E-Book Management System

Software Requirement Specification Document

SRS Document

E-Book Management System

Submitted by

Mashfiq Shahriar Zaman - 17301167

Ibna Kowsar - 17301130

Ajmain Alam - 17101054

Saqibul Alam -17301058

Submitted to

Afrina Khatun

Lecturer, Brac University

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1 Introduction to E-Book Management System

An eBook management System is software built to handle the primary housekeeping functions of reading books. The management system provides the users and the administrators an easy and efficient way to read and manage books in online. As technology develops nothing remains untouched and paper is no exception. EBook are beginning to change the way information is created, disseminated and viewed. These downloadable files from the eBook system include everything typically found in print formats including numbered pages, table of contents, pictures, graphics and layout just like any other print book.

1.1 Purpose

The document is about the Software Requirements Specification (SRS) for the EBook Management System. The purpose of this document is to give a detailed description of the requirements for the EBook Management System. This software requirements specification document enlists all necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This has been prepared after detailed communications with the project team and stakeholders.

The document includes a set of use cases that describe interactions the users will have with the software. It will also explain the system constraints, interface and interactions between the software and the users. The document also includes the testing of the software system so that the errors can be solved before occurring.

1.2 Overview

- The remainder of this document includes six chapter and appendices
- The second chapter introduces the inauguration of EBook Management System like the introduction, meeting and various types of viewpoint and collaboration.

- The third chapter discusses about the requirement engineering like usage scenario, functional requirements and non- functional requirements.
- The fourth chapter introduces the modeling based scenario of the system.
- The fifth chapter discusses the class based modeling concept
- The last chapter describes the testing case of the management system.

1.3 Conclusion

In this document we have discussed our project overview including project purpose. In next chapter the inception of this project will be described.

CHAPTER 2

Inauguration of Student Information Management System

2.1 Introduction

The first phase of requirements engineering is inception. The phase named inception helps to initiate the meetings to build the first draft about the project planning.

The inception phase helps to find not only the upcoming errors of the system but also it alleviates to describe the requirements properly. Moreover, this phase helps to develop solutions for the problems. The inception phase of the eBook management system defines how the system would get started and what the scope is. The phase would also find the nature of the upcoming problems and the solutions. Again, the main purpose of the inception phase is to find out the requirements of the users to discover their requirements, prioritizing the requirements and resolving requirements conflicts. To get all the necessary information we have done the following inception phases:

- Planning Meeting
- Identifying Users
- Identifying multiple viewpoints
- Working towards Collaboration
- Conclusion

2.2 Planning Meeting

To get a proper view of the requirements we have arranged several meetings with the users. The meetings helped us to know about the needs and which requirements should be given more priority. Depending on the size of the project and its complexity the meeting may take several days.

 Date: 3 October, 2019

Place: Computer Lab

Subject Matter: Identifying the users

Group Members:

- Mashfiq Shahriar Zaman -17301167
- Ibna Kowsar -17301167
- Ajmain Alam - 17101054
- Saqibul Alam - 17301058

 Date: 15 October, 2019

Place: Library

Subject: Collecting requirements from the user

Group Members:

Above all the members.

 Date: 27 October, 2019

Place: Research Lab

Subject: Identifying the conflict requirements and prioritizing the requirements.

Group Members:

Above all the members.

Identifying the users

The users of the eBook management system are those persons who want to read books from anywhere and in anytime. The people who do not want the hassles of going to a library and borrow books from the library from a certain period of time. The users of the eBook management system will be able to read books in online and after being a registered user of the system they will be able to download the books. We want to give people access to wide range of books. The following queries help us to identify the users:

- Who will use the system?
- Who will be benefited from the system?
- Who will maintain the system?
- Who contains information about the system?

Queries to the Users

We have set a group of questions in a particular way so that the users could easily understand the questions and give information depending on the asked questions. The questions helped us to get to know about the user requirements of the system and design the system in a proper way so that the user will get more benefits.

2.3 Recognizing Multiple Viewpoints

User Viewpoint:

- User will be able to read books online
- Can request for more books
- User- friendliness
- Same layout and design as the printed books
- Easy access
- Supports in multiple operating system

Admin Viewpoint:

- Secured database

- Database backup
- Easy maintenance like inserting, updating, deleting books and users
- Separate accounts for registered and non-registered users
- Easily understandable user interface

System viewpoint:

- Fast recovery in case of server crash
- Handle data traffic
- Multiple languages to give access to a wide range of users
- Server will be running for 24/7
- Less reload time

2.4 Working towards Collaboration:

To find out the requirements of the users we have asked several questions to the user. After gathering all the information, we have found out that some requirements of the user are conflicting with each other and some requirements are common. Therefore, we need to follow the stated steps given below:

- Finding out the requirements which are common
- Finding the conflicting requirements
- Identifying the special requirements
- Grouping the related requirements
- Prioritizing requirements
- Taking final decisions

Conclusion:

The inception phase helped use to get information about the requirements of the users and prioritizing the requirements. Moreover, the inception phase is a major part of the system to group similar requirements and removes redundancy. The next chapter will discuss about finding requirements and user scenario.

CHAPTER 3

Requirement Engineering

3.1 Usage Scenario:

E-Book Management System is a web-based library management system that will allow users, registered or unregistered to have total access to books of all genres. This online based book reading system will give the user flexibility and hassle free reading experience. The user has the freedom to access the web based application from anywhere where the service is available and anytime. An eBook is a book which is stored in the digital format. The users can download it to a computer, PC, Mac, laptop, PDA or any kind of computer and can be read on the screen. The eBook will have numbered pages, table of contents, pictures and graphics just exactly like a printed book. The only difference is that after being registered in the system the registered user will be able to download the eBook.

Users of this application will be categorized as following:

- Registered
- Non - Registered
- Admin

Users will be able to register to the application where they will be assigned with a unique ID and password. They can then login to access the facilities of a registered user. A registered user can maintain a wish-list where they can add books to read in the future as well as delete from the list. They can search the database based on different criteria such as, book name, author, year published etc. After reading they can also rate the books to let other users know about them. If there are such books that a user wishes to read but is not available in the database, they can request for books to the admin.

