

CSCI-5448 Final Report

# Online Library System

Team 40:

Ankita Singh  
Krithika Balan  
Omkar Pawar

# 1. Implemented Features

Non - Functional Requirements				
ID	Requirement	Topic Area	Actor	Priority
NFR-001	UI should be simple and should also be easily accessible to all users and levels of expertise.	Usability	System	Medium
NFR-002	A comprehensive list of FAQs on how to use the system will be available for the customers.	Usability	System	Low
NFR-003	As long as the database is up and running, the system can be restarted in case of failure without loss of data.	Reliability	System	High
NFR-004	System will be supported on multiple platforms.	Reliability	System	High

User Requirements				
ID	Requirement	Topic Area	Actor	Priority
UR-001	As a customer, I want an option to sign up, so that I can create an account on the portal	Account	Customer	High
UR-002	As a customer, I want an option to log in using my credentials so that I can access my account	Authentication	Customer	High
UR-005	As a customer, I want to see a dashboard displaying categorized list of books based on my preferred genres so that I can browse for available books.	Dashboard	Customer	Medium
UR-006	As a customer I want a dashboard so that i can manage the books that I have borrowed	Dashboard	Customer	High

UR-007	As a customer, I want to be able to search for books based on titles, authors, genres or ISBN	Books	Customer	High
UR-009	As a customer, I want to see a book's description so that I can view details of the book and other users' reviews before I borrow it.	Books	Customer	Low
UR-010	As a customer, I want to have an option to contact library staff so that I can ask queries.	Queries	Customer	Medium
UR-012	As a library staff, I want to be able to login, so that I can access my account.	Account	Staff	High
UR-013	As a library staff, I want to see the questions posted by customers so that I can reply with the answers.	Queries	Staff	Medium
UR-014	As a library staff, I want dashboard so that I can add and remove books in circulation.	Dashboard	Staff	High
UR-019	As a library staff, I want an update book functionality so that I can update the details of book in circulation.	Dashboard	Staff	High

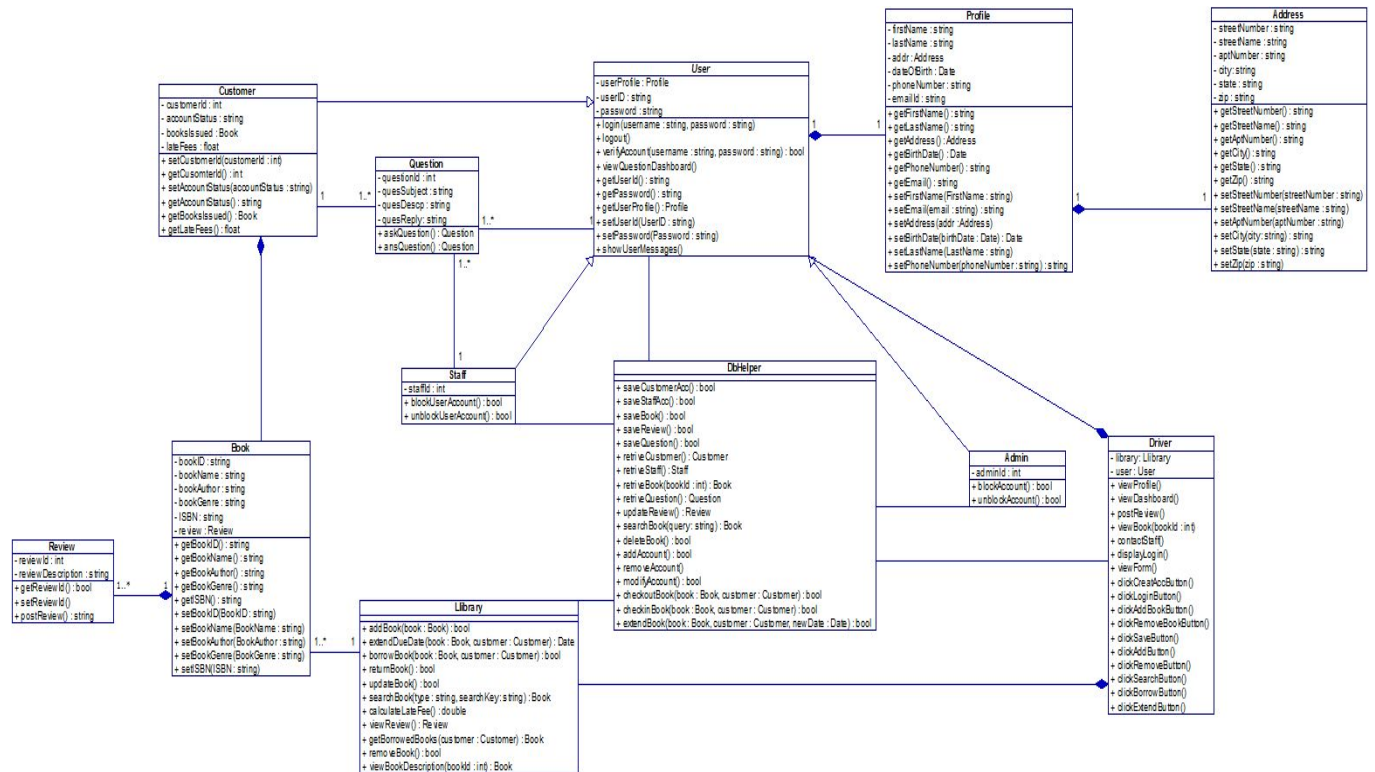
## 2. Features that we did not implement:

