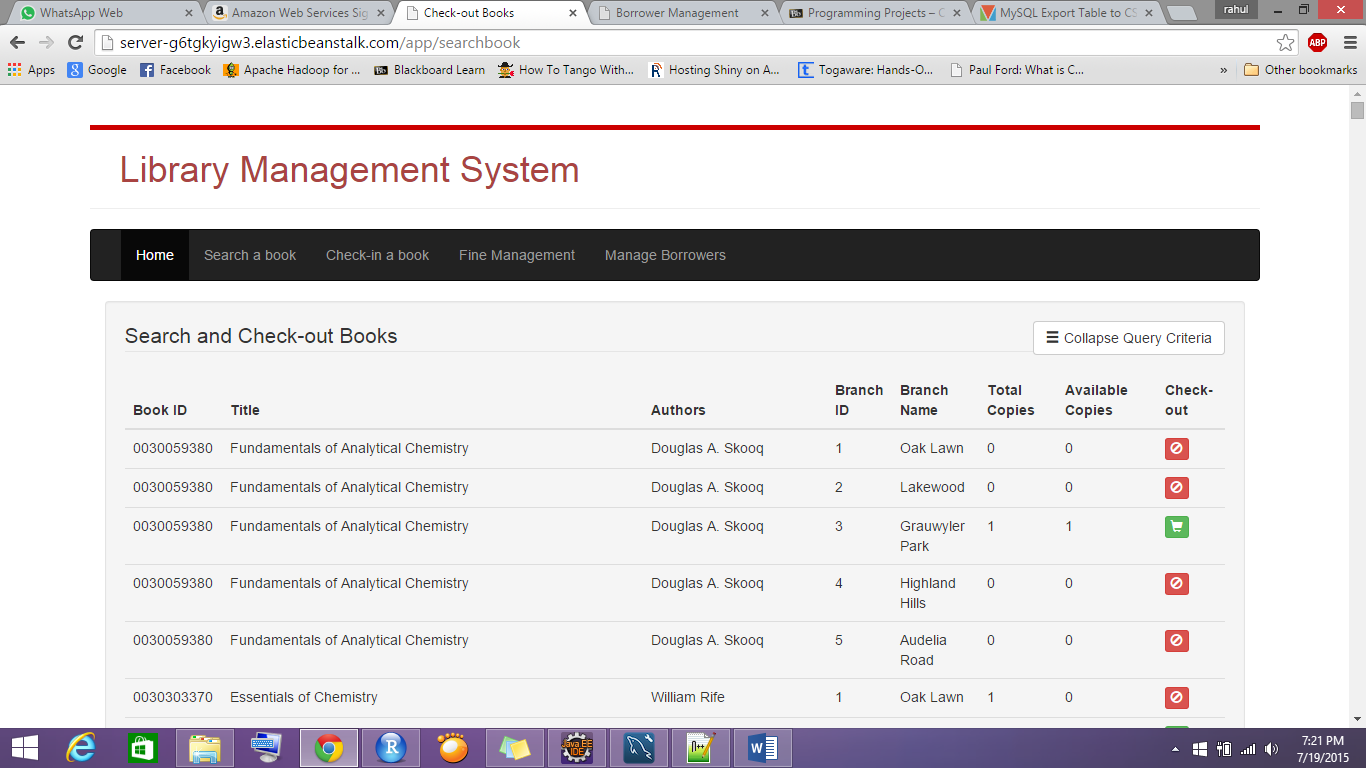


With this menu, User (Librarian) can search book using any combination of three criteria. To search for part of name, Input should be enclosed with wild character (%). For example, if author name is ‘Jared Leto’ then search can be done using any of below,