Besides this, if a user does not register in the system, they can still read but for a limited time. The users who will not be recognized as registered user cannot be able to download any books or they cannot request for any new books or add ratings or comments. The registered user will have

some privileges. The non-registered user will be named as guest user. However, the guest user can also do the registration at a later time and enjoy the advantages of registered user.

The admin will moderate the users and the books. They will collect information regarding the wish-lists of different users see if there any books that need to be added to the database. They will also be able to delete any books that are outdated. Admin can remove inactive users from the database as-well-as add new ones. They can also update the catalogue.

3.2 Functional Requirements

Depending on the scenario above the functional requirements of the system are stated below:

1. Users will be able to login

2. Admin will be able to add new books

3. Users will have to register

4. They will also be able to add books to their wish-lists, as well delete them

6. Admin can delete inactive users

7. He can also delete outdated books

8. Users can read and request for books

10. Users can search for books based on different criteria

11. The login has to be authenticated

12. Top readers will be identified

13. Registered users will be request for unavailable books

3.3 Non-functional requirements

The non-functional requirements of the system are given below:

1. Encrypted login will be done to ensure security.

2. Database backup

3. The user interface will be attractive and easy to use

5. Able to deal with viruses like sending message when the system will be affected by viruses

6. Less reload time on any interface

7. Fast recovery in case of server crash

10. Can handle huge amount of traffic.

11. Server will be running 24/7

12. Multiple languages to give access to a wide range of users

13. System will have different users

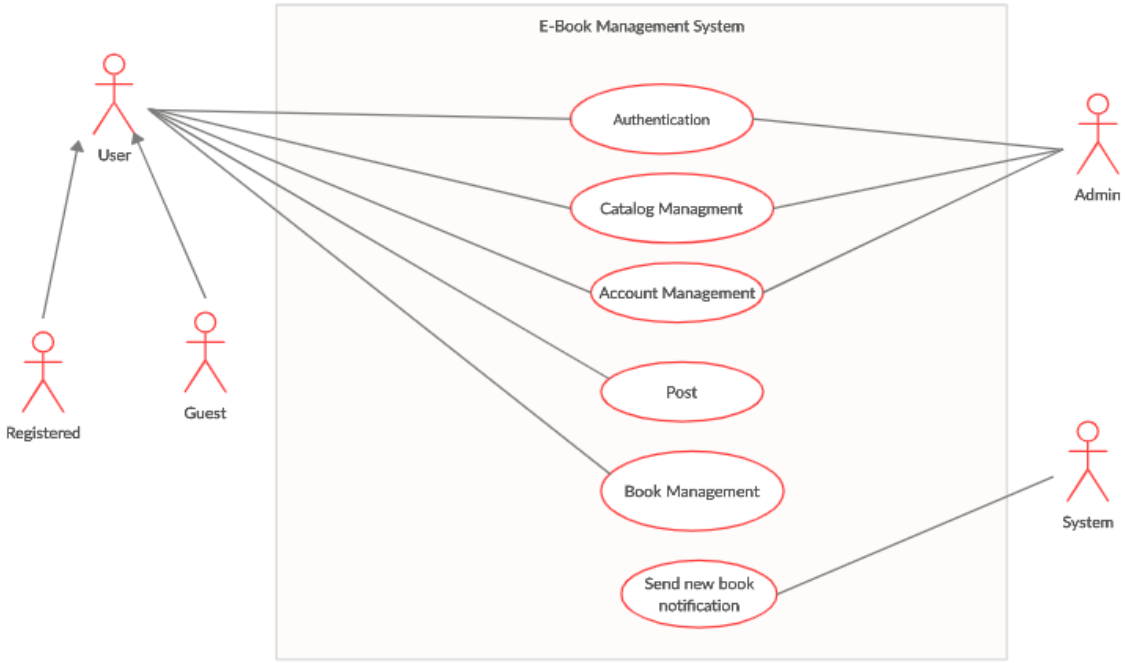
Chapter 4

4. Scenario Based Modeling of E-Book Management System

4.1 Introduction

Requirements modeling are essentially the planning stage of a software system or application. The path by which such a system or application comes to life will generally begin with a business or entity identifying a problem they're facing that requires a software solution, and they approach a software development team to come up with this solution. Scenario Based Modeling is the process of identifying the requirements this software solution must meet in order to be successful. Requirements modeling contains several sub-stages, typically: scenario-based modeling, flow-oriented modeling, data modeling, class-based modeling and behavioral modeling. Also, as the term 'modeling' implies, all of these stages typically result in producing diagrams that visually convey the concepts they identify. The most common method for creating these diagrams is UML. In Use Case we created multiple primary and secondary actors to show the requirements of the management system in detail.

4.2 UseCase Diagrams and Scenario



Level 0: E-Book Management System

FIGURE 4.2.1 - E-Book Management System (Level 0)

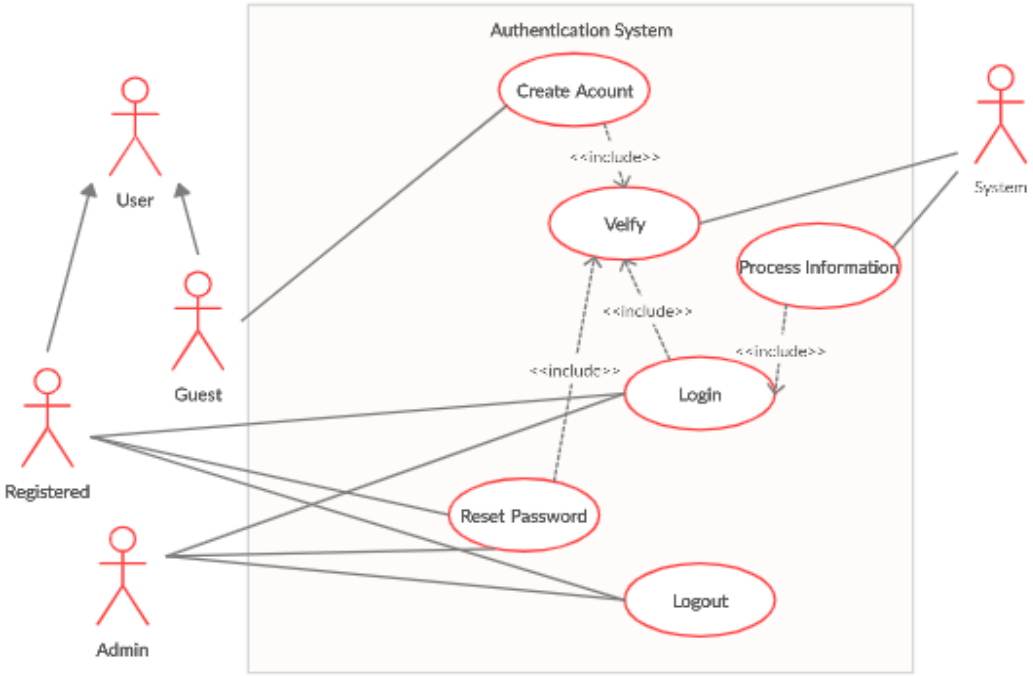
Use case Name: E-Book management system

Use case ID: 0

Primary Actor: User (Registered, Guest)

