



CS3332 – Software Engineering I

Developing A Library Management Software of Ha Noi University of Science and Technology (HUST)

Group 1			
Group Members	No.	Name	ID
	1	Ta Duc Duy	1624634
	2	Tran Hai Minh	1642853
	3	Le Duc Tuyen	1624790
	4	Truong Thanh Hung	1624817
	5	Vu Duy Hao	1628402
Lecturer	Dr. Du Dinh Vien		

Contents

1	General Description of Project	1
1.1	Introduction to the Ta Quang Buu Library	1
1.2	Drivers for Building Software.....	1
1.3	Objectives.....	2
1.4	Project Scopes	2
1.5	Team Members' Contribution	3
2	Software Requirements Specifications.....	4
2.1	Functional Requirements	4
2.1.1	Functional Requirements for Hust Student	4
2.1.2	Functional Requirements for Librarian	4
2.2	Non-Functional Requirements	5
2.4	Overall Use Case Diagram.....	6
2.5	List of Actors.....	8
2.6	List of Use Cases.....	8
2.7	Bussiness Rules	9
2.8	Use Case Specification.....	9
2.8.1	Register new account.....	9
2.8.2	Log in.....	11
2.8.3	Reset password	12
2.8.4	Change password.....	13
2.8.5	Log out.....	15
2.8.6	View profile.....	15
2.8.7	Edit profile details	16
2.8.8	View notifications	18
2.8.9	Ask for borrowing book	19
2.8.10	View all personal requests.....	20
2.8.11	Seach personal requests.....	21
2.8.12	Cancel personal requests	22
2.8.13	Add new book.....	24
2.8.14	Remove book.....	25
2.8.15	Update book details	26
2.8.16	Search book	28
2.8.17	Add reader	29
2.8.18	Remove reader.....	31
2.8.19	Update reader.....	32

2.8.20	Search reader	33
2.8.21	Ban reader.....	34
2.8.22	Unban reader	35
2.8.23	View history of borrowing and returning book.....	36
2.8.24	Issue book.....	38
2.8.25	View requests to borrow book.....	39
2.8.26	Seach requests to borrow book.....	40
2.8.27	Return book	41
2.8.28	View reports and statistics.....	43
3	Software Design and Implementation	45
3.1	System Architecture.....	45
3.1.1	MVC Model.....	45
3.1.2	Repository Pattern	45
3.1.3	Singleton Pattern	46
3.2	Activity Diagram.....	47
3.2.1	Activity Diagram for Register new account.....	47
3.2.2	Activity Diagram for Log in.....	48
3.2.3	Activity Diagram for Reset password	49
3.2.4	Activity Diagram for Change password.....	50
3.2.5	Activity Diagram for Log out.....	51
3.2.6	Activity Diagram for View profile	52
3.2.7	Activity Diagram for Edit profile details	53
3.2.8	Activity Diagram for View notifications	54
3.2.9	Activity Diagram for Ask for borrowing book	55
3.2.10	Activity Diagram for View all personal requests.....	56
3.2.11	Activity Diagram for Search personal requests.....	57
3.2.12	Activity Diagram for Cancel personal requests	58
3.2.13	Activity Diagram for Add new book.....	59
3.2.14	Activity Diagram for Remove book	60
3.2.15	Activity Diagram for Update book details	61
3.2.16	Activity Diagram for Search book	62
3.2.17	Activity Diagram for Add reader	63
3.2.18	Activity Diagram for Remove reader.....	64
3.2.19	Activity Diagram for Update reader.....	65
3.2.20	Activity Diagram for Search reader	66
3.2.21	Activity Diagram for Ban reader.....	67

3.2.22	Activity Diagram for Unban reader.....	68
3.2.23	Activity Diagram for View history of borrowing and returning book....	69
3.2.24	Activity Diagram for Issue book	70
3.2.25	Activity Diagram for View requests to borrow book from reader.....	71
3.2.26	Activity Diagram for Search requests to borrow book from reader	72
3.2.27	Activity Diagram for Return book	73
3.2.28	Activity Diagram for View reports and statistics.....	74
3.3	Sequence Diagram	75
3.3.1	Sequence Diagram for Create new account (Sign Up)	75
3.3.2	Sequence Diagram for Login	76
3.3.3	Sequence Diagram for Reset Password.....	77
3.3.4	Sequence Diagram for Log Out.....	77
3.3.5	Sequence Diagram for View Profile Details	78
3.3.6	Sequence Diagram for Edit Profile Details	78
3.3.7	Sequence Diagram for Borrow Book	79
3.3.8	Sequence Diagram for View Personal Request.....	79
3.3.9	Sequence Diagram for Search Personal Request	80
3.3.10	Sequence Diagram for Cancel Personal Request	80
3.3.11	Sequence Diagram for Add Book	81
3.3.12	Sequence Diagram for Search Book	82
3.3.13	Sequence Diagram for Update Book Details	83
3.3.14	Sequence Diagram for Add Reader.....	84
3.3.15	Sequence Diagram for Remove Reader	84
3.3.16	Sequence Diagram for Update Reader Details.....	85
3.3.17	Sequence Diagram for Search Reader.....	86
3.3.18	Sequence Diagram for Ban/Unban Reader	86
3.3.19	Sequence Diagram for (Student) View History of Borrowing and Returning Book	87
3.3.20	Sequence Diagram for (Librarian) View History of Borrowing and Returning Book	87
3.3.21	Sequence Diagram for View Personal Request.....	88
3.3.22	Sequence Diagram for View Statistic	88
3.4	Database Design.....	89
3.4.1	users	89
3.4.2	readers.....	90
3.4.3	books.....	90
3.4.4	categories	90

3.4.5	authors	91
3.4.6	borrow.....	91
3.6	Package Diagram	92
4	Software Testing.....	93
4.1	Overall Description	93
4.1.1	Testing Levels	93
4.1.2	Test Plan	93
4.1.3	Test Scope.....	93
4.1.4	Test item	93
4.1.5	Testing Risk Register	94
4.1.6	Test Approach (Strategy)	95
4.1.7	Management and Metrics	96
4.1.8	Test Estimation.....	96
4.1.9	Test Deliverables	97
4.2	User Guides	110
4.2.1	Deliverables Package.....	110
4.2.2	Software Requirements	110
4.2.3	Installation Guide	110
5	Evaluation and Results	115
5.1	Evaluation of Library Management Software	115
5.1.1	Easy of Use.....	115
5.1.2	Effectiveness.....	115
5.1.3	Flexibility.....	115
5.2	Results of Implementing Library Management Software	115
5.2.1	Improved Management Efficiency.....	115
5.2.2	Reporting and Performance Analysis.....	115
5.3	Summary	116
6	References	117

Table of Tables

Table 1. Members' Contribution	3
Table 2. List of Actors	8
Table 3. List of Use Cases	9
Table 4. List of Business Rules.....	9

Table of Figures

Figure 1. Overall Use Case Diagram	6
Figure 2. Use Case Diagram for Hust Student.....	7
Figure 3. Use Case Diagram for Librarian	7
Figure 4. Activity Diagram for UC-01	47
Figure 5. Activity Diagram for UC-02	48
Figure 6. Activity Diagram for UC-03	49
Figure 7. Activity Diagram for UC-04	50
Figure 8. Activity Diagram for UC-05	51
Figure 9. Activity Diagram for UC-06	52
Figure 10. Activity Diagram for UC-06	53
Figure 11. Activity Diagram for UC-08	54
Figure 12. Activity Diagram for UC-09	55
Figure 13. Activity Diagram for UC-10	56
Figure 14. Activity Diagram for UC-11	57
Figure 15. Activity Diagram for UC-12	58
Figure 16. Activity Diagram for UC-13	59
Figure 17. Activity Diagram for UC-14	60
Figure 18. Activity Diagram for UC-15	61
Figure 19. Activity Diagram for UC-16	62
Figure 20. Activity Diagram for UC-17	63
Figure 21. Activity Diagram for UC-18	64
Figure 22. Activity Diagram for UC-19	65
Figure 23. Activity Diagram for UC-20	66
Figure 24. Activity Diagram for UC-21	67
Figure 25. Activity Diagram for UC-22	68
Figure 26. Activity Diagram for UC-23	69
Figure 27. Activity Diagram for UC-24	70
Figure 28. Activity Diagram for UC-25	71
Figure 29. Activity Diagram for UC-26	72
Figure 30. Activity Diagram for UC-27	73
Figure 31. Activity Diagram for UC-28	74
Figure 32. Sequence Diagram for Create New Account (Sign Up)	75
Figure 33. Sequence Diagram for Log in	76
Figure 34. Sequence Diagram for Reset Password.....	77
Figure 35. Sequence Diagram for Log Out.....	77
Figure 36. Sequence Diagram for View Profile Details	78
Figure 37. Sequence Diagram for Update Profile Details	78
Figure 38. Sequence Diagram for Borrow Book	79
Figure 39. Sequence Diagram for View Personal Request.....	79
Figure 40. Sequence Diagram for Seach Personal Request.....	80
Figure 41. Sequence Diagram for Cancel Personal Request	80
Figure 42. Sequence Diagram for Add Book	81
Figure 43. Sequence Diagram for Search Book	82
Figure 44. Sequence Diagram for Update Book Details	83
Figure 45. Sequence Diagram for Add Reader.....	84
Figure 46. Sequence Diagram for Remove Reader	84

Figure 47. Sequence Diagram for Update Reader Details.....	85
Figure 48. Sequence Diagram for Search Reader.....	86
Figure 49. Sequence Diagram for Ban/Unban Reader	86
Figure 50. Sequence Diagram for (Student) View History of Borrowing and Returning Book	87
Figure 51. Sequence Diagram for (Librarian) View History of Borrowing and Returning Book	87
Figure 52. Sequence Diagram for View Personal Request.....	88
Figure 53. Sequence Diagram for View Statistic	88
Figure 54. Database Diagram	89
Figure 55. Package Diagram.....	92

Preface

With the rapid growth of information technology and communication industries worldwide and in our country, it has greatly pushed the process of digitalization in the country. Many software and IT applications have been applied effectively in real life. This makes our tasks get done quickly, efficiently, and saves time. Especially in management, computer management software has helped people manage scientifically, accurately, and save costs. We no longer need large storage spaces for books, documents, etc. In general, computers have replaced much of human labor.

I, along with my team, aim to bring technological advances to the Vietnamese people. Specifically, we, as HUSTers, would like to leverage these technological advances to bring convenience and professionalism to our university, Hanoi University of Science and Technology (HUST).

It is not difficult to realize that Ta Quang Buu Library is a symbol of HUST University and plays a major role in every aspect of the university. Therefore, if we apply technology to this library, helping it provide more meaningful services, we will create significant social value. This is the reason why my team and I have selected the topic "**Developing a Library Management Software for HUST.**" Due to the limited time to carry out the project and my lack of deep management experience, there may be some mistakes and shortcomings in the content presented. We sincerely hope to receive the support and feedback from teachers and friends to improve our project.

We also would like to express our sincere gratitude to our teacher, Dr. Du Dinh Vien, who directly and kindly guided and helped us complete this final project of our major.

1 General Description of Project

1.1 Introduction to the Ta Quang Buu Library

The Ta Quang Buu Library is the primary academic library of Hanoi University of Science and Technology, serving thousands of students, faculty, and researchers. It is a vital resource center, offering a vast collection of books, journals, digital resources, and research materials across various academic disciplines. The library plays a crucial role in supporting the university's educational and research missions by providing access to a wide range of information resources and fostering a culture of learning and inquiry.

As the university grows and the demand for library services increases, managing the library's operations using traditional methods has become increasingly challenging. The need for an efficient and modern system to manage the library's resources, streamline operations, and enhance user experience has become evident, prompting the development of the Library Management System.

1.2 Drivers for Building Software

The decision to develop this software is driven by several key factors that highlight the need for a modern and efficient library management system. Firstly, the increasing number of students and faculty at HUST has led to a growing demand for library services. The traditional methods of managing the library's vast collection of resources are becoming insufficient, resulting in longer wait times, difficulties in locating materials, and challenges in managing user records. By developing a comprehensive software solution, we aim to address these challenges by streamlining library operations, thereby improving the overall efficiency and user experience.

Secondly, the integration of digital resources into the library is essential in the modern educational landscape. As more academic resources become available online, the need for a system that can seamlessly manage both physical and digital collections is more pressing than ever. The proposed library management software will enable the Ta Quang Buu Library to efficiently manage digital resources, making them more accessible to students and faculty. This integration will not only enhance the library's service offerings but also align with HUST's broader goals of digital transformation.

Another significant driver for developing this software is the need to enhance the accuracy and reliability of library data management. Currently, the manual processes involved in cataloging, tracking, and updating library records are prone to errors. These errors can lead to mismanagement of resources, lost materials, and inaccurate reporting. The new software will automate these processes, reducing the likelihood of human error and ensuring that the library's data is always up-to-date and accurate. This will, in turn, enable better decision-making and resource allocation within the library.

Furthermore, the development of this software is motivated by the desire to foster a more engaging and interactive user experience. By incorporating features such as online book reservations, automated notifications, and personalized user accounts, the software will make it easier for students and faculty to interact with the library. This will encourage more frequent use of the library's resources and contribute to a more vibrant academic community at HUST.

In summary, the development of library management software for the Ta Quang Buu Library is driven by the need to improve operational efficiency, integrate digital resources, enhance data accuracy, and provide a better user experience. By addressing these key drivers, the software will not only meet the current needs of the library but also position it to adapt to future challenges and opportunities. This project represents a significant step towards modernizing the library and ensuring that it continues to play a vital role in supporting the academic and research activities of HUST.

1.3 Objectives

The primary objective of the Library Management System project is to create a comprehensive software solution that automates and improves the management of library resources at the Ta Quang Buu Library. The system aims to simplify and streamline the processes of tracking users, issuing, returning, and tracking library materials. By implementing the Library Management System, the library seeks to enhance its operational efficiency, provide better services to users, and integrate digital and physical resources more effectively.

1.4 Project Scopes

The project aims to build a Library Management System, which only serves for two types of users:

- Hust Student: the student is working at Ha Noi University of Science and Technology (HUST)
- Librarian: the librarian is working at Ta Quang Buu Library

Additionally, the project will focus on studying the core business process of Ta Quang Buu Library, including

- Managing borrowing/returning book
- Managing readers
- Managing books

1.5 Team Members' Contribution

No.	Full Name	Role	Contributed Task
1	Ta Duc Duy	Leader	Collect input requirements, write the report
2	Tran Hai Minh	Coder	Implement use cases: UC-09, UC-10, UC-11, UC-12, UC-23, UC-24, UC-25, UC-26, UC-27, UC-28
3	Le Duc Tuyen	Tester	Design test cases
4	Truong Thanh Hung	Coder	Implement use cases: UC-13, UC-14, UC-15, UC-16, UC-17, UC-18, UC-19, UC-20, UC-21, UC-22, UC-23
5	Vu Duy Hao	Coder	Implement use cases: UC-01, UC-02, UC-03, UC-04, UC-05, UC-06, UC-07, UC-08

Table 1. Members' Contribution

2 Software Requirements Specifications

2.1 Functional Requirements

2.1.1 Functional Requirements for Hust Student

The students, who are working at Ha Noi University of Science and Technology (Hust), can use the following features.

- Register new account
- Log in
- Reset password
- Change password
- Log out
- View profile
- Edit profile details
- Search book
- Ask for borrowing book
- View history of borrowing and returning book
- View all personal request
- Cancel personal requests
- Search personal requests
- View notifications

2.1.2 Functional Requirements for Librarian

The librarians, who are working at Ta Quang Buu Library, can use the following features.

- Log in
- Reset password
- Change password
- Log out
- View profile
- Edit profile details
- Add new book
- Remove book
- Update book details
- Search book
- Add reader
- Remove reader
- Update reader
- Search reader
- Ban reader
- Unban reader

- View history of borrowing and returning book
- Issue book
- View all requests to borrow book from reader
- Search requests to borrow book from reader
- Return book
- View reports and statistics

2.2 Non-Functional Requirements

- The application must have a user-friendly interface
- The application must be easy to use for the users.
- The application must have high security
- The application must be easy to maintain

2.4 Overall Use Case Diagram

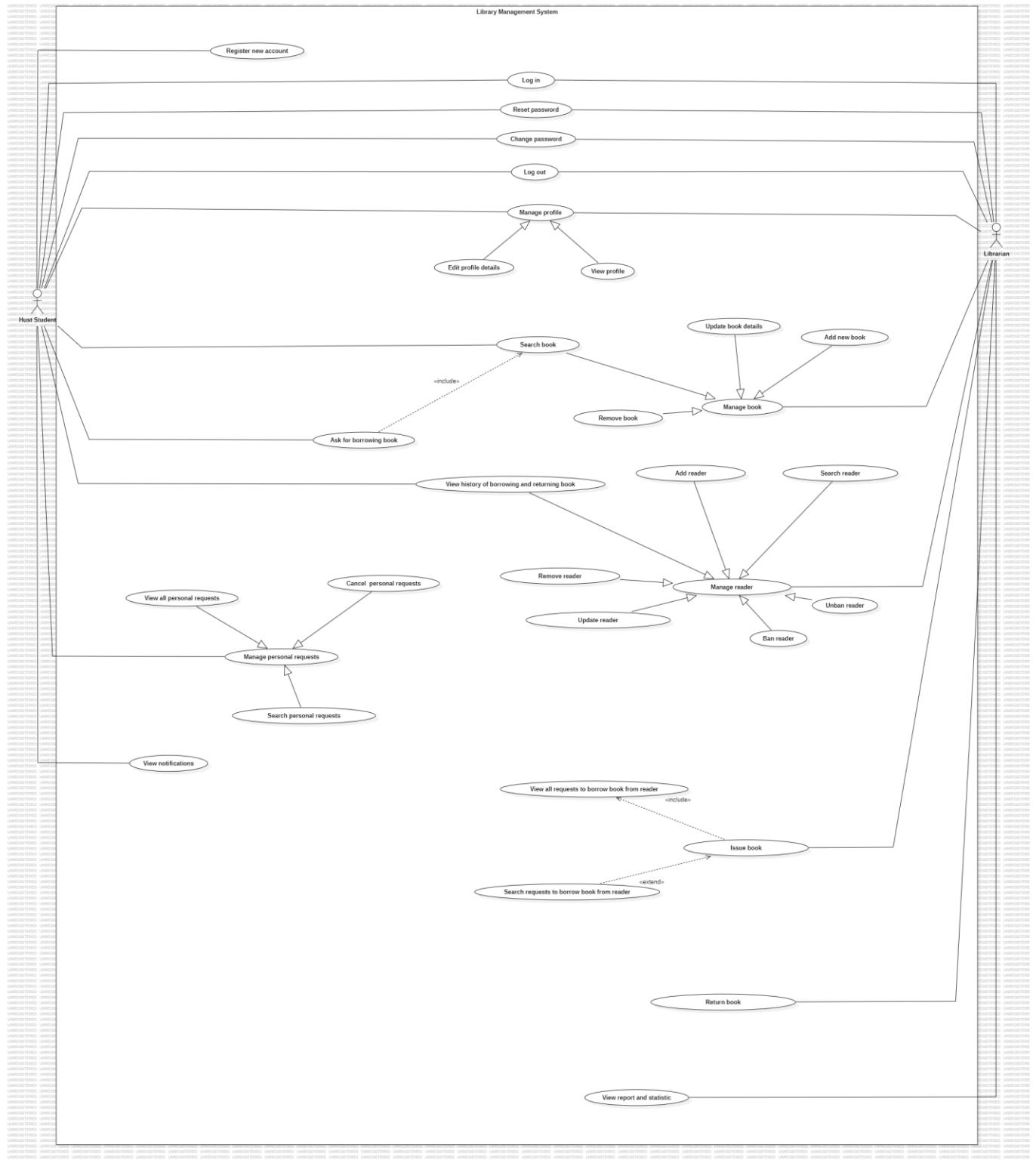


Figure 1. Overall Use Case Diagram



Figure 2. Use Case Diagram for Hust Student

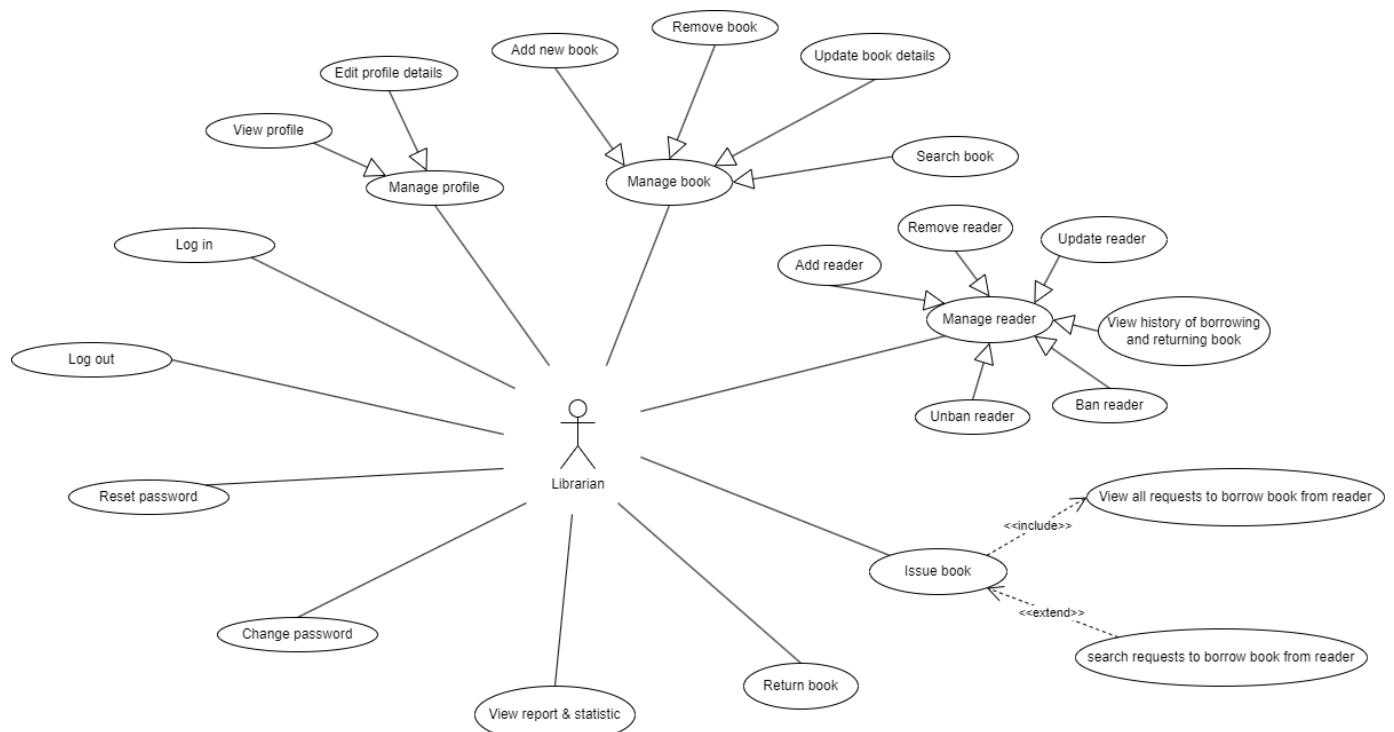


Figure 3. Use Case Diagram for Librarian

2.5 List of Actors

No.	Actor	Description
1	Hust Student	They are students who are learning at Ha Noi University of Science and Technology (HUST)
2	Librarian	They are librarians who are working at Ta Quang Buu Library

Table 2. List of Actors

2.6 List of Use Cases

ID	Use Case	Actor(s)
UC-01	Register new account	Hust Student
UC-02	Log in	Hust Student, Librarian
UC-03	Reset password	Hust Student, Librarian
UC-04	Change password	Hust Student, Librarian
UC-05	Log out	Hust Student, Librarian
UC-06	View profile	Hust Student, Librarian
UC-07	Edit profile details	Hust Student, Librarian
UC-08	View notifications	Hust Student
UC-09	Ask for borrowing book	Hust Student
UC-10	View all personal requests	Hust Student
UC-11	Search personal requests	Hust Student
UC-12	Cancel personal requests	Hust Student
UC-13	Add new book	Librarian
UC-14	Remove book	Librarian
UC-15	Update book details	Librarian
UC-16	Search book	Hust Student, Librarian
UC-17	Add reader	Librarian

UC-18	Remove reader	Librarian
UC-19	Update reader	Librarian
UC-20	Search reader	Librarian
UC-21	Ban reader	Librarian
UC-22	Unban reader	Librarian
UC-23	View history of borrowing and returning book	Hust Student, Librarian
UC-24	Issue book	Librarian
UC-25	View requests to borrow book from reader	Librarian
UC-26	Search requests to borrow book from reader	Librarian
UC-27	Return book	Librarian
UC-28	View reports and statistics	Librarian

Table 3. List of Use Cases

2.7 Business Rules

ID	Description
BR-01	Using a maximum of 30 characters for the Name of user
BR-02	Using a maximum of 30 characters for Email
BR-03	Using a maximum of 11 digits (0-9) for Phone Number
BR-04	Using a maximum of 100 characters for the Address
BR-05	Using a maximum of 100 characters for Username
BR-06	Using a maximum of 100 characters for Password
BR-07	The new password (sent in Gmail) contains 8 characters long
BR-08	The maximum length of the new password is 1000 characters
BR-09	Using a maximum of 100 characters for the book's title
BR-10	Using a maximum of 30 characters for the book's category
BR-11	Using a maximum of 30 characters for the book's author

Table 4. List of Business Rules

2.8 Use Case Specification

Use Case Specification	
Use Case No.:	UC-01
Use Case Name:	Register new account

Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student
Summary:	This feature applies only for Hust Student, but not for Librarian. This feature allows the user to create a new account.
Trigger:	The user clicks on “Create new account”
Preconditions:	1. The user does not have a registered account before
Post-conditions:	1. Success: The user creates a new account successfully. 2. Failure: The system displays an error message if the user provides incomplete or incorrect information.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The user clicks on “Create new account” in the homepage	The system displays the form, as below - Username - Password - Re-password - Fullname - Phone Number - Email - Address - Birth
2	The user fills out the given form	
3	The user submits the form by clicking on “Sign up”	The system validates the submitted form. [Exception 1] [Exception 2] [Exception 3]
4		The system displays the message “Sign up successfully”

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response

1	The user enters an existing Username	The system displays the message “Username already existed”, and then prompts the user to re-enter the form.
2	The user enters an existing Email	The system displays the message “Email already existed”, and then prompts the user to re-enter the form.
3	The user enters an existing Phone Number	The system displays the message “Phone Number already existed”, and then prompts the user to re-enter the form.

Business Rules:	<ul style="list-style-type: none"> • BR-01 • BR-02 • BR-03 • BR-04 • BR-05 • BR-06
------------------------	--

2.8.2 Log in

Use Case Specification	
Use Case No.:	UC-02
Use Case Name:	Log in
Created By:	Tạ Đức Duy
Date:	July 28 th , 2024
Actors:	Hust Student, Librarian
Summary:	This use case allows 2 types of users (Hust Student, Librarian) to log in to the library management system. The system verifies the user's credentials and grants access to the system's resources based on the user's role and permissions.
Trigger:	The user clicks on “Log in”
Preconditions:	The user has already registered an account.
Post-conditions:	<ol style="list-style-type: none"> 1. The user is authenticated 2. The user can now perform actions according to their role. 3. History of log-in is recorded.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The user clicks on “Log in” on the homepage	<p>The system asks the user to fill out the following form</p> <ul style="list-style-type: none"> - Role (Hust Student/ Librarian) - Username

		- Password
2	The user fills out the given form	
3	The user clicks on "Log in"	The system verifies the user account in the database. Upon successful authentication, the system directs the user to their respective dashboard based on their role. [Exception 1]

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The user enters an incorrect username or password.	The system displays an error message indicating the credentials are invalid and prompts the user to try again.

Business Rules:	<ul style="list-style-type: none"> BR-05 BR-06
-----------------	--

2.8.3 Reset password

Use Case Specification	
Use Case No.:	UC-03
Use Case Name:	Reset password
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student, Librarian
Summary:	This use case allows a user who has forgotten their password or wishes to change it to reset their password in the library management system. The user can choose to reset their password by verifying their identity through the registered email address.
Trigger:	The user clicks on "Forgot password"
Preconditions:	<ol style="list-style-type: none"> The user must have an active account in the library management system. The system must have the user's email address and verified.

Post-conditions:	<p>3. Success: The user's password is successfully reset, and they can log in with the new password.</p> <p>4. Failure: The system displays an error message if the password reset process cannot be completed (e.g., verification failure, system error).</p>
-------------------------	--

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The user selects the "Forgot Password" or "Reset Password" option on the login page.	The system prompts the user to enter the registered email.
2	The user enters the email	The system verifies this email [Exception 1]
3		The system sends a new 8-characters-long password to this email
4	The user checks the registered email and uses the new password sent into this email to login again.	

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The user enters an invalid or incorrect email address.	The system displays an error message and prompts the user to enter valid email.

Business Rules:	<ul style="list-style-type: none"> • BR-07
------------------------	---

2.8.4 Change password

Use Case Specification	
Use Case No.:	UC-04
Use Case Name:	Change password
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student, Librarian
Summary:	This use case allows a user to change their password in the library management system for security purposes.

Trigger:	The user clicks on “Change password” in the “System” section.
Preconditions:	1. The user must be logged into the system
Post-conditions:	1. Success: The user's password is successfully updated, and they can log in with the new password. 2. Failure: The system displays an error if the change fails (e.g., incorrect current password, new password doesn't meet requirements).

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The user clicks on “Change password” in the “System” section.	The system shows the following editable form - Enter current password - Enter newpassword - Re-enter newpassword
2	The user fills out the given form	
3	The user clicks on “OK”	The system verifies the submitted form. [Exception 1] [Exception 2] [Exception 3]
4		The system updates the password and displays a confirmation message.

