

NATIONAL INSTITUTE OF TECHNOLOGY, KARNATAKA



**Project report for the course: CS814
Odd semester: Mtech 2020-2021**

LIBRARY MANAGEMENT SYSTEM (LMS)

Submitted by:

Saurabhkumar Desale 202CS029
Gita Dwivedi 202CS009

**Department of computer science and engineering
National Institute of Technology Karnataka
P.O Srinivasnagar, surathkal, Mangalore-575025
Karnataka, India
January 2021**

Table of contents

1. Abstract.....	3
2. Introduction.....	4
2.1 Architecture of application.....	5-7
2.2 Authorization.....	7-9
3. Conclusion.....	10
4. References.....	11

1 Abstract

Library management system is an application for assisting librarians and students for accessing the library of an institute. This application is developed to provide management of different facilities provided by the library. It has a feature of a user account for librarians, admins and students. In addition to this it has basic features of add, remove, issue, and return of books.

Development of this project includes back-end database management and front-end application development. Various permissions to different users such as librarian, admin and student will be given according to the allowed permissions to that particular user i.e. this project implements role-based access control(RBAC) policy to control the access of the system based on the user's role.

2 Introduction

Online Library Management System(LMS) is a system which maintains the information about the books present in the library, their authors, the members of the library to whom books are issued etc. This is very difficult to organize manually. Maintenance of all this information manually is a very complex task. Owing to the advancement of technology, organization of an Online Library becomes much simpler. The Online Library Management has been designed to computerize and automate the operations performed over the information about the members, book issues and returns and all other operations. This computerization of the library helps in many instances of its maintenance. It reduces the workload of management as most of the manual work done is reduced. This will contain every student related information which book is issued to whom and date on which book is issued and return date. Using this library management system software this will easy to maintain and update the information within seconds.

LMS has various users and all users have different permissions based on their roles. This is implemented by using RBAC policy.

Architecture of application

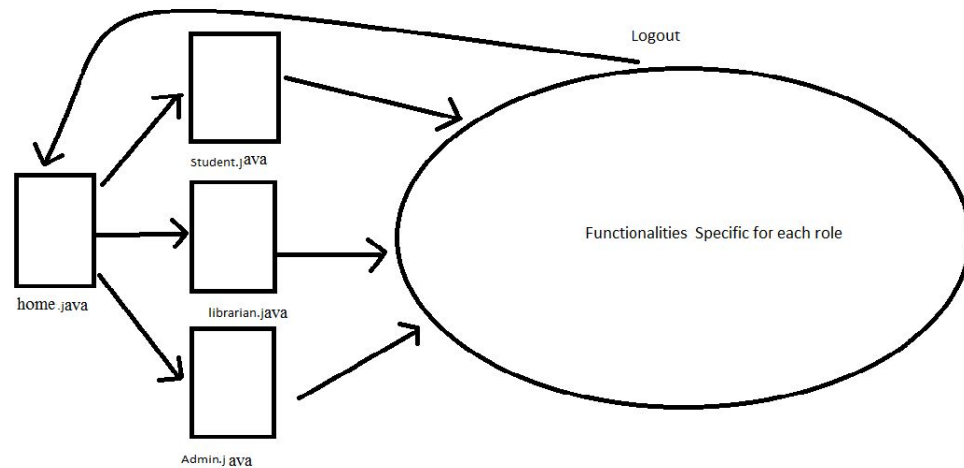


Figure: Architecture of LMS

Home.java is the home page in the LMS, it has all the functionalities for authentication of the LMS. student.java, librarian.java and admin.java will have functionalities pertaining to login into LMS. LMS provides different functionalities for different users according to their roles.

Student:

Students can search, check availability, issue and return books.

Functionalities provided here are :

search: student can search books, it has also functionality of checking availability of book

issue: student can issue book based on availability

return: student can return book

change password: for changing password

logout: it is used to logout from the system

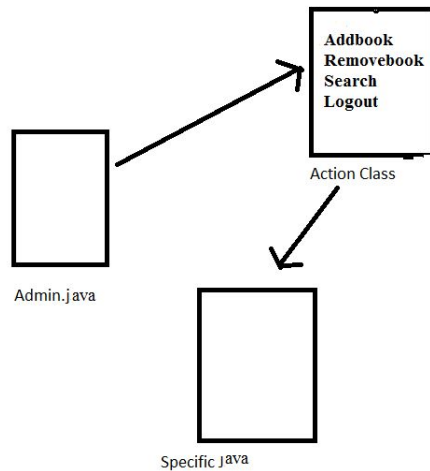


Figure:student.java

Librarian:

Librarians can add, remove, change password and search books.

