CU LMS Application

Report

Submitted By

Rubel Patwary
171442607, 44th Batch
Department of CSE
City University
Dhaka-1215, Bangladesh

Submitted To

Richard Philip
Senior Lecturer
Department of CSE
City University
Dhaka-1215, Bangladesh

Index

Chapte-1: Introduction 2
1.1 Problem Statement 2
1.2 Proposed Solution 2
1.3 Project Structure 2
Chapter-2: Related Projects 4
2.1 Project-1: Evergreen 4
2.2 Project-2: LibLime Koha 5
Chapter-3: Analysis 6
3.1 Requirement Gather 6
3.1.1 Document Reading 6
3.1.2 Project Requirement Observations 6
3.1.3 Interview 6
3.2 Feasibility Study 7
3.2.1 Economic Feasibility 7
3.2.2 Technical Feasibility 7
3.2.3 Organizational Feasibility 7
3.3 Functional and Non-Function Requirements 8
3.3.1 Functional Requirements 8
3.3.2 Non-Functional Requirements 8
Chapter-4: Diagrams 9
4.1 Use Case Diagram 9
4.2 Activity Diagram 13
4.3 Sequence Diagram 14
4.4 Class Diagram 15
Chapter-5: Evaluation 16
Chapter-6: Conclusion 16

Chapte-1: Introduction

1.1 Problem Statement

A Library Management System is built to reduce the cost and human resource to maintain a Library and to increase user experience through fast and efficient process. A Library Management System helps to keep the records of whole transactions of the books available in the library. Library Management System makes everyday library tasks more efficient. This means more work can be done in less time. Consequently, this decreases operational costs. This also minimizes paperwork and manual tasks, thus allowing library personnel to concentrate on other things such as interaction with users. The major benefit of Library Management System is, tracking facilities. A library management system is able to track the amount of books in the library, amount of books borrowed by a user, availability of books so easily.

1.2 Proposed Solution

Software systems are built to make human tasks easier, efficient securely. So our project is mainly focused on the User Friendly Interaction and Security of the system. As this project is mainly focused on User Friendly Flavor so we tried to keep the User Interface as simple as possible. So any user can access and interact with the system more efficiently than any other system. Our application designed as module based for future development. Which makes our application more loosely coupled than any other systems. This project is created in JAVA, which makes the application platform independent. In 1.3 we have described about the project structure in more detailed.

1.3 Project Structure

Our Project is based on JAVA Backend. We have Used Spring Framework for this project. If some small libraries use traditional library management system, they may waste resources into the system. According to the present situation of multi-hierarchical architecture development of information system, we had analyzed thoroughly the spring framework. It integrated the frameworks to design a set of sufficient flexible, loose coupling, expandable Library Management System.

Through the Spring Framework We have separated the application into several layer. Below a diagram for the Project Structure is given:

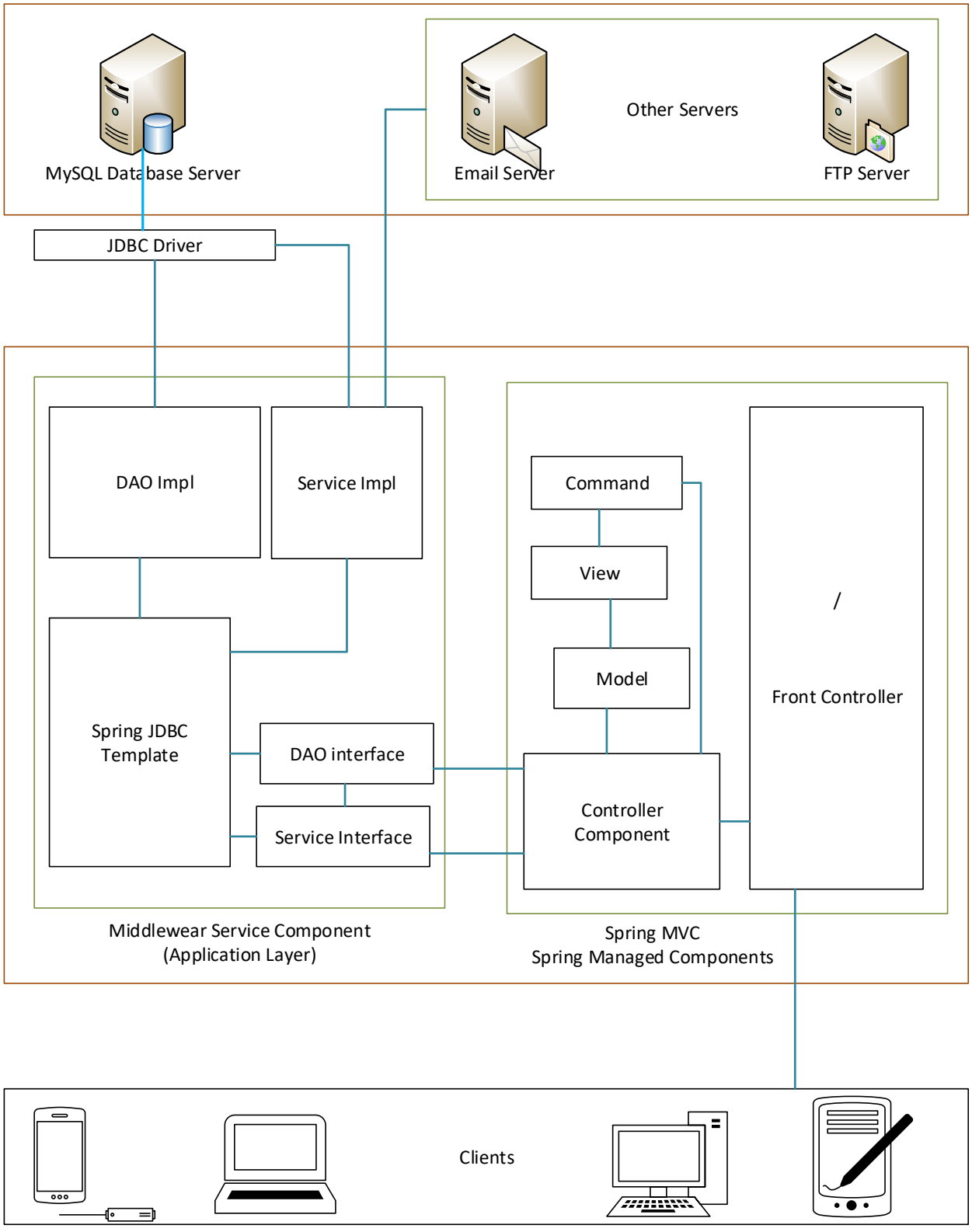

Fig 1.3.a: Project Structure Overview.

Chapter-2: Related Projects

2.1 Project-1: Evergreen

The Evergreen Project develops an open source ILS (integrated library system) used by more than 2,000 libraries around the world. Evergreen's features include:

- Circulation: for staff to check items in and out to patrons
- Cataloging: to add items to the library's collection and input information, classifying and indexing those items. Evergreen is known for an extremely flexible indexing system that allows for a high level of customization and by default uses Library of Congress MODS as its standard.
- Online public access catalog (OPAC): a public catalog, or discovery interface, for patrons to find and request books, view their account information, and save book information in Evergreen "bookbags." The OPAC received a makeover in early 2009 with the new, optional skin, Craftsman. There is also an optional Children's OPAC. Various patron services such as paying bills by PayPal and Stripe, optional retaining of circulation history, book bags and more.
- Self Service - Evergreen comes with self-checkout and registration options that can be activated by the libraries.
- OPAC exposes structured web data by schema.org standards to aid discovery by major search engines.
- Acquisitions: for staff to keep track of those materials purchased; invoices, purchase orders, selection lists, etc.
- Authorities
- Serials
- Web based staff client that is OS independent
- Added Content services Chillifresh, Content Cafe, Novelist, OpenLibrary and Syndetics natively supported with others supportable.
- Native APIs for writing custom clients.
- Statistical Reporting: flexible, powerful reporting for retrieval of any statistical information stored in the database.
- SIP 2.0 support: for interaction with computer management software, self-check machines, and other applications.
- Search/Retrieve via URL and Z39.50 servers

Evergreen also features the Open Scalable Request Framework (OpenSRF, pronounced 'open surf'), a stateful, decentralized service architecture that allows developers to create applications for Evergreen with a minimum of knowledge of its structure.

2.2 Project-2: LibLime Koha

The development of LibLime Koha is steered by a growing number of libraries throughout the world. These libraries, either on their own, or collaborating in groups, sponsor the development of new features to support their workflows. LibLime Koha's impressive feature set continues to evolve and expand to meet the needs of its sponsoring libraries.

Easy-to-use circulation policies, strong patron management, intuitive navigation, and extensive permissions for staff accounts.