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The user enters an incorrect current password.	The user shows message “Incorrect current password”
2	The new password doesn't meet the security requirements (e.g., too short, not complex enough).	The user is prompted to enter a valid password.
3	Entering a new password and Re-entering a new password doesn't match	The user shows the error message and prompts the user to re-enter the field “Re-entering new password”

Business Rules:	• BR-08
------------------------	---------

2.8.5 Log out

Use Case Specification	
Use Case No.:	UC-05
Use Case Name:	Log out
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student, Librarian
Summary:	This use case describes the process by which a user securely logs out of the library management system, ending their session.
Trigger:	The user indicates that they want to log out of the system.
Preconditions:	1. The user must be logged into the system.
Post-conditions:	1. Success: The user's session is terminated, and they are redirected to the login page or a confirmation page indicating they have logged out.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The user selects the "Log Out" option from the "System" section	The system redirects the user to the login page or displays a log-out confirmation page.

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	N/A	

Business Rules: N/A

2.8.6 View profile

Use Case Specification	
Use Case No.:	UC-06

Use Case Name:	View profile
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student, Librarian
Summary:	This use case allows a user to view their personal profile details, such as name, email, phone number, etc.
Trigger:	The user chooses to view their profile information by selecting the "Infomation" option on the homepage.
Preconditions:	1. The user must be logged in to the library management system.
Post-conditions:	1. Success: The user successfully views their profile information.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The user chooses to view their profile information by selecting the "Infomation" option on the homepage.	<p>The system retrieves the user's profile information from the database. Then the system displays the user's profile details, including</p> <ul style="list-style-type: none"> - Full name - ID - Birth - Address - Registered email - Registered phone number

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	N/A	
Business Rules:		N/A

2.8.7 Edit profile details

Use Case Specification

Use Case No.:	UC-07
Use Case Name:	Edit profile details
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student, Librarian
Summary:	This use case allows the user to update their personal profile details, such as name, email, phone number, and other relevant information in the library management system.
Trigger:	The user clicks on “Information” and he/she wants to edit the personal profile details.
Preconditions:	1. The user has logged in successfully.
Post-conditions:	1. Success: User’s profile details are updated and saved in the system. 2. Failure: The system displays an error message if the profile details cannot be updated (e.g., due to missing required fields or system error).

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The user clicks on “Information” in the homepage	<p>The system displays the current user information under the uneditable form, as belows</p> <ul style="list-style-type: none"> - Reader ID (un-editable) - Full Name (un-editable) - Address (un-editable) - Email (un-editable) - Phone Number (un-editable) - Birth (un-editable)
2	The user clicks on “Update information”	<p>The system shows current user information under editable form, as follows</p> <ul style="list-style-type: none"> - Reader ID (un-editable) - Full Name (editable) - Address (editable) - Email (editable) - Phone Number (editable) - Birth (editable)
3	The user edits necessary fields in the form	
4	The user submits the form by clicking on “Save”	<p>The system validates the input and updates the profile information in the database.</p> <p>[Exception 1]</p>

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	If input validation fails (e.g., incorrect email format)	The system displays an error message, and then asks the user to correct the input and resubmit.

Business Rules:	<ul style="list-style-type: none"> • BR-01 • BR-02 • BR-03 • BR-04 • BR-05 • BR-06
------------------------	--

2.8.8 View notifications**Use Case Specification**

Use Case No.:	UC-08
Use Case Name:	View notifications
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student
Summary:	This use case allows a user to view notifications related to their account activities, such as due date reminders, reservation confirmations, or system updates.
Trigger:	After immediately the user logged in to the system.
Preconditions:	1. User must be logged in.
Post-conditions:	1. Success: The user views notifications

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	After the user successfully logs in to the system	The system displays necessary notifications

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	N/A	

Business Rules: N/A

2.8.9 Ask for borrowing book

Use Case Specification

Use Case No.:	UC-09
Use Case Name:	Ask for borrowing book
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student
Summary:	This use case allows a user to send a request to borrow books to the librarian
Trigger:	The user clicks on “Available book” (search book function) and he/she wants to send a request to borrow some books
Preconditions:	1. The user must be logged in and verified as Hust Student
Post-conditions:	1. Success: The borrowing request is successfully sent and awaits approval by the librarian.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step .	Actor Action	System Response
1	The user clicks on “Available book” section	The system shows all available books
2		The system shows the search feature

3	The user searches for the desired books.	The system shows related books that are available to borrow.
4	The user selects a desired book	The system asks the user to enter the date to return this book.
5	The user enters the date to return the book	
6	The user clicks on “Request borrow”	The system displays the confirmation message “Your request is sent” [Exception 1]

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The user enters the invalid date to return the book.	The system displays an error message highlighting the issues and prompts the user to correct them.

Business Rules: N/A

2.8.10 View all personal requests

Use Case Specification	
Use Case No.:	UC-10
Use Case Name:	View all personal requests
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student
Summary:	This feature allows the reader to view his/her history of requests to borrow the book that are being waiting for the librarian's response.
Trigger:	The user clicks on “Manage personal requests” on the homepage, and he/she wants to view all his/her requests (that are being waiting for the librarian's response)
Preconditions:	1. The user must be logged in and verified as Hust Student
Post-conditions:	1. Success: History of requests including requests to borrow the book and requests to renew the book.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response												
1	The student clicks on “Manage personal requests” on the homepage	<p>The system displays the history of requests to borrow the book. For example,</p> <table border="1"> <thead> <tr> <th colspan="4">Requests to borrow the book</th> </tr> <tr> <th>Book ID</th> <th>Title</th> <th>Start Borrow Date</th> <th>Due Date</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Requests to borrow the book				Book ID	Title	Start Borrow Date	Due Date				
Requests to borrow the book														
Book ID	Title	Start Borrow Date	Due Date											

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	N/A	

Business Rules: N/A

2.8.11 Search personal requests

Use Case Specification	
Use Case No.:	UC-11
Use Case Name:	Search personal requests
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student
Summary:	This feature allows the reader to search his/her requests that were made before
Trigger:	The user clicks on “Manage personal requests” on the homepage, and he/she wants to search for requests.
Preconditions:	1. The user must be logged in and verified as Hust Student

Post-conditions:	1. Success: The history of related requests is shown			
Main Success Scenario/Main Flow/Normal Flow/Main Path:				
Step.	Actor Action	System Response		
1	The student clicks on “Manage personal request” on the homepage	The system prompts the student to enter a keyword to search		
2	The student enters a keyword	The history of related requests is shown. [Exception 1]		
Alternative Flows:				
No.	Actor Action	System Response		
1	N/A			
Exceptions:				
No.	Actor Action	System Response		
1	If no request found	The system displays the messages “No request found”, and prompts the user to enter another keyword.		
Business Rules:	N/A			
2.8.12 Cancel personal requests				
Use Case Specification				
Use Case No.:	UC-12			
Use Case Name:	Cancel personal requests			
Created By:	Ta Duc Duy			
Date:	July 28 th , 2024			
Actors:	Hust Student			
Summary:	This feature allows the reader to cancel one or more than one requests that were made before.			
Trigger:	The user clicks on “Manage personal requests” on the homepage, and he/she wants to cancel some requests that were already sent to the librarian.			
Preconditions:	1. The user must be logged in and verified as Hust Student 2. The user made some requests before			

Post-conditions:	1. Success: The selected requests are canceled 2. Failure: The system shows the message “No request to cancel” if no request is found.
-------------------------	---

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response												
1	The student clicks on “Manage personal requests” on the homepage	<p>The system all requests that were made before. For example,</p> <table border="1"> <tr> <td colspan="4">Requests to borrow the book</td> </tr> <tr> <td>Book ID</td><td>Title</td><td>Start Borrow Date</td><td>Due Date</td> </tr> <tr> <td></td><td></td><td></td><td></td> </tr> </table>	Requests to borrow the book				Book ID	Title	Start Borrow Date	Due Date				
Requests to borrow the book														
Book ID	Title	Start Borrow Date	Due Date											
2	The student selects one or more than one desired requests to cancel [Alternative 1]													
3	The student clicks on “Cancel request”	The system displays the message “Do you want to cancel the request?”												
4	The student clicks on “Ok”	The system removes information of the selected requests from the database												
5		The system displays the confirmation message “Canceled request(s)”												

Alternative Flows:

No.	Actor Action	System Response
1	1. The student uses the search function to search for the desired requests.	The system shows all related requests [Exception 1]
	2. The student selects one or more than one desired requests to cancel	
	3. Come back to the step 3 in the main flow	

Exceptions:

No.	Actor Action	System Response
-----	--------------	-----------------

1	1. No request is found	The system shows the message “No request is found”, and then prompts the user to enter another keyword.
Business Rules:	N/A	

2.8.13 Add new book

Use Case Specification	
Use Case No.:	UC-13
Use Case Name:	Add new book
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows the librarian to add one or more than one book into the database
Trigger:	The librarian clicks on “Book management” and he/she wants to add new books to the database
Preconditions:	The user already logged in and was verified as a librarian’s role
Post-conditions:	1. Success: The new book is successfully added to the database. 2. Failure: The system displays an error message if the book cannot be added

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Book management” in the homepage	The system shows the following editable form: - Book ID (generated randomly and un-editable) - Title - Author - Category - Quantity - Published Date
2	The librarian fills out the given form	
3	The librarian clicks on “Add book”	The system validate the submitted form, and then the system informs “Added new book” [Exception 1]

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The librarian enters incomplete or incorrect book details.	The system displays an error message highlighting the issues and prompts the librarian to correct them.

Business Rules:	<ul style="list-style-type: none"> • BR-09 • BR-10 • BR-11
-----------------	---

2.8.14 Remove book

Use Case Specification

Use Case No.:	UC-14
Use Case Name:	Remove book
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows a librarian to remove all books whose ID is selected by the librarian
Trigger:	The librarian clicks on “Book management”, and he/she wants to remove all books basing on book ID.
Preconditions:	<ol style="list-style-type: none"> 1. The librarian must be logged in to the library management system. 2. The book must exist in the system.
Post-conditions:	<ol style="list-style-type: none"> 1. Success: The books are successfully removed from the system 2. Failure: The system displays an error message if the book cannot be removed (e.g., it does not exist in the system).

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Book management” option in the homepage	The system displays all books

2	The librarian searches for the book in the system using the search function.	The system displays all related book details [Exception 1]
3	The librarian selects a desired book among searched books.	
4	The librarian clicks on “Remove book”.	The system prompts the librarian to confirm the removal.
5	The librarian confirms the removal. [Alternative 1]	The system removes all books whose ID is selected by the librarian.

Alternative Flows:

No.	Actor Action	System Response
1	The librarian cancels the removal process.	The system does not remove the book and returns to the previous screen.

Exceptions:

No.	Actor Action	System Response
1	The librarian attempts to search for a book that does not exist in the system.	The system displays an error message stating that the book cannot be found.

Business Rules:	N/A
-----------------	-----

2.8.15 Update book details

Use Case Specification	
Use Case No.:	UC-15
Use Case Name:	Update book details
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This use case allows a librarian to update the details of an existing book in the library system. This action is performed when there are changes in the book's information, such as a new edition, updated author details, or correction of errors.
Trigger:	The librarian clicks on “Book management”, and indicates that he/she want to update the details of an existing book
Preconditions:	<ol style="list-style-type: none"> 1. The librarian must be logged in to the library management system. 2. The book must already exist in the system.

Post-conditions:	1. Success: The book's details are successfully updated in the system. 2. Failure: The system displays an error message if the details cannot be updated (e.g., invalid information, system error).
-------------------------	--

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on "Book management"	The system shows all books
2	The librarian searches for the book they wish to update in the system.	The system displays the related books [Exception 1]
3	The librarian selects the book he/she wants to edit.	The system provides the edit-book-details Form (the information of the selected book is automatically put into this form by default), as below - Book ID (un-editable) - Title - Author - Category - Quantity - Published Date
4	The librarian updates the necessary fields by filling out the given form.	
5	The librarian clicks on "Update book"	The system displays a confirmation message. [Exception 2]
6	The librarian confirms updating the book by clicking on "Ok" [Alternative 1]	The system validates the submitted form, then the system updates the book details in the database, and then sends the message "Updated book".

Alternative Flows:

No.	Actor Action	System Response
1	The librarian cancels the update before final confirmation.	The system does not update the book details and returns to the previous screen.

Exceptions:

No.	Actor Action	System Response
1	The librarian attempts to update a book that does not exist in the system.	The system displays an error message stating that the book cannot be found.

2	The librarian enters incomplete or invalid details while updating the book.	The system displays an error message highlighting the issues and prompts the librarian to correct them.
Business Rules:	<ul style="list-style-type: none"> • BR-09 • BR-10 • BR-11 	

2.8.16 Search book

Use Case Specification	
Use Case No.:	UC-16
Use Case Name:	Search book
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student, Librarian
Summary:	This use case allows users (librarians, students) to search for books within the library system.
Trigger:	The user clicks on “Book management” (if librarian) or “Available book” (if student), and indicates that he/she want to find a specific book or books matching certain criteria.
Preconditions:	1. The user must have access to the library management system.
Post-conditions:	1. Success: The system displays a list of books that match the search criteria. 2. Failure: The system displays a message indicating no books were found matching the search criteria.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	<ul style="list-style-type: none"> • Librarian actions: The librarian clicks on “Book management” in the homepage • Student actions: The student clicks on “Available book” in the home page 	The system displays the search interface with various search criteria fields.
2	The user enters the desired search criteria	The system validates the entered search criteria. [Exception 1]
3	The user submits the search request.	The system searches the library's database for books that match the

		criteria. If found, then the system shows the search results, including details like title, author, availability, etc. If not found, the system shows message "not found".
--	--	--

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The user enters invalid or incomplete search criteria.	The system prompts the user to correct the input and resubmit the search.

Business Rules:	N/A
-----------------	-----

2.8.17 Add reader

Use Case Specification	
Use Case No.:	UC-17
Use Case Name:	Add reader
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows the librarian to add a new reader (Hust Student) to the system. At the same time, the system creates a new account for this new reader.
Trigger:	The librarian clicks on "Reader management", and he/she wants to add a new Hust Student.
Preconditions:	User must be logged in and verified Librarian
Post-conditions:	<ol style="list-style-type: none"> Success: A new reader information is added to the system Failure: System shows corresponding error messages

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on "Reader management"	The system shows the editable form, as below

		<ul style="list-style-type: none"> - Reader ID (generated randomly, and un-editable) - Username - Reader Name - Email - Phone Number - Address - Birth
2	The librarian uses the information of new reader to fill out the given form.	
3	The librarian clicks on “Add reader”	<p>The system validates the submitted form</p> <p>[Exception 1]</p> <p>[Exception 2]</p> <p>[Exception 3]</p>
4		<p>The system creates a new account for this new reader, based on the input Username (password is generated randomly)</p>
5		The system shows message “Add reader”

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The librarian entered the username that already existed	The system shows the message “The username existed”
2	The librarian entered the email that already existed	The system shows the message “The email existed”
3	The librarian entered the phone number that already existed	The system shows the message “The phone number existed”

Business Rules:	<ul style="list-style-type: none"> • BR-01 • BR-02 • BR-03 • BR-04 • BR-05
-----------------	---

2.8.18 Remove reader

Use Case Specification	
Use Case No.:	UC-18
Use Case Name:	Remove reader
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows the librarian to remove the reader (Hust Student) from the system. At the same time, the system removes the corresponding account of this reader.
Trigger:	The librarian clicks on “Reader management”, and he/she wants to remove every about a somewhat reader.
Preconditions:	The user must be logged in and verified Librarian
Post-conditions:	1. Success: Every about the reader is removed 2. Failure: System shows corresponding error messages

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Reader management”	The system shows all readers so far
2	The librarian selects one reader to remove [Alternative 1]	
3	The librarian clicks on “Remove reader”	The system shows the message “Do you want to remove this reader?”
4	The librarian clicks on “Ok”	The system removes everything about this reader from the database
5		The system shows the message “Removed reader”

Alternative Flows:

No.	Actor Action	System Response
1	The librarian uses the search function to search for the desired reader	The system shows all related readers [Exception 1]
2	Come back to the step 3 in the main flow	

Exceptions:

No.	Actor Action	System Response
1	If not found	The system shows the message "No result found"
Business Rules:		N/A

2.8.19 Update reader

Use Case Specification	
Use Case No.:	UC-19
Use Case Name:	Update reader
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows the librarian to update the reader's details including Name, Username, Phone Number, Email, etc.
Trigger:	The librarian clicks on "Reader management", and he/she wants to update the reader's details
Preconditions:	The user must be logged in and verified Librarian
Post-conditions:	1. Success: A new reader information is added to the system 2. Failure: System shows corresponding error messages

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on "Reader management"	The system shows the editable form, as below - Reader ID (un-editable) - Username - Reader Name - Email - Phone Number - Address - Birth
2	The librarian user search function to search for the desired reader	The system shows all related readers
3	The librarian selects one reader to edit	The system automatically fills out the given form by the selected reader's information

4	The librarian edits necessary fields in the given form	
5	The librarian clicks on “Update reader”	The system validates the submitted form [Exception 1] [Exception 2] [Exception 3]
6		The system updates the database
7		The system shows the message “Updated reader”

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The librarian entered the username that already existed	The system shows the message “The username existed”
2	The librarian entered the email that already existed	The system shows the message “The email existed”
3	The librarian entered the phone number that already existed	The system shows the message “The phone number existed”

Business Rules:	<ul style="list-style-type: none"> • BR-01 • BR-02 • BR-03 • BR-04 • BR-05
-----------------	---

2.8.20 Search reader

Use Case Specification	
Use Case No.:	UC-20
Use Case Name:	Search reader
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian

Summary:	This feature allows the librarian to search readers, based on many criteria.
Trigger:	The librarian clicks on “Reader management”, and he/she wants to search for some readers
Preconditions:	The user must be logged in and verified Librarian
Post-conditions:	<ol style="list-style-type: none"> 1. Success: The related readers are shown 2. Failure: No reader found

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Reader management”	The system shows all readers so far
2	The librarian enters a keyword to search for a reader's details	The system shows all related readers [Exception 1]

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	If not found	The system shows the message “No result found”

Business Rules:	N/A
-----------------	-----

2.8.21 Ban reader

Use Case Specification	
Use Case No.:	UC-21
Use Case Name:	Ban reader
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows the librarian to block a user account. After the user account is blocked, the user cannot log in or cannot access the system anymore.
Trigger:	The librarian clicks on “Reader management”, and he/she wants to ban a user account.

Preconditions:	The user must be logged in and verified Librarian
Post-conditions:	1. Success: The account of the selected user is blocked 2. Failure: System shows corresponding error messages

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Reader management”	The system shows all readers so far
2	The librarian selects one reader to ban [Alternative 1]	The system shows the state of ban/unban of this reader
3	The librarian clicks on “Ban reader”	The system shows the message “Do you want to ban this reader?”
4	The librarian clicks on “Ok”	The system does not allow the account of this banned user to access to the system anymore.
5		The system shows the message “Banned reader”

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	N/A	

Business Rules: N/A

2.8.22 Unban reader

Use Case Specification	
Use Case No.:	UC-22
Use Case Name:	Unban reader
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian

Summary:	This feature allows the librarian to unblock a user account. After the user account is unblocked, the user can access the system as norm.
Trigger:	The librarian clicks on “Reader management”, and he/she wants to ban a user account.
Preconditions:	The user must be logged in and verified Librarian
Post-conditions:	3. Success: The account of the selected user is blocked 4. Failure: System shows corresponding error messages

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Reader management”	The system shows all readers so far
2	The librarian selects one reader to ban [Alternative 1]	The system shows the state of ban/unban of this reader
3	The librarian clicks on “Ban reader”	The system shows the message “Do you want to unban this reader?”
4	The librarian clicks on “Ok”	The system does not allow the account of this banned user to access to the system anymore.
5		The system shows the message “The reader is unbanned”

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	N/A	
Business Rules:		N/A

2.8.23 View history of borrowing and returning book

Use Case Specification	
Use Case No.:	UC-23
Use Case Name:	View history of borrowing and returning book

Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Hust Student, Librarian
Summary:	<ol style="list-style-type: none"> This feature allows students to view their history of borrowing and returning books (e.g. Borrow Date, Due Date, Return Date) This feature allows the librarian to view the history of borrowing and returning books of readers (e.g. Borrow Date, Due Date, Return Date)
Trigger:	<ol style="list-style-type: none"> The student clicks on “Available book”, and he/she wants to view the history of borrowing and returning books. The librarian clicks on “Reader management”, and he/she wants to view the history of borrowing and returning books of all readers.
Preconditions:	The user already logged in to the system
Post-conditions:	<ol style="list-style-type: none"> The student views his/her history of borrowing and returning books. The librarian views the history of borrowing and returning books of all readers.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action (Hust Student)	System Response
1	The student clicks on “History borrow”	<p>The system shows the history of borrowing and returning books, including</p> <ul style="list-style-type: none"> - Book ID - Book Title - Start-Borrow Date - Due Date - Return Date

Step.	Actor Action (Librarian)	System Response
1	The librarian clicks on “Reader management”	The system shows all readers
2	The librarian selects one reader to view his/her history of borrowing and returning book. [Alternative 1]	
3	The librarian clicks on “History borrow”	<p>The system shows the history of borrowing and returning books of the selected reader, including</p> <ul style="list-style-type: none"> - Book ID - Book Title - Start-Borrow Date - Due Date - Return Date

Alternative Flows:

No.	Actor Action	System Response
1	1. The librarian uses the search functions to search for the desired reader	The system shows all related readers [Exception 1]
	2. Come back to the step 3 in the main flow	
Exceptions:		
No.	Actor Action	System Response
1	If not found	The system shows the message “No result found” and prompts the librarian enter new keyword
Business Rules:		N/A

2.8.24 Issue book

Use Case Specification	
Use Case No.:	UC-24
Use Case Name:	Issue book
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows the librarian to issue books for corresponding readers, based on their requests to borrow books.
Trigger:	The librarian clicks on “Issue Book” on the homepage, and he/she wants to issue books for corresponding readers, based on their requests to borrow books.
Preconditions:	<ol style="list-style-type: none"> 1. The librarian already logged in to the system 2. The reader must send his/her requests to borrow books before.
Post-conditions:	<ol style="list-style-type: none"> 1. The reader borrows the book successfully 2. The history of borrowing books is recorded

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Issue book” on the homepage	The system shows all requests to borrow books from readers
2	The librarian selects one request [Alternative 1]	

3	The librarian clicks on “Approve” [Alternative 2]	The system sends a confirmation message to the corresponding reader’s email. For example, “Book Name, ID, has been approved by the librarian. Please go to the librarian to take the book”
4		The system shows the message “Issued book successfully”

Alternative Flows:

No.	Actor Action	System Response
1	1. The librarian uses the search function to search for the desired request	The system shows all related requests [Exception 1]
	2. Come back to the step 3 in the main flow	
2	The librarian clicks on “Reject”	The system sends a rejection message to the corresponding reader’s email. For example, “Book Name, ID, has been rejected.”

Exceptions:

No.	Actor Action	System Response
1	If not found	The system shows the message “No corresponding request found”, and prompts the librarian to enter another keyword.

Business Rules:	N/A
-----------------	-----

2.8.25 View requests to borrow book

Use Case Specification	
Use Case No.:	UC-25
Use Case Name:	View requests to borrow book
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows the librarian to view all requests to borrow books from the readers

Trigger:	The librarian clicks on “Issue Book” on the homepage, and he/she wants to view all requests to borrow books from the readers
Preconditions:	1. The librarian already logged in to the system
Post-conditions:	1. The librarian can view all requests to borrow books from the readers.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Issue book” on the homepage	<p>The system shows all requests to borrow books from readers. The information of each request should include</p> <ul style="list-style-type: none"> - Reader ID - Reader Name - Book ID - Book Title - Borrow Date - Due Date

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	N/A	

Business Rules: N/A

2.8.26 Search requests to borrow book

Use Case Specification	
Use Case No.:	UC-26
Use Case Name:	Search requests to borrow book
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian

Summary:	This feature allows the librarian to search for requests to borrow books from the readers by various types of keywords.
Trigger:	The librarian is in the “Issue book” section, and he/she wants to search for requests to borrow books from the readers by various types of keywords
Preconditions:	1. The librarian already logged in to the system
Post-conditions:	2. The librarian can view all desired requests

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “Issue book” on the homepage	The system shows the search-request function
2	The librarian enters a keyword to search for the desired requests	The system shows all related requests [Exception 1]

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	If not found	The system shows the message “No corresponding request found”, and then prompts the librarian to enter another keyword.

Business Rules:	N/A
------------------------	-----

2.8.27 Return book

Use Case Specification	
Use Case No.:	UC-27
Use Case Name:	Return book
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian

Summary:	This feature allows the librarian to return a book. The borrowers will bring their borrowed books to the librarian. After that, the librarian makes confirmation and return the book to the library's inventory.
Trigger:	The librarian clicks on "Return book", and he/she wants to return a book, which is returned by the borrower.
Preconditions:	<ol style="list-style-type: none"> 1. The librarian already logged in to the system 2. The borrower gives the librarian his/her information
Post-conditions:	<ol style="list-style-type: none"> 1. Success: The librarian confirms that the borrower returns the books successfully 2. Failure: The system shows corresponding error messages.

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on "Return book" on the homepage	The system asks the librarian to enter the information of the book returned, and the information of the reader who returns the book.
2	The librarian enters information of the book returned, and the information of the reader who returns the book.	
3	The librarian clicks on "Return book"	The system verifies information of the book returned, and the information of the reader who returns the book. [Exception 1]

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	The information of the returner and the information of the book returned do not match	The system shows the message "Returned book and Returner do not match", then the system prompts the librarian to enter again.
Business Rules:		N/A

2.8.28 View reports and statistics

Use Case Specification	
Use Case No.:	UC-28
Use Case Name:	View reports and statistics
Created By:	Ta Duc Duy
Date:	July 28 th , 2024
Actors:	Librarian
Summary:	This feature allows a librarian to view statistics including the number of readers, available books, books currently borrowed, overdue readers, and return-book-on-time readers.
Trigger:	The librarian clicks on “View statistics”
Preconditions:	1. The librarian must be logged in to the library management system.
Post-conditions:	1. The system shows the number of available books 2. The system shows the number of books currently borrowed 3. The system shows the number of overdue readers 4. The system shows the number of return-book-on-time readers

Main Success Scenario/Main Flow/Normal Flow/Main Path:

Step.	Actor Action	System Response
1	The librarian clicks on “View statistics” option in the homepage	The system displays the following statistics 1. The system shows the number of available books 2. The system shows the number of books currently borrowed 3. The system shows the number of overdue readers 4. The system shows the number of return-book-on-time readers

Alternative Flows:

No.	Actor Action	System Response
1	N/A	

Exceptions:

No.	Actor Action	System Response
1	N/A	

Business Rules:	N/A
------------------------	-----

3 Software Design and Implementation

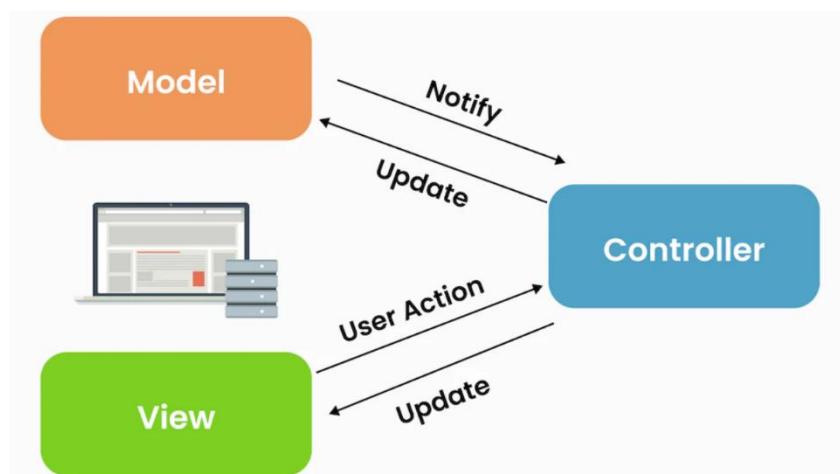
3.1 System Architecture

3.1.1 MVC Model

In this Library Management Project, we use the **MVC (Model – View – Control)** model to build the code. The **Model-View-Controller** model is a software architectural design used to separate an application into three interconnected components that are **Model**, **View** and **Control**. We will interact with the **View**, **Model** will be the object store the data and the **Control** will interact between the **View** and the **Model**. We chose this model because it has a lot of advantages:

- Separate the model (dealt with once at a time):
 - **Model**: handling the object and data.
 - **View**: Handling the UI we interacting with.
 - **Control**: Processing and controlling the interaction between View and Model.

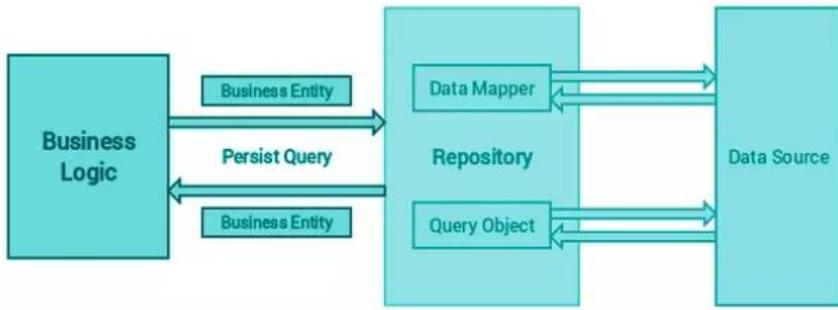
This helps in organizing the code effectively.



- Easy to be maintained.
- Easy for testing. Model can be tested independently and the same for others two.
- Easy to improve the Project.

3.1.2 Repository Pattern

In the Project we applied the Repository Pattern which is the between layer of processing logic and data operation.



We choose this Pattern because of its advantages:

- **Separation of Concerns:** it separates data access logic from business logic, making the codebase cleaner and more manageable.
- **Maintainability:** Centralizing data access logic in repositories simplifies maintenance and update, as changes are confined to a single location.
- **Consistency:** Ensures a consistent data access strategy across the application, promoting code reuse and reducing duplication.

3.1.3 Singleton Pattern

Singleton pattern is a design that ensures a class has only one instance throughout the application and provides a global point of access to that instance.

The key components:

1. **Private Static Instance:** A static variable to hold the single instance of the class.
2. **Private Constructor:** A constructor that is private to prevent direct instantiation from outside the class.
3. **Public Static method:** A method that provides access to the single instance, creating it if necessary.