User Requirements				
ID	Requirement	Topic Area	Actor	Priority
UR-003	As a customer, I want to update my profile so that i can provide my details	Account	Customer	Medium
UR-004	As a customer, I want to be able to select my preferred genres so that I can see books related to those genres on my home page.	Profile	Customer	Low
UR-008	As a customer, I want an option to post reviews for books so that I can write reviews for the book that I have read.	Books	Customer	Low
UR-011	As a customer, I want to be able to remove account when I no longer need it.	Account	Customer	Low
UR-015	As a library staff, I want to able to manage customer accounts so that I block/unblock accounts when needed	Account	Staff	Medium
UR-016	As a library staff, I want an option to post reviews, so that I can write reviews for books in circulation.	Book	Staff	Medium
UR-018	As an administrator, I want to able to manage user accounts so that I can block/unblock users as needed.	Account	Admin	High
UR-020	As a library staff, I want to able to manage customer accounts so that I add/remove customer accounts.	Account	Staff	Medium

### 3. Class Diagram

Original Class diagram:

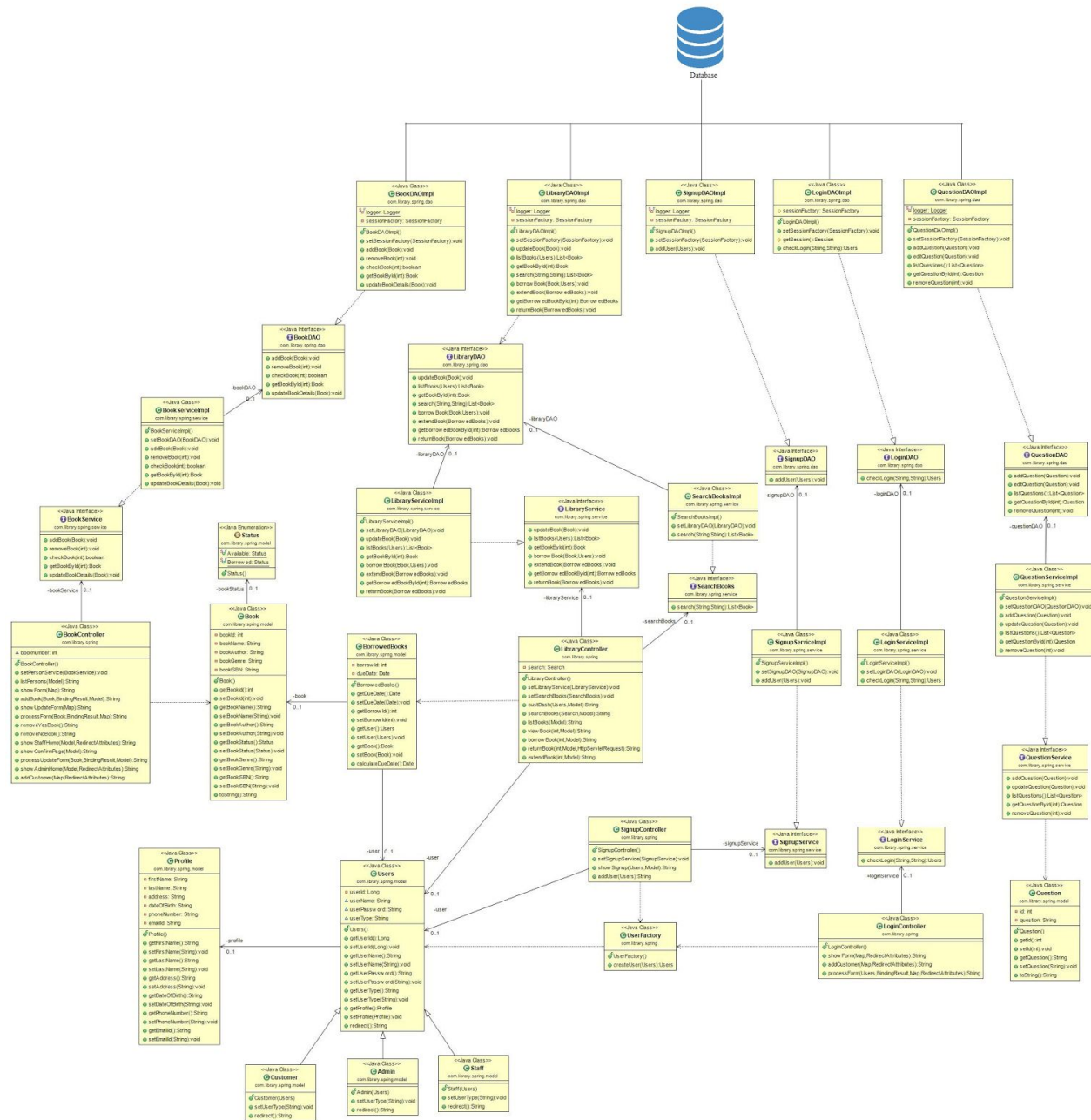
Available on GitHub at: [ClassDiagram\\_OnlineLibrarySystem](#)



