



Project Report
On
Book Reservation System(library)

Submitted to:

Prof. Eng. Wahab Hamou-Lhadj

COEN6312 – Model-Driven Software Engineering

Submitted By:

Raynard Omongbale – 40150036

Alireza Sameni – 40151521

Issam Sedki – 40171941

Heyu Guan – 40113703

Ying Sun – 40115153

Yu Song – 40114330

February 9, 2021

ABSTRACT

This project focuses on building a book reservation system that would provide users with easy access to reservation services through a clear and efficient management method. This project development is based on Scrum methodology.

This report includes the team members' experience, a description of the requirements for the entire system, and a design class diagram as well as a description of the system. The system overview includes a detailed feature list, user stories, and functional and non-functional requirements. The system design class diagram was constructed using Papyrus.

TABLE OF CONTENTS

ABSTRACT	2
LIST OF FIGURES	4
LIST OF TABLES	5
TEAM INFORMATION	6
1. REQUIREMENTS DOCUMENT	1
1.1 Descriptions of the Book Reservation System	1
1.2 Product User Stories	3
1.2.1 User Stories of Normal Users	3
1.2.2 User stories of administrators	5
1.2.3 User stories of the database	7
2. DESIGN DOCUMENT	8
2.1 System Class Diagram	8
2.2 User Interface	10
2.3 Project Versioning	10
Reference	11
APPENDIX I – EXAMPLES OF COMMUNICATION AND COLLABORATION OVER ZOOM AND WHATSAPP GROUP	12

LIST OF FIGURES

<i>Figure 3 Class Diagram of Book Reservation System</i>	<u>8</u>
<i>Figure 1 User Login Interface</i>	<u>10</u>
<i>Figure 2 User Registration Interface</i>	<u>10</u>

LIST OF TABLES

<i>Table 1 Team Members and their descriptions</i>	<u>6</u>
<i>Table 2 User stories of normal users</i>	<u>4</u>
<i>Table 3 User stories of administrators</i>	<u>6</u>
<i>Table 4 User stories of the database</i>	<u>7</u>

TEAM INFORMATION

Below is a list of the team members and the expertise of each member:

	Member's name	Student ID	Expertise language
1	Raynard Ombale (team leader)	40150036	Software Engineer (C++, Python, Golang, VHDL)
2	Issam Sedki	40171941	senior software programmer and architect (Java, angular, OOP, togaf,...)
3	Alireza Sameni	40151521	Embedded firmware engineer. Verilog, C, C++, JAVA
4	Ying Sun	40115153	JAVA, C, C++
5	Yu Song	40114330	C#, asp.net, Python, basic JAVA
6	Heyu Guan	40113703	JAVA, Python

Table 1 Team Members and their descriptions

1. REQUIREMENTS DOCUMENT

1.1 Descriptions of the Book Reservation System

This section is about the requirements description of the book reservation system.

The main uses of the book reservation system are:

- i. Upon registration, all the users can access the library system with a login (email) and a password.
- ii. The password is not stored in plain text in the database.
- iii. The system generates a unique id for each registered user.

Student:

A student can register in the library with his student id. Once registered, each student can borrow up to 5 books/periodicals at a time. If library material is not returned after the expected duration, a fine is applied for each day of delay (20 cents/per day). The student cannot borrow a new book if he owes the library a fine.

A student has a name, email, phone number.

Teacher:

A teacher can register in the library with his school id. Once registered, the teacher can borrow any number of books/periodicals.

A teacher has a name, an email, a phone number.

Library administrator:

The administrator is responsible for the well-functioning of the library. Among other things, he can:

- approve/decline registrations.
- blacklist a user.
- Add or remove a book from the library inventory.
- Ask a user to return a book.

The administrator has a name, an email, a phone number.

Library material management:

Books and periodicals can be added to the inventory by the administrator at any time. The system can keep track of who added the material to the inventory and the date the operation was done in.

Library material is added to a specific section of the library, depending on its genre.

The books can be rented for a max of 1 month, and the periodicals can be rented for one week max.