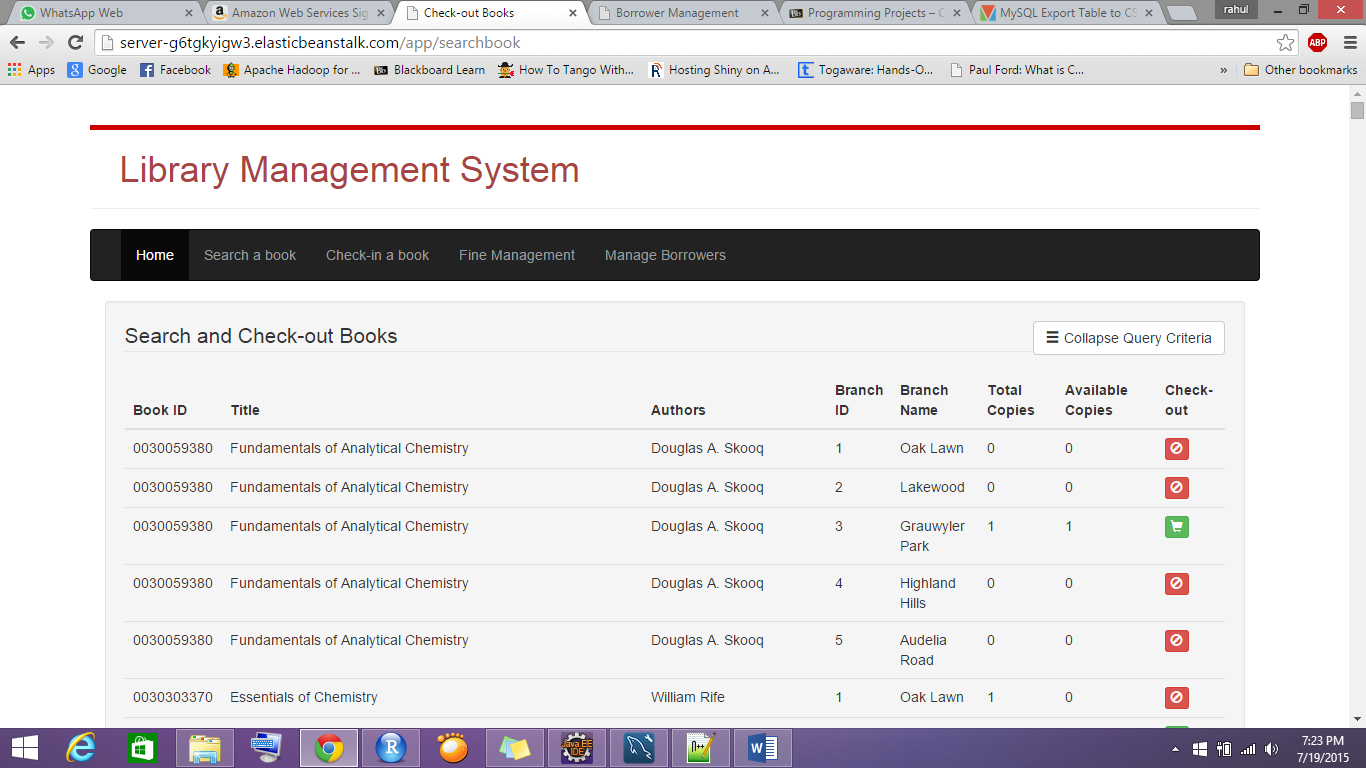
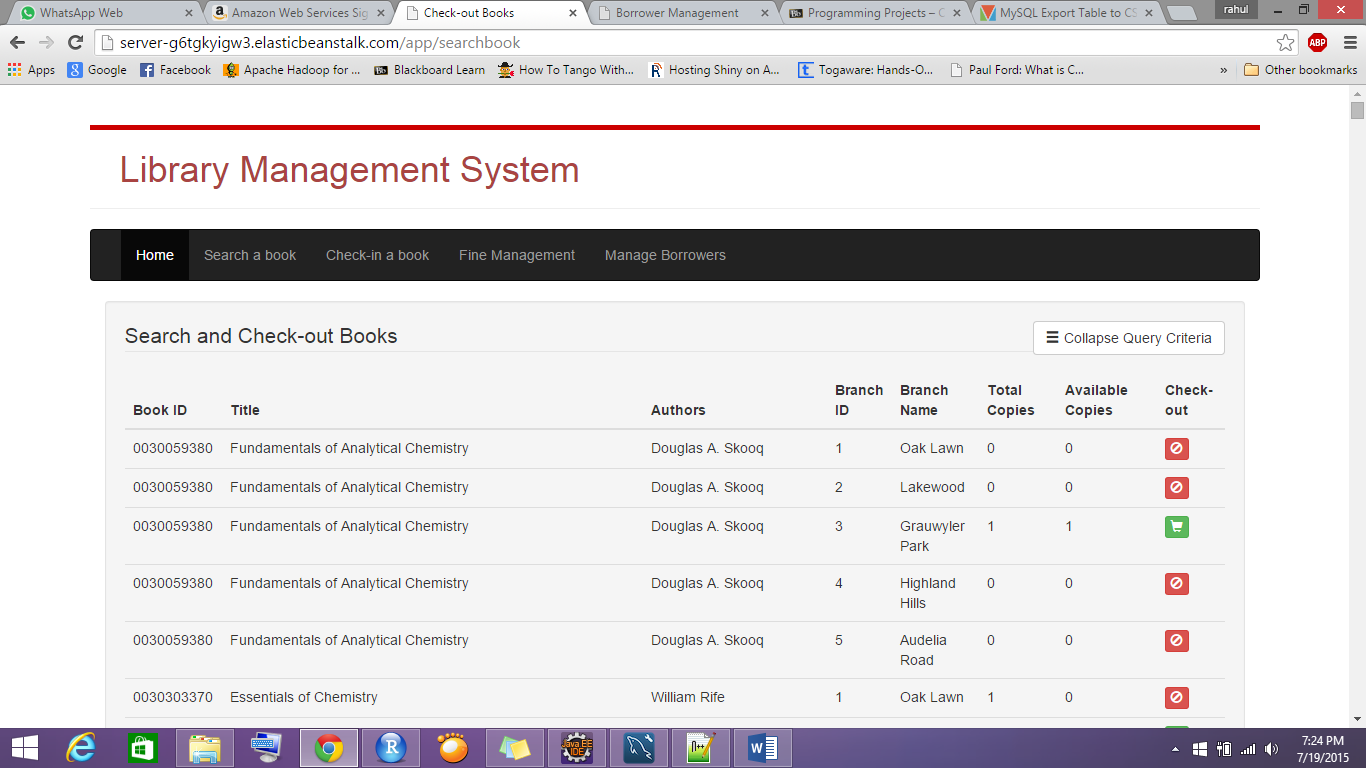
%J% / %Leto / Jared% and so on..

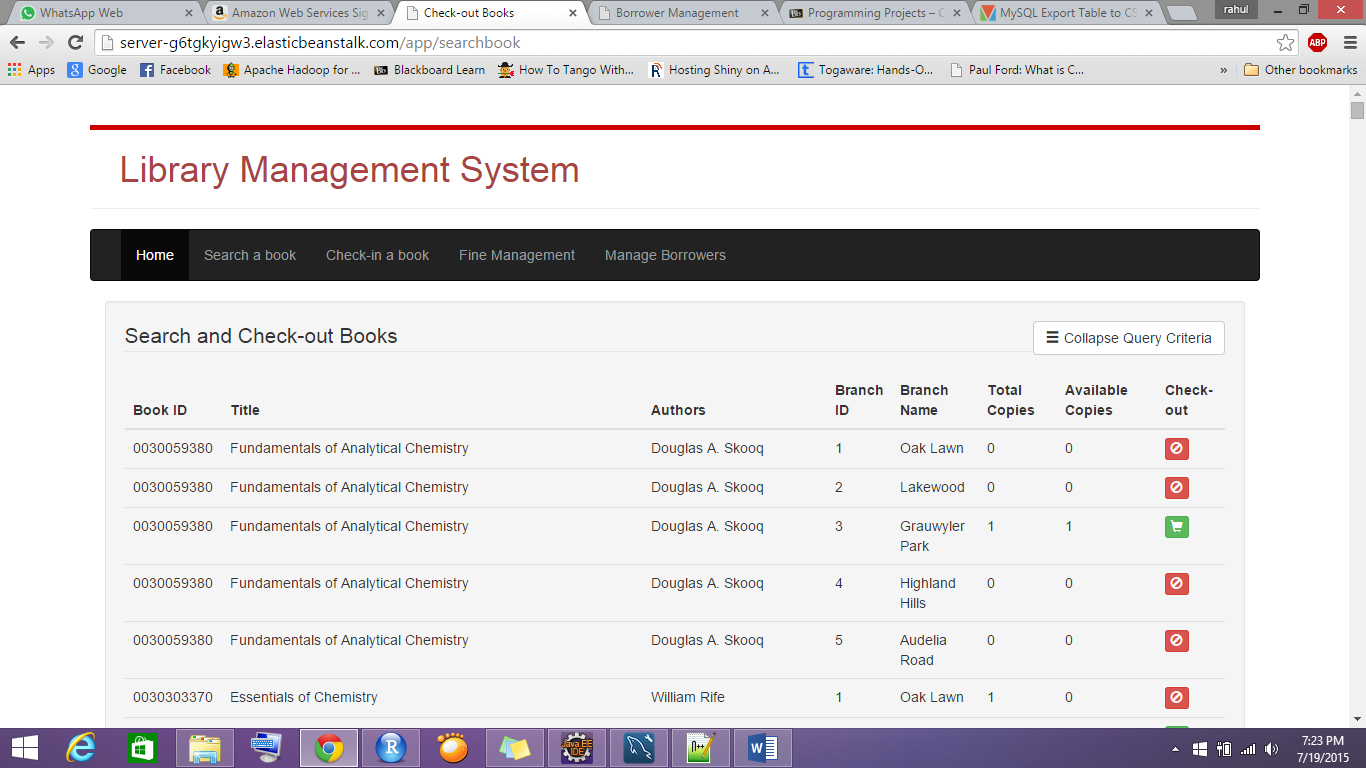
If no criteria is given, it will search for all books and show as list.