# Final Class Diagram

The class diagram can be found here:

[https://github.com/ankita-singh4/Library-Portal/blob/master/ClassDiagram\\_Final.jpg](https://github.com/ankita-singh4/Library-Portal/blob/master/ClassDiagram_Final.jpg)

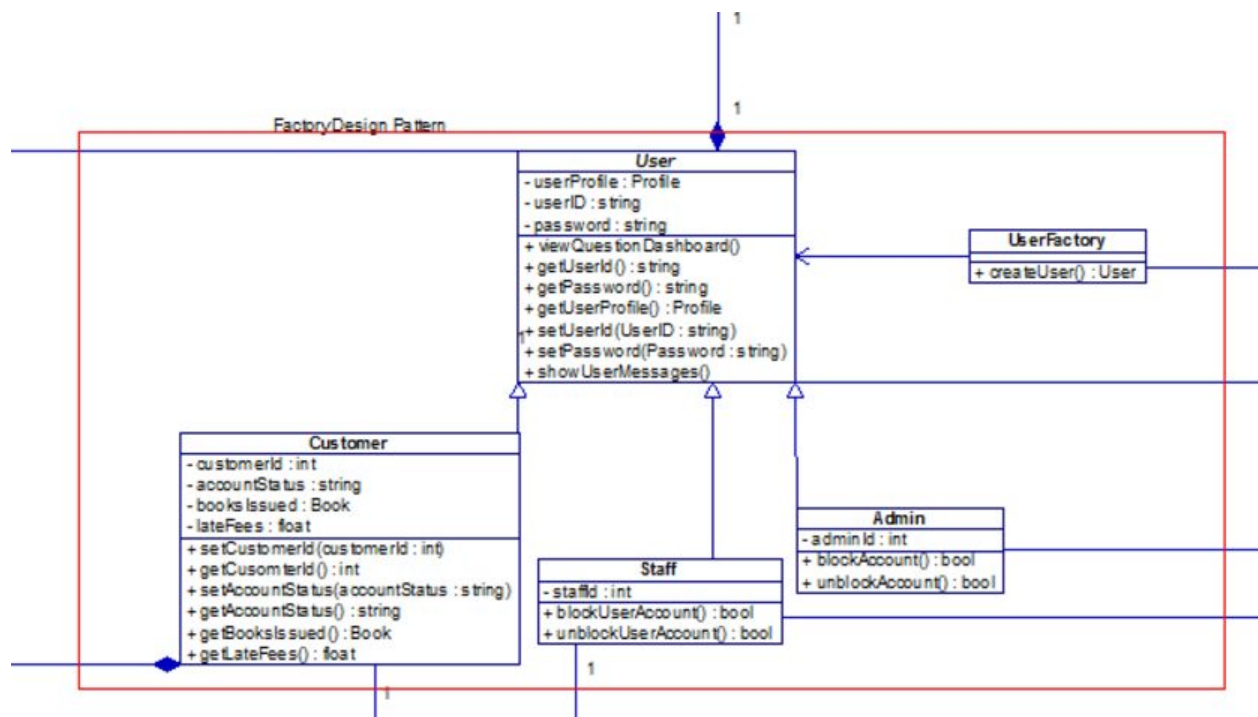


If observed carefully, our class diagram has changed because we have used spring and hibernate.. Since we had our class diagram ready and also since we have our requirements gathered and analysed much earlier, it was easy for us to speed up the process of implementation. We tried using the principle of low cohesion and high coupling during the implementation process. We also tried sticking to the single responsibility and separation of concerns while implementing the classes.