We chose to use this pattern because of:

- **Reduced Memory Usage:** Minimizes memory consumption by avoiding the creation of multiple instances.
- **Consistency:** Guarantees that all parts of the application work with the same instance, ensuring consistency of state or configuration.
- **Global Access Point:** Provides a single, global point of access to the instance, which simplifies interactions with the object from different parts of the application.

3.2 Activity Diagram

3.2.1 Activity Diagram for Register new account

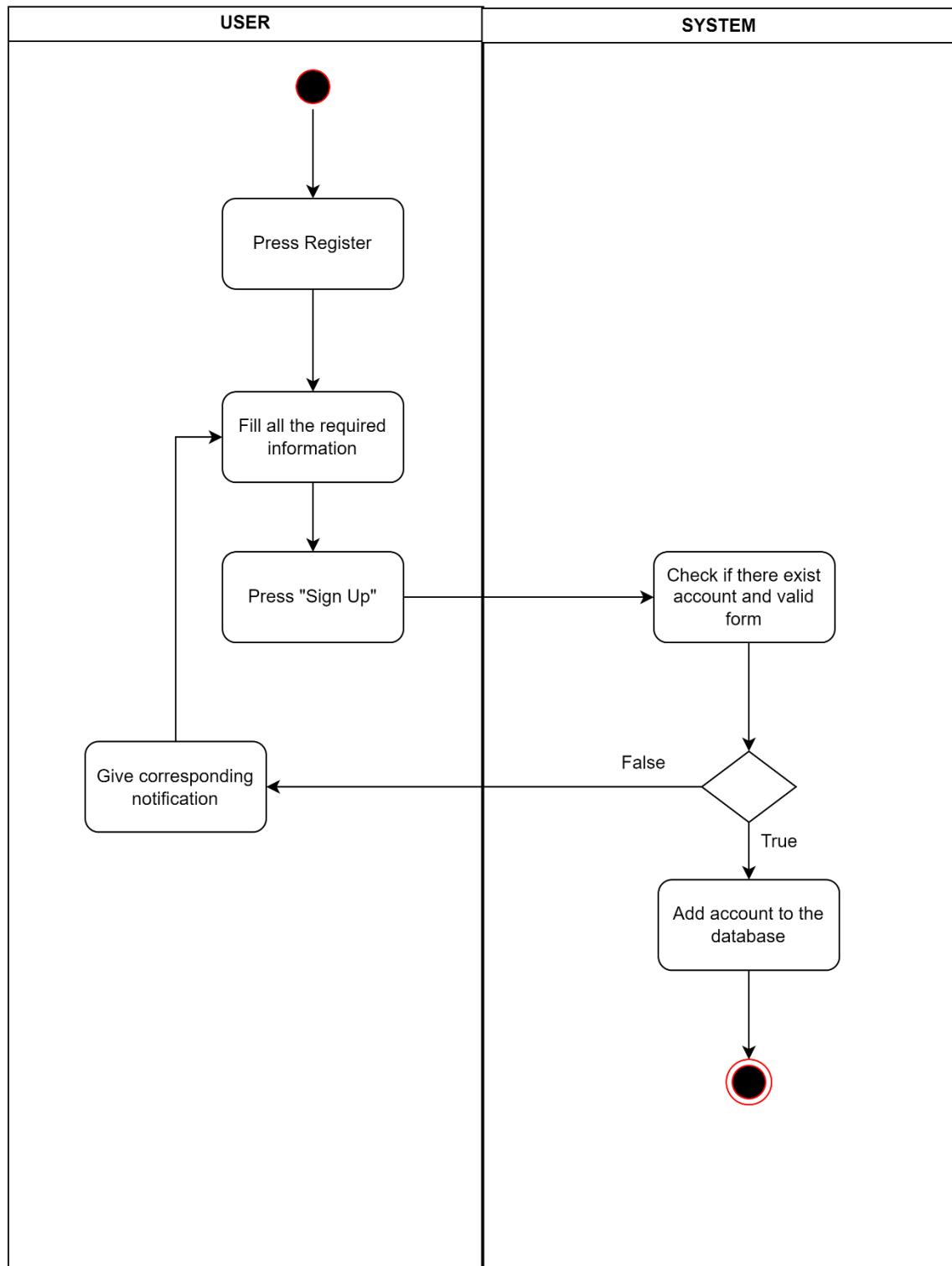


Figure 4. Activity Diagram for UC-01

3.2.2 Activity Diagram for Log in

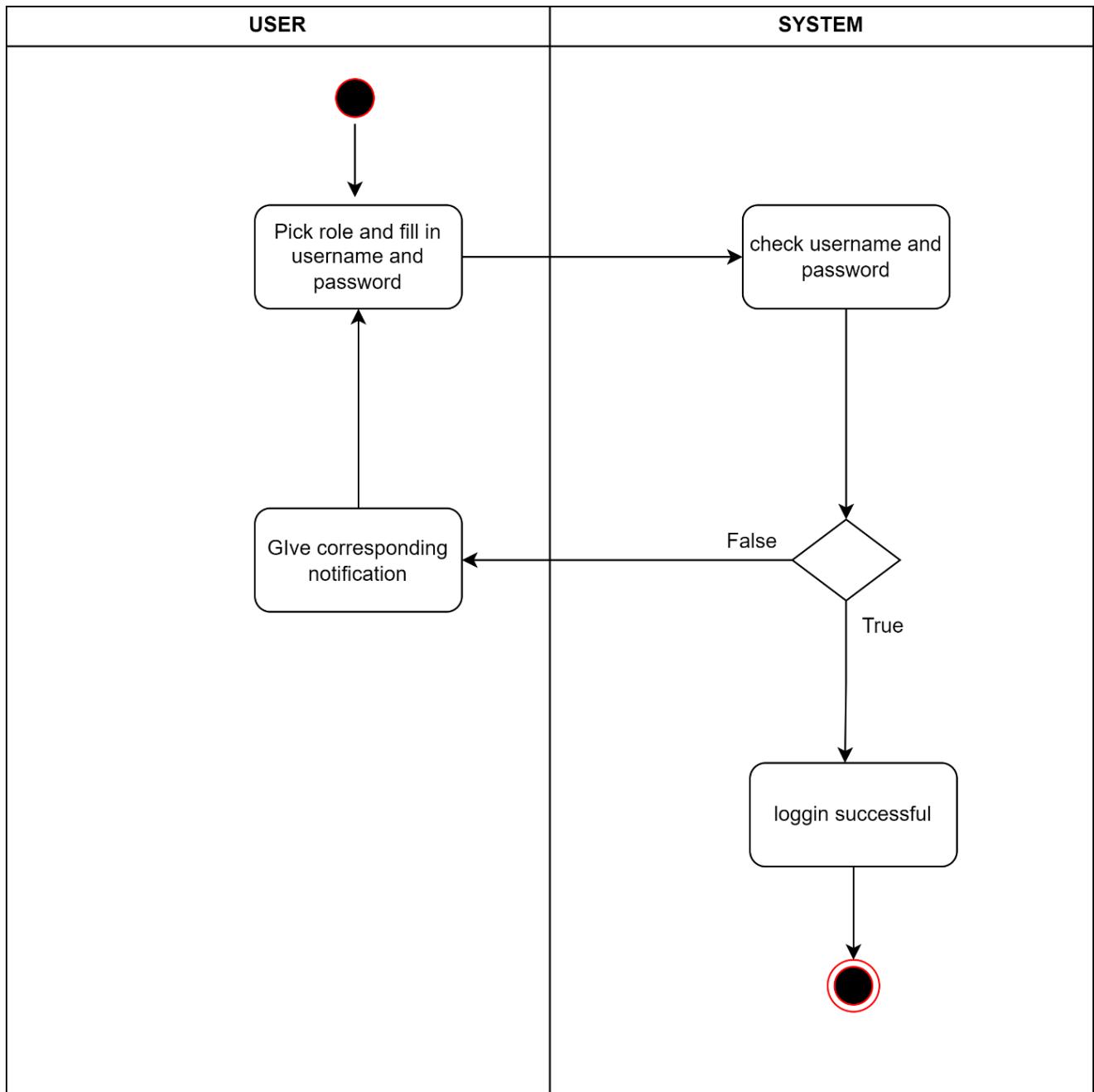


Figure 5. Activity Diagram for UC-02

3.2.3 Activity Diagram for Reset password

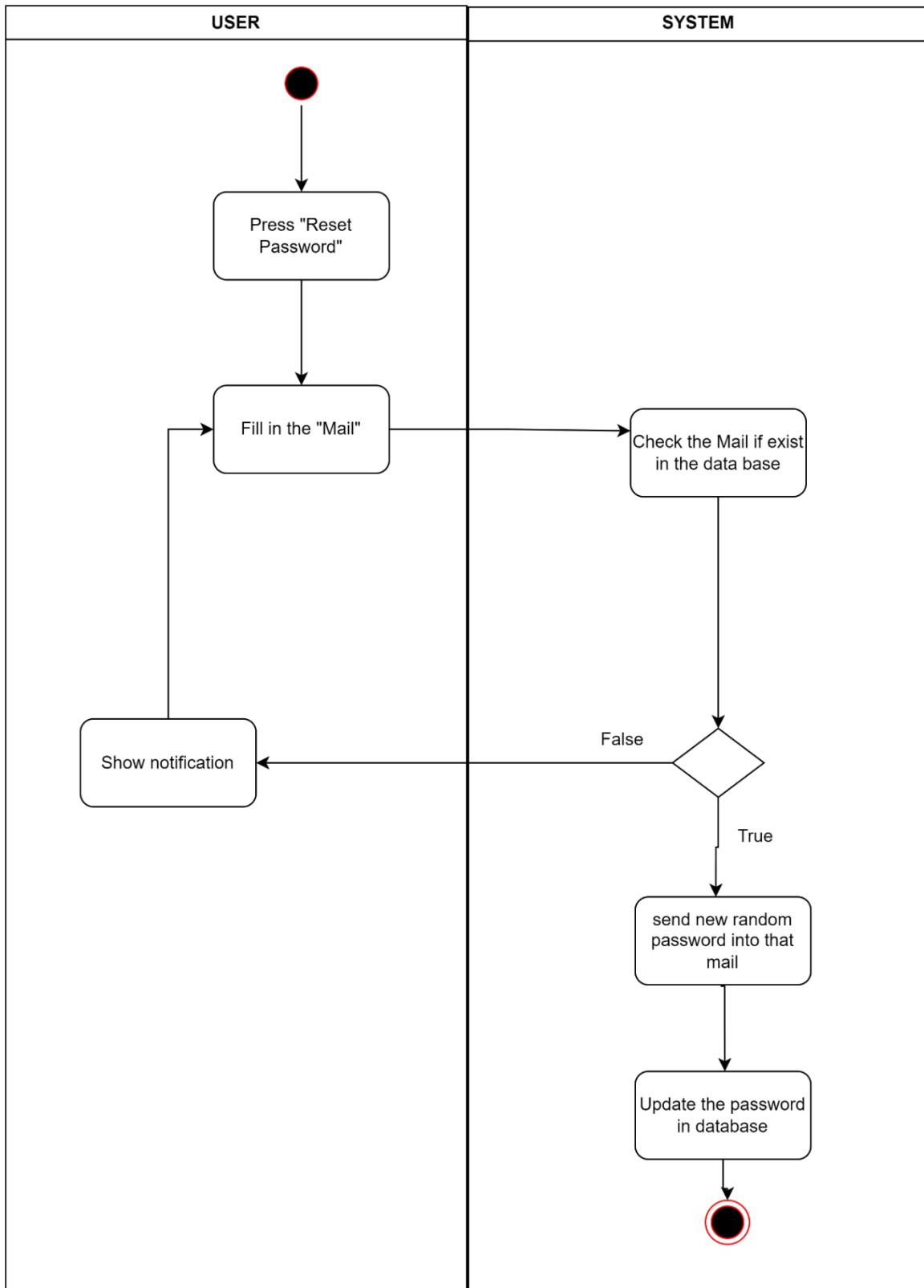


Figure 6. Activity Diagram for UC-03

3.2.4 Activity Diagram for Change password

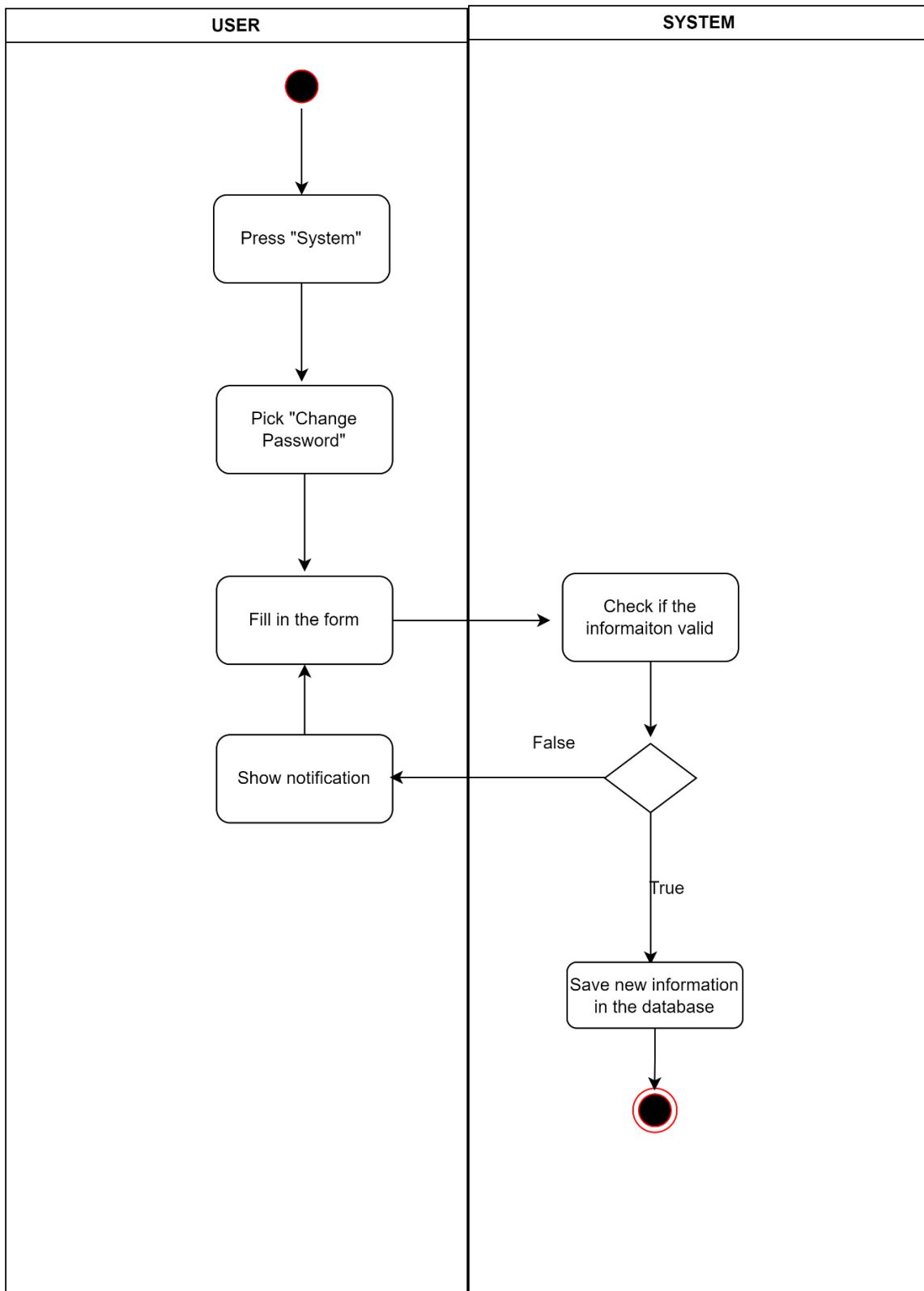


Figure 7. Activity Diagram for UC-04

3.2.5 Activity Diagram for Log out

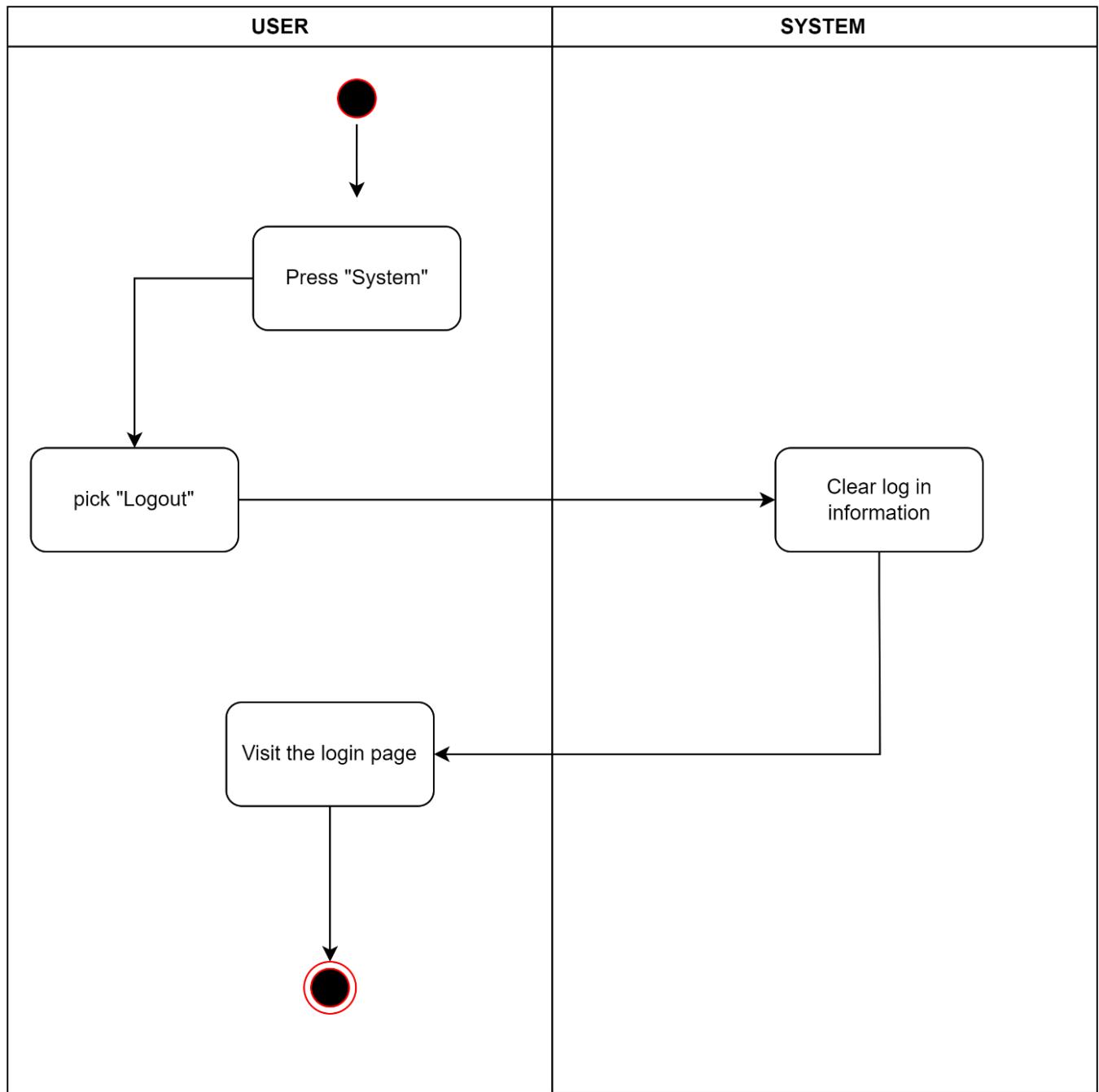