Secondary Actor: Admin, System

1. Authentication System:



Level 1: Authentication System

FIGURE 4.2.2- Authentication System (Level 1)

Use case Name: Authentication System

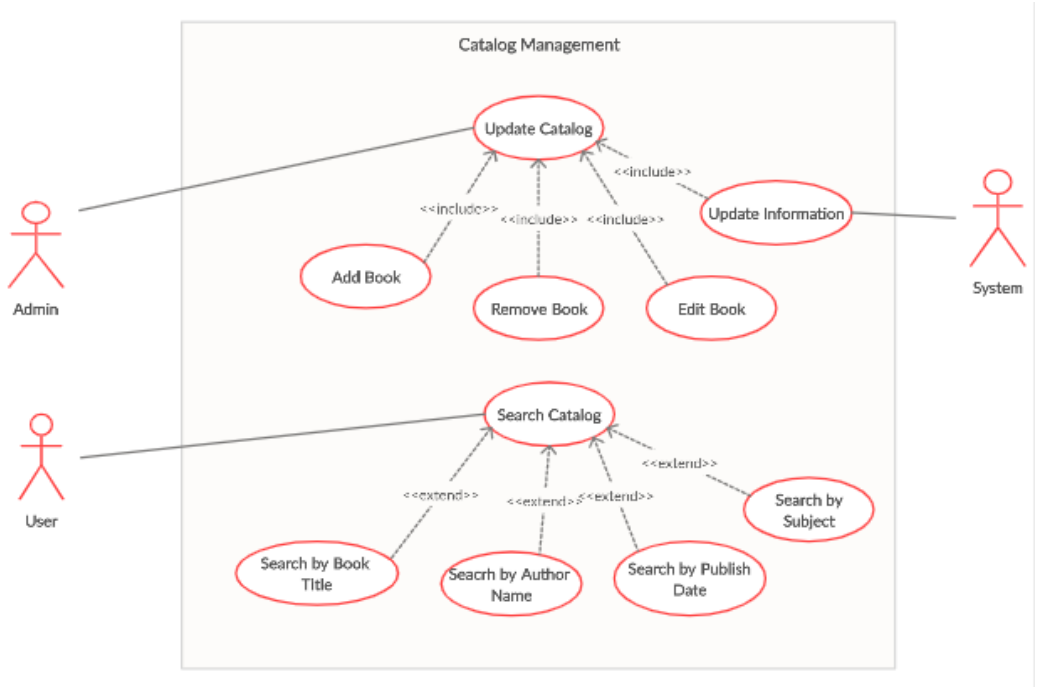
Use case ID: 1

Primary Actor: User (Registered, Guest), Admin

Secondary Actor: System

Scenario for level 1: The authentication system helps to verify a user after giving their id and password by checking the validity of the given id and password with the database. Also, for new users while creating their account the system checks for validity of the given information and processes the information.

2. Catalog Management:



Level 2: Catalog Management

FIGURE 4.2.3- Catalog Management (Level 2)

Use case Name: Authentication System

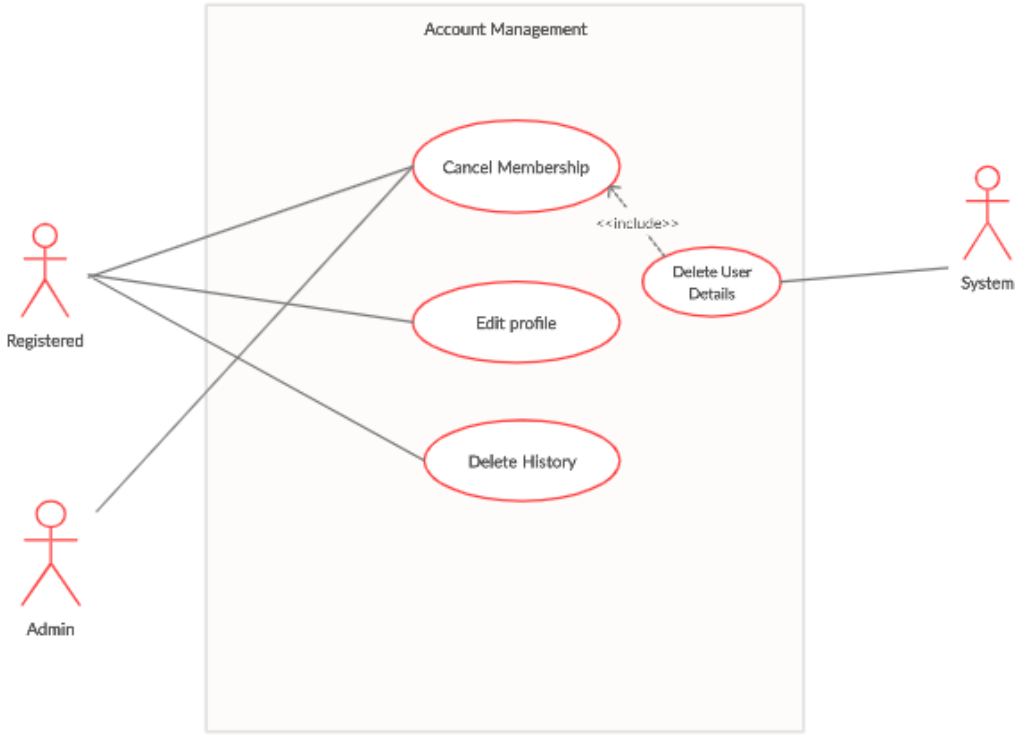
Use case ID: 2

Primary Actor: User, Admin

Secondary Actor: System

Scenario for level 2: Admin can add, delete, edit or update any information regarding the books. Moreover, the users can search their books through filtering by name of the book, name of the author and date of the publication or even by any subject name.