1.2 Product User Stories

1.2.1 User Stories of Normal Users

ID	Story	Functional/Non-functional	Priority
1	As a user, I should be able to login into my own profile through my Username and Password so that I can see my information.	Functional	3
	<ul style="list-style-type: none"> - Given a “log in” button, the correct username and password is needed at the same time; then the personal center page should be displayed - Given a “Personal Center” button, the personal information page is displayed when the user clicks the button. 		
2	As a user, I should be able to edit my profile so in case there are some mistakes or changes occurring	Functional	3
	<ul style="list-style-type: none"> - Given that the user wants to edit their profile when the “Profile” page is opened, then the “Edit Profile” button should be available for the user to use. - Given an “Edit Profile” button, when clicked the user should be able to edit any of the information they provided in their profile. - Given a “Save” button, when clicked, any information added or changed by the user should be saved. 		
3	As a user, I should be able to search for the book I want to borrow in the library so that I can find it as soon as possible.	Functional	1
	Display a list of related books based on the keywords entered by the book title entered		
4	As a user, I can check the status of the book so that I can know if the book can be borrowed or how many books are left in the library.	Functional	1
	Displays the status of the book. Unavailable books and books of poor quality cannot be borrowed.		
5	As a user I should be able to view the profile of a book in the library.	Functional	2
	Given a user want to check if the book is the right version, then they have the right to have access to the profile of a book, according to the comparison they are able to obtain the right book.		

6	As a user I should be able to borrow a book when the book is available so that I can use it on time.	Functional	1
	Given a user want to borrow a book, then there should be a “Borrow” button on the user interface so the user can click the button to borrow the book.		
7	As a user I should be able to return a book when I finish using it so that I can return it on time	Functional	1
	Given a user want to return a book, then there should be a “Return” button on the user interface so the user can click the button to return the book		
8	As a user, I expect the book reservation system stays stable during the time I need it the most so that I can obtain my books on time	Non-functional	1
9	As a user I don’t want my information link to a third party so that I my privacy is protected by the system	Non-functional	2
10	As a user I want to use the service for 24h for all day so that I can reserve the target book in advance	Non-functional	2

Table 2 User stories of normal users

1.2.2 User stories of administrators

ID	Story	Functional/Non-functional	Priority
11	As an administrator, I should be able to add books in the systems when there are new books introduced into the library so that the whole systems can be updated on time	Functional	1
	Given an "ADD" function under the admin authority, a blank information form for this brand-new book comes up when admins utilize this function.		
12	As an administrator, I should be able to remove books in the systems when there are some books are eliminated in the library so that the whole systems can be updated on time	Functional	1
	Given an "Remove" function under the admin authority, when the administrator utilizes this function, a book can be deleted from the system.		
13	As an administrator, I should be able to edit books' profile so in case mistakes or changes are showing up.	Functional	2
	Given an "Edit" function under the admin authority, when the administrator utilizes this function, the profile of the book can be edited.		
14	As an administrator, I should be able to add user's information when new students are enrolling to the library.	Functional	2
	Given an "Accept" function under the admin authority, when the administrator utilizes this function, users' registration can be accepted.		
15	As an administrator, I should be able to delete user's information when there are students who drop out of the library	Functional	2
	Given an "delete" function under the admin authority, when the administrator utilizes this function, the registration of users can be deleted.		
16	As an administrator, I should be able to put the users with low credit on the blacklist, so that they cannot borrow any materials in the future.	Functional	3

	Given an “blacklist” function under the admin authority, the users with low credit will be put in the blacklist when the administrator utilizes this function.		
17	As an administrator, I should be able to remove the users on the blacklist, so that they can borrow any materials in the future.	Functional	3
	Given an “remove blacklist” function under the admin authority, when the administrator utilizes this function, the users on the blacklist can recover their rights to borrow materials.		
18	The administrator does any action; the system should be responsive as quickly as possible.	Non-functional	1
19	The system should be easy to use by the administrator and clear enough to reduce any user errors.	Non-functional	2
20	The administrator’s information should be kept safe and not shared to any other third party	Non-functional	2
21	The program should be available to the administrator for 24 hours per day, so if there any changes needed for the inventory, they can easily do that	Non-functional	1