Figure 8. Activity Diagram for UC-05

3.2.6 Activity Diagram for View profile

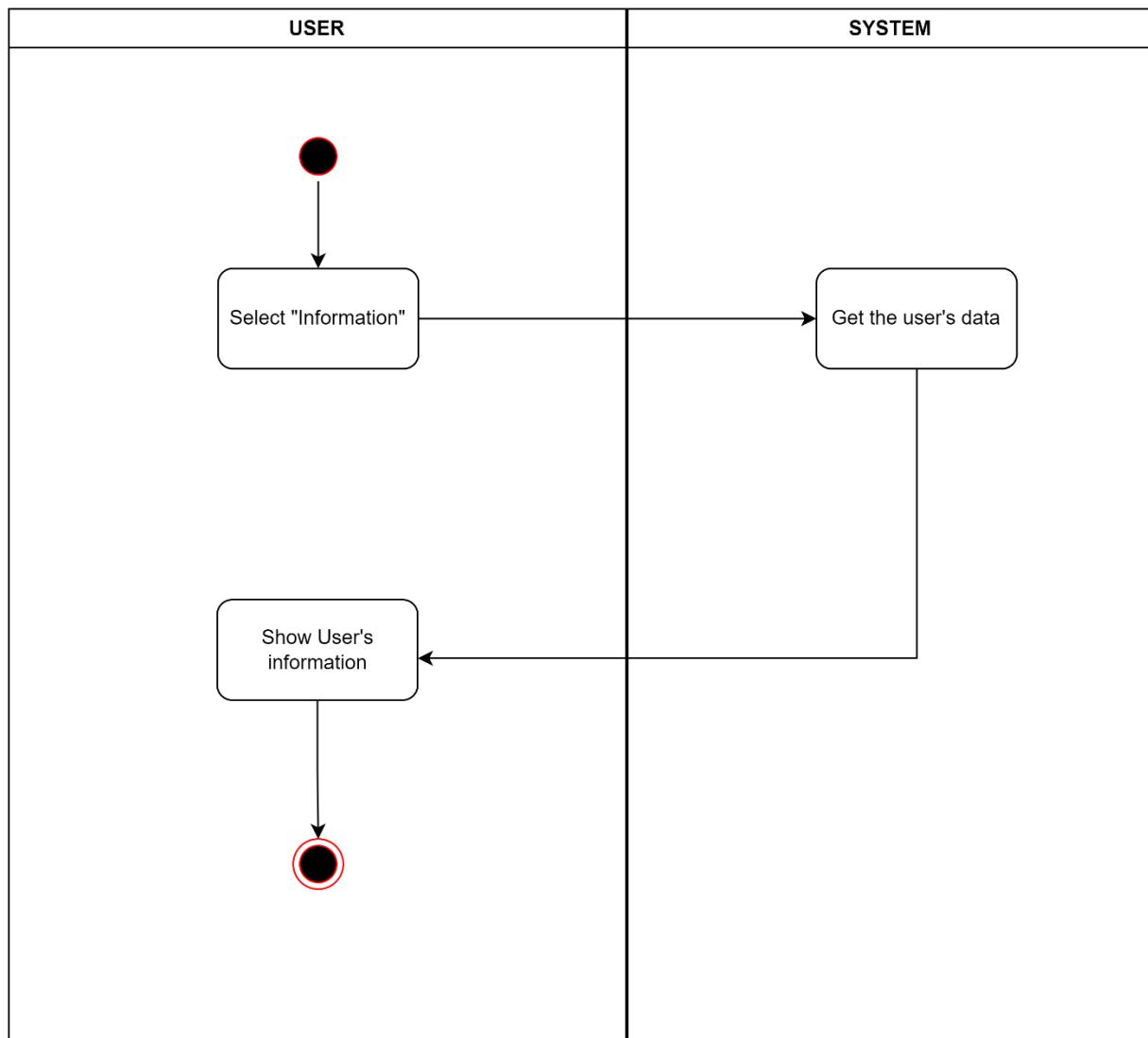


Figure 9. Activity Diagram for UC-06

3.2.7 Activity Diagram for Edit profile details

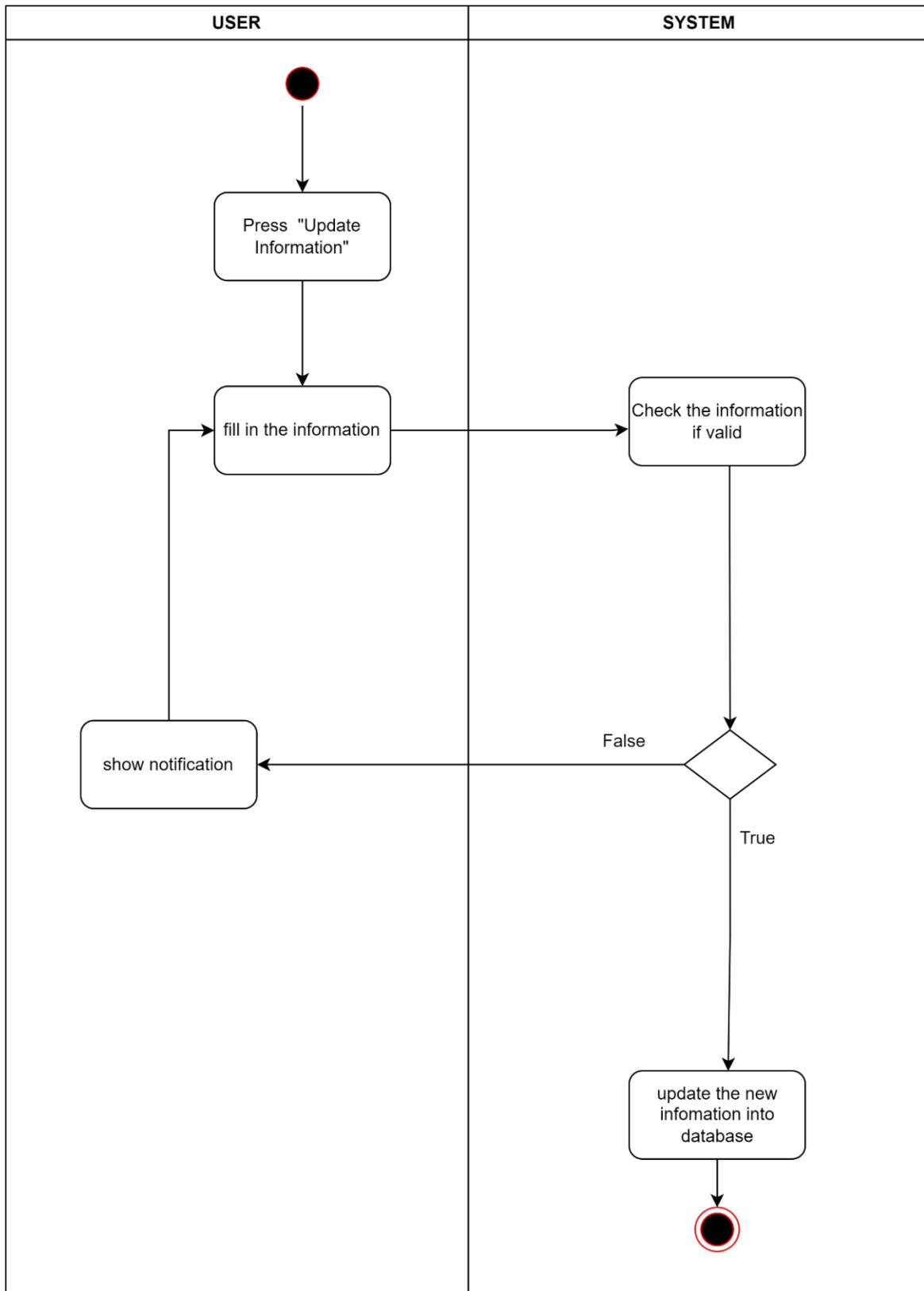


Figure 10. Activity Diagram for UC-06

3.2.8 Activity Diagram for View notifications

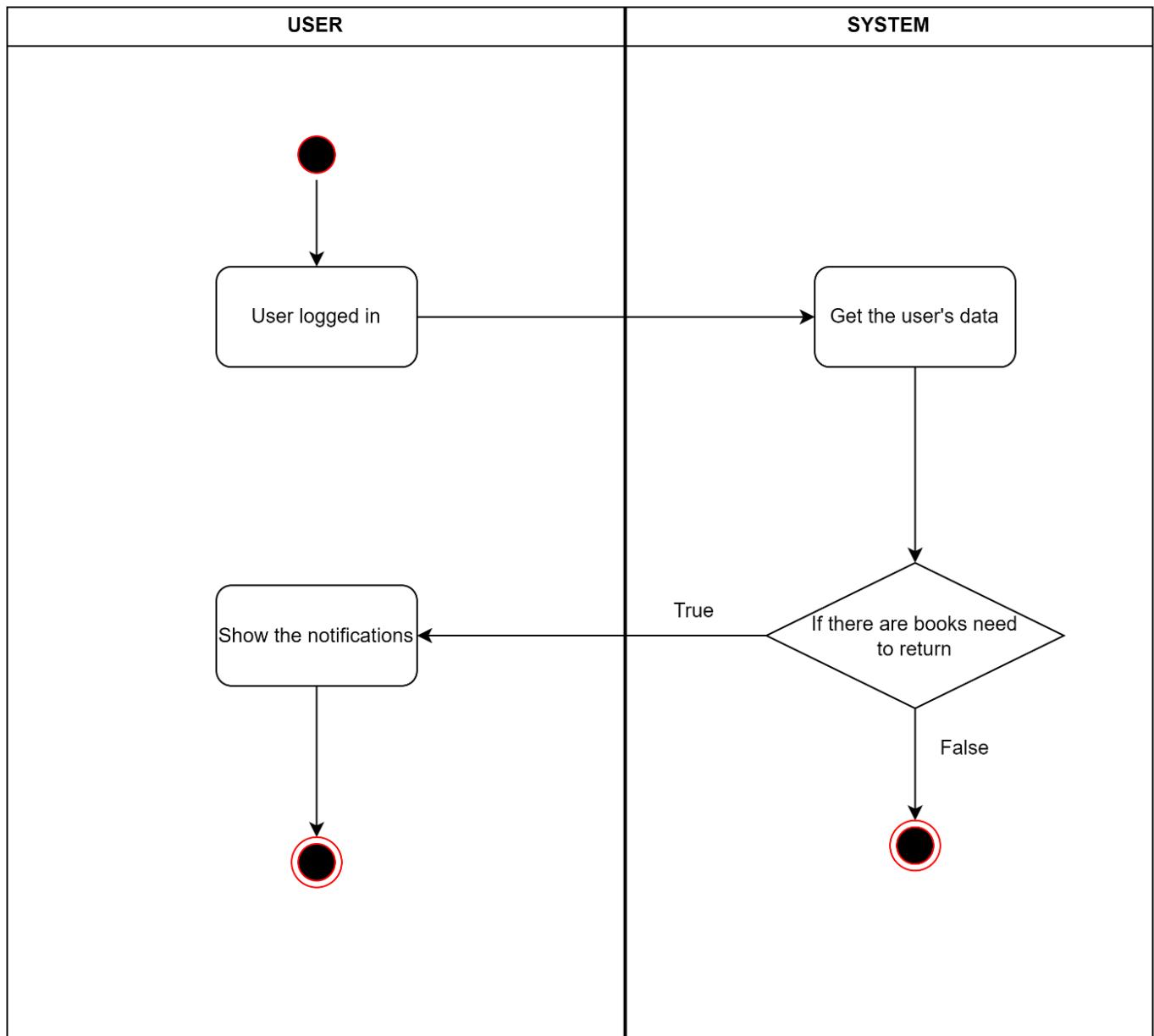


Figure 11. Activity Diagram for UC-08

3.2.9 Activity Diagram for Ask for borrowing book

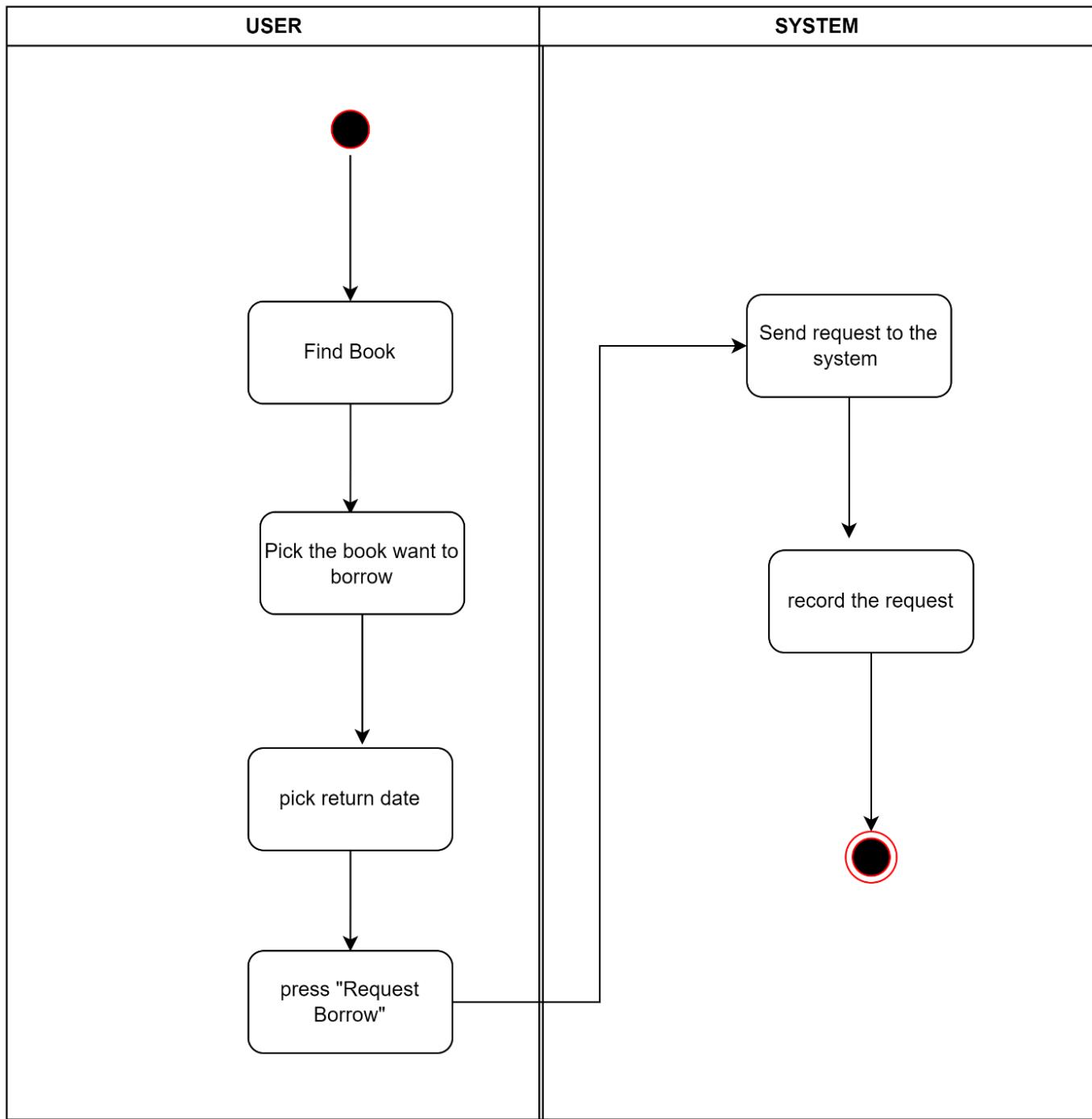


Figure 12. Activity Diagram for UC-09

3.2.10 Activity Diagram for View all personal requests

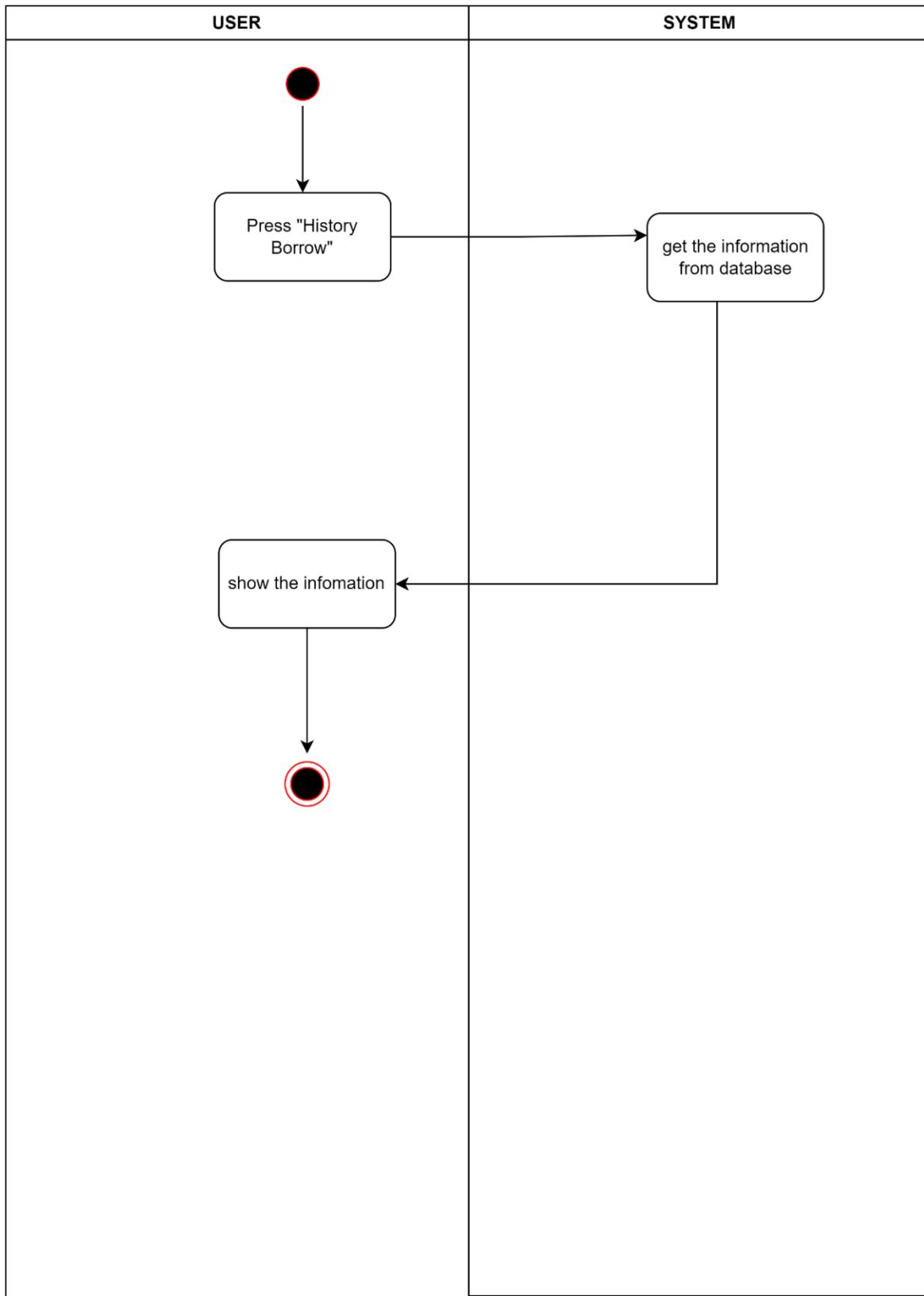


Figure 13. Activity Diagram for UC-10

3.2.11 Activity Diagram for Search personal requests

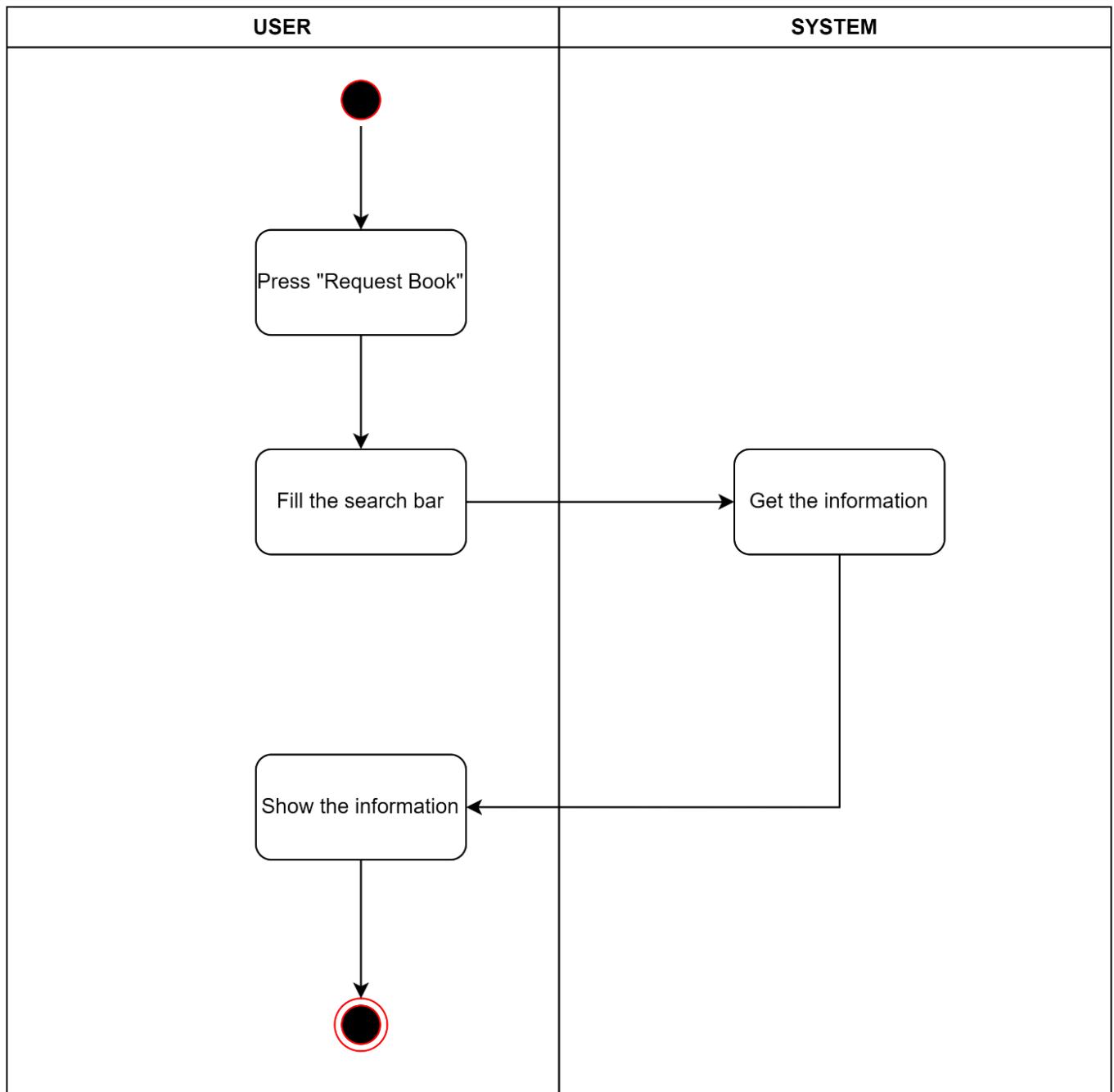


Figure 14. Activity Diagram for UC-11

3.2.12 Activity Diagram for Cancel personal requests

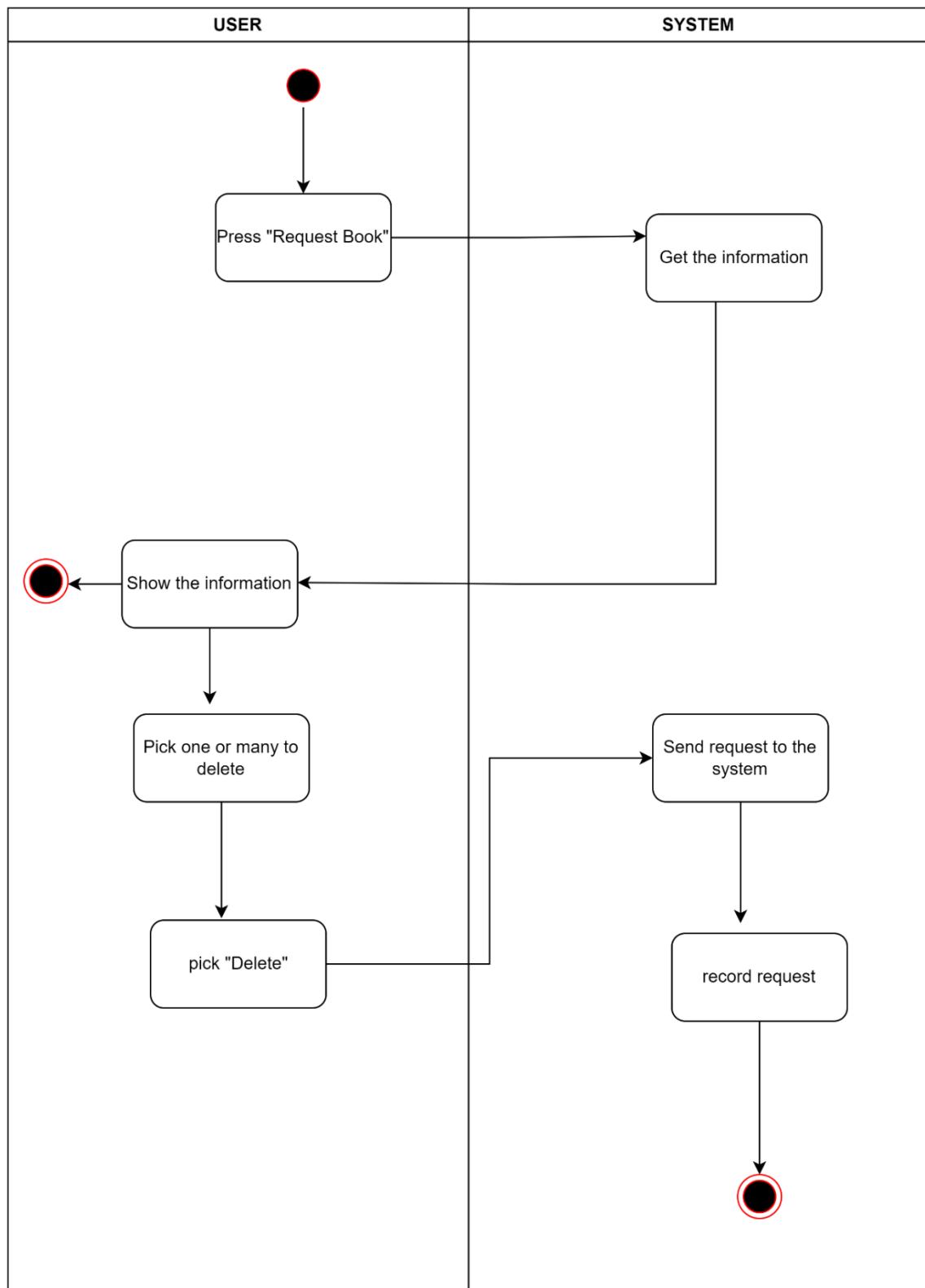


Figure 15. Activity Diagram for UC-12

3.2.13 Activity Diagram for Add new book

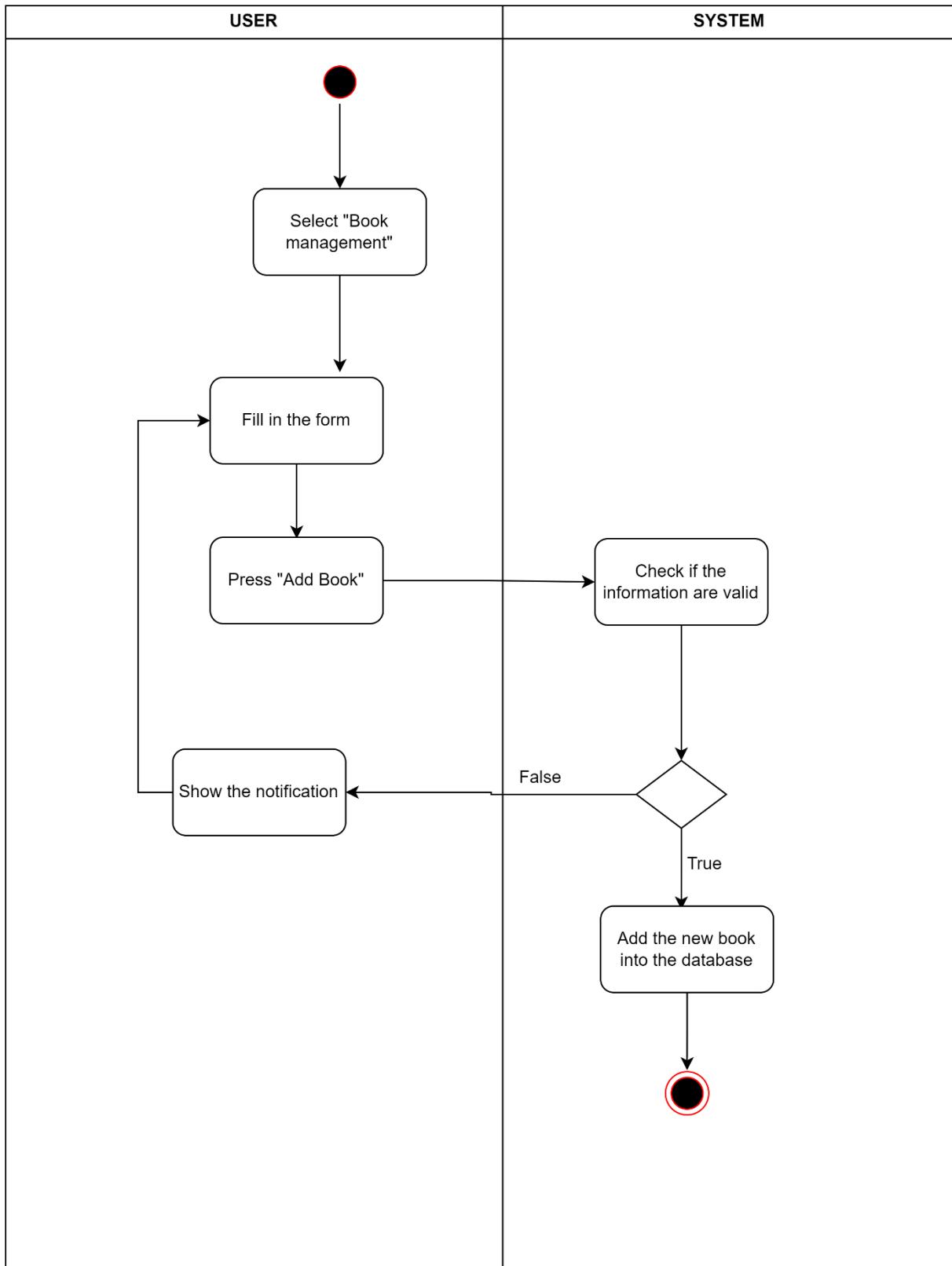


Figure 16. Activity Diagram for UC-13