3. **Account Management:**



Level 3: Account Management

FIGURE 4.2.4- Account Management (Level 3)

Use case Name: Account System

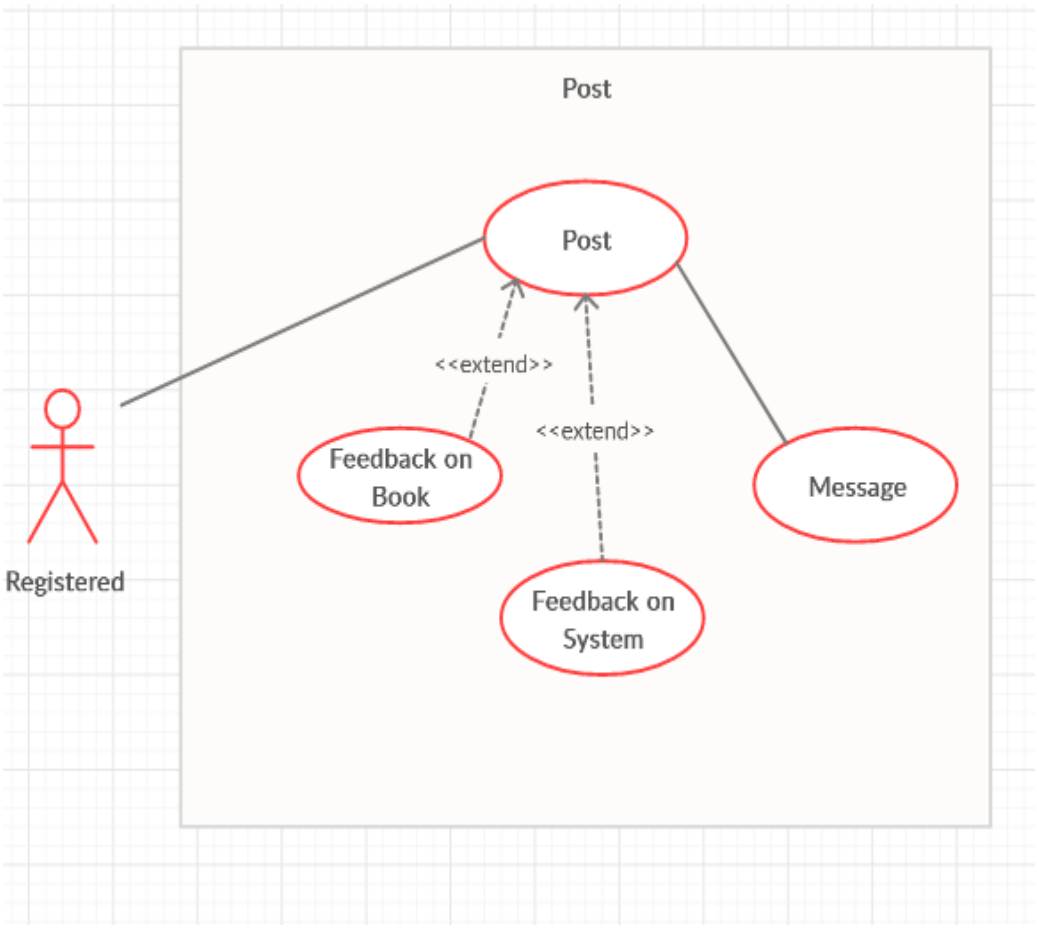
Use case ID: 3

Primary Actor: Registered, Admin

Secondary Actor: System

Scenario level 3: Registered user and admin can edit their account and accordingly the system will update their information. Again, they can cancel their membership which means they can delete their account and system will delete their information also users can delete their browsing history if they wish to.

4. **Post:**



Level 4: Post

FIGURE 4.2.5- Post (Level 4)