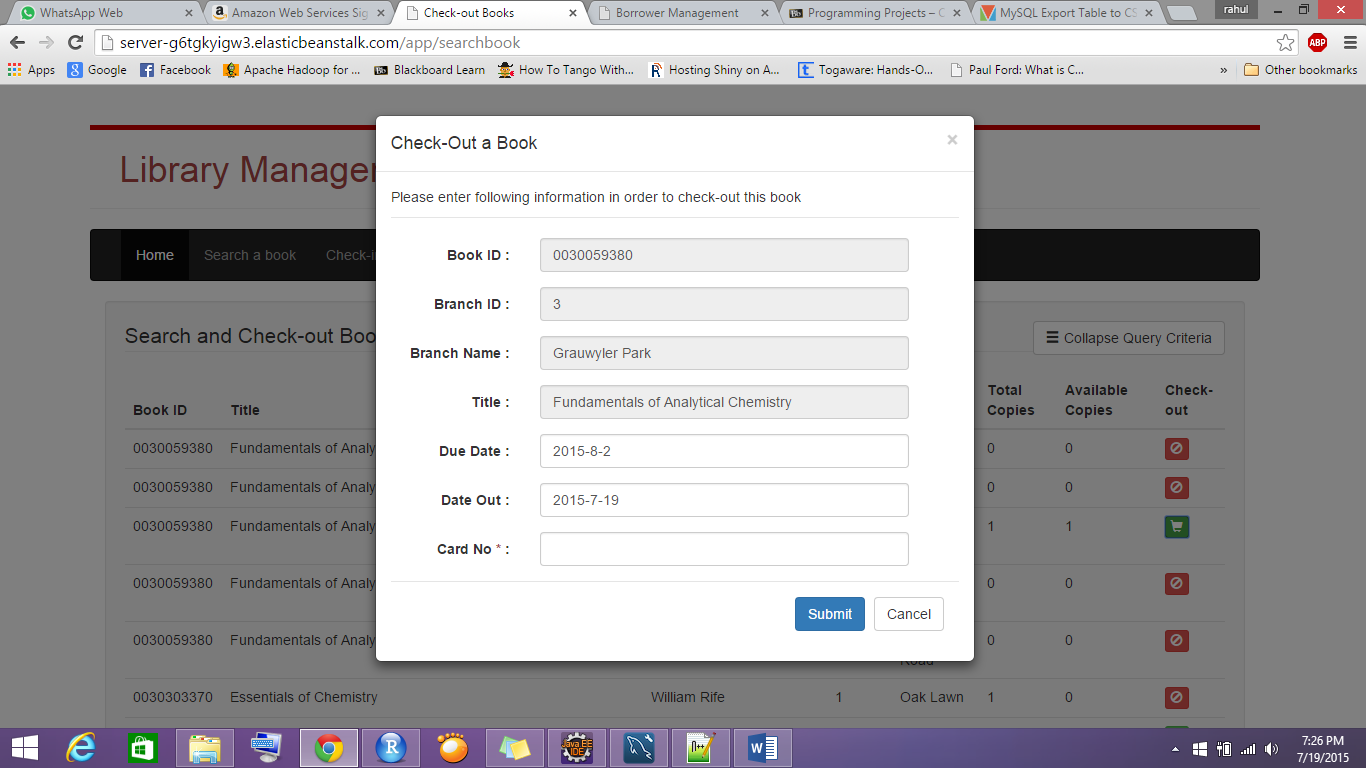
Query criteria can be collapsed to go through search results easily.