Most of our original class diagram remained the same. We just made a few changes after refactoring and also after we decided to implement design patterns. We removed a few redundant classes and since we used hibernate for our ORM we used we had to remove our DB helper class. Also since we used Spring MVC we had to remove most of our classes related to UI and event handling.

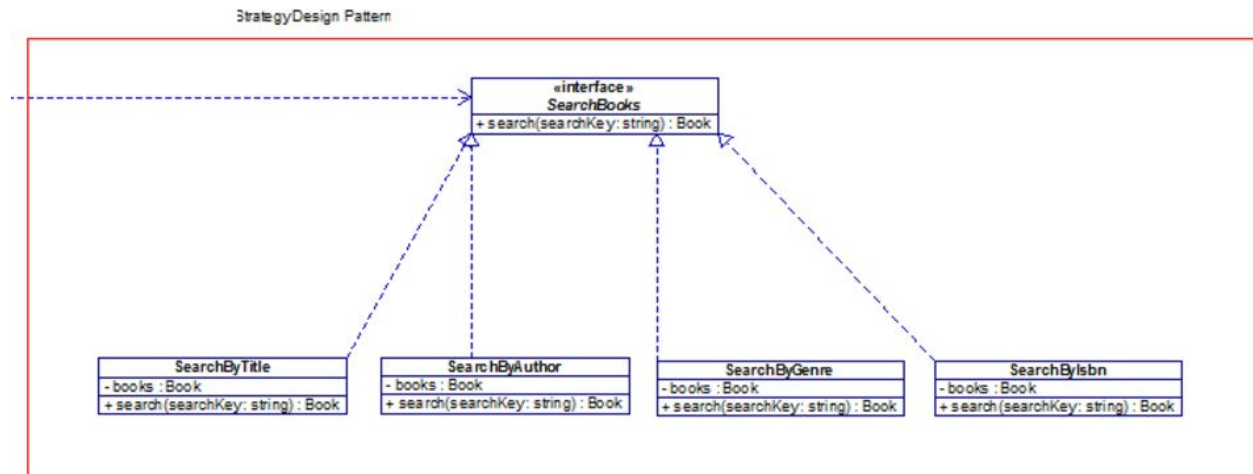
4. Did you make use of any design patterns in the implementation of your final prototype? If so, how? Show the classes from your class diagram that implement each design pattern (each design pattern as a separate image in the .PDF). If not, where could you make use of design patterns your system? Show a class diagram of how you could implement each design pattern and compare how it would change from your current class diagram.

Yes, we did make use of design patterns. While refactoring for part 3 of the project we realised that we could make use of few design patterns. We did further analysis of our requirements and design and also considered the scale of our project. Based on this, the design pattern that we implemented is, the factory design pattern. We used this design pattern for the User class. In our application, we have 3 types of users namely Customer, Staff and Admin. We realised that we could use factory



Another design pattern that we thought we could implement was the strategy design pattern. We realised that we could apply strategy design pattern to implement search functionality. Since we wanted to be able to search a book by different search criteria we thought that implementing this design pattern would be ideal in our case. But during the implementation we found that there was not much difference in term of algorithm that we were using for searching. Since the difference in algorithms is the crux of Strategy design pattern, we decided against the use of it. Also, since the scale of our application was small we did not implement it. Our application is open to design modification and we plan to implement it in future.





## 5. What have you learned about the process of analysis and design now that you have stepped through the process to create, design and implement a system?

During the process of analysis and design we realised that proper collection and analysis of requirements gave us clear understanding of the application and helped us to implement the project faster. We were clear as to what features our application will have. We used UML diagrams to give us a better understanding of the flow of data, and also in what steps to implement the application.

With good understanding of the requirements we were able to create well defined and simple use cases. This helped us later in the project when we were assigning individual tasks and we were able to divide and implement the work better. We were able to integrate them better, without spending much time on bug-fixes.

We also realised that during the course of implementation, the requirements are not fixed and are bound to change. With this in mind we realised that having fixed and rigid requirements and designs would hinder the flow of work. Instead, a flexible, simple and open to modification style of design is more efficient in developing a good product.

Along with OOAD concepts, we were also exposed to the entire product development cycle. This was a good learning experience. It gave us more clarity and a better understanding of how to design and implement an end to end application.

We used Spring MVC framework and Hibernate for the ORM. Not only did we learn new technology but also using these eased our implementation. We were able to focus more on business and application logics rather than spending time on configuring underlying frameworks and architecture.