Use case Name: Post

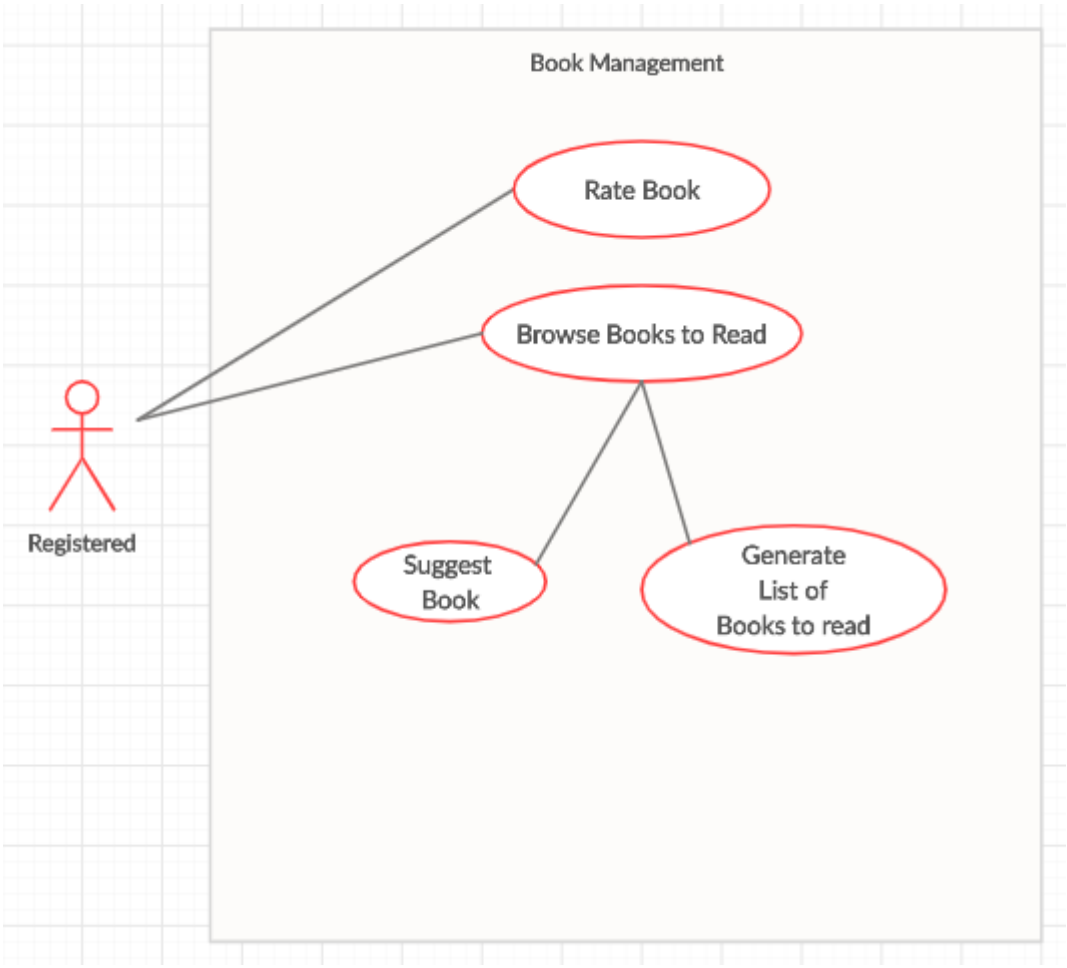
Use case ID: 4

Primary Actor: Registered

Secondary Actor: Nil

Scenario for level 4: Registered users has the privilege to post any feedback regarding the books they are reading so that others can know about the book and also they can give feedback on the system as well. For instances, if all the pages GUI are available, also how to improve system so that it can ensure more friendly environment for readers.

5. **Book Management:**



Level 5: Book Management

FIGURE 4.2.6- Book Management (Level 5)

Use case Name: Book Management

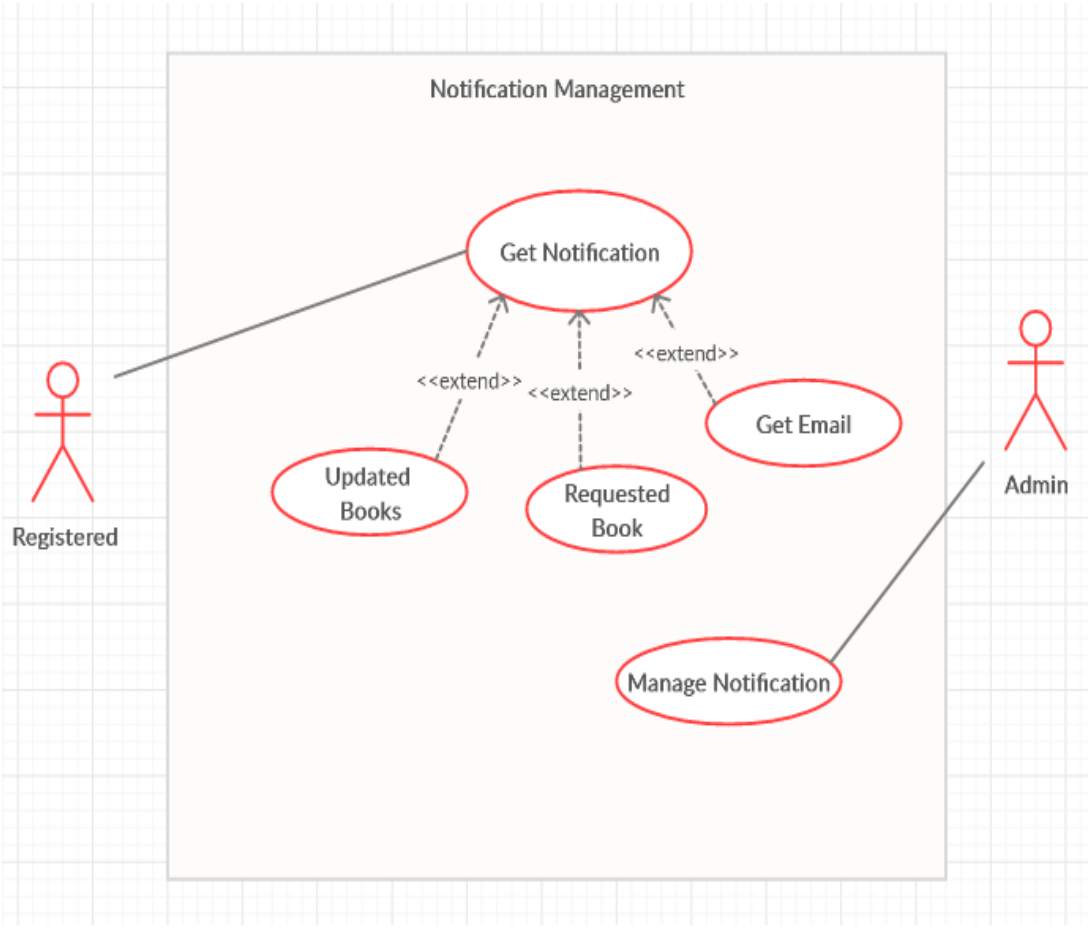
Use case ID: 5

Primary Actor: Actor

Secondary Actor: Nil

Scenario for level 5: Registered users can rate books after reading each book to let other readers about the book of the system and if they want they can suggest a book to any individual and also they can generate a list of book which will be known as wish-list where they will be able to see the list of book they are interested in.

6. **Notification Management:**



Level 6: Notification Management

FIGURE 4.2.7- Notification Management (Level 6)

Use case Name: Notification Management

Use case ID: 6

Primary Actor: Registered

Secondary Actor: Admin

Scenario for level 6: Registered user will receive a notification if there is any updated book on the system and they will get this service only if they permit the system. As in our previous section we showed registered user can request for a new book which is not available, so after adding the book the corresponding user will be notified.

Chapter 5

5 Class based modeling concept

5.1 Identifying Analysis classes:

Potential classes:

- External entities (e.g., other systems, devices, people) that produce or consume information to be used by computer-based system.
- Things (e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
- Occurrences or events (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
- Roles (e.g., manager, engineer, salesperson) played by people who interact with the system.
- Organizational units (e.g., division, group, and team) that are relevant to an application.
- Places (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
- Structures (e.g., sensors, four-wheeled vehicles, or computers that define a class of objects or related classes of objects.