Table 3 User stories of administrators

1.2.3 User stories of the database

Story		Functional/Non-functional	Priority
22	As a developer I want the database to have all the users' information: Username, Password, Name, Email, Phone number, if on the blacklist, and ID so that I can find the needed information in the database.	Functional	1
23	As a developer I want the database to have all the administrators' information: Username, Password, Name, Email, Phone number, if on the blacklist, and stuff ID so that I can find the needed information in the database.	Functional	1
24	As a developer I want the database to have all the materials' information: Name, ID, availability, genre, maximum reservation, if on the blacklist, so that I can find the needed information in the database.	Functional	1
25	As a developer, I want the database to adapt to any changes done by the admins or users to get lost.	Functional	1
26	The database should be able to scale with the number of customers without crashing.	Non-functional	1
27	There should not be any data breeching incident occurring for the database	Non-functional	1

Table 4 User stories of the database

2. DESIGN DOCUMENT

The purpose of this document is to describe the implementation of the Book Reservation System. This system helps in the complete process of reservation and stocking of the library with books for users.

2.1 System Class Diagram

This section shows the representation of the system in terms of structure, classes, attributes, association between the different classes, inheritance etc.

The following class diagram was generated using Papyrus.

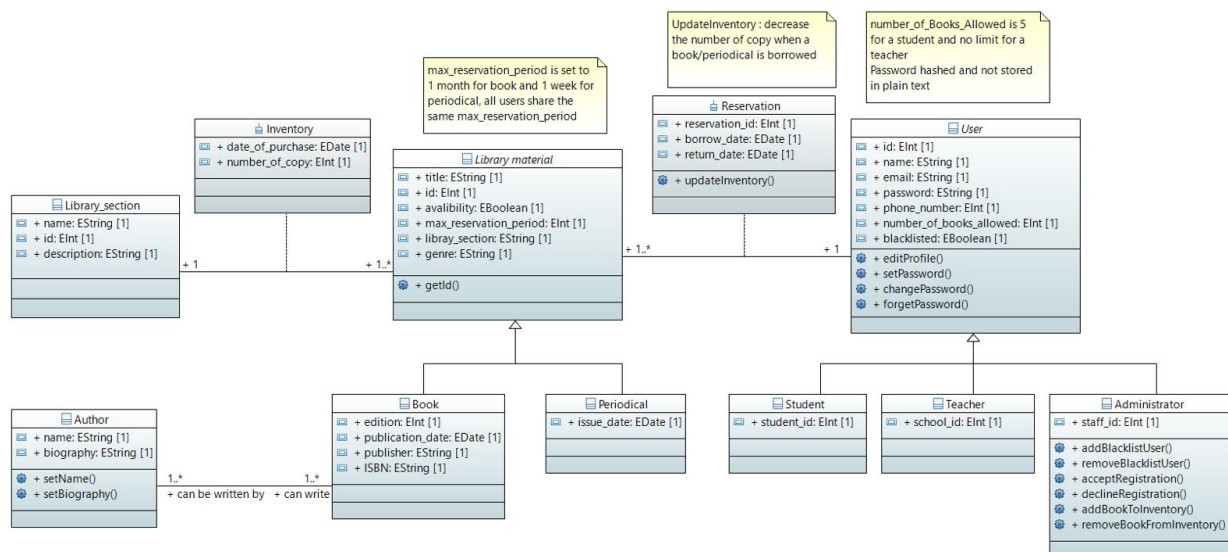


Figure 1 Class Diagram of Book Reservation System

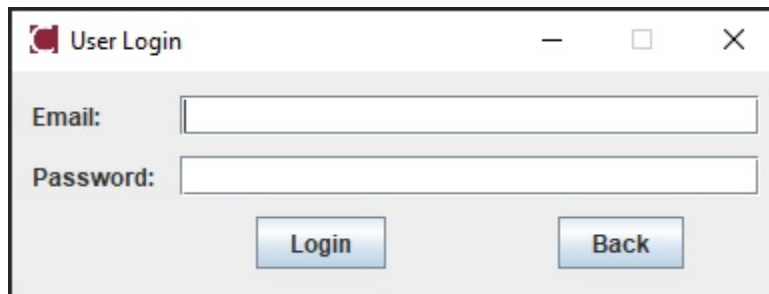
The class diagram above shows: -

- 1) The class diagram has two abstract classes: User and Library material
- 2) The class diagram has two associations: Reservation and Inventory.
- 3) Reservations can keep track of the date and all information associated with a user reserving a book.