3.2.14 Activity Diagram for Remove book

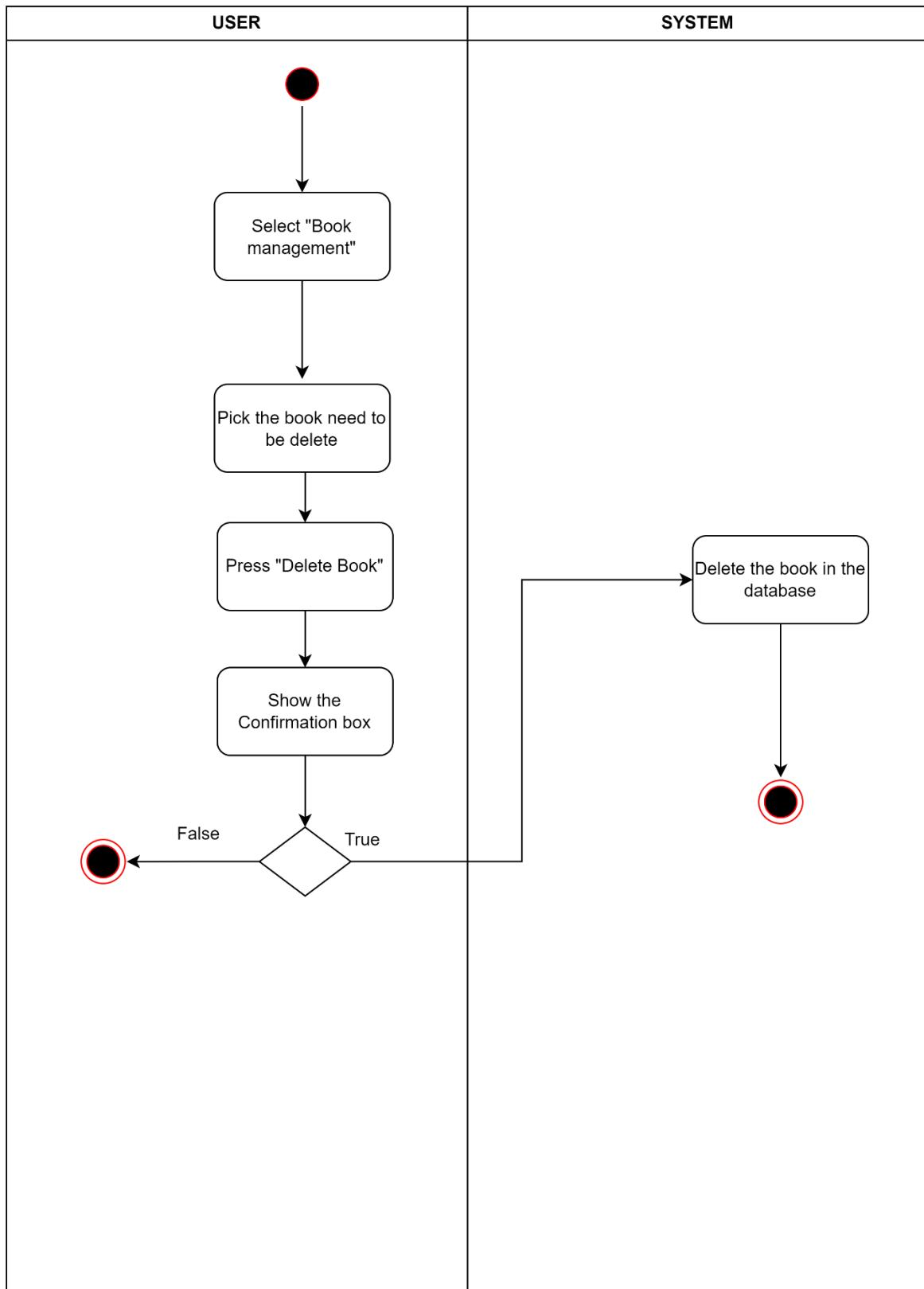


Figure 17. Activity Diagram for UC-14

3.2.15 Activity Diagram for Update book details

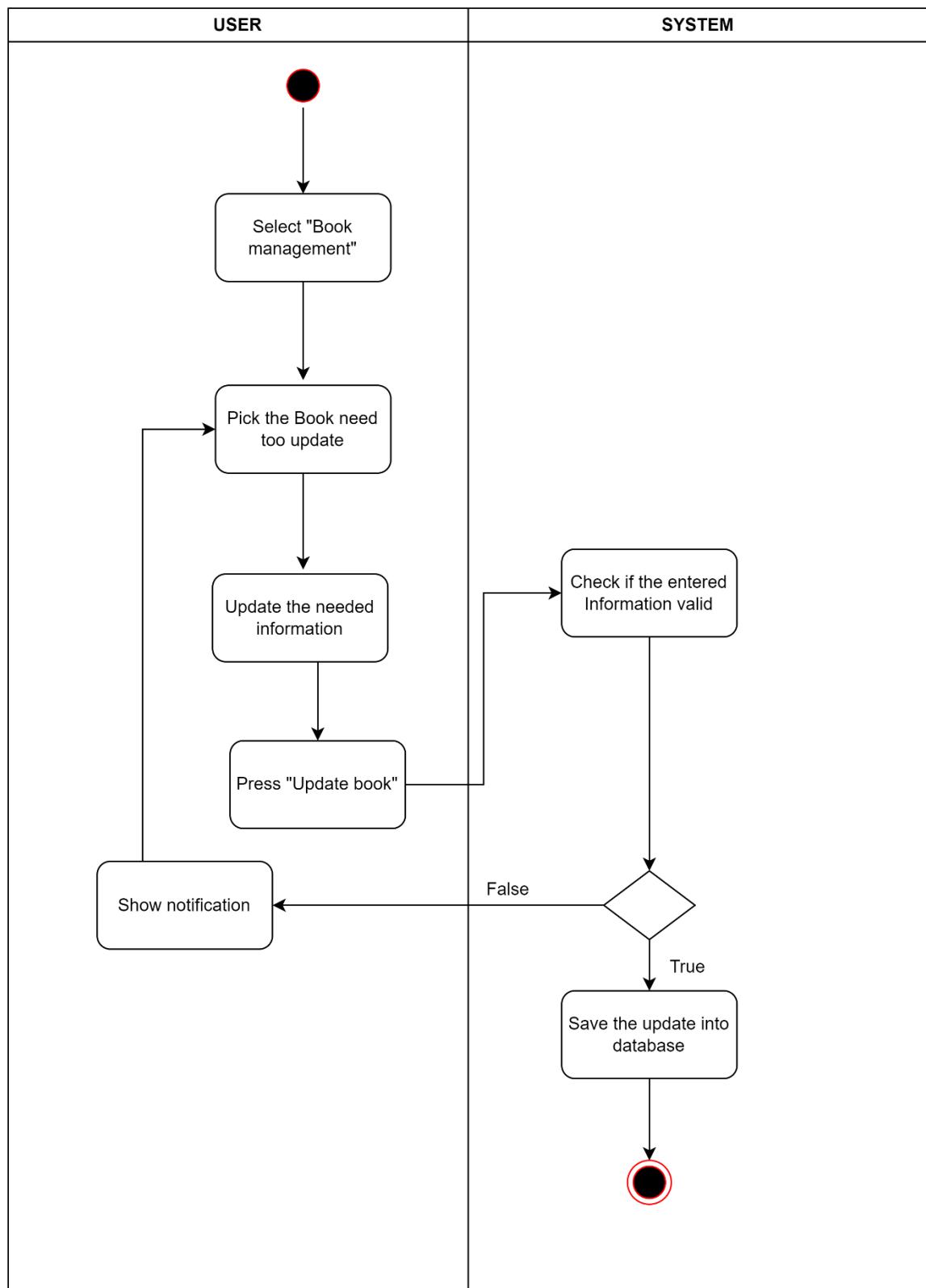


Figure 18. Activity Diagram for UC-15

3.2.16 Activity Diagram for Search book

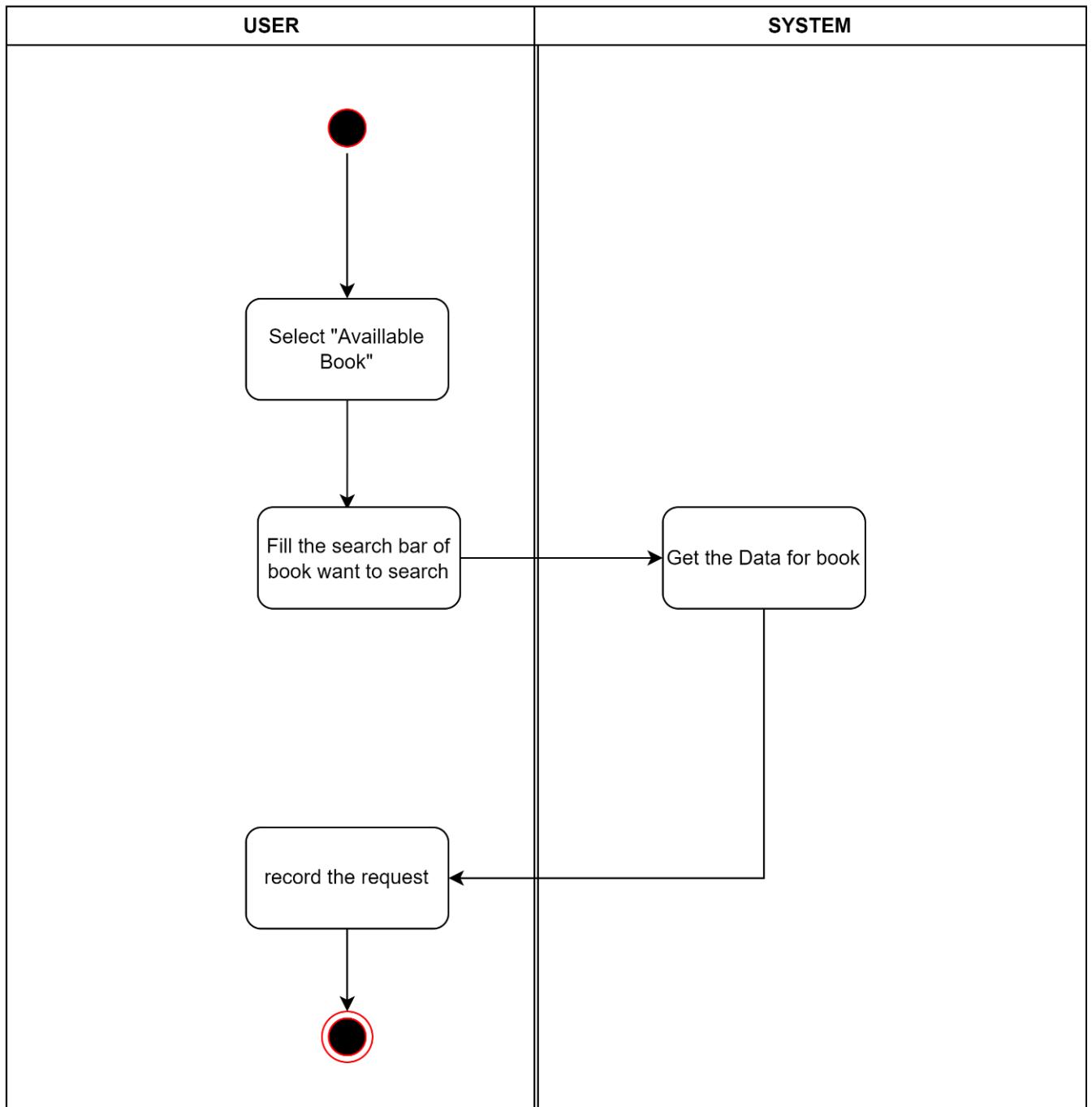


Figure 19. Activity Diagram for UC-16

3.2.17 Activity Diagram for Add reader

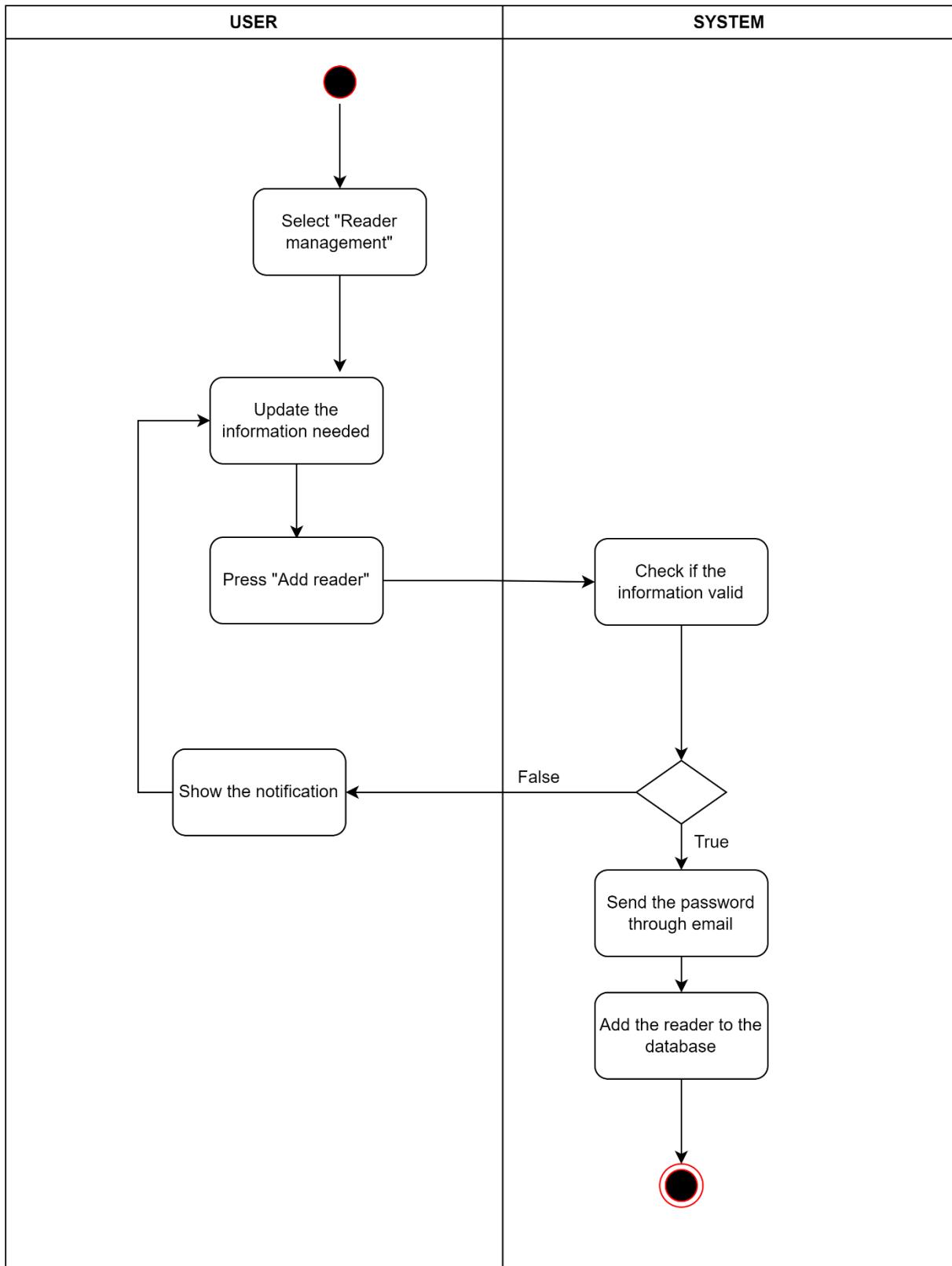


Figure 20. Activity Diagram for UC-17

3.2.18 Activity Diagram for Remove reader

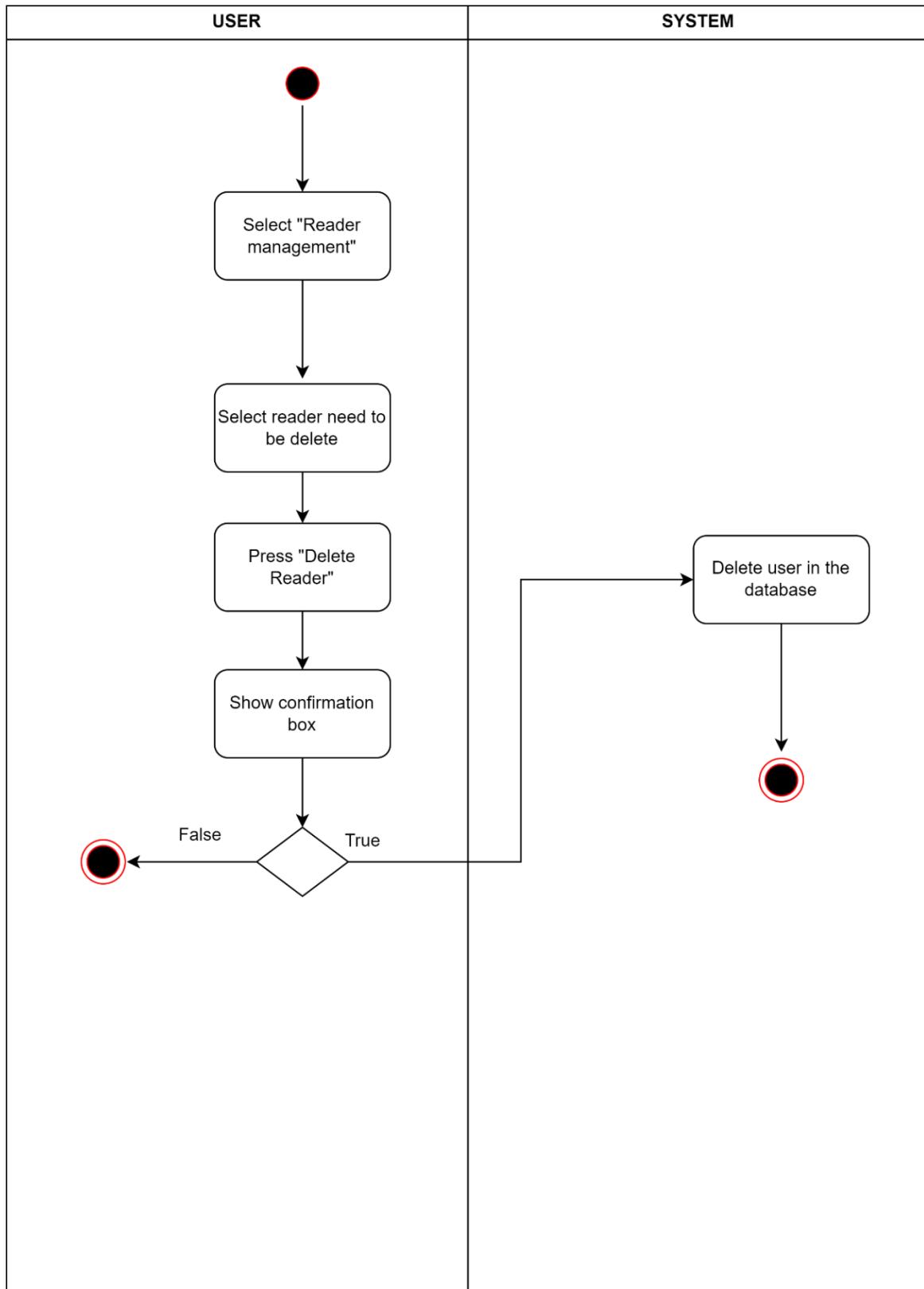


Figure 21. Activity Diagram for UC-18

3.2.19 Activity Diagram for Update reader

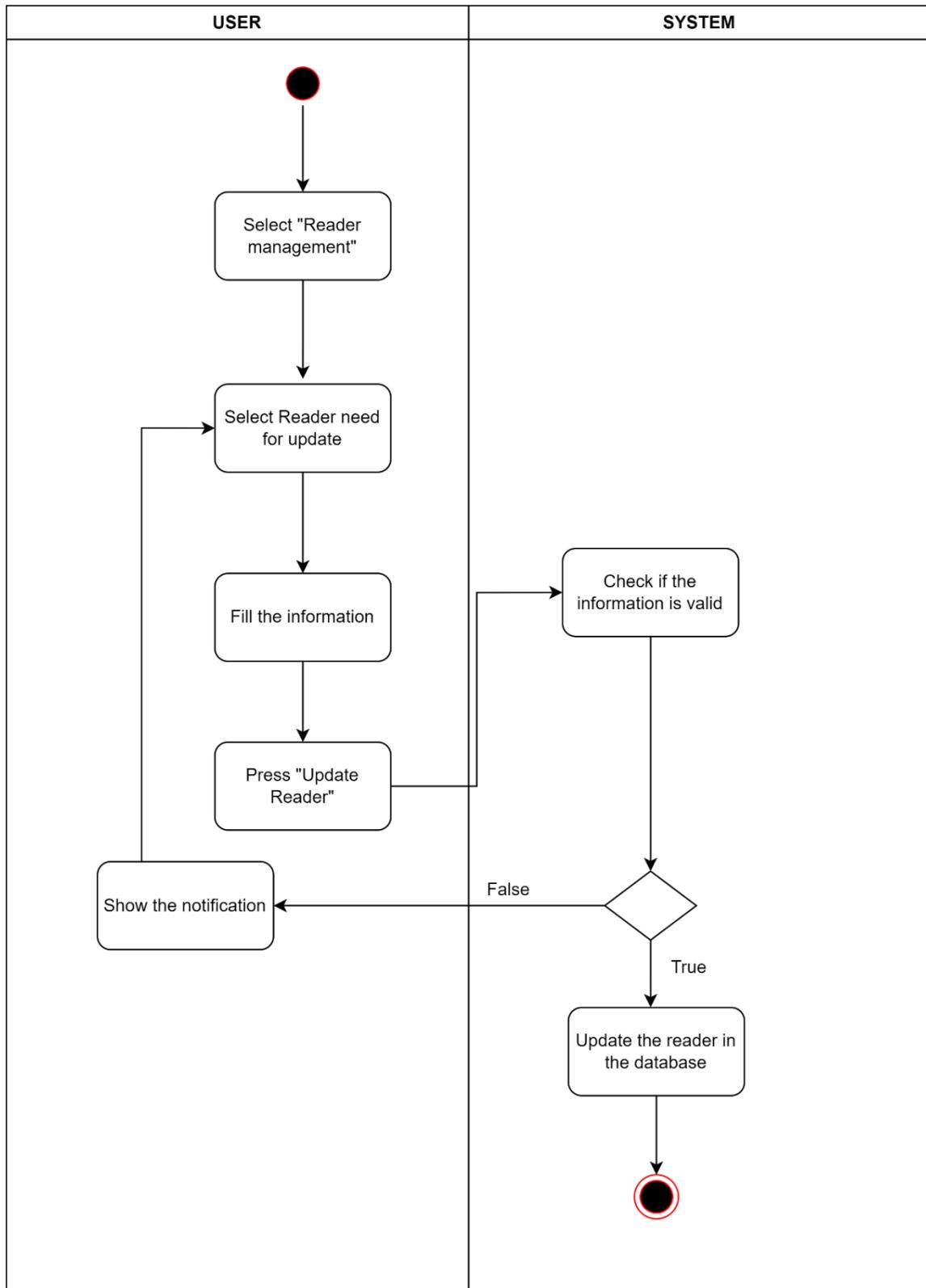


Figure 22. Activity Diagram for UC-19

3.2.20 Activity Diagram for Search reader

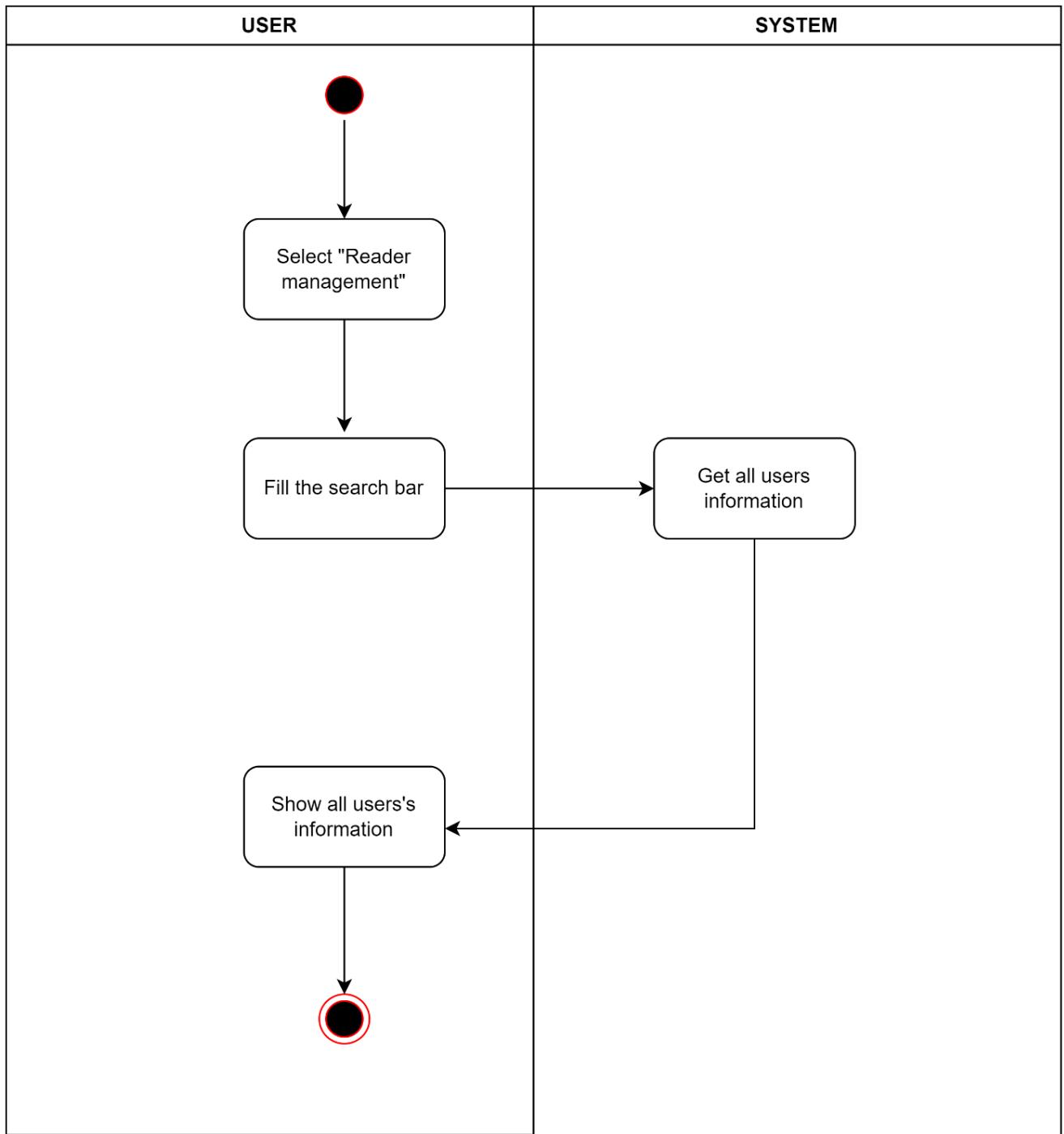


Figure 23. Activity Diagram for UC-20

3.2.21 Activity Diagram for Ban reader

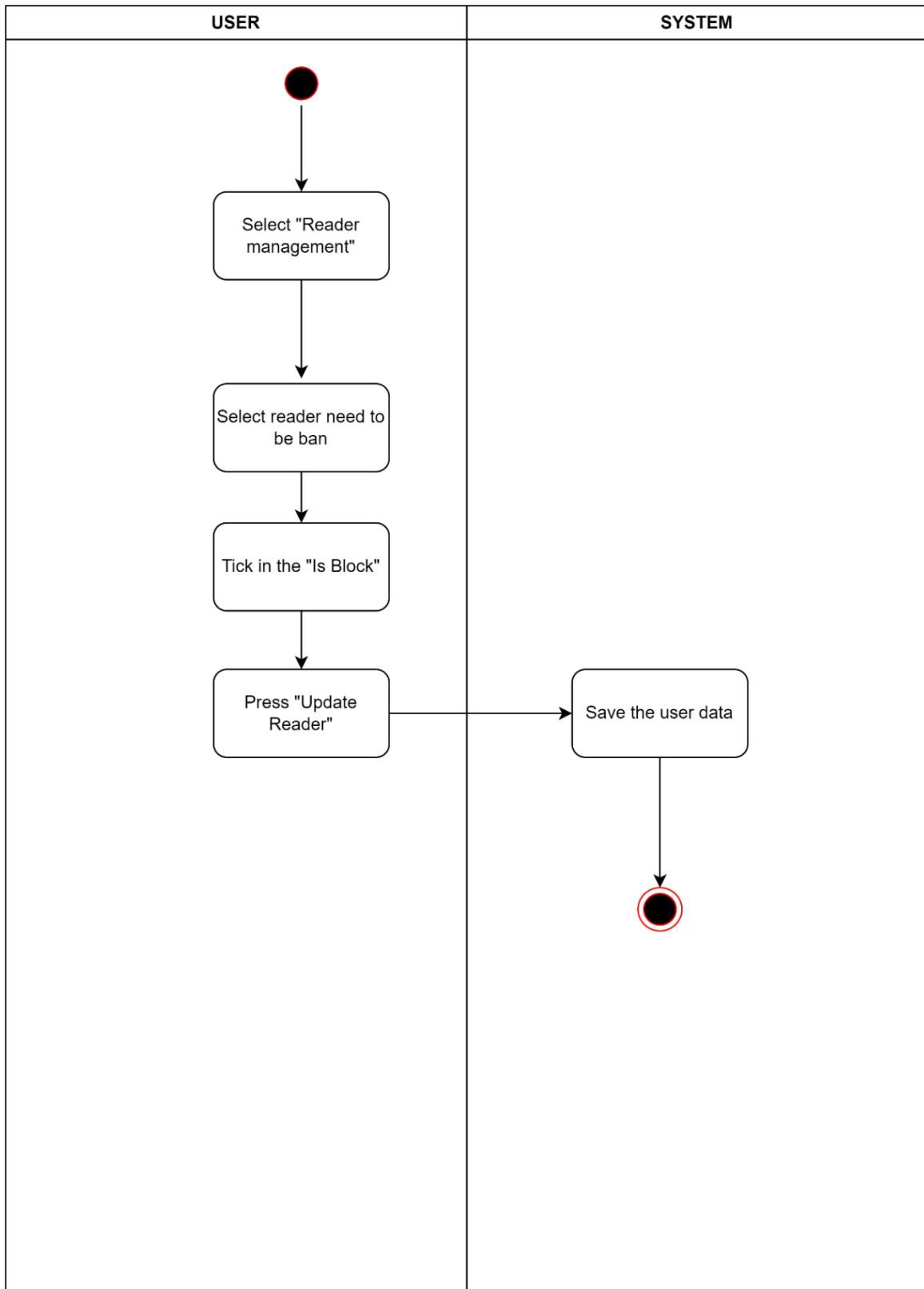


Figure 24. Activity Diagram for UC-21

3.2.22 Activity Diagram for Unban reader

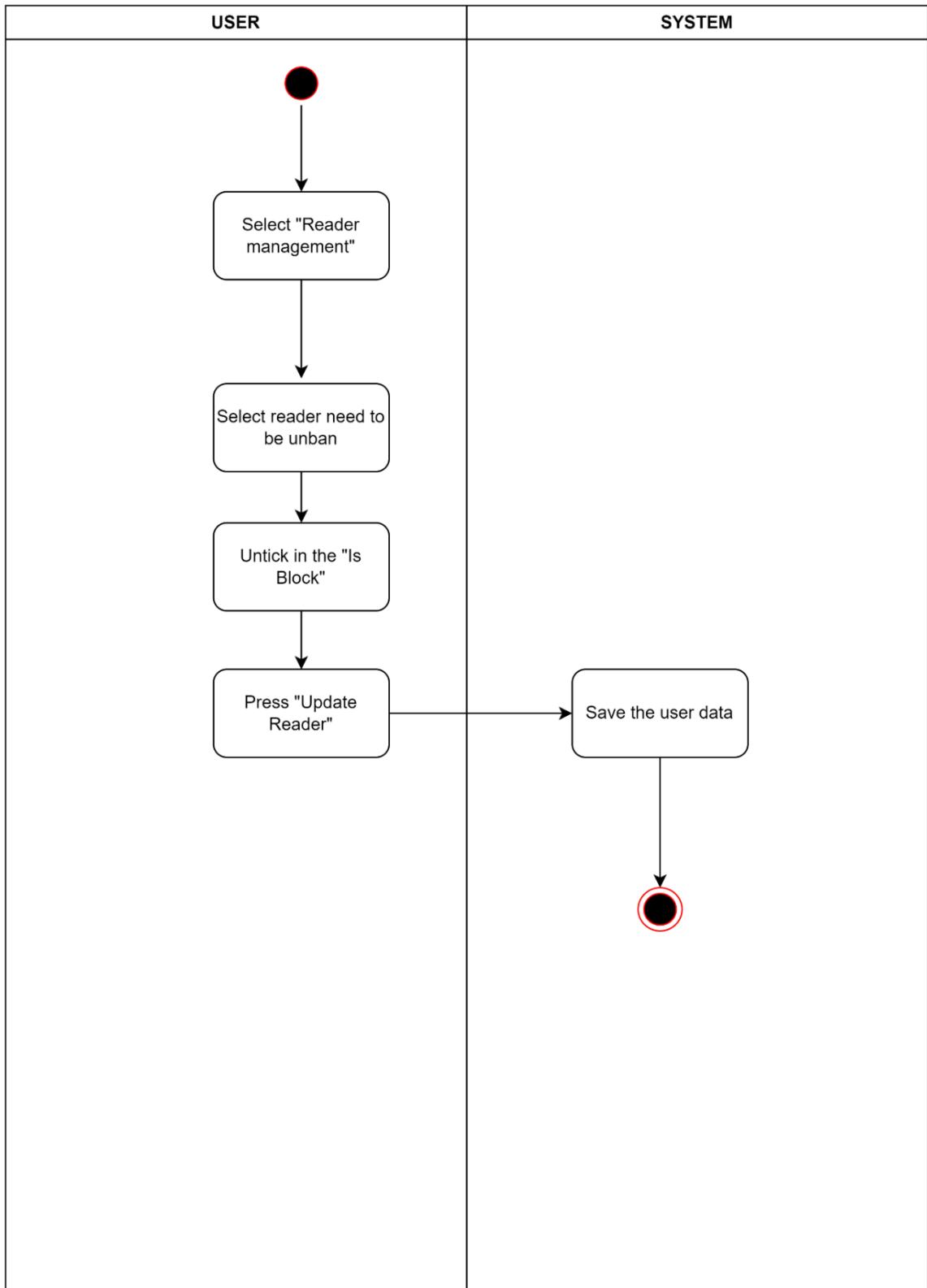