* 1. Check-out a book –

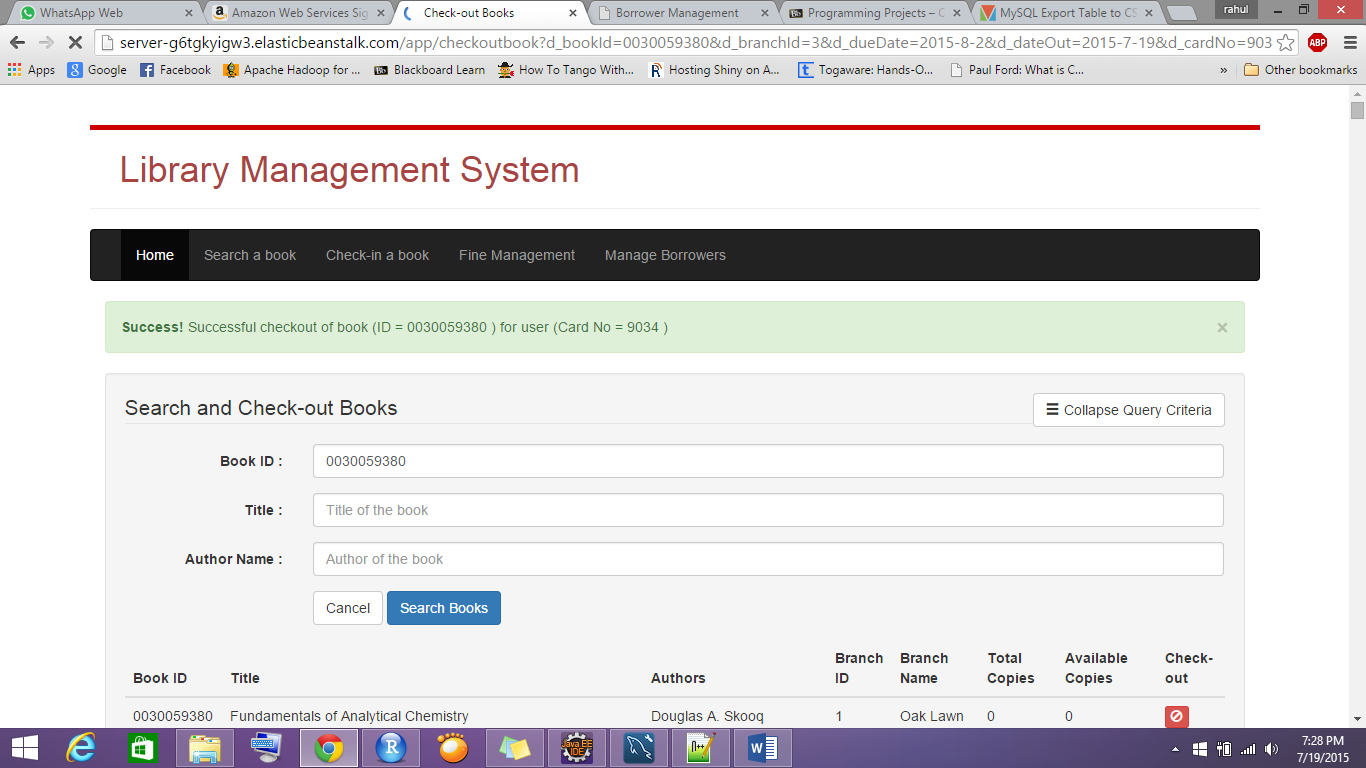
If particular book has available copies more than 0, then that book can be check-out for borrower by clicking on  button. If book is not available then  button shows up which will not allow user to check-out book for a borrower.

After clicking on  button, a pop-up will appear,



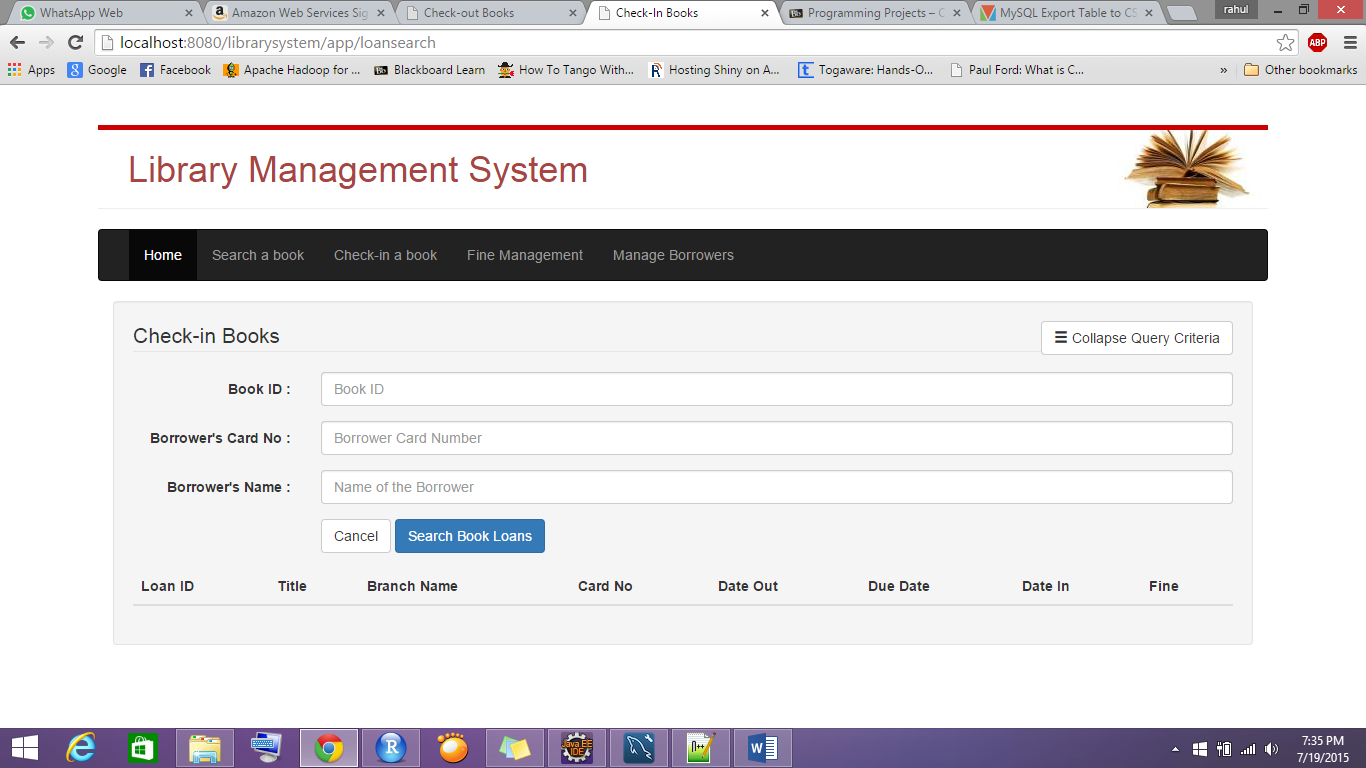
Here, user have to enter a valid card No and click on submit button.

If successful, below message will show up.