- 4) Inventory can keep track of the number of copies added and the date of addition.
- 5) A library material belongs to only one library section (ex: geography, physics, math.)

- 6) A library section can contain many books.
- 7) Many authors can write a book.
- 8) A user can make a reservation for more than a book at a time.
- 9) More than one user cannot reserve a book.
- 10) All types of users can make a reservation for the same period, and each material has a max-reservation-period associated with it.

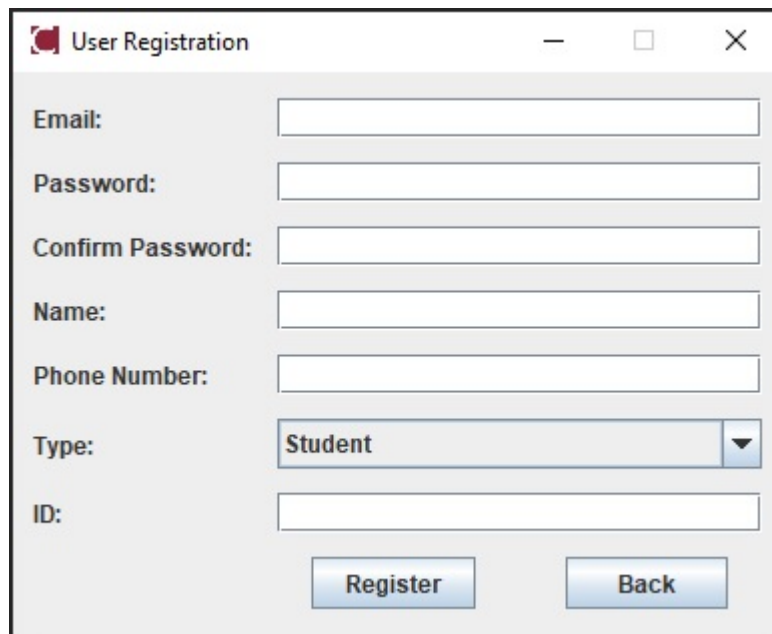
2.2 User Interface

The following two figures are two examples of user interfaces of this system.



A screenshot of a 'User Login' window. The window has a title bar with a red icon, the text 'User Login', and standard window controls (minimize, maximize, close). The main area is light gray and contains two text input fields: 'Email:' and 'Password:'. Below these fields are two buttons: 'Login' and 'Back'.

Figure 2 User Login Interface



A screenshot of a 'User Registration' window. The window has a title bar with a red icon, the text 'User Registration', and standard window controls (minimize, maximize, close). The main area is light gray and contains several text input fields: 'Email:', 'Password:', 'Confirm Password:', 'Name:', 'Phone Number:', and 'ID:'. There is also a dropdown menu for 'Type:' with 'Student' selected. At the bottom are two buttons: 'Register' and 'Back'.

Figure 3 User Registration Interface

2.3 Project Versioning

In this project we make use of the GitHub for project versioning.

GitHub repository URL: https://github.com/raynard2/6312_LibraryProject.git

Reference

[1] Ian Sommerville, Software Engineering, 10th ed., Pearson, March 24, 2015.

[2] Roger S Pressman and Dr. Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 8th Ed., Mcgraw-Hill Education, January 23, 2014.

[3] Ian Gorton, Essential Software Architecture, 2nd Ed., May 17, 2014.

[4] OMG Unified Modeling Language specification V2.5, June 2015

APPENDIX I – EXAMPLES OF COMMUNICATION AND COLLABORATION OVER ZOOM AND WHATSAPP GROUP