5.2 Selection of classes from potential classes:

1) **Retained information.** The potential class will be useful during analysis only if information about it must be remembered so that the system can function.
2) **Needed services**. The potential class must have a set of identifiable operations that can change the value of its attributes in some way.
3) **Multiple attributes**. During requirement analysis, the focus should beon "major" information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.
4) **Common attributes**. A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.
5) **Common operations**. A set of operations can be defined for the potential class and these operations apply to all instances of the class.

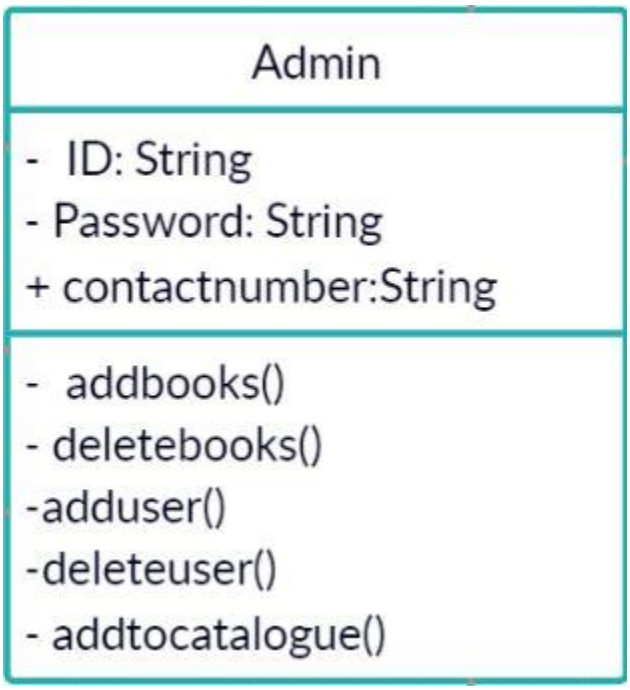
6) **Essential requirements**. External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirements model.

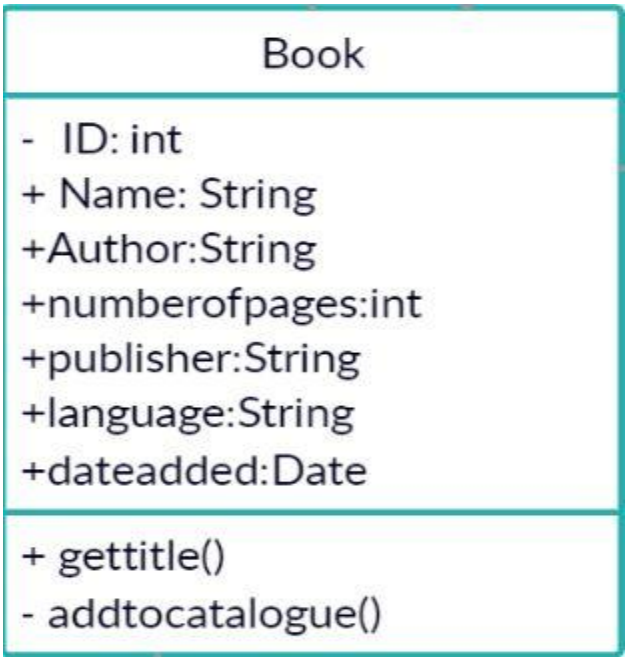
5.3 Potential class table:

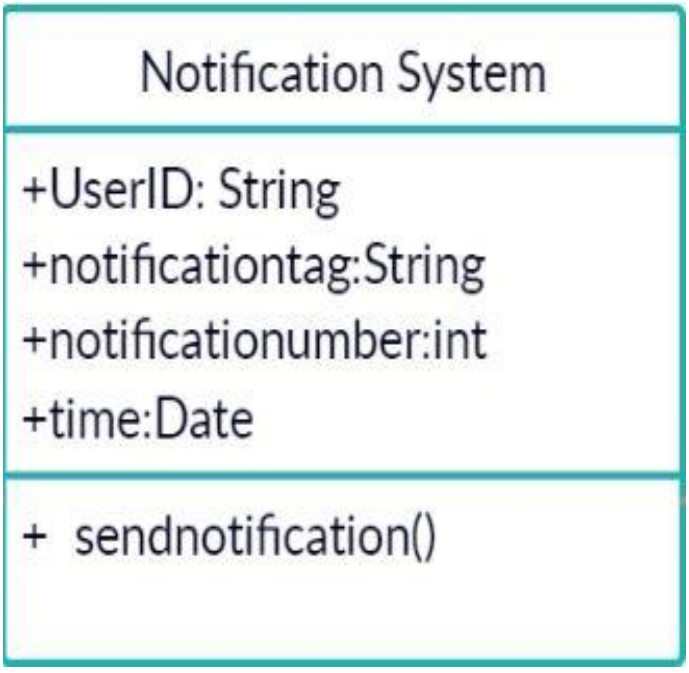
Potential class	Classification	Class Analysis	Result
User	External entities	1,2,3,4,5,6	Accepted
Rating	Things	1,2,3,4,5	Accepted
Notification	Occurrences or events	1,2,3,4,5	Accepted
Book	Things	1,2,3,4,5,6	Accepted
Catalogue	Things	1,2,3,4,5	Accepted
Author	Roles	1,2,3	Denied