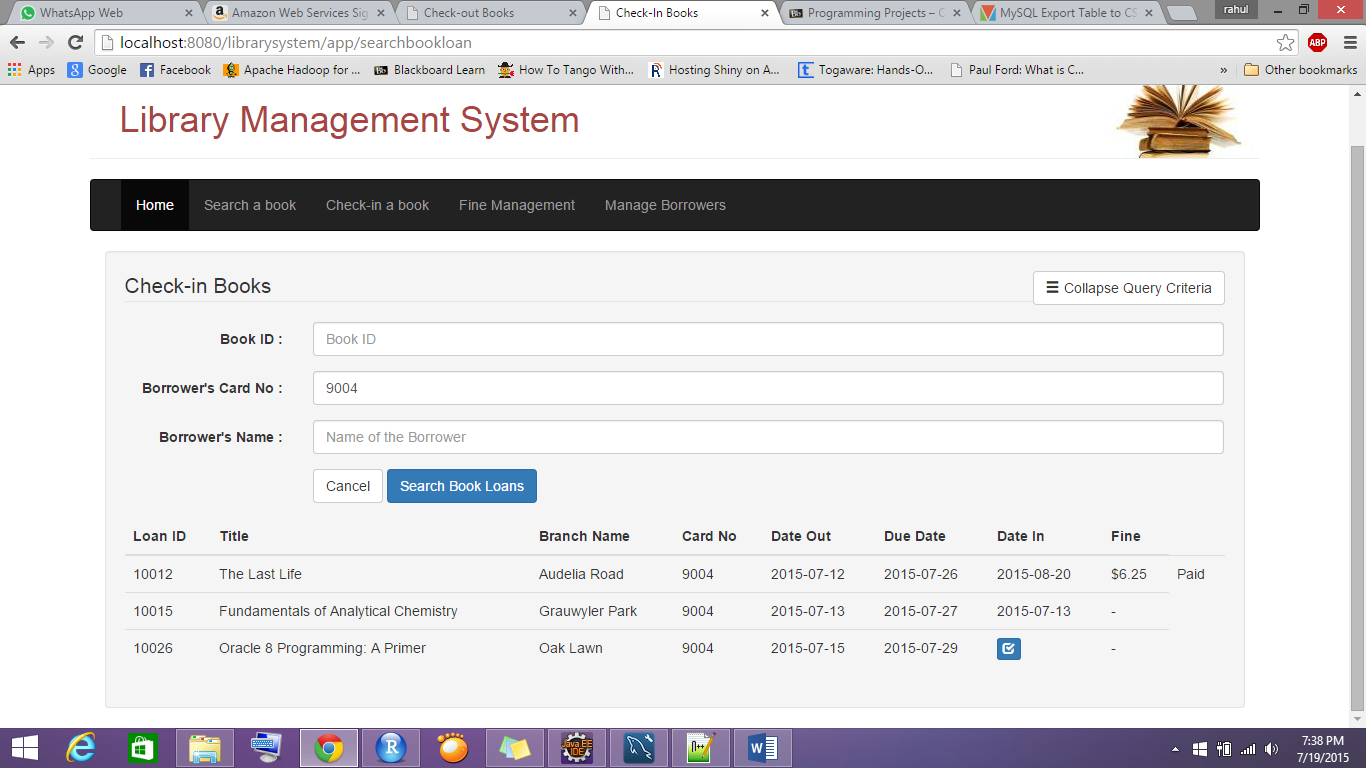


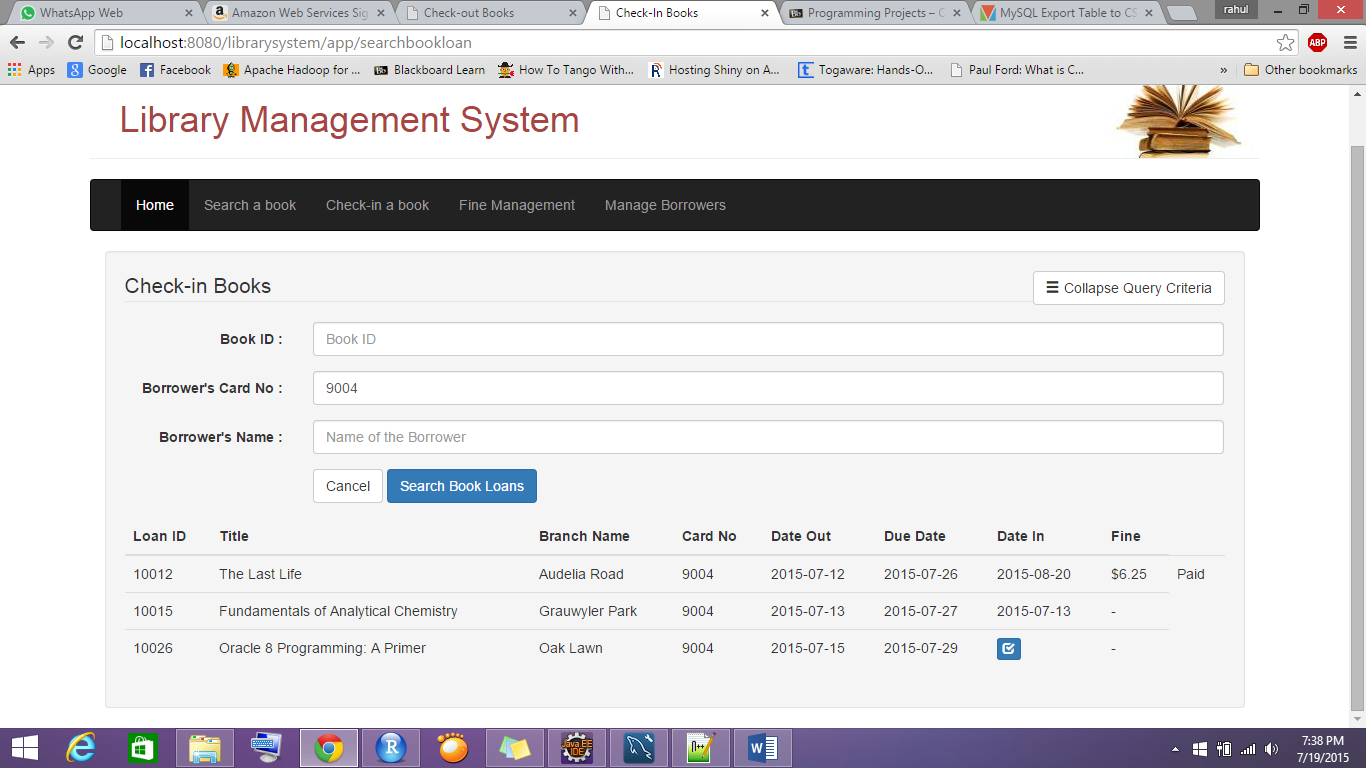
* 1. Check-in a book –

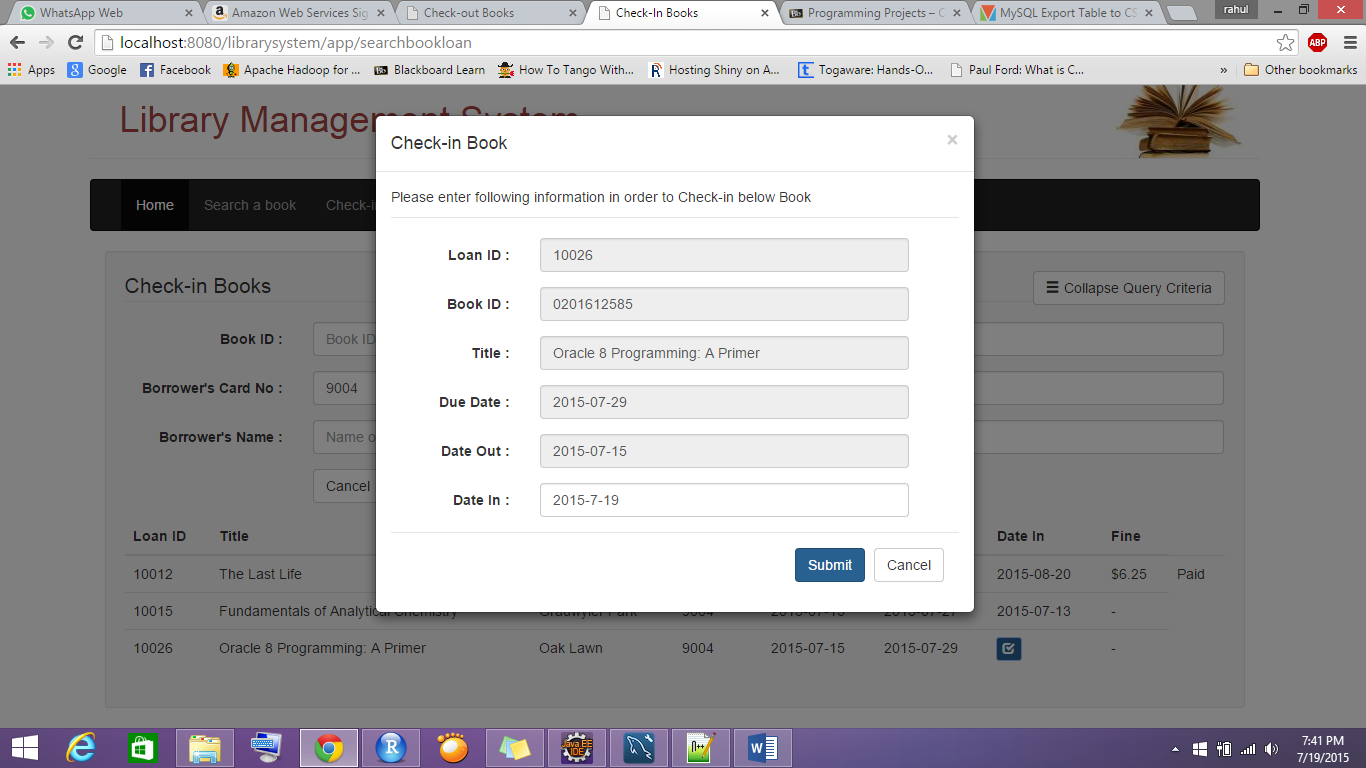