Following are some Features that LibLime Koha Includes:

- Parent-child relationship for patron records, as well as a 'copy' patron feature to quickly add families.
- A Clubs and Services feature that allows libraries to manage reading groups, book clubs, and other community outreach programs. This feature is easily managed by library staff.
- Extensive support for holds, including an option to 'suspend' and 'reactivate' a hold, an option to place holds from a patron's OPAC account, an option to allow staff to re-organize the holds queue, and an option to place holds at either the title or item level.
- Enhanced matching policy rules for the 001 and 035 tags, allowing libraries to update older records with a newer version.
- Libraries to 'undo' entire import batches from the catalog in a single click, rather than having to delete on a one-by-one basis.
- SIP2 configuration for a wide variety of vendors and their products, including ITG, 3M, EnvisionWare, Talking Tech, Overdrive, TechLogic, and Librarica. LibLime Koha also works with EzProxy as a dual authentication source for remote database access.
- OPAC, staff, administrative features and self-checkout interfaces are all based on standards-compliant World Wide Web technologies--XHTML, CSS and Javascript--making LibLime Koha a completely Web-based solution.

Chapter-3: Analysis

3.1 Requirement Gather

3.1.1 Document Reading

We have read several books including online resources to get the exact processes to design the entire application. In past years desktop application was the only dominant in the market. But recent years the web based multi user login application is getting high response from the users. So we had decided to build the application web base so anyone can access the application from anywhere just only using the internet and any browser. As most of the web based application is publicly available we came to learn from reading various online blogs that providing security into the system is one of the critical part of the entire development cycle. We had read various books based on system analysis and design in order to design a legacy application and here we have created one the “CU LMS” Application.

3.1.2 Project Requirement Observations

Requirement Observations are given below:

1. Multi User Application
2. Role Base User Access
3. Export Book Information in CSV
4. Import Book Information from CSV
5. SSL Based Communication
6. Store secure information as encrypted
7. Student/Teacher Account Activation/De-Activation
8. CRUD Operations over books information’s
9. CRUD Operations over Users information’s
10. Bulk Delete Operation for Books.
11. Bulk Delete Operation for Users.

3.1.3 Interview

Before starting the project our team visited several Libraries including the Library at Dhaka University. We had talked to several Librarian about the application context. Some of the Librarian was already using several LMS to maintain their Libraries. So they have suggested us several changes and up gradation of our own application. Below these major points are given which shall be integrated in future Release.

a) Access / Restrict User by Dynamic Role Assignment. This feature allows an Admin to customize specific role by permitting or restricting accessible areas.
b) User/Book Information Reporting.
c) Bulk Delete Operation into the system. An Admin is allowed to delete multiple user/books at the same time.
d) Export Book/User Data as CSV format or in Excel formatted file.
e) Import Book/User Data from CSV format or in Excel formatted file.

3.2 Feasibility Study

Feasibility analysis guides the organization in determining whether or not to proceed with a project. Feasibility analysis also identifies the important risks associated with the project that must be addressed if the project is approved. As with the system request, each organization has its own process and format for the feasibility analysis, but most include three techniques: technical feasibility, economic feasibility, and organizational feasibility. Below all three feasibility is described for the proposed LMS System.

3.2.1 Economic Feasibility

In this type of feasibility, the cost of hardware, software and overall budget is evaluated to run the new system. Tangible and intangible benefits are also considered in the evaluation. Following some Economic Feasibility are given based on the CU LMS Project.

a) Development Cost: As the entire system is built by a group of students and declared as open source project. So there is no development cost into the entire project Development.
b) Annual Operating Cost: As the application is open source so there is no Annual Cost based on the application context. But as the application uses Hardware, Bandwidth and Other Resources, so these might cost a little annually. These costs are listed below:

Sl	Title	Cost
1.	Development costs	0.0
2.	Hardware Cost (Approx.)	10,000
3.	Bandwidth Cost (Approx.)	12,000
4.	Electricity Cost (Approx.)	5,000
5.	Other	3,000
	Total (Approx.)	30,000

3.2.2 Technical Feasibility

In this type of feasibility, the present hardware and software compatibility with the new one is checked out to run the new system. Below some points of Technical Feasibility are given based on this project.

a) Familiarity with Functional area: As we had gathered a lot knowledge and spent most of our times in analysis part so there is low risk of reaching the goal of this project.
b) Familiarity with Technology: Our group members are specialized in deferent technologies. So we have separated our task according to their specialization.
c) Project Size: This project is not big. So there is less complexity to build the entire project.

3.2.3 Organizational Feasibility

In this type of feasibility, the issues like, operational scope for the fast acceptability of the alternative solution, human issues, social issues, internal issues (organizational conflicts) and legal issues are to be checked out.

a) Senior management: The senior management will be able to track every transaction of books and other operations so easily by this application.
b) Users: This application targets only 3 types of users, Admin/Librarian, Teacher and Student.