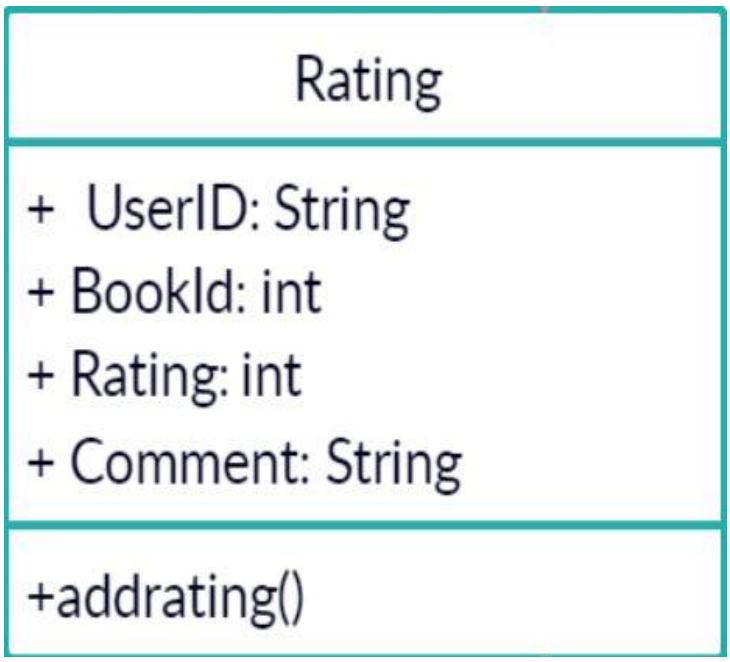
FIGURE 5.3.1 – Potential Class Table

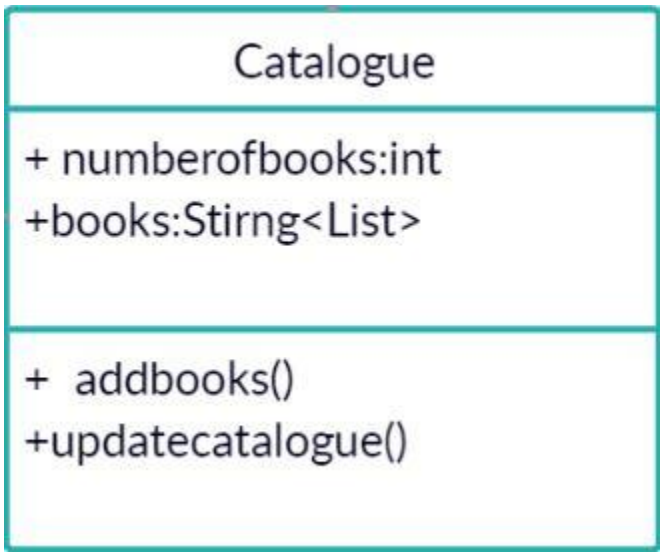
5.4 Class Diagram for E-Book Management System:

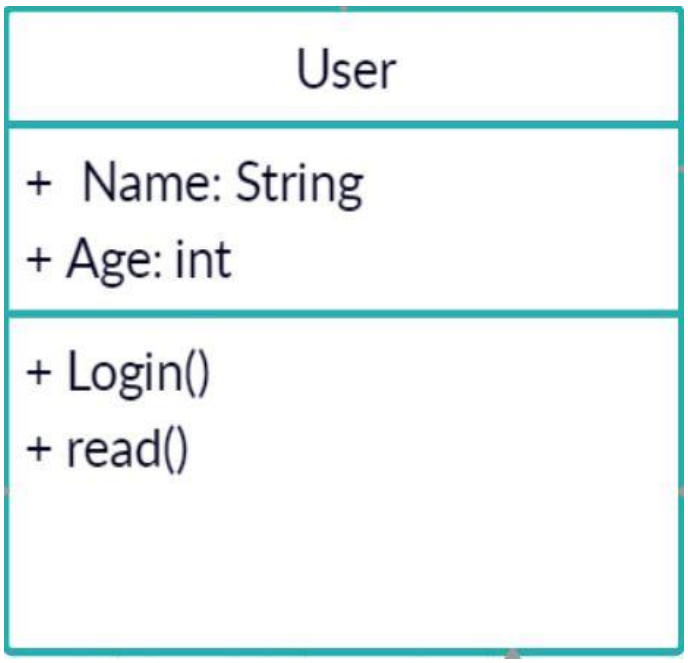



Book
- ID: int
+ Name: String
+Author:String
+numberofpages:int
+publisher:String
+language:String
+dateadded:Date
+ gettitle()
- addtocatalogue()

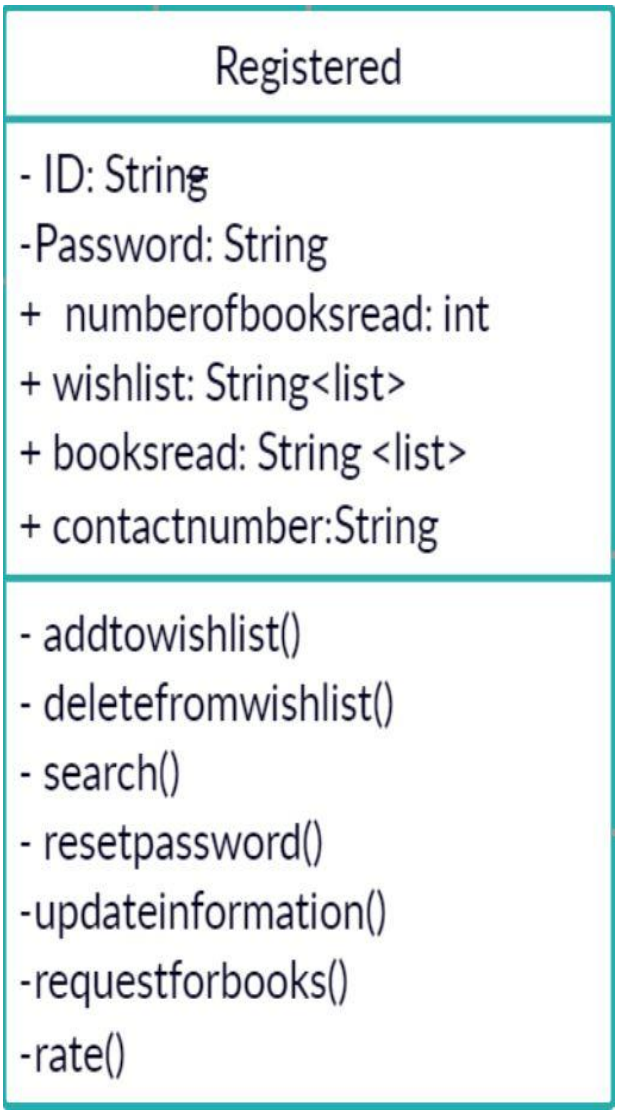

Notification System
+UserID: String
+notificationtag:String
+notificationumber:int
+time:Date
+ sendnotification()


Rating
+ UserID: String
+ BookId: int
+ Rating: int
+ Comment: String
+addrating()