Figure 25. Activity Diagram for UC-22

3.2.23 Activity Diagram for View history of borrowing and returning book

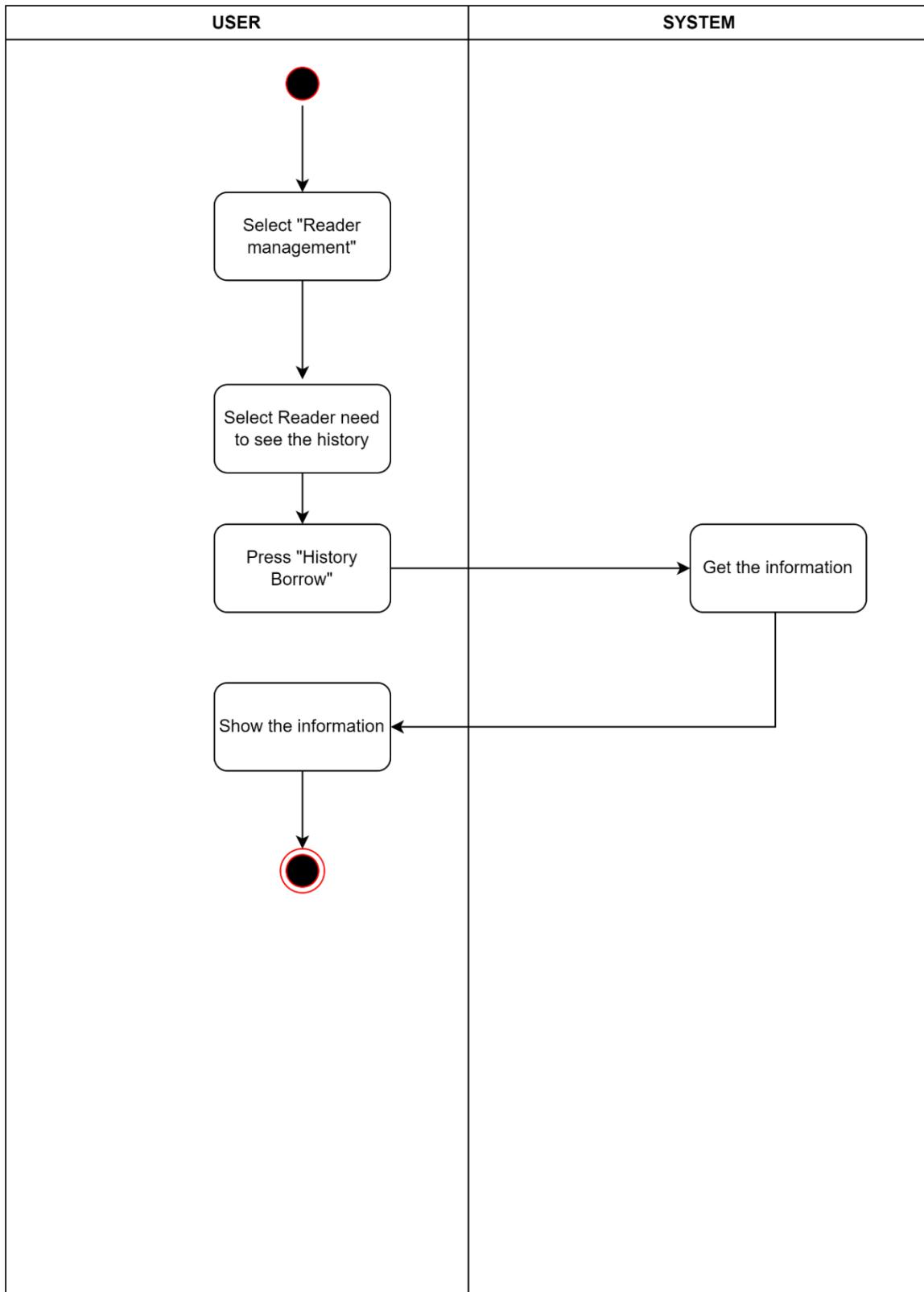


Figure 26. Activity Diagram for UC-23

3.2.24 Activity Diagram for Issue book

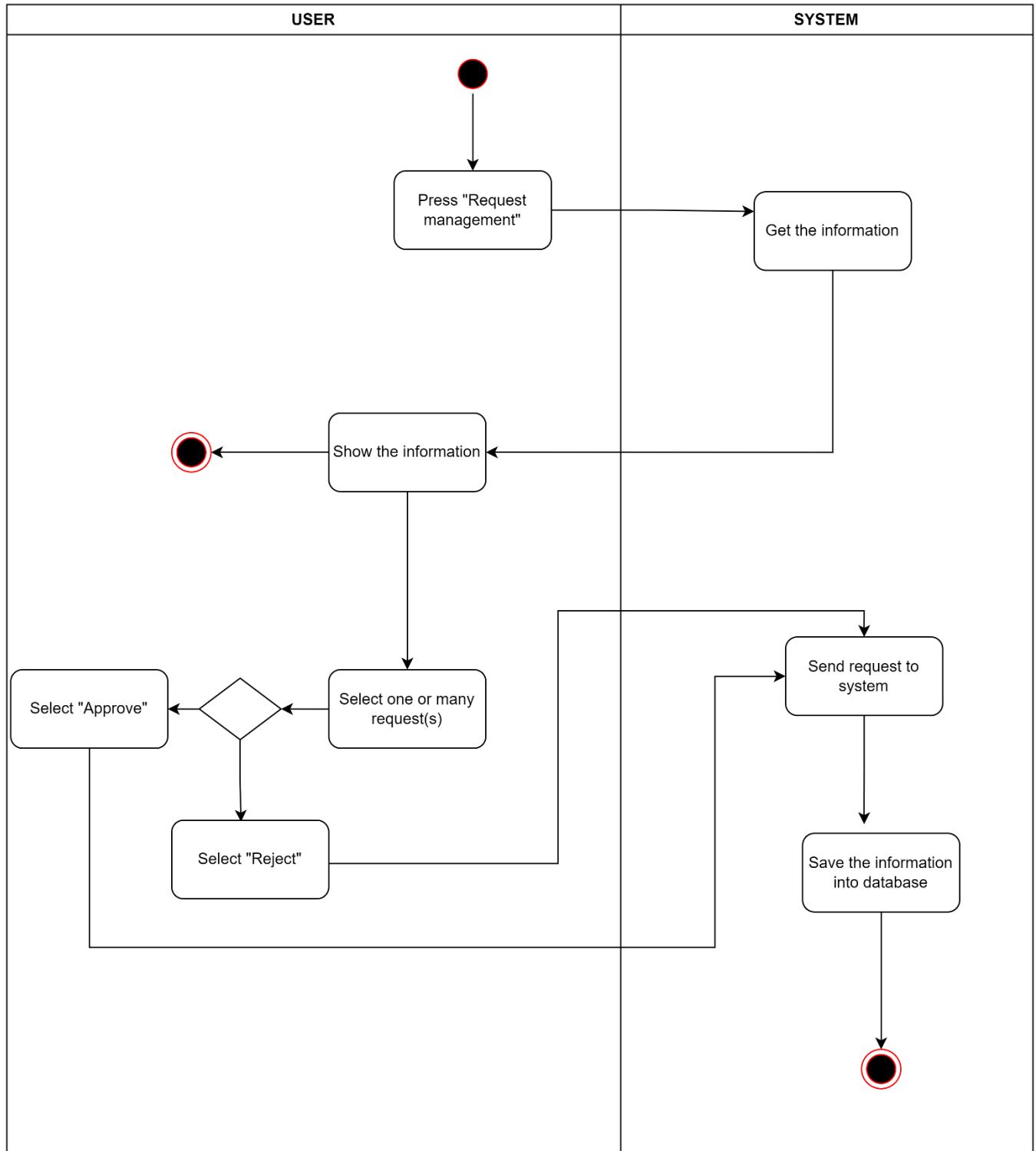


Figure 27. Activity Diagram for UC-24

3.2.25 Activity Diagram for View requests to borrow book from reader

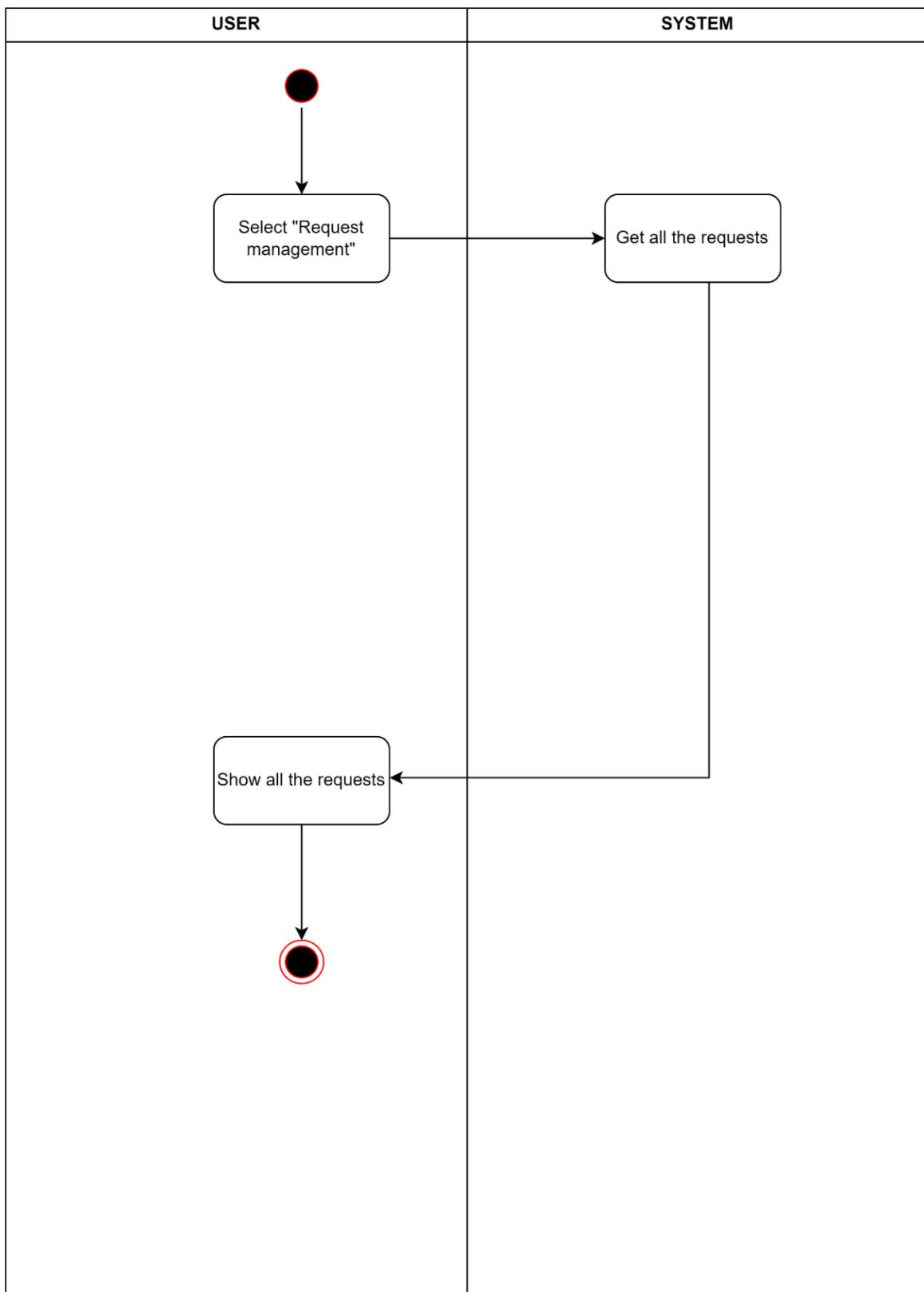


Figure 28. Activity Diagram for UC-25

3.2.26 Activity Diagram for Search requests to borrow book from reader

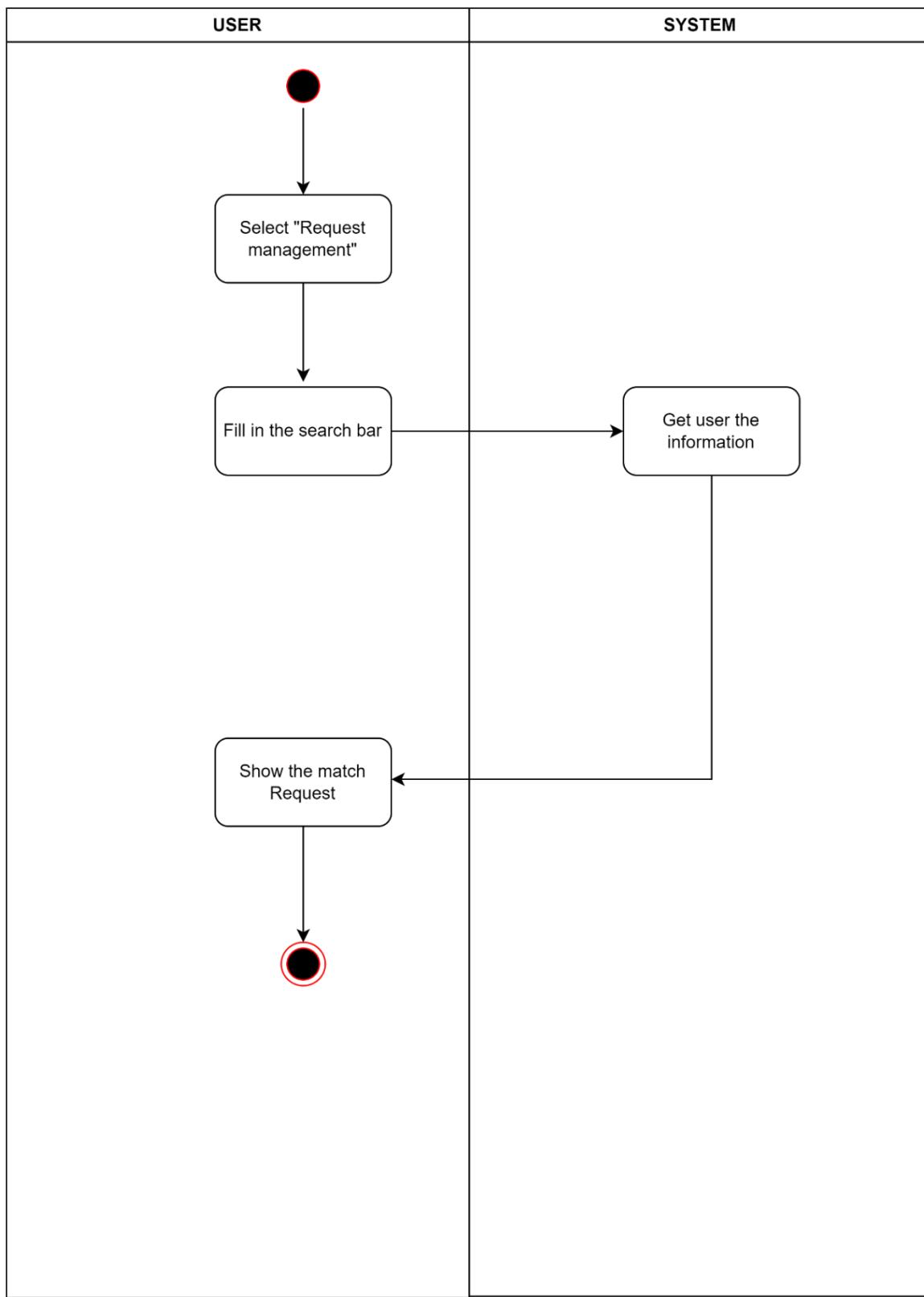


Figure 29. Activity Diagram for UC-26

3.2.27 Activity Diagram for Return book

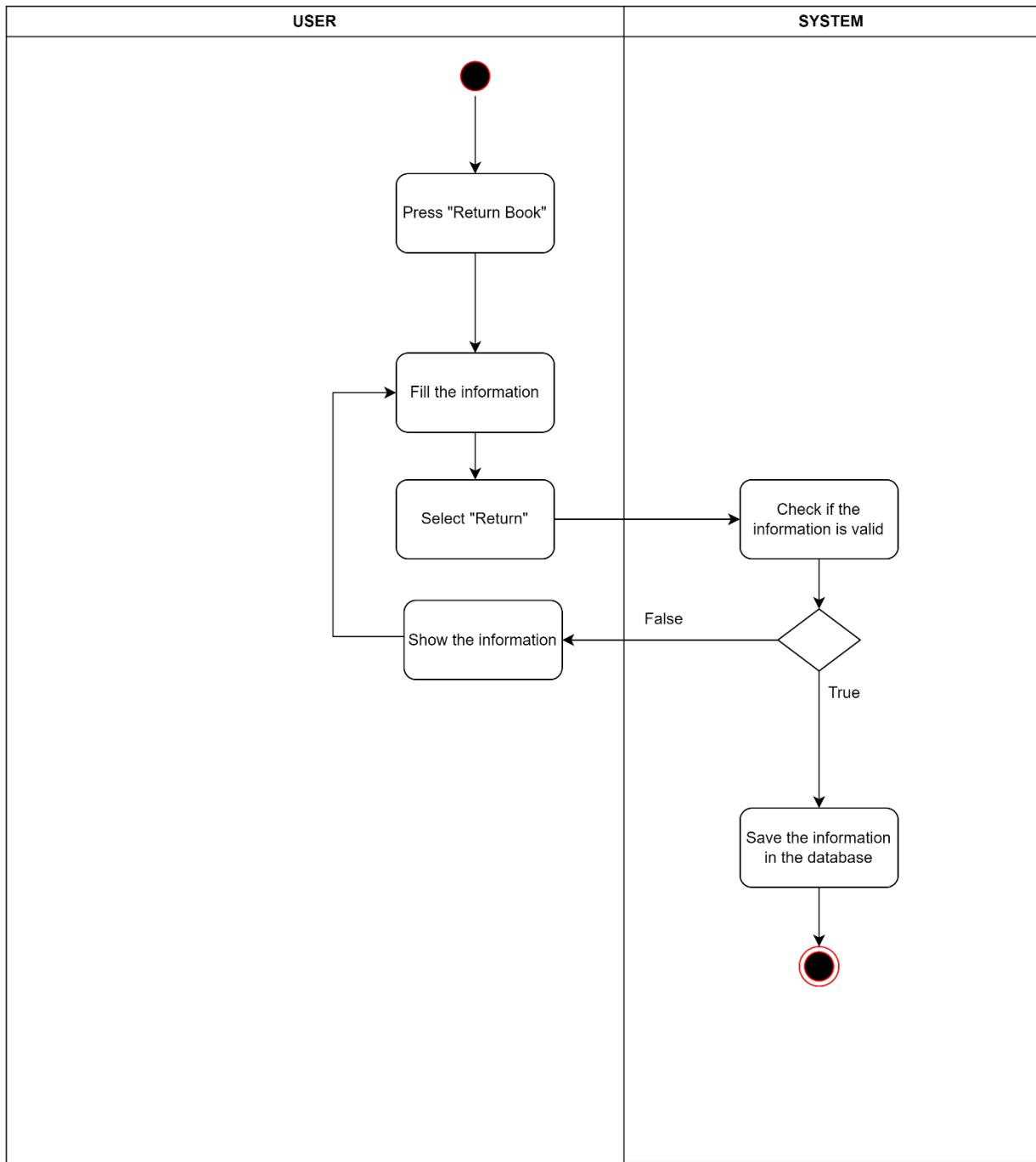


Figure 30. Activity Diagram for UC-27

3.2.28 Activity Diagram for View reports and statistics

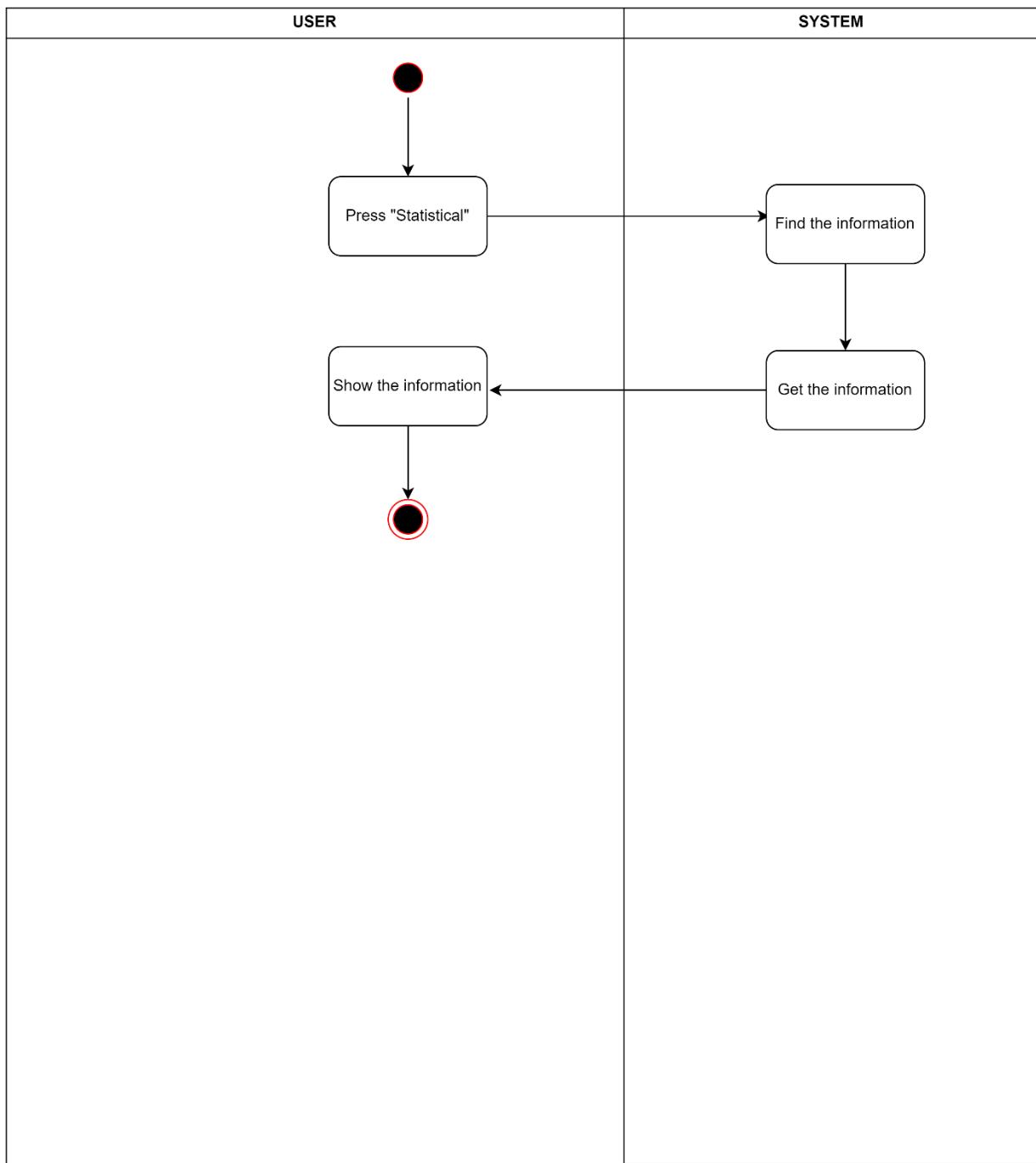


Figure 31. Activity Diagram for UC-28

3.3 Sequence Diagram

3.3.1 Sequence Diagram for Create new account (Sign Up)

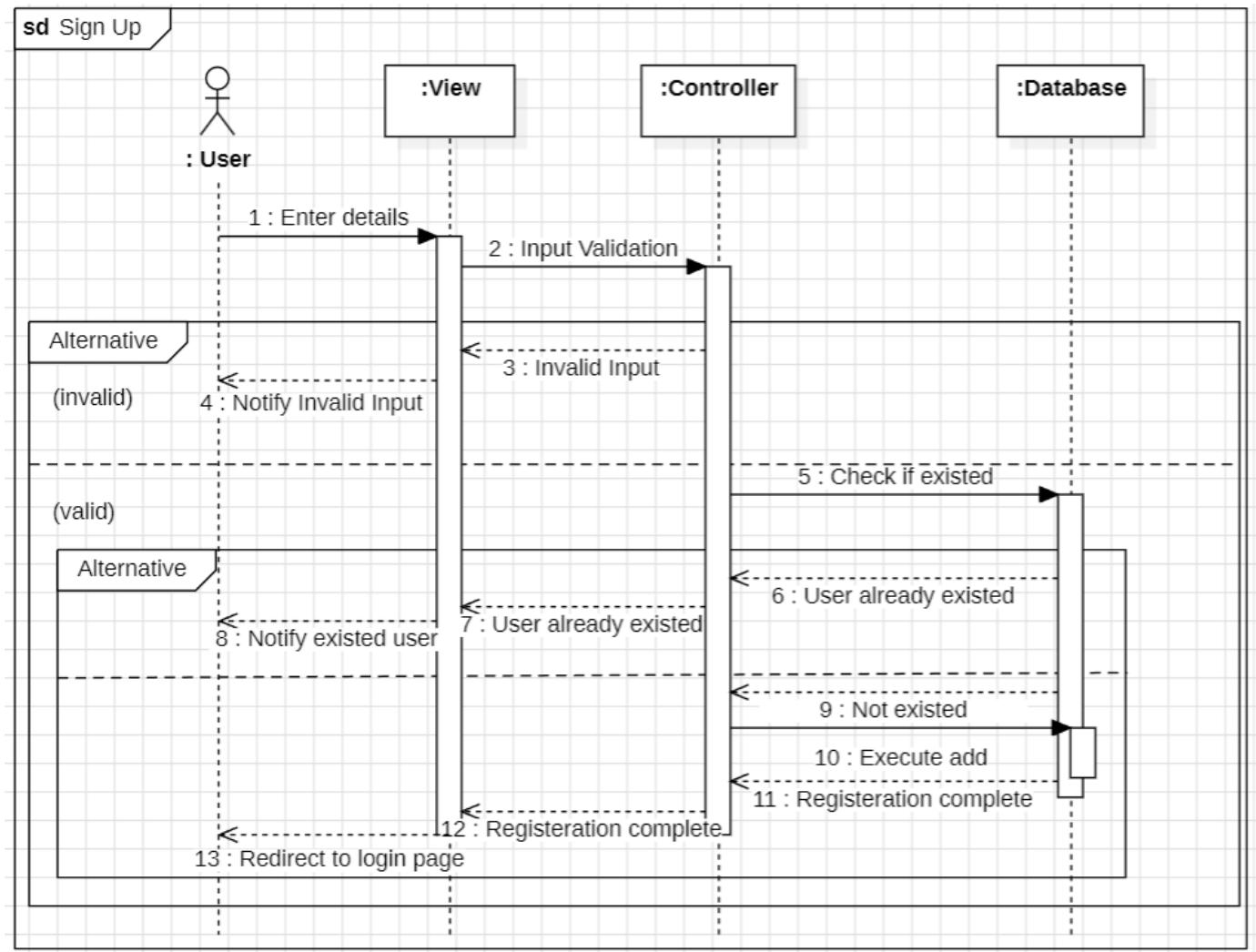


Figure 32. Sequence Diagram for Create New Account (Sign Up)