User can search book loans using query criteria as shown below –



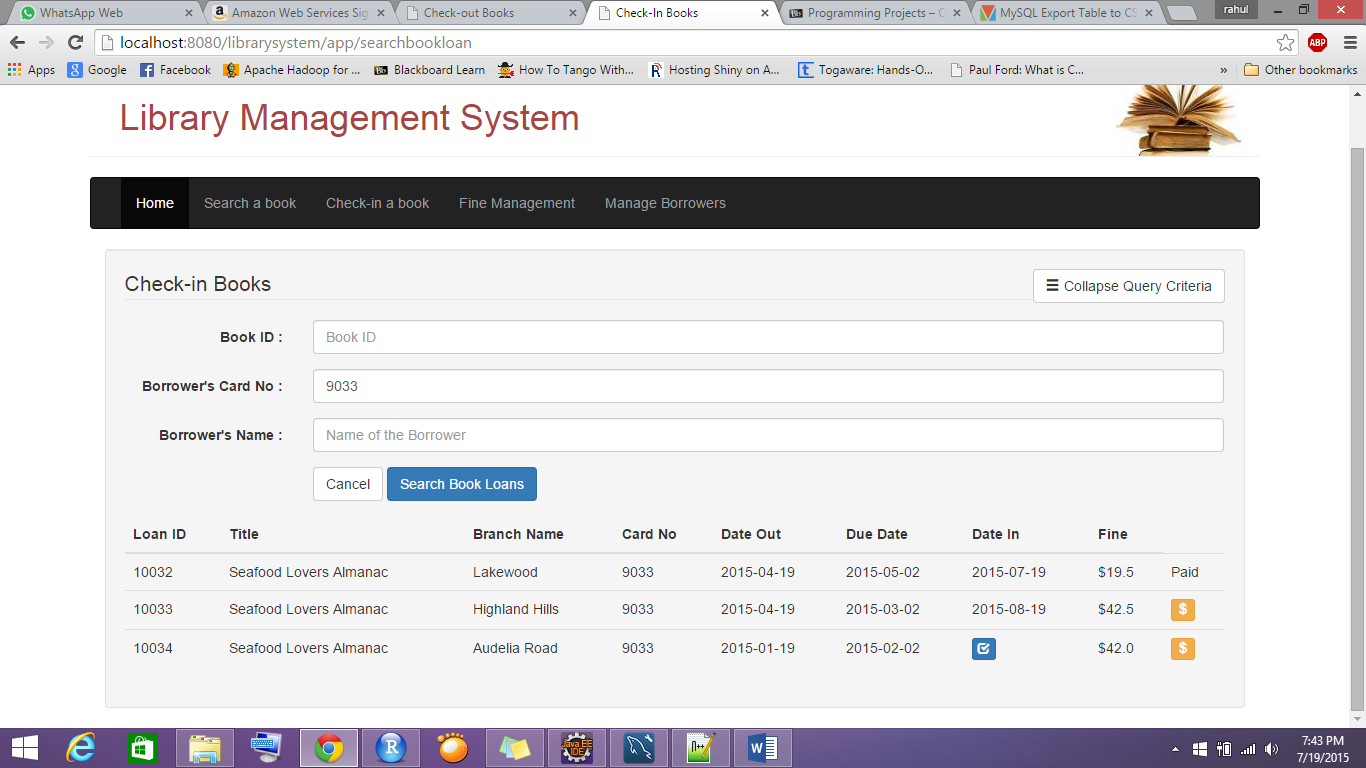
If no criteria given, it will show all book loans.