3.3 Functional and Non-Function Requirements

3.3.1 Functional Requirements

SL	Requirement	Description
1.	Multi-User Login	This feature enables multiple users to interact with the system simultaneously with valid User Id and Password. No invalid user / guest user should have access into the system.
2.	Role Based	This feature allows to identify a user type. It allows and restricts a user from accessing restricted parts of the system for that particular user.
3.	Register User	Before using the system each user must be registered into the system
4.	Add/Modify Users	This feature shall allow only the admin to Add/Modify a User.
5.	Bulk Delete Users	This feature allows to delete multiple users at a time.
6.	Add/Modify Books	This feature shall allow only the admin to Add/Modify a Books.
7.	Bulk Delete Books	This feature allows to delete multiple Books at a time.
8.	Request Book	Only users (Teacher/Students) are allowed to Request for a Particular Book. The request is only valid if the user still have the eligibility of book request. If the requested book is not available or the user had already borrowed maximum amount of book then, the request will be rejected.
9.	Return Book	User must return the book physically to the admin. The Administrator will change the book status along with user status.
10.	View User/Book List	This feature allows to view list of all users/books.

3.3.2 Non-Functional Requirements

SL	Requirement	Description
1	Efficiency Requirement	The system should run efficiently without any crashing. When the system is implemented all users will easily access the LMS with a faster data transaction.
2	Reliability Requirement	The system should accurately perform member registration, member validation, report generation, book transaction and search.
3	Server Requirement	a) CPU: Quad core 2GHz+ CPU b) Minimum Memory : 4 GB c) Recommended Memory : 8 GB d) Minimum Disk Space: 20 GB e) Recommended Disk Space: 50GB f) Operating System : Windows/Linux (Recommended : Linux)