3.3.2 Sequence Diagram for Login

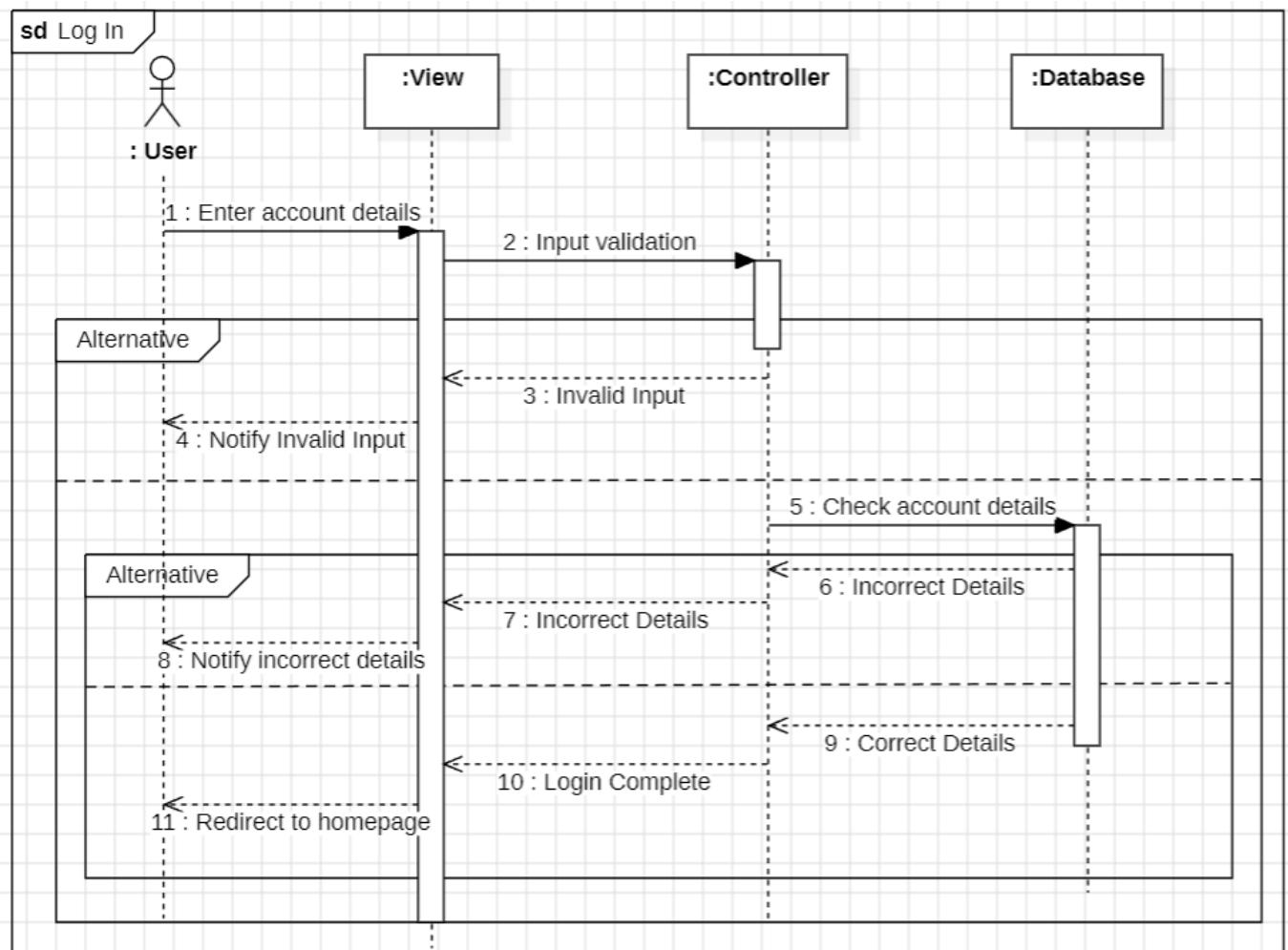


Figure 33. Sequence Diagram for Log in

3.3.3 Sequence Diagram for Reset Password

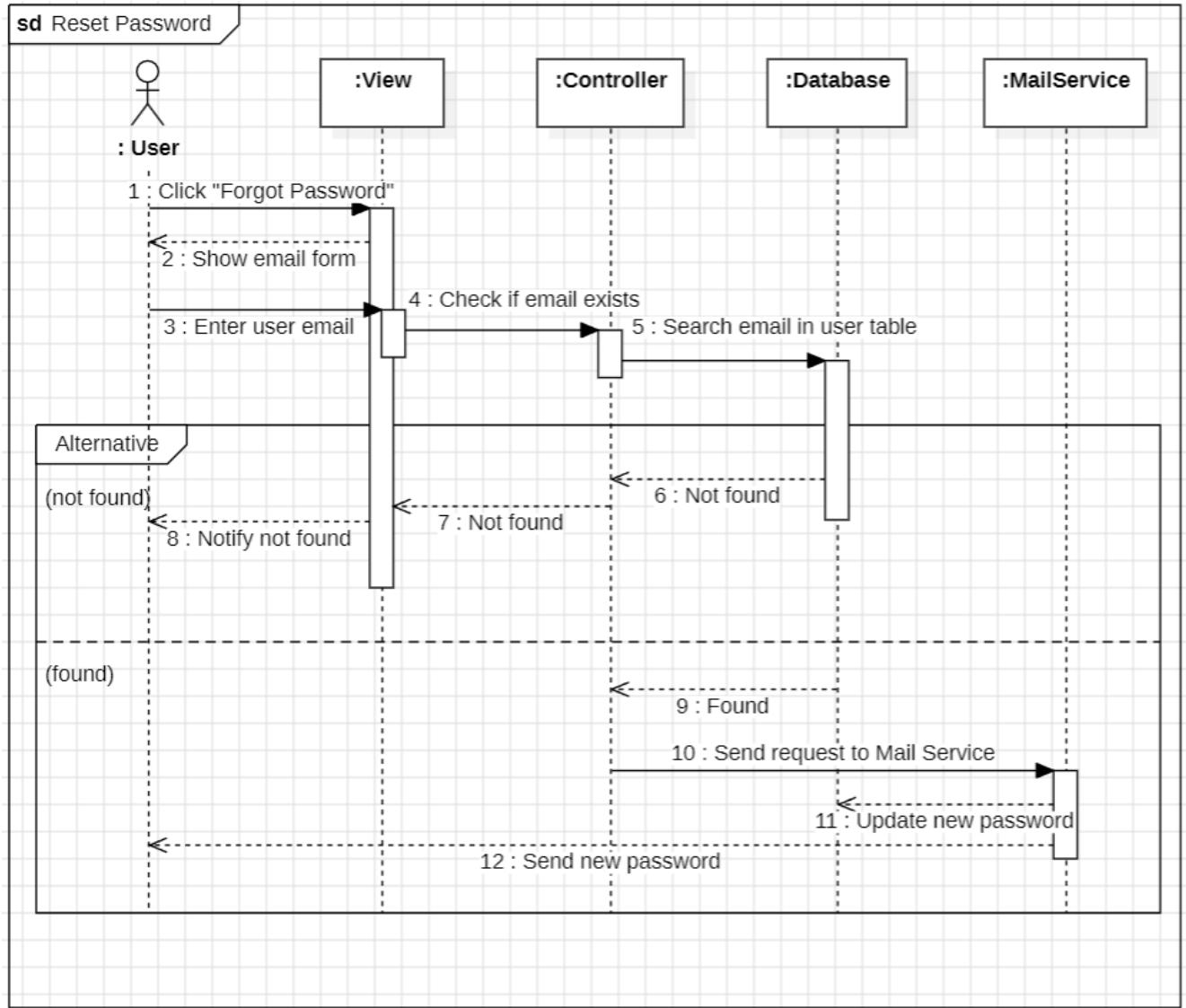


Figure 34. Sequence Diagram for Reset Password

3.3.4 Sequence Diagram for Log Out

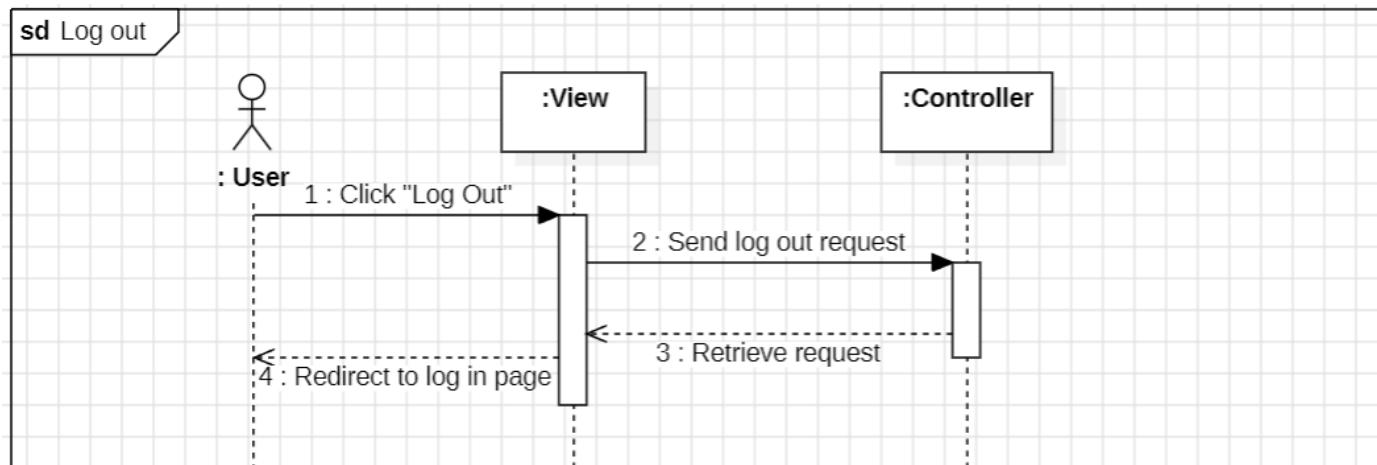


Figure 35. Sequence Diagram for Log Out

3.3.5 Sequence Diagram for View Profile Details

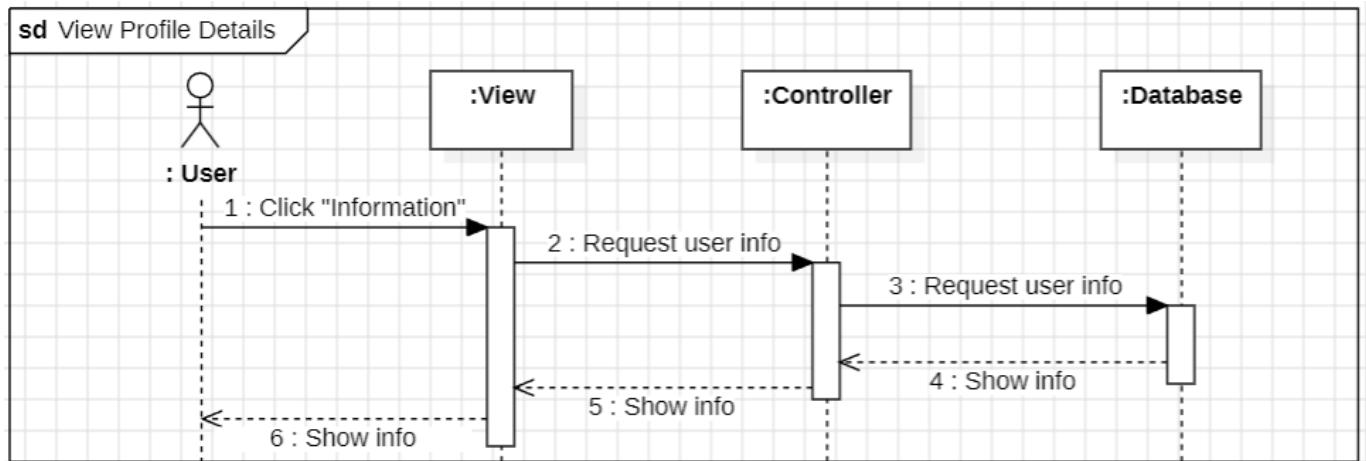


Figure 36. Sequence Diagram for View Profile Details

3.3.6 Sequence Diagram for Edit Profile Details

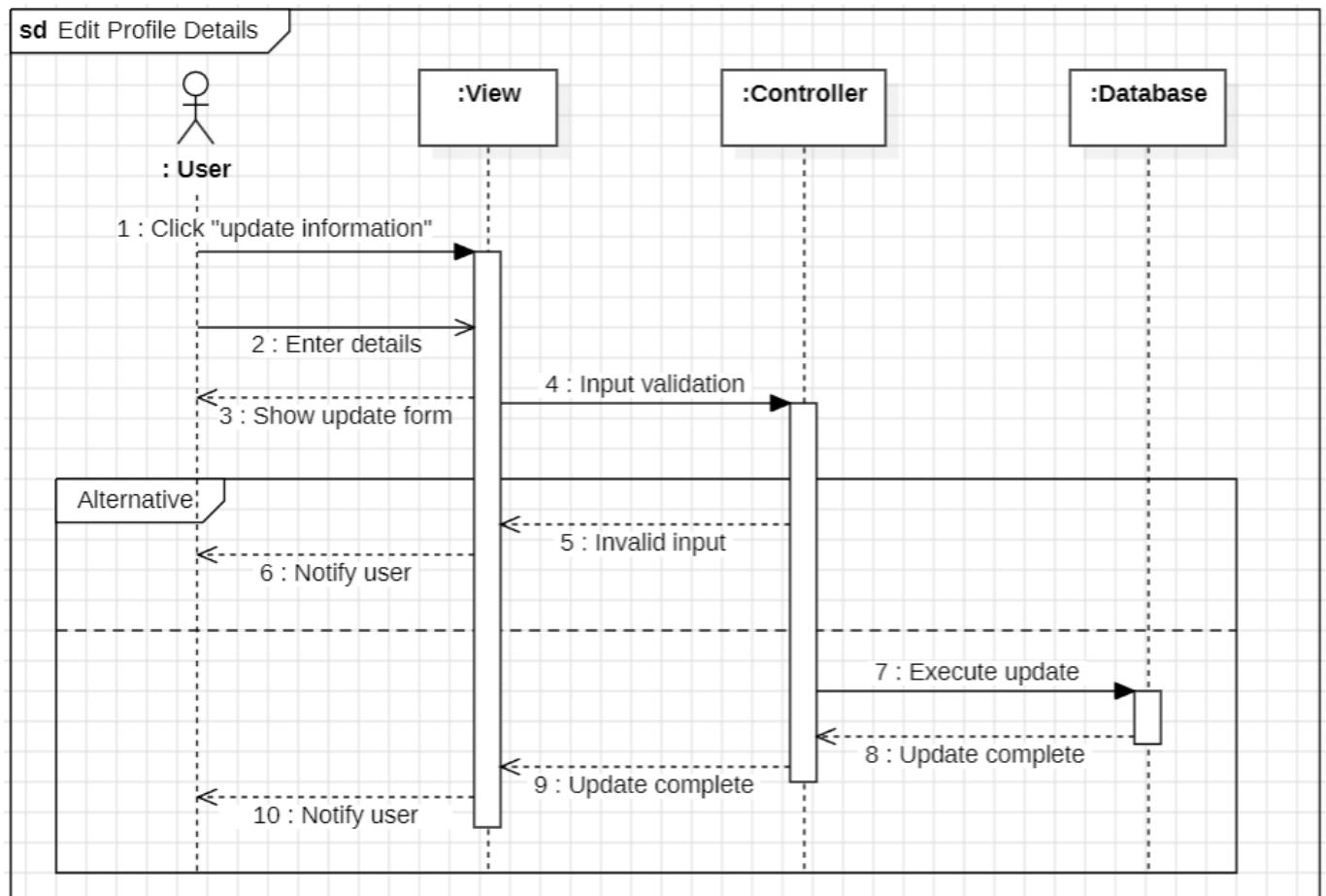


Figure 37. Sequence Diagram for Update Profile Details

3.3.7 Sequence Diagram for Borrow Book

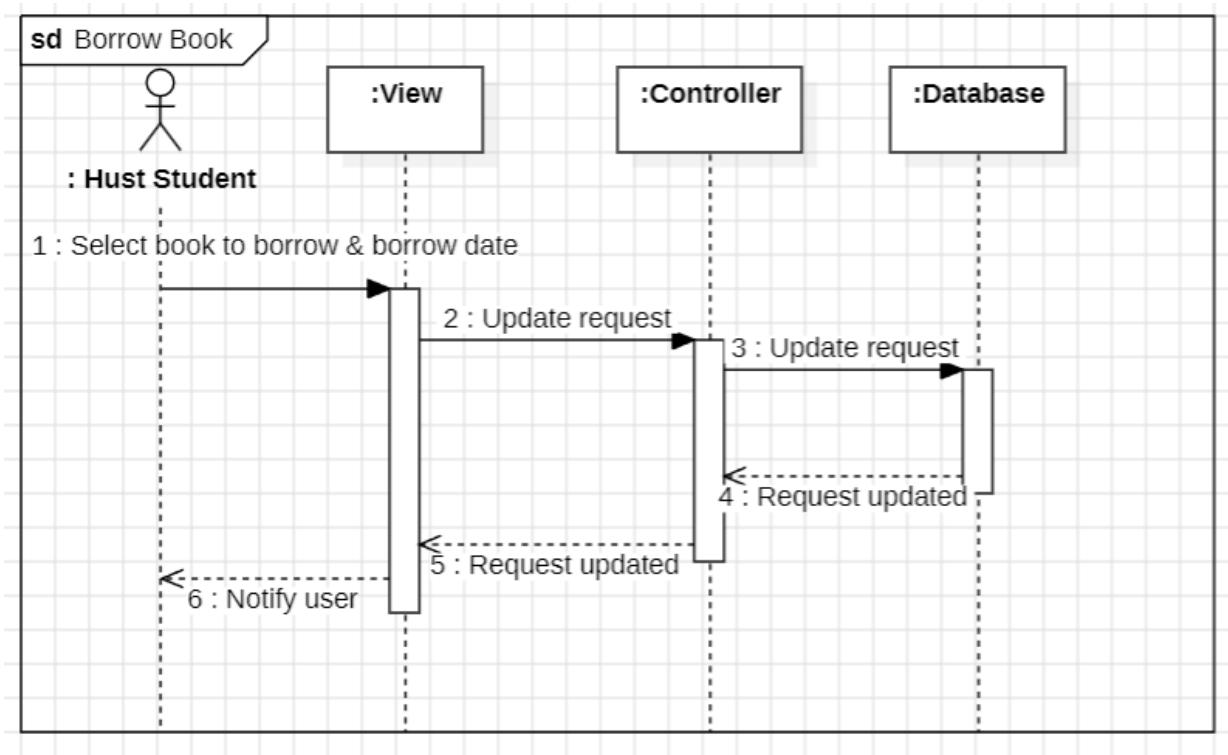


Figure 38. Sequence Diagram for Borrow Book

3.3.8 Sequence Diagram for View Personal Request

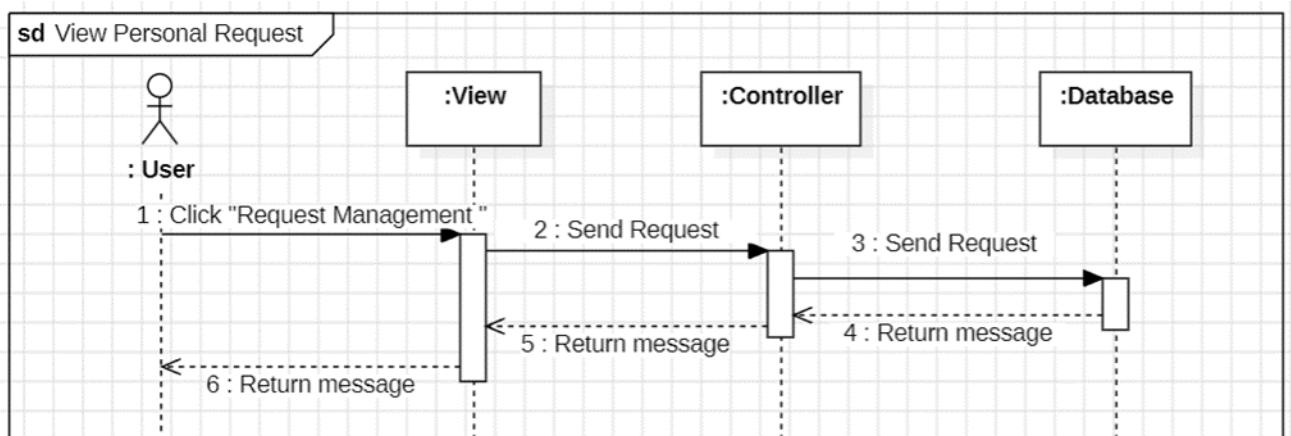


Figure 39. Sequence Diagram for View Personal Request

3.3.9 Sequence Diagram for Search Personal Request

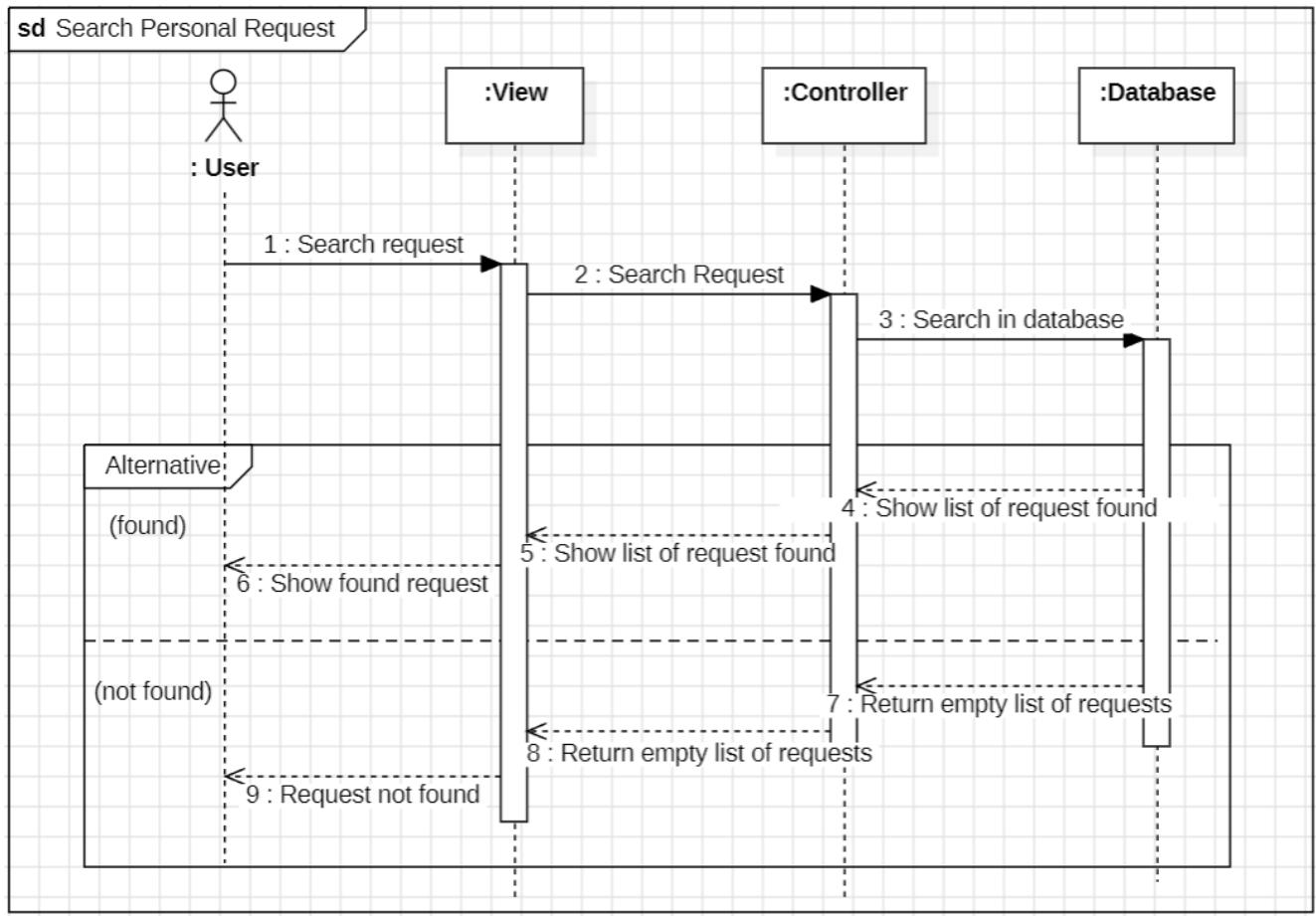


Figure 40. Sequence Diagram for Seach Personal Request

3.3.10 Sequence Diagram for Cancel Personal Request

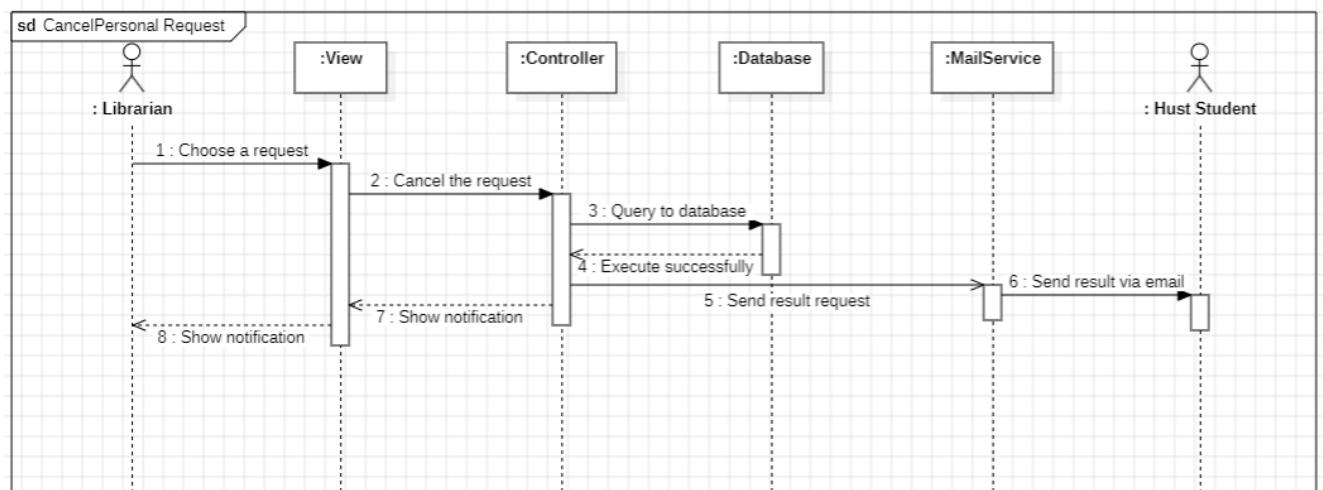


Figure 41. Sequence Diagram for Cancel Personal Request

3.3.11 Sequence Diagram for Add Book

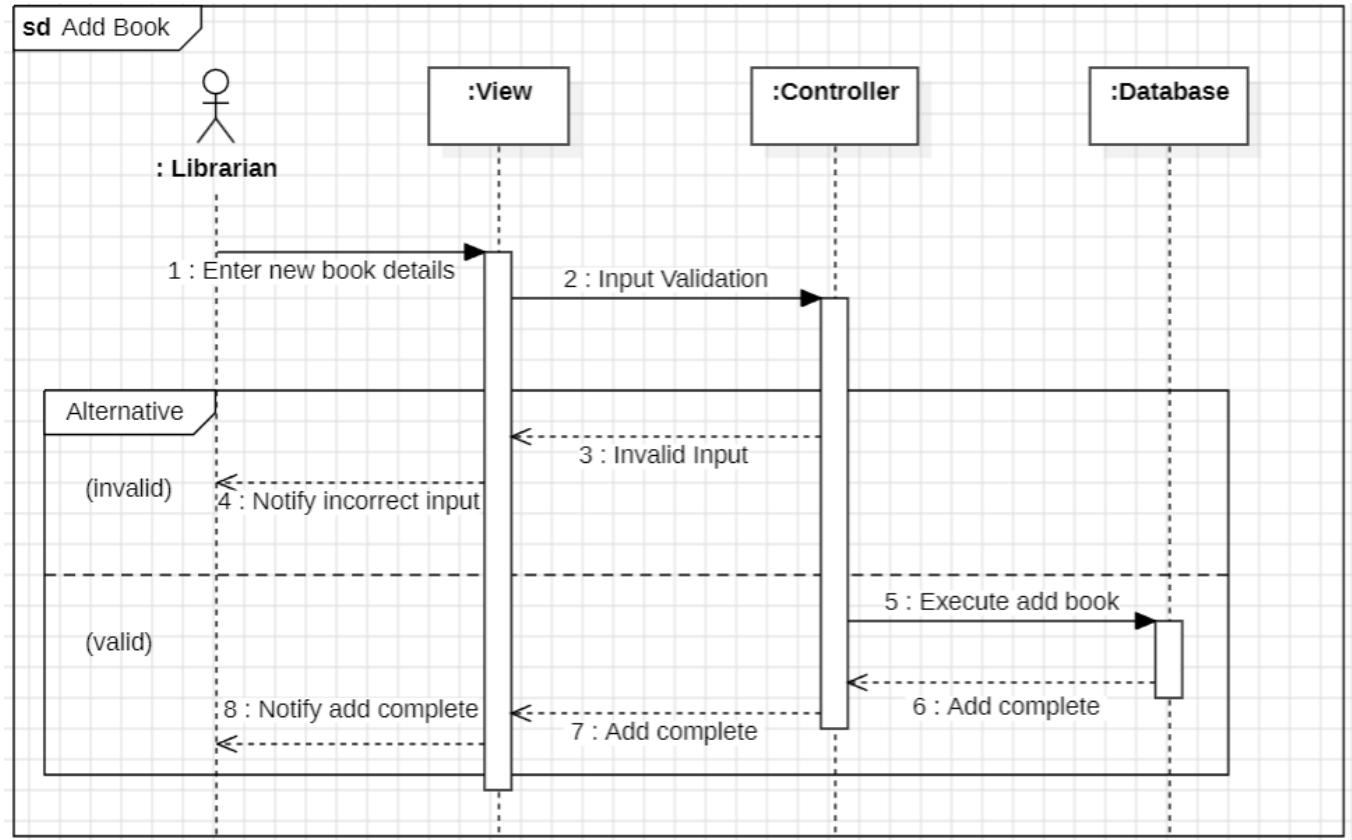


Figure 42. Sequence Diagram for Add Book

3.3.12 Sequence Diagram for Search Book

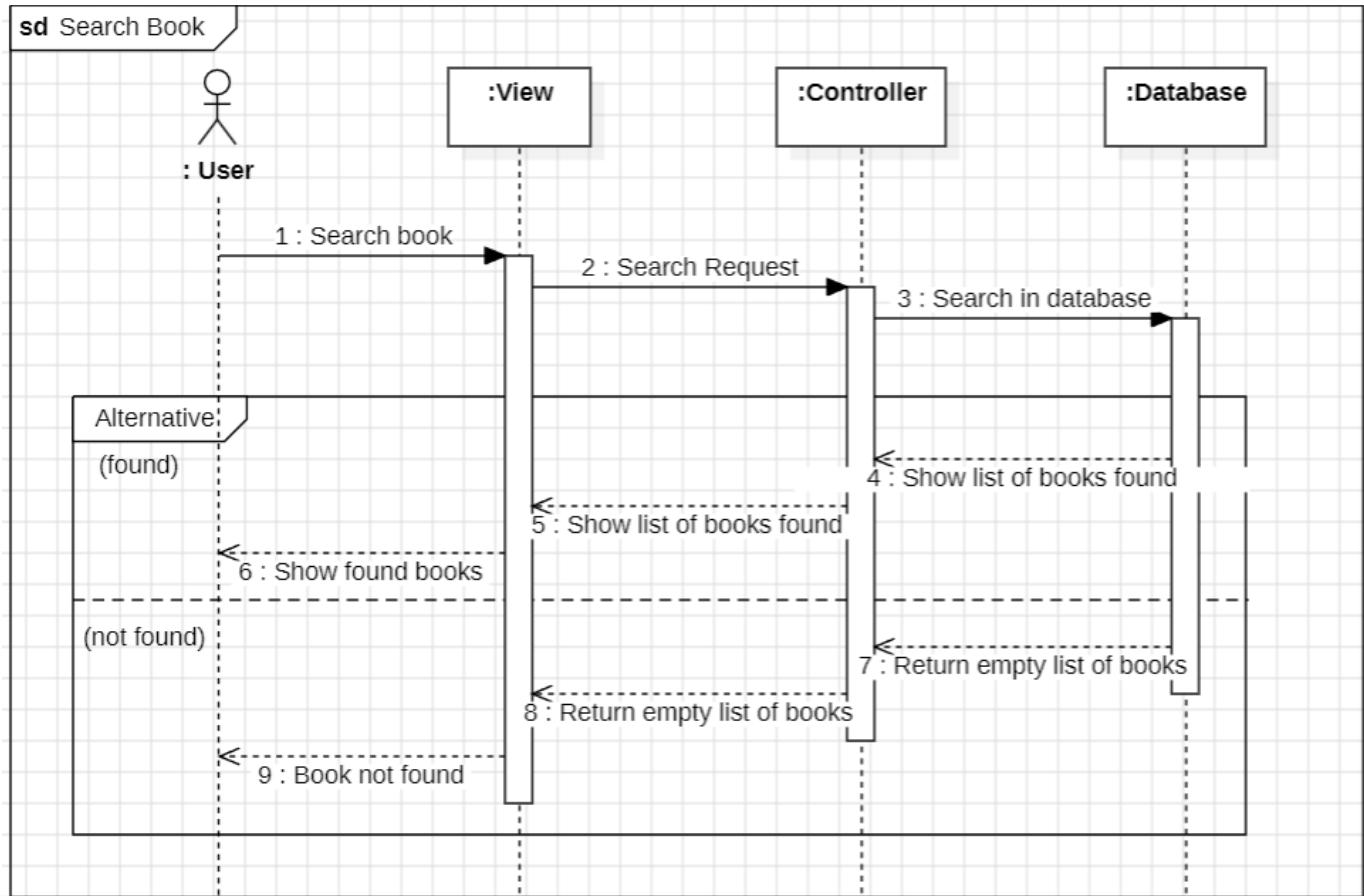


Figure 43. Sequence Diagram for Search Book

3.3.13 Sequence Diagram for Update Book Details

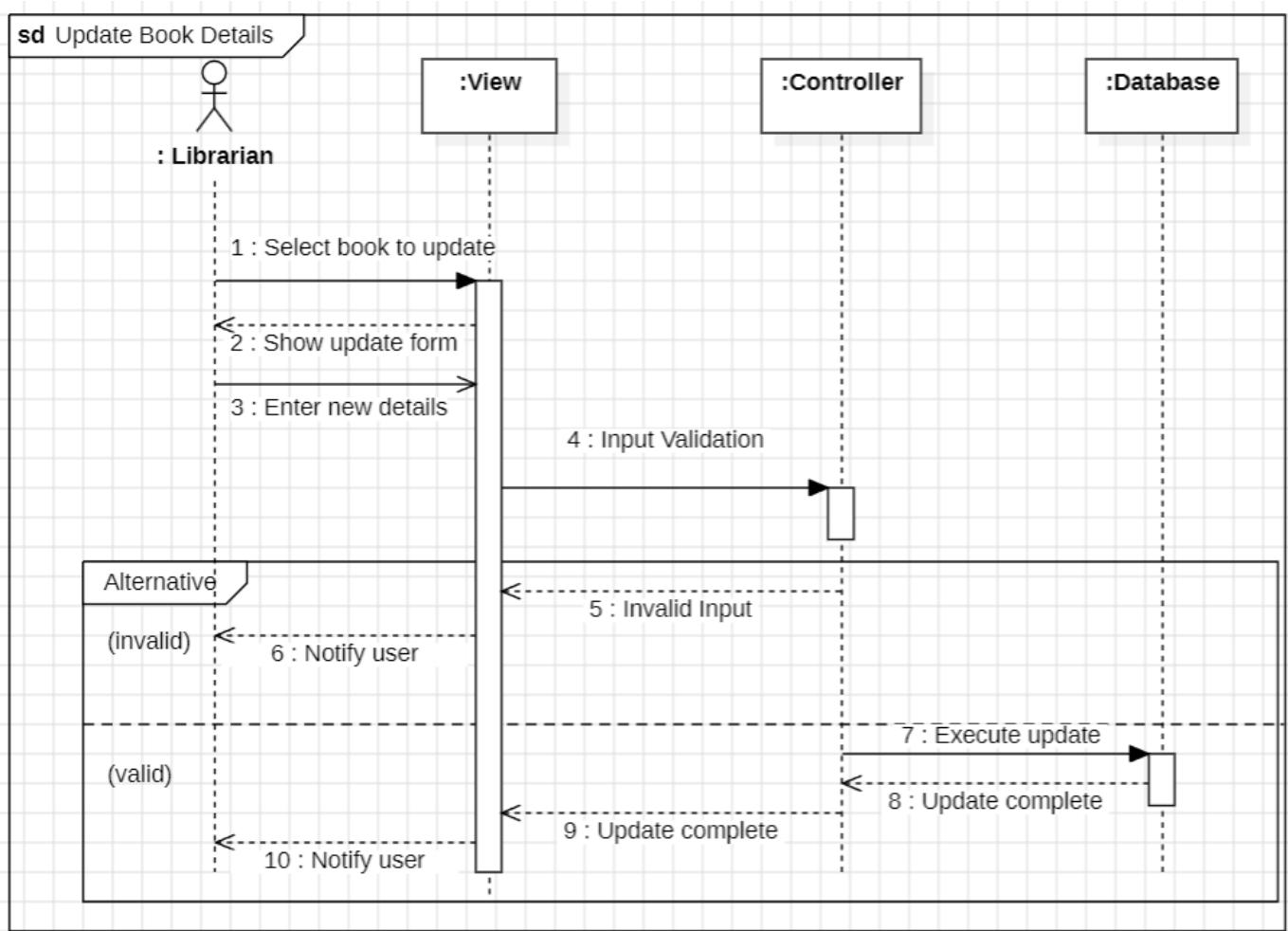


Figure 44. Sequence Diagram for Update Book Details

3.3.14 Sequence Diagram for Add Reader

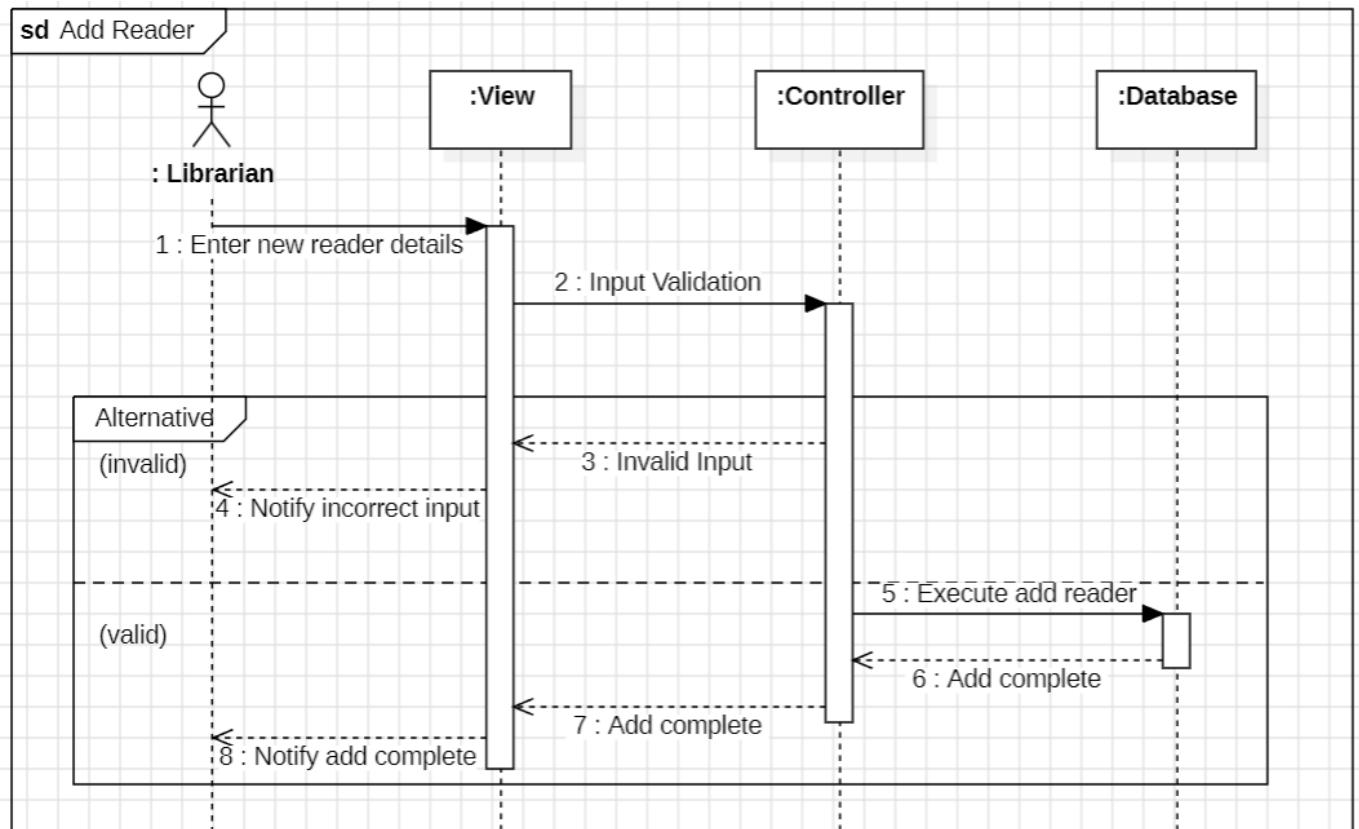


Figure 45. Sequence Diagram for Add Reader

3.3.15 Sequence Diagram for Remove Reader

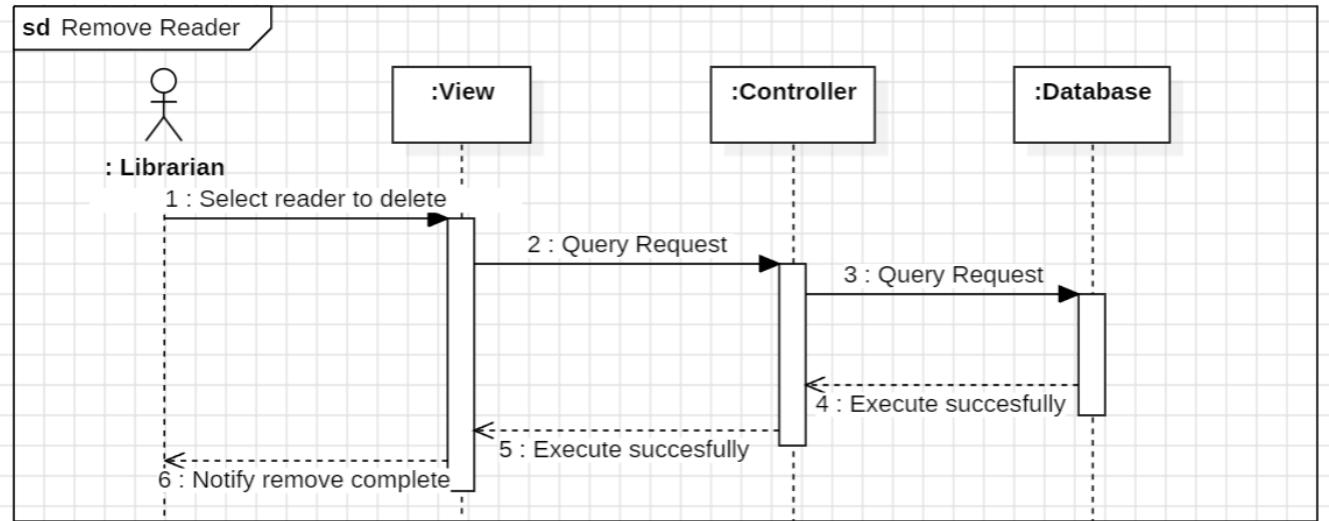


Figure 46. Sequence Diagram for Remove Reader

3.3.16 Sequence Diagram for Update Reader Details

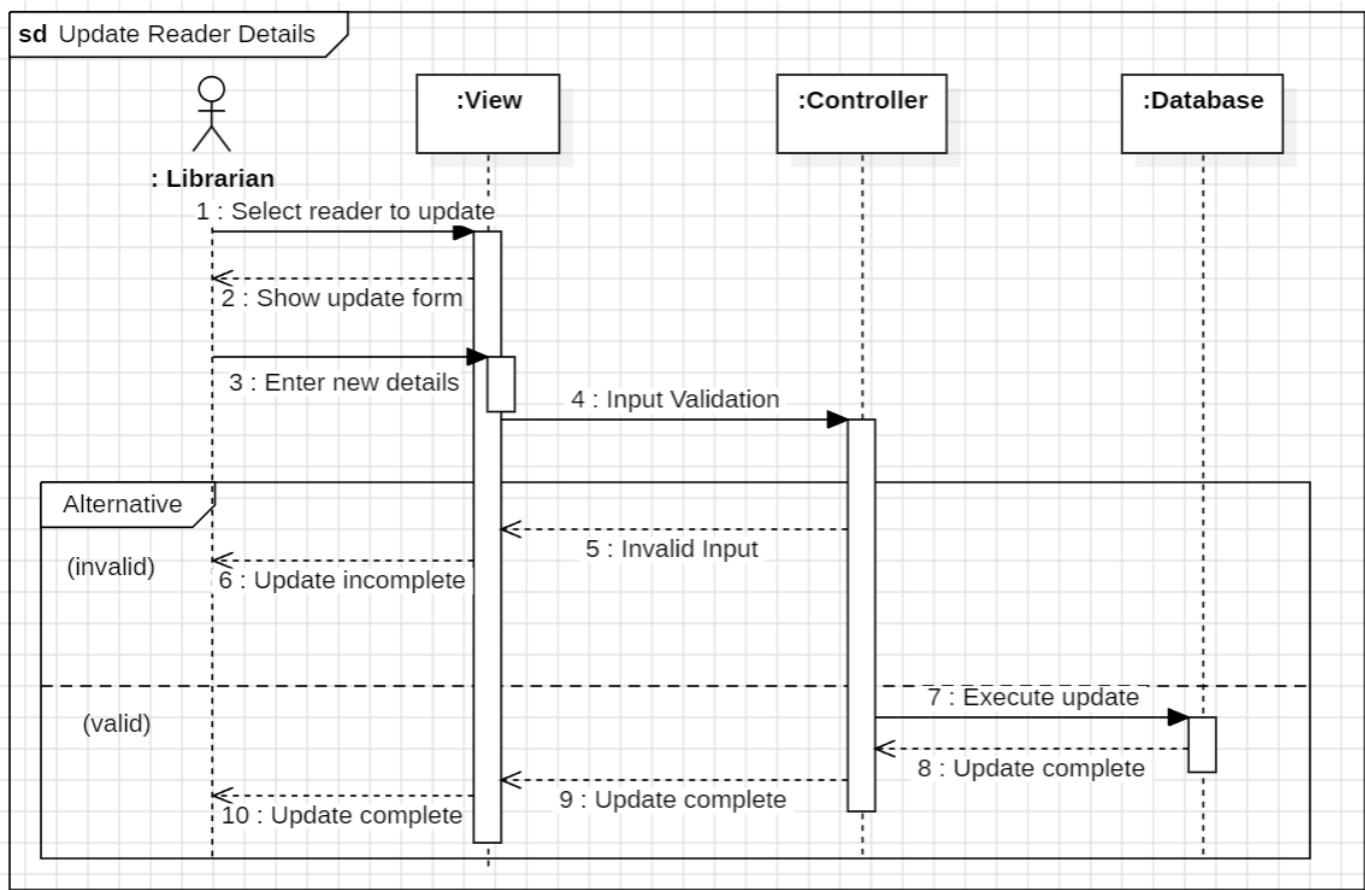


Figure 47. Sequence Diagram for Update Reader Details

3.3.17 Sequence Diagram for Search Reader

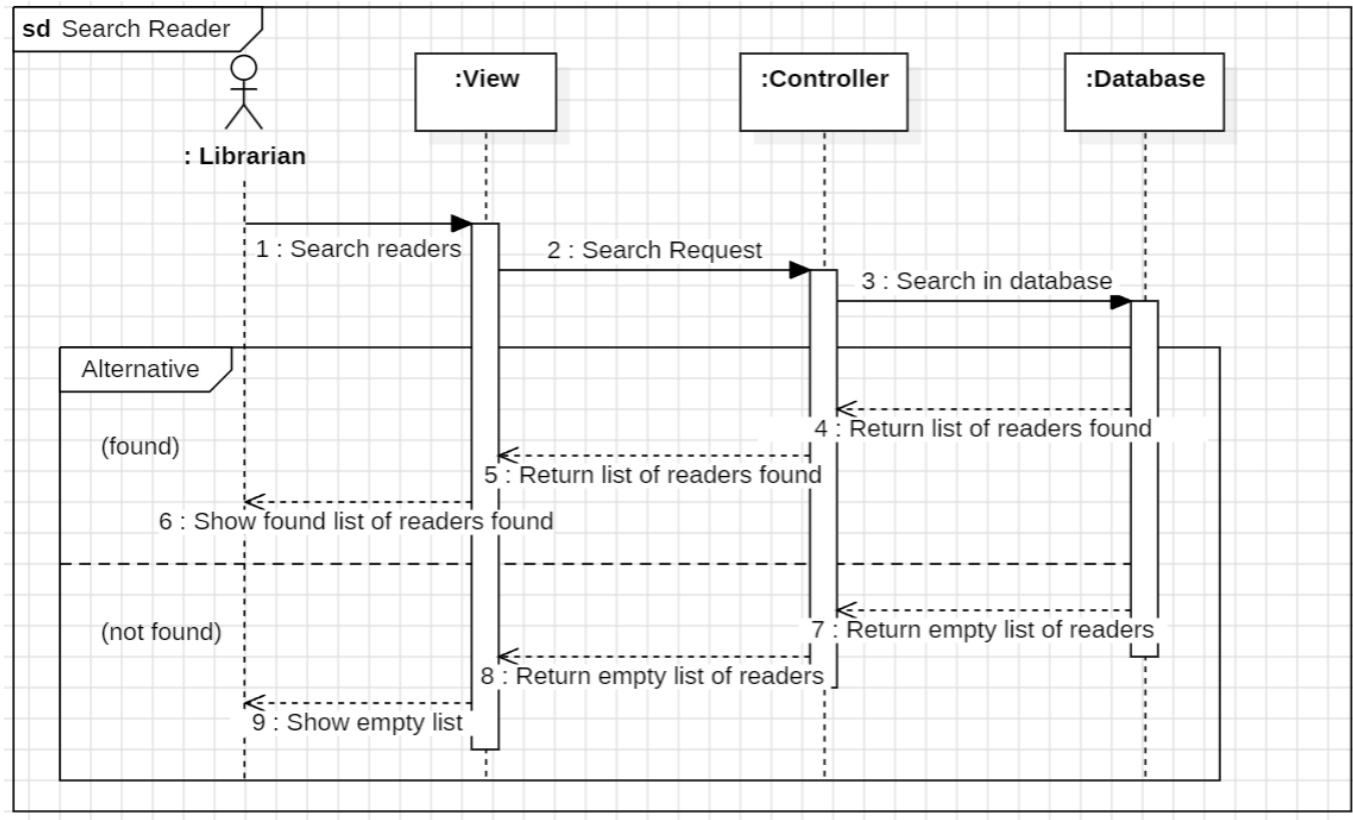


Figure 48. Sequence Diagram for Search Reader

3.3.18 Sequence Diagram for Ban/Unban Reader

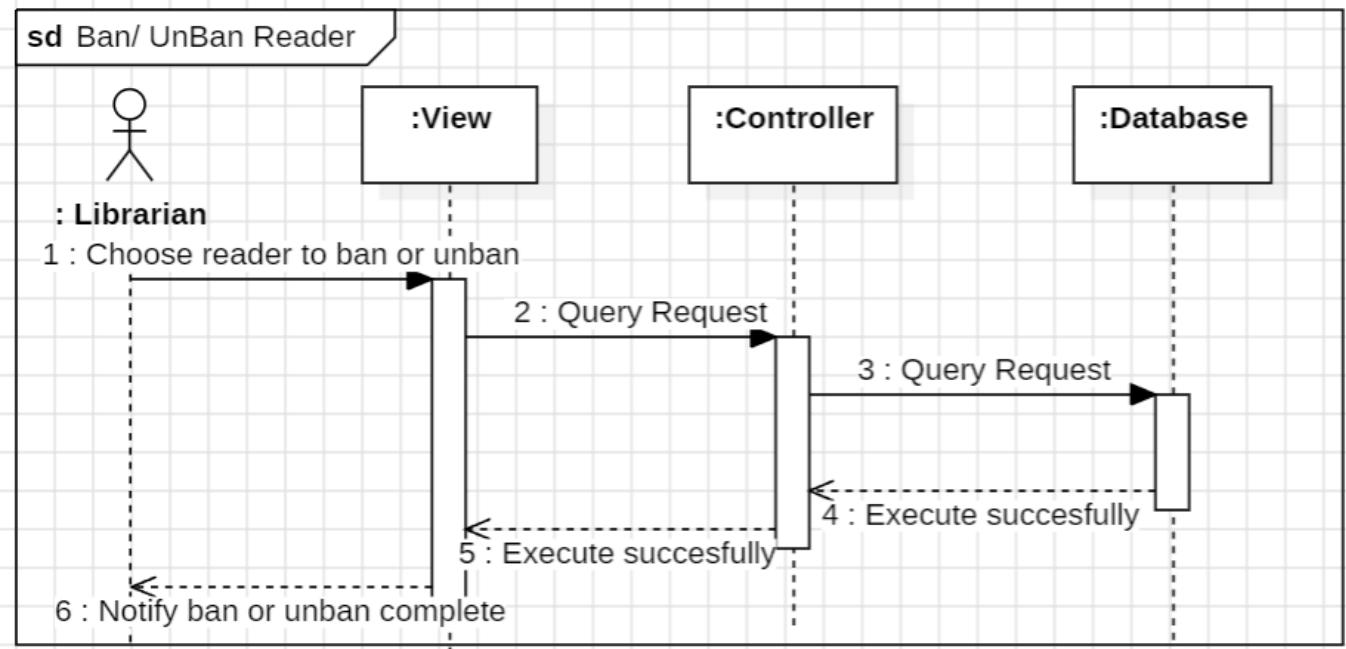


Figure 49. Sequence Diagram for Ban/Unban Reader

3.3.19 Sequence Diagram for (Student) View History of Borrowing and Returning Book

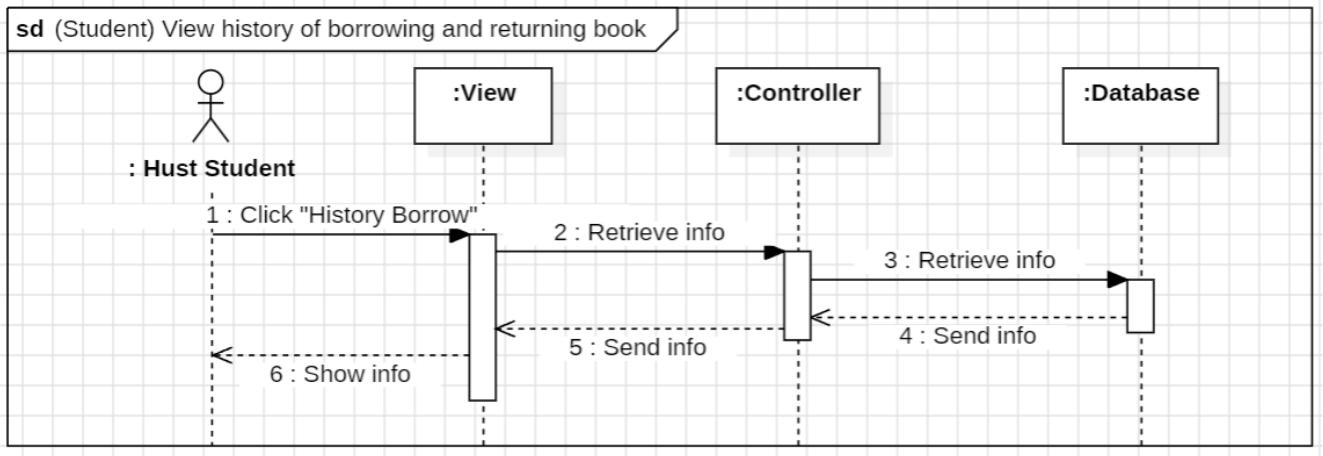