Then user can check-in using button . After clicking on this button a pop-up will appear which needs a check-in date. After clicking on submit, book check-in into the system.

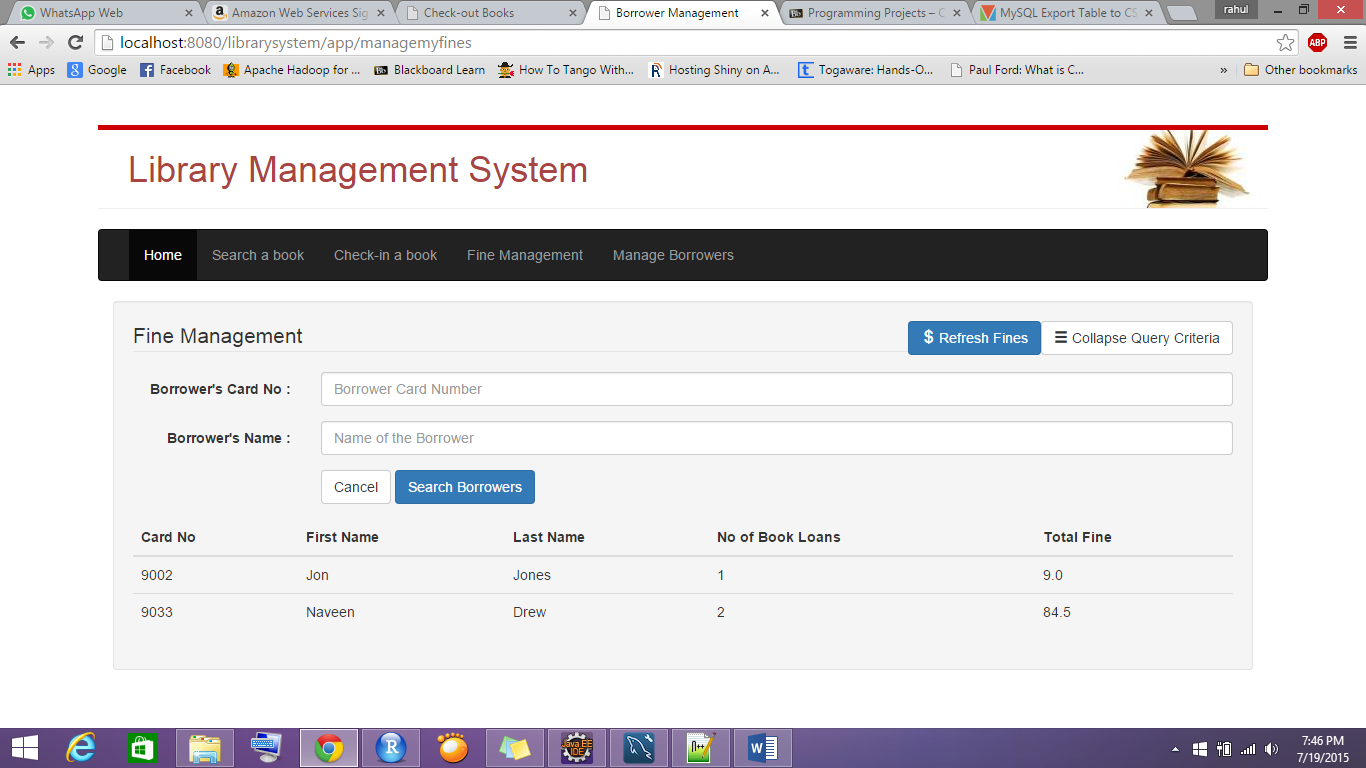


* 1. Payment of late fees –

To confirm the payment for given book loan, user have to click on  button and after clicking on submit button on pop-up, payment gets submitted into system.