Chapter-4: Diagrams

4.1 Use Case Diagram

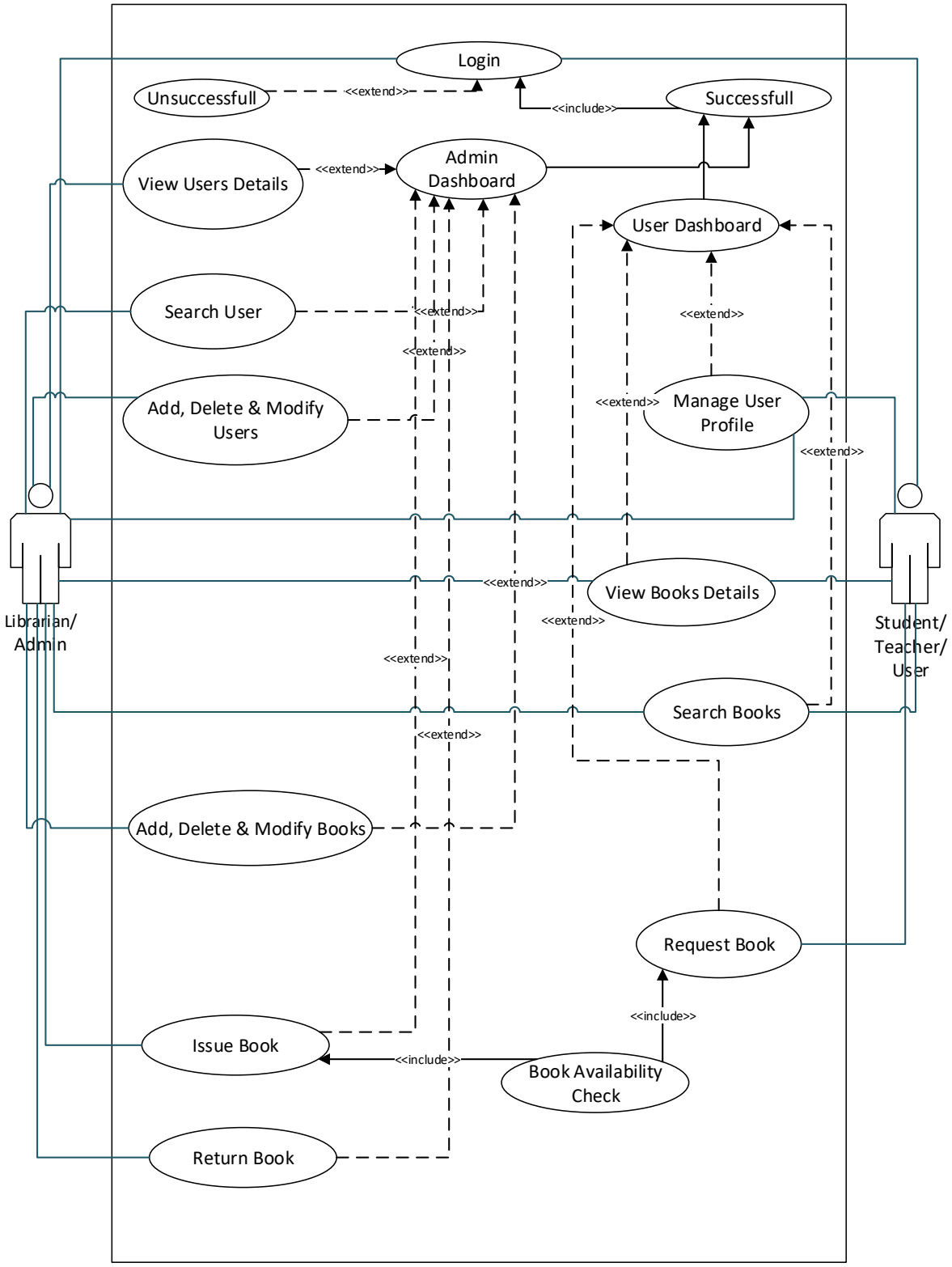

Fig 4.1.a: Library Management System Use Case Diagram.

Use Case Steps Description:

Use Case:	Login
Introduction :	To interact with the system, The application shall validate provided credentials with this system. It also defines the actions a user can perform based on User Role.
Actors:	Librarian/Admin Student/Teacher/User
Pre-Conditions:	User must have a compatible browser to access the system through web.
Post-Conditions:	User should be redirected to the user dashboard page
Basic Flow:	1. Show Login Page 2. Enter User Id and Password 3. User Credentials Validation Successful 4. Redirect to User Dashboard
Alternative Flow:	1. Show Login Page 2. Enter User Id and Password 3. User Credentials Validation Un-successful 4. Redirect to Login Page
Special Requirement:	User Must be a Registered User

Use Case:	View/Search Users
Introduction :	This use case enable a user to view users list or search particular user matching the pattern of the search text.
Actors:	Admin/Librarian
Pre-Conditions:	1. User Must be Logged In 2. User must have Admin Role
Post-Conditions:	1. Show All Users list 2. Show users that have been matched with the provided search texts.
Basic Flow:	1. Show All Users in a list with details of these users.
Alternative Flow:	1. Show specific users whose username is matched with the search pattern.
Special Requirement:	User Must Have Admin Role

Use Case:	Add, Delete & Modify Users
Introduction :	This use case permits a user to add, delete or modify any Users.
Actors:	Admin/Librarian
Pre-Conditions:	1. User Must be Logged In 2. User must have Admin Role
Post-Conditions:	1. Shows a form to add new user if user clicks into the add button. 2. Removes a user if any user is selected and clicked on Delete Button.

	3. Shows a form with existing data of a user and let's to modify the contents.
Basic Flow:	Basic CRUD operations.
Alternative Flow:	If any invalid data is submitted into the add/modify form then the submission will be rejected with an error message.
Special Requirement	User Must Have Admin Role

Use Case:	Add, Delete & Modify Books
Introduction :	This use case permits a user to add, delete or modify any Books.
Actors:	Admin/Librarian
Pre-Conditions:	1. User Must be Logged In 2. User must have Admin Role
Post-Conditions:	1. Shows a form to add new user if user clicks into the add button. 2. Removes a user if any user is selected and clicked on Delete Button. 3. Shows a form with existing data of a user and let's to modify the contents.
Basic Flow:	Basic CRUD operations.
Alternative Flow:	If any invalid data is submitted into the add/modify form then the submission will be rejected with an error message.
Special Requirement	User Must Have Admin Role

Use Case:	Issue Book
Introduction :	By this use case a user is able to issue a book to someone.
Actors:	Admin/Librarian
Pre-Conditions:	1. User Must be Logged In 2. User must have Admin Role 3. User must have requested for a book
Post-Conditions:	1. If Book is available the click on the "Issue Book" Button 2. A success message should be on the top of the page.
Basic Flow:	1. User Request should be accepted if book is available
Alternative Flow:	1. If requested book is not available then an error message will pop up and the request will be deleted.
Special Requirement	

Use Case:	Return Book
Introduction :	This use case stores the data
Actors:	Admin/Librarian
Pre-Conditions:	1. User Must be Logged In 2. User must have Admin Role
Post-Conditions:	1. Book Information must be matched 2. User total borrowed book count will be modified and the status of the book will be changed to available.