Functionalities provided here are :

addBook: books and copies of books of same title can be added

removeBook: books and copies of books can be removed

search: for searching of a book

change password: for changing password

logout: it is used to logout from the system

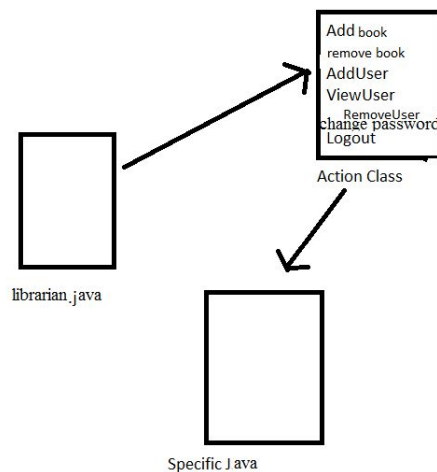


Figure:Librarian.java

Admin:

Admin is responsible for adding or removing accounts of students and librarians. Only Admin can give credentials to new users and also role assignments.

Functionalities provided here are :

addBook: This is used to add new book into the database

removeBook : It is used to remove book which is present in a database

addUser : It is used to add a new User into the system

viewUser : It is used to view all the users information which are present into the system

removeUser: It is used to remove existing user from the system

change password: for changing password

Logout: It is used to logout from the system and invalidate sessions

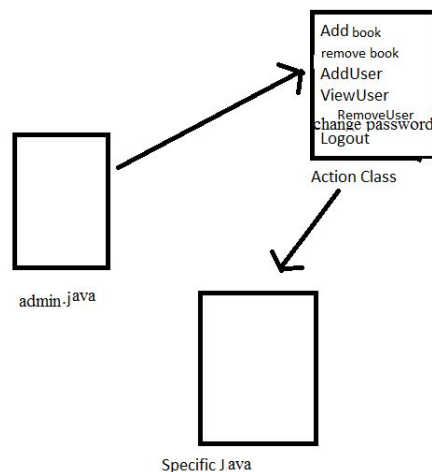


Figure: admin.java

2.2 Authorization:

2.2.1 Need of RBAC based Authorization:

A Role-Based Access Control is a method of restricting access within an organization based upon the role of the respective user. A user's role within the organization decides what kind of permissions he/she possesses. In role-based access control model roles are based upon several factors, including job position, responsibility and job competency. As such organizations can designate whether the user is an end-user, the administrator, or the special user, etc. Limiting the access to the system is important as many users might not be permanent or permitting access to the third parties, like customers, and vendors, makes it difficult for managing access effectively. If organizations use the RBAC model, they are better able to secure the sensitive information,

data, and critical applications. Organizations can control access to the data, information depending upon the role of the user. So only privileged users can access the sensitive information.

2.2.2 Benefits of using RBAC:

1. **Improving operational efficiency:** With RBAC, companies can decrease the need for paperwork and password changes when they hire new employees or switch the roles of existing employees. RBAC lets organizations quickly add and change roles, as well as implement them across platforms, operating systems (OSes) and applications.
2. **Giving administrators increased visibility:** RBAC gives network administrators and managers more visibility and oversight into the business, while also guaranteeing that authorized users and guests on the system are only given access to what they need to do their jobs.
3. **Reducing costs:** By not allowing user access to certain processes and applications, companies may conserve or more cost-effectively use resources, such as network bandwidth, memory and storage.

2.2.3 Components of RBAC used in LMS:

In LMS, we have used password based authentication (log-in) to access the system. We have three types of users namely Students, Librarian and Administrator which generates three roles in our RBAC based application as mentioned above. Each user, after successful login, has his separate session. The user student has access only to functionalities like Search book, issue and return book etc. User librarian has access to functionalities like add book, remove book and search book etc. The administrator has access to functionalities like Add User, View User, Remove user, Add book, Remove book etc. Clearly we separated permissions and functionalities based upon the role of the user.

In LMS, common functionalities to all the roles are Change password and Search. To implement this we have used a flag which is set by a particular user according to his/her role. Similarly, functionalities such as Add book and Remove book are shared by Librarian and Administrator.

By sharing some components of the modules reduce cost of the application as well as resources required for those extra modules.

Components of Administrative Model: The administrator can add/remove a new book in LMS, new user in LMS, view existing users and remove users from LMS.

Implementation of Authorization:

We have used log-in based authorization. Each user will log-in to the system using email id and password. An email is assigned to the role of user to it and an email id is unique. Depending upon the role, after successful login the user will be granted appropriate privileges according to its role.

3. CONCLUSION

This LMS provides a computerized version of the library management system which will benefit the students as well as the staff of the library.

RBAC policy is useful when we have a very large number of users associated with different roles. In the end, we can say that we have successfully demonstrated the RBAC policy in our library management system. We have identified users by their respective roles, separated their functionalities, permissions according to their role. The RBAC policy implementation in our application helped us to separate student's and librarian's policies, permissions, etc from that of highly privileged administrator policies, permissions. It improved the operational efficiency from an administrator's perspective. But also reduced the operational cost of the organization otherwise three different systems would have to be maintained and synchronization between these three of them would be a difficult task.

4. REFERENCES

- R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, "Role-based access control models," in Computer, vol. 29, no. 2, pp. 38-47, Feb. 1996, doi: 10.1109/2.485845.
- <https://dev.mysql.com/doc/refman/en/>
- <https://www.tutorialspoint.com/java/index.html>
- https://www.w3schools.com/html/html_scripts.asp