Catalogue
+ numberofbooks:int
+books:Stirng<List>
+ addbooks()
+updatecatalogue()


User
+ Name: String
+ Age: int
+ Login()
+ read()


Unregistered
+ sessiontime: Date
- Register(): Type


Registered
- ID: String
-Password: String
+ numberofbooksread: int
+ wishlist: String<list>
+ booksread: String <list>
+ contactnumber:String
- addtowishlist()
- deletefromwishlist()
- search()
- resetpassword()
-updateinformation()
-requestforbooks()
-rate()

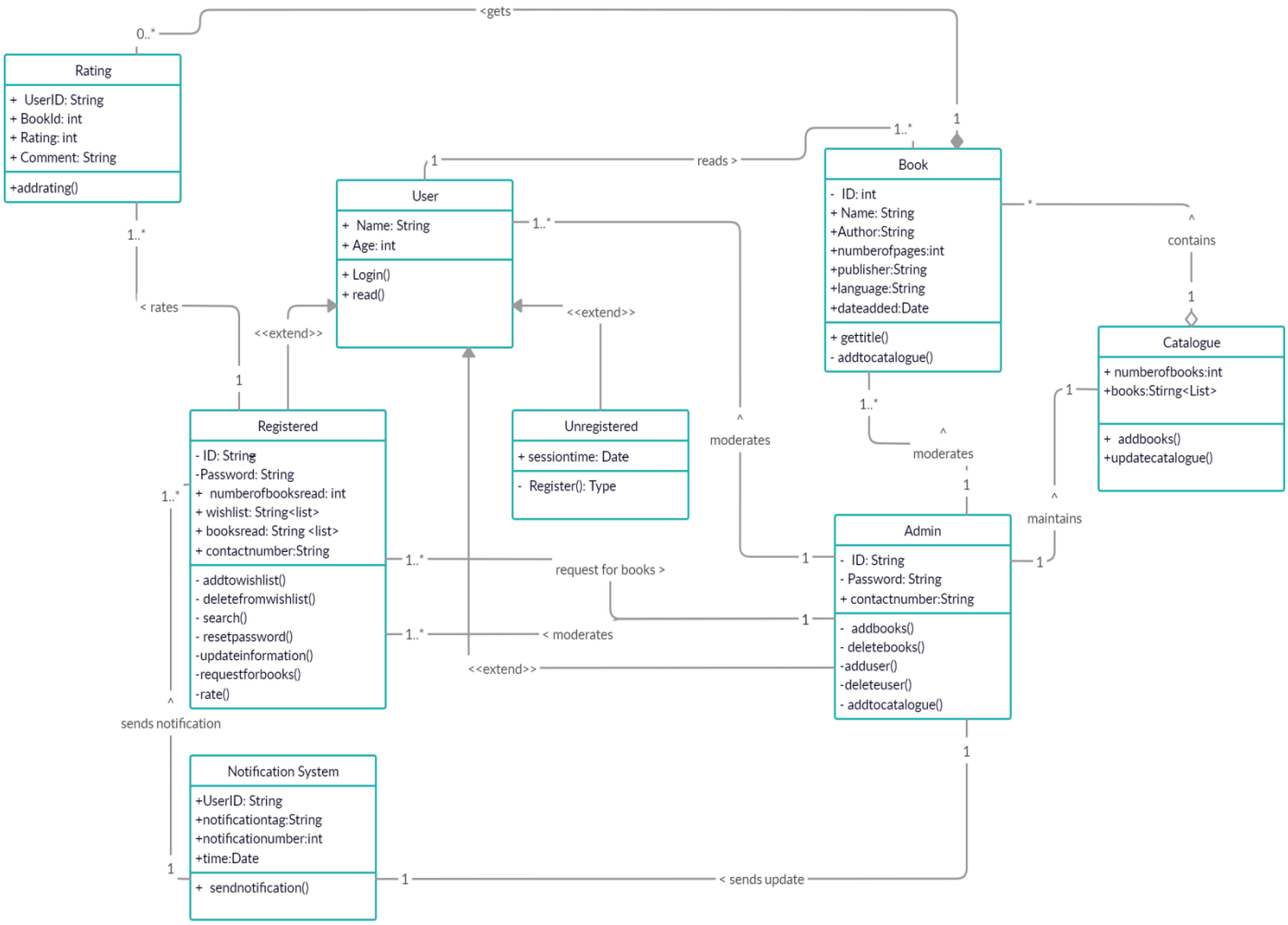


Figure 5.4.1 – Class Diagram

Chapter 6

Testing

6.1 Functional Testing

6.1.1 Unit Testing:

```
public class account
{
        String ID;
        String password;
        String contact;
        List <String> wishlist;

        public account(String id, String pass, String number){
           ID=id;
           password=pass;
           contact=number;
        }
        public void addtowishlist(Book book){
           wishlist.add(book);
        }
}
public class accounttest{
   static account acc;

   public void setupacc(){
      acc=new account("123456","qwerty","0987654321");
       }
   public void testaddtowishlist(){
      acc.addtowishlist("Narnia");
      assertEquals("Narnia",acc.wishlist);
```

```
    }
}
```

6.2 Non-Functional Testing

Let’s assume that the threshold of the system is 500 requests

6.2.1 Load Testing

Load testing is the process of testing the behavior of the system by applying maximum load in terms of Software accessing and manipulating large input data. It can be done in both normal and peak load conditions. Loading the system with requests up to and until the threshold to see how the system handles the requests and to see if there is enough database storage capability. The Load testing of this systemcan be performed with the help of automated tools such as Load Runner, AppLoader, IBM Rational Performance Tester, Apache JMeter, Silk Performer, Visual Studio Load Test etc.

6.2.2 Stress Testing

The Stress testing includes the testing of software behavior under abnormal conditions like taking away the resources, applying load beyond the actual load limit. Loading the system with requests beyond the threshold to how the system behaves when the system breaks; to see if data was backed up correctly when recovered and for how long the system stays down. This can be also be done by turning on or off the networks ports or the database.

CONCLUSION

To conclude, E-book reader is a web application that will allow the users of the app to read books anytime, anywhere given they have access to the internet. We have tried our best to evaluate all possible requirements that users may have and integrated those to our plans regarding the creation of the application. We hope that the given figures will help the developers in completing the project.