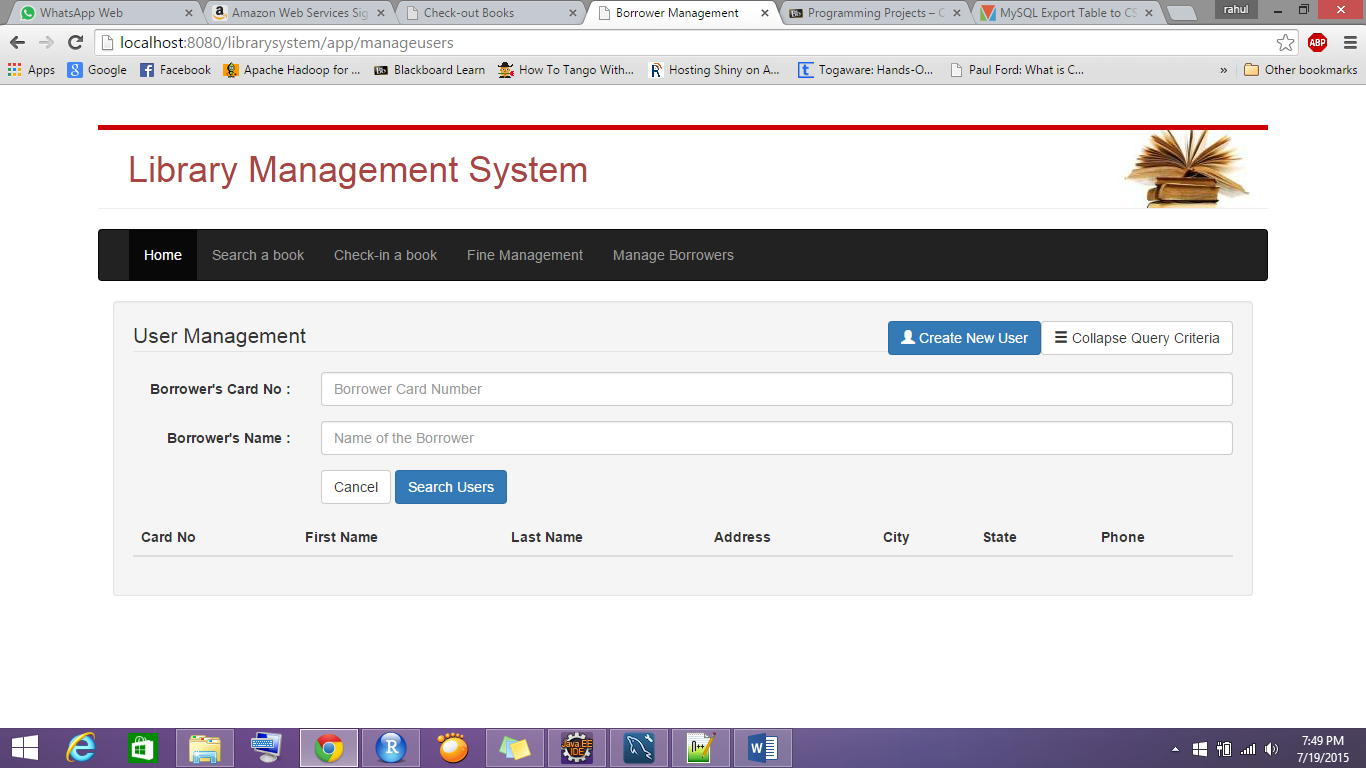
* 1. Fine Management –

All fines can be updated using button ‘$ Refresh Fines’. Using search button, user can search through all unpaid fines in system.

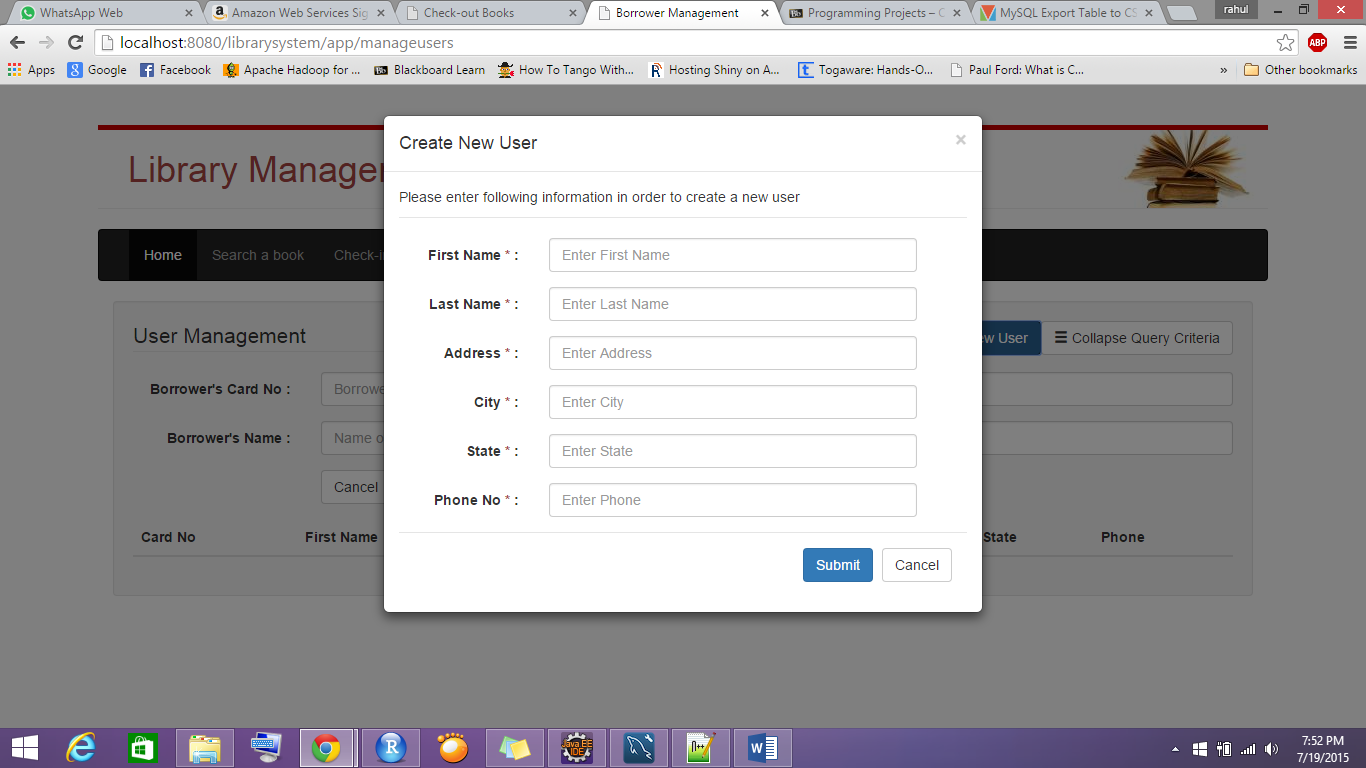


* 1. Borrower management –

User can search through borrowers using Card No or Name.



User can be created using ‘Create New User’ button.



**Design Decisions and Justification –**

1. In my design, I have embedded all the check-in, Check-out and Payment functionality in list page itself. Because it’s self-explanatory and user friendly.
2. I have provided a way to select all rows if no criteria is given, this will provide user to go through all data at once.
3. I have provided an easy way for check-out and check-in where all the data is pre-filled and user just have to enter for only one field.
4. I have placed ‘Refresh Fines’ button on Fine management page to simplify the functionality on Fine management page.

Overall, I tried to design a system which would be very easy to work with.

**Technical Dependencies –**

1. I created this whole web application using Java – Spring framework and JSP. I used Twitter bootstrap framework for UI design.
2. Maven Should be installed in system to run this application

Backend Framework – Spring MVC, JDBC

Frontend Framework – Twitter Bootstrap, JSP

Server – TOMCAT

Database – MySql