Basic Flow:	1. System should update the status of both user and the book.
Alternative Flow:	If book information does not match then an error message will be shown.
Special Requirement	

Use Case:	View Book
Introduction :	This use case shows list of Books along book information's accordingly.
Actors:	Librarian/Admin Student/Teacher/User
Pre-Conditions:	1. User must be logged in
Post-Conditions:	1. A list of book will be shown 2. Show Book that have been matched with the provided search texts.
Basic Flow:	1. Show All Users in a list with details of these users.
Alternative Flow:	Show specific Books whose Book Name or Author Name is matched with the search pattern
Special Requirement	

Use Case:	Request Book
Introduction :	This use case allows a User to request a book from the library to borrow.
Actors:	Student/Teacher/User
Pre-Conditions:	1. User must be logged in 2. User must not have exceeded number of total allowed book 3. Requested Book must be available
Post-Conditions:	Request will be sent to the Admin user to approve the request
Basic Flow:	Book request should be sent to the admin
Alternative Flow:	If user have exceeded maximum allowed book or if the requested book is not available then an error message will pop up.
Special Requirement	

4.2 Activity Diagram

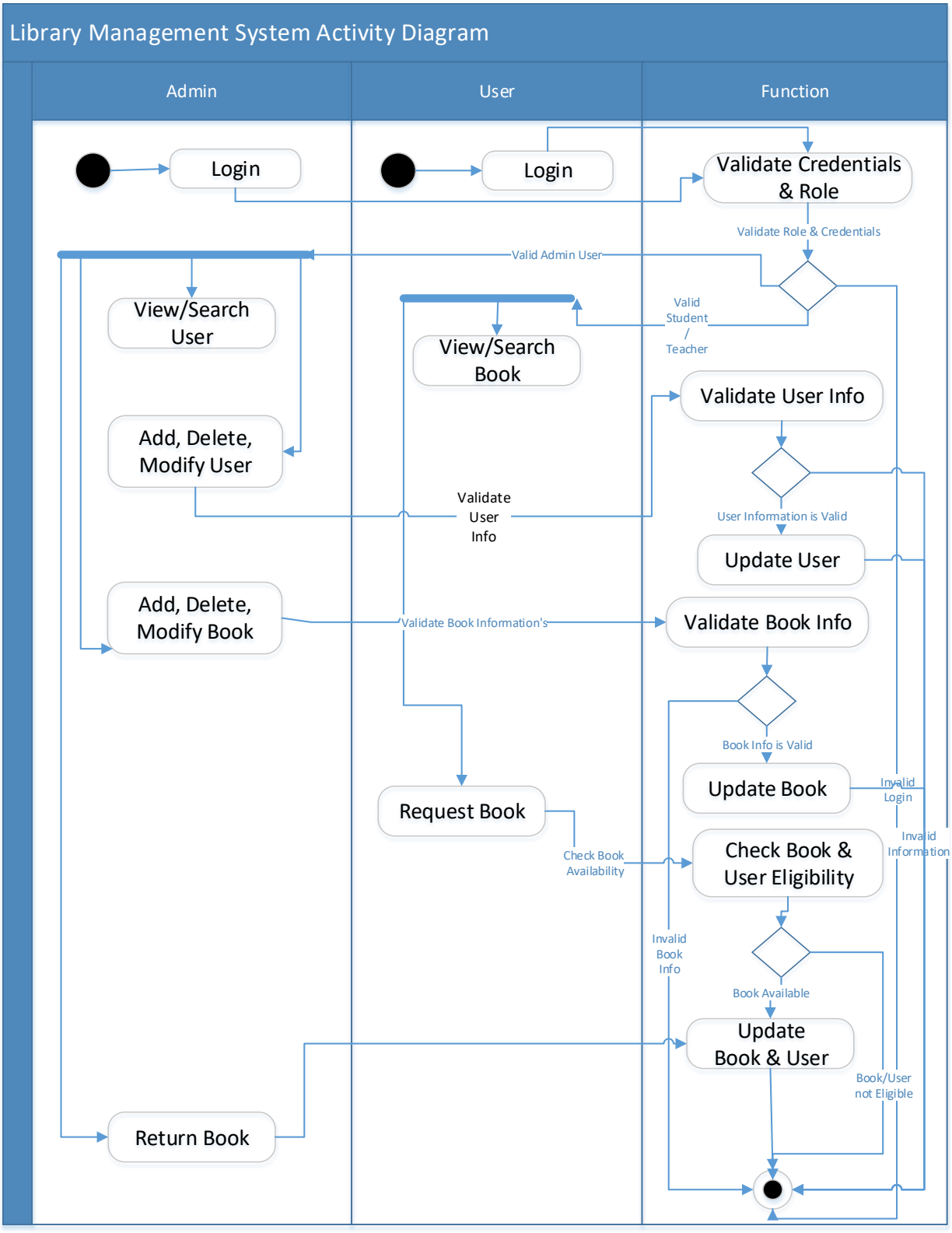

Fig 4.2.a: Library Management System Activity Diagram

4.3 Sequence Diagram

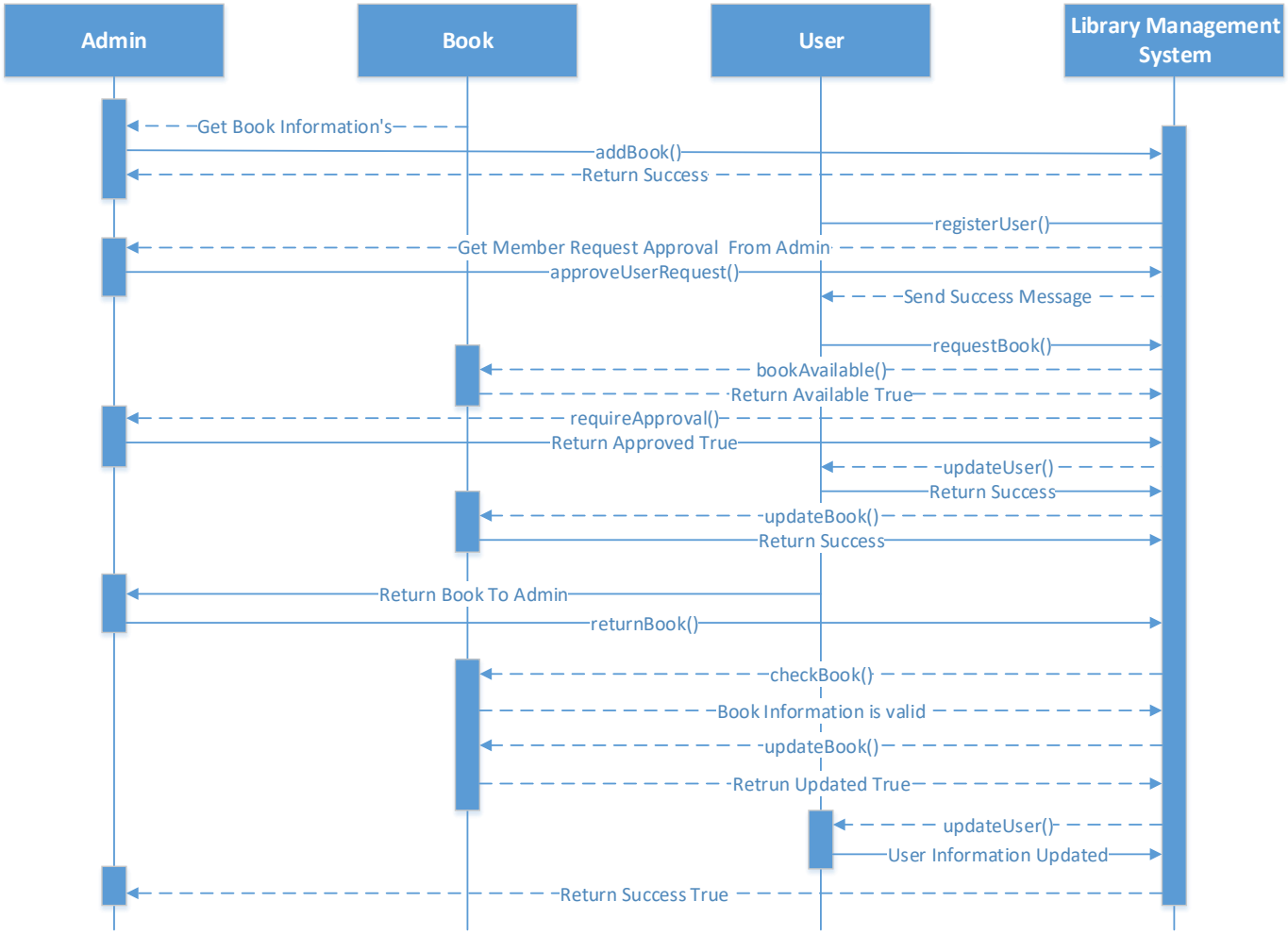

Fig 4.3.a : Library Management System Sequence Diagram

4.4 Class Diagram

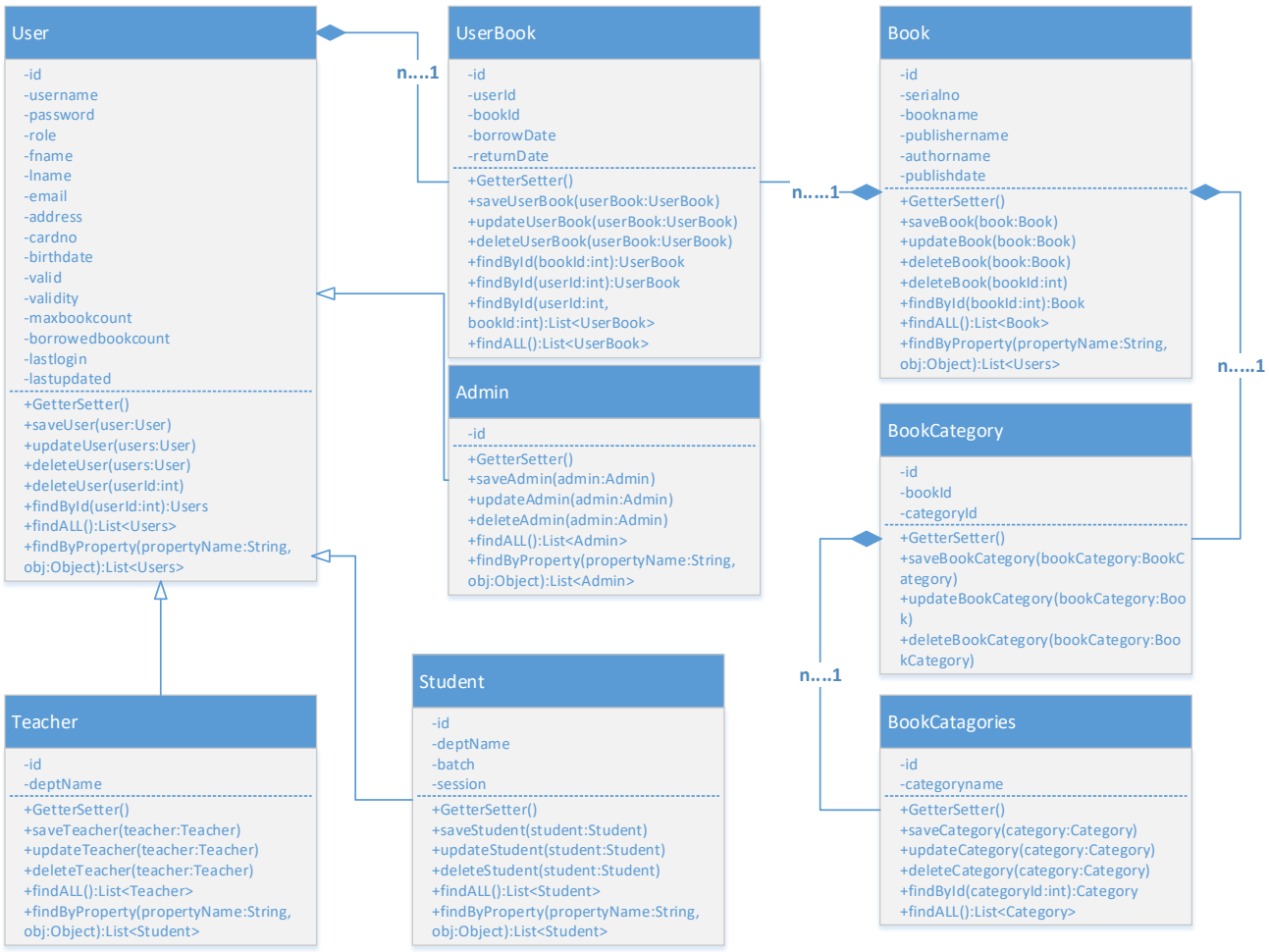

Fig 4.4.a: Library Management Class Diagram

Chapter-5: Evaluation

As this project is released recently, having some good usability also have some limitations. Following given some Evaluation Rating based on Technological and User Point of View:

OS Dependency	As the application is built with java, so the application can be used in any server operating system.
Simplicity	Each Options is kept as much as simple, which helps user to use this application more easily.
Security	The application can be run through SSL based connection to secure the connection with clients. We have also included AES password encryption so user's password are kept secret into the system.
Loosely Coupled Architecture	The entire system is designed in Loosely Coupled Architecture. In the entire application we tried to keep different components or elements have relatively little knowledge or interactive dependency on other parts of the application.

Chapter-6: Conclusion

Comparing with other software competitors in the market with their software qualities, we are hopeful that CU LMS is one of the best application which started with strong startup. CU LMS is built module basis it have a rich documentation which shall help any developer to easily understand and develop the application according to their needs. We believe that any project that are related to education should be free of cost. And we have put our efforts accordingly to serve the community as well as the country.