Figure 50. Sequence Diagram for (Student) View History of Borrowing and Returning Book

3.3.20 Sequence Diagram for (Librarian) View History of Borrowing and Returning Book

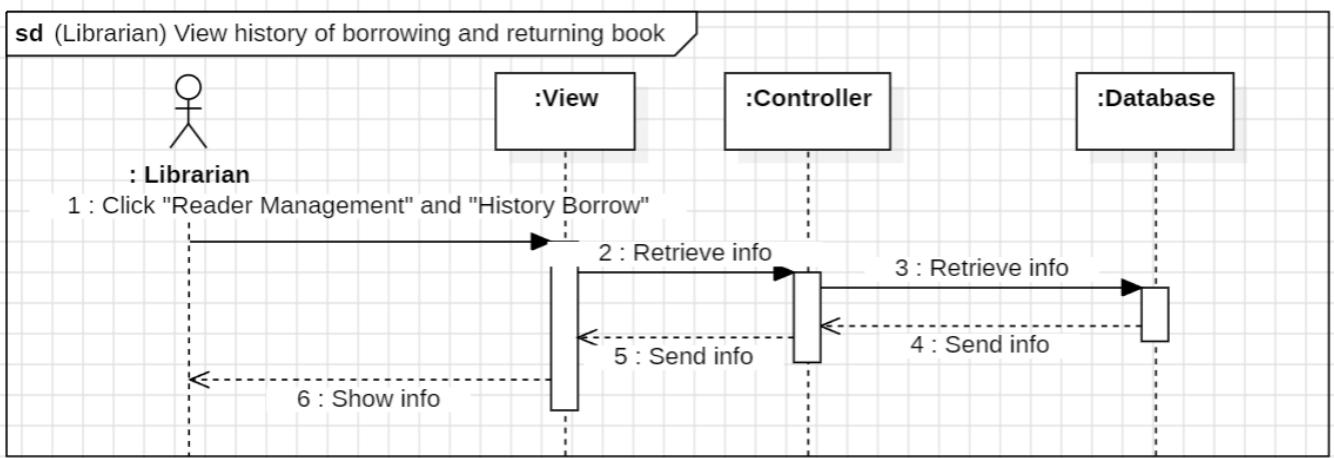


Figure 51. Sequence Diagram for (Librarian) View History of Borrowing and Returning Book

3.3.21 Sequence Diagram for View Personal Request

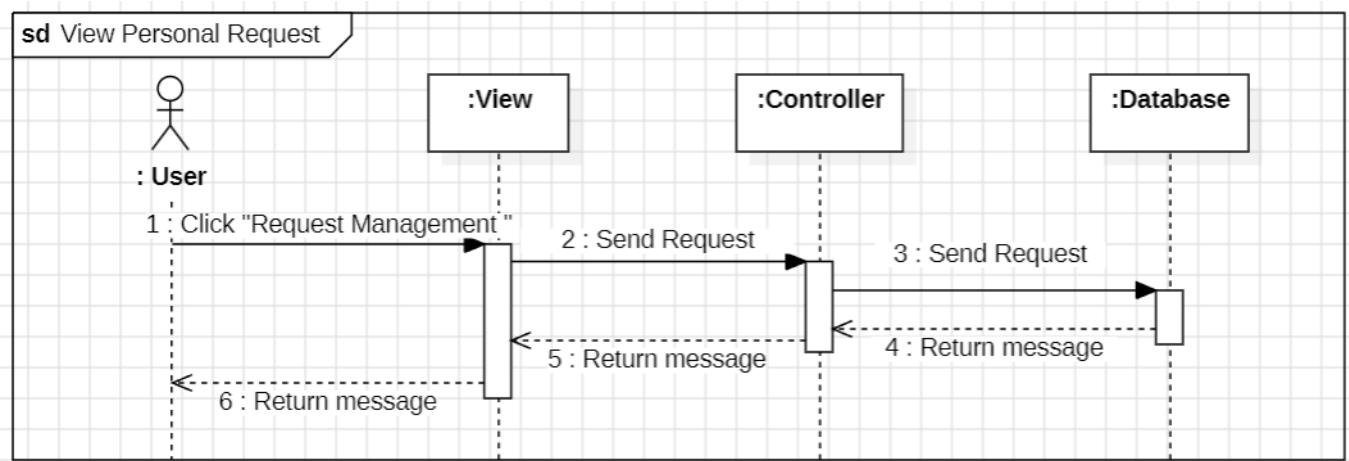


Figure 52. Sequence Diagram for View Personal Request

3.3.22 Sequence Diagram for View Statistic

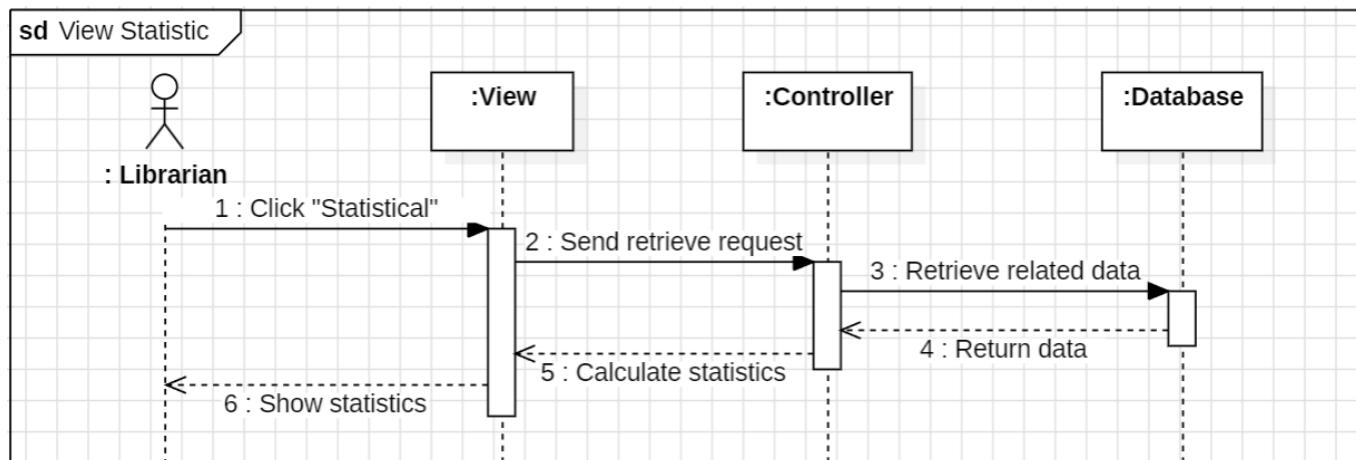


Figure 53. Sequence Diagram for View Statistic

3.4 Database Design

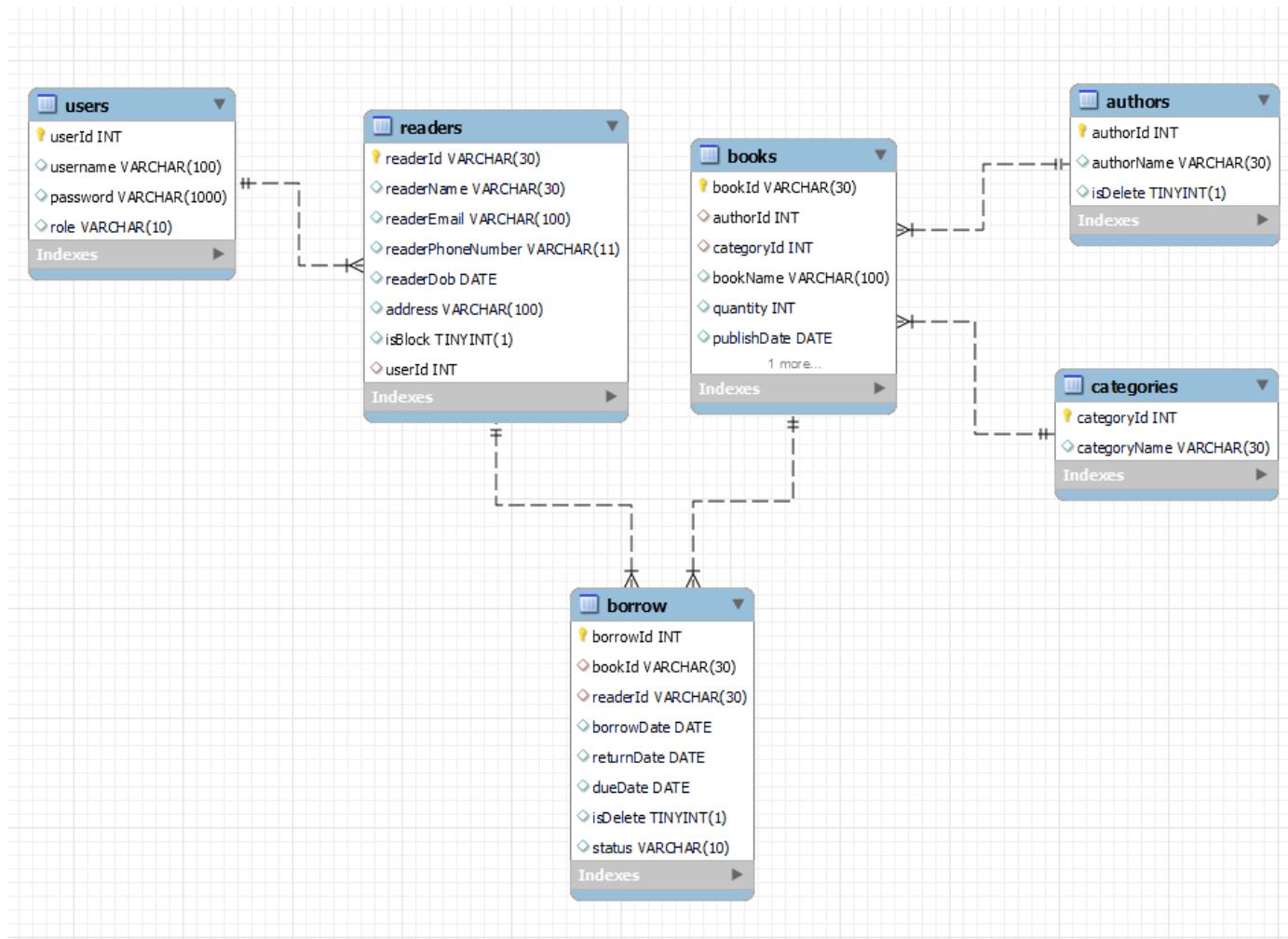


Figure 54. Database Diagram

3.4.1 users

Field Name	Data Type	Description
userId	int	Account ID
username	varchar(100)	Username
password	varchar(1000)	Password
role	varchar(10)	Role of each user (Reader/Librarian)

3.4.2 readers

Field Name	Data Type	Description
readerId	varchar(30)	Identification Code for each reader
readerName	varchar(30)	Reader's Name
readerEmail	varchar(30)	Reader's Email
readerPhoneNumber	varchar(11)	Reader's Phone Number
readerDob	date	Reader's Date of Birth
address	varchar(100)	Reader's Address
isBlock	tinyint(1)	Flag to indicate if the reader has been deleted.
userId	int	Account ID

3.4.3 books

Field Name	Data Type	Description
bookId	varchar(30)	The ID of each book
authorId	int	Author of each book
categoryId	int	Category of each book
bookName	varchar(100)	Title of each book
quantity	int	Quantity of each book
publishDate	date	Published Date of each book
isDelete	tinyint(1)	Flag to indicate if the book has been deleted.

3.4.4 categories

Field Name	Data Type	Description
categoryId	int	The ID for each category
categoryName	varchar(30)	Name of a category

3.4.5 authors

Field Name	Data Type	Description
authorId	int	The ID for each author
authorName	varchar(30)	Author's Name
isDelete	tinyint(1)	Flag to indicate if the author has been deleted.

3.4.6 borrow

Field Name	Data Type	Description
borrowId	int	The ID for each request to borrow book from the reader
bookId	varchar(30)	The ID of the requested-to-borrow book
readerId	varchar(30)	The ID of the reader who sends a request to borrow books
borrowDate	date	The date that the reader asks to borrow the book
returnDate	date	The date that the reader returns the book
dueDate	date	The date that the reader promises to return the book
isDelete	tinyint (1)	Flag to indicate if the request has been deleted.
status	varchar(10)	Status of request (approved or rejected)

3.6 Package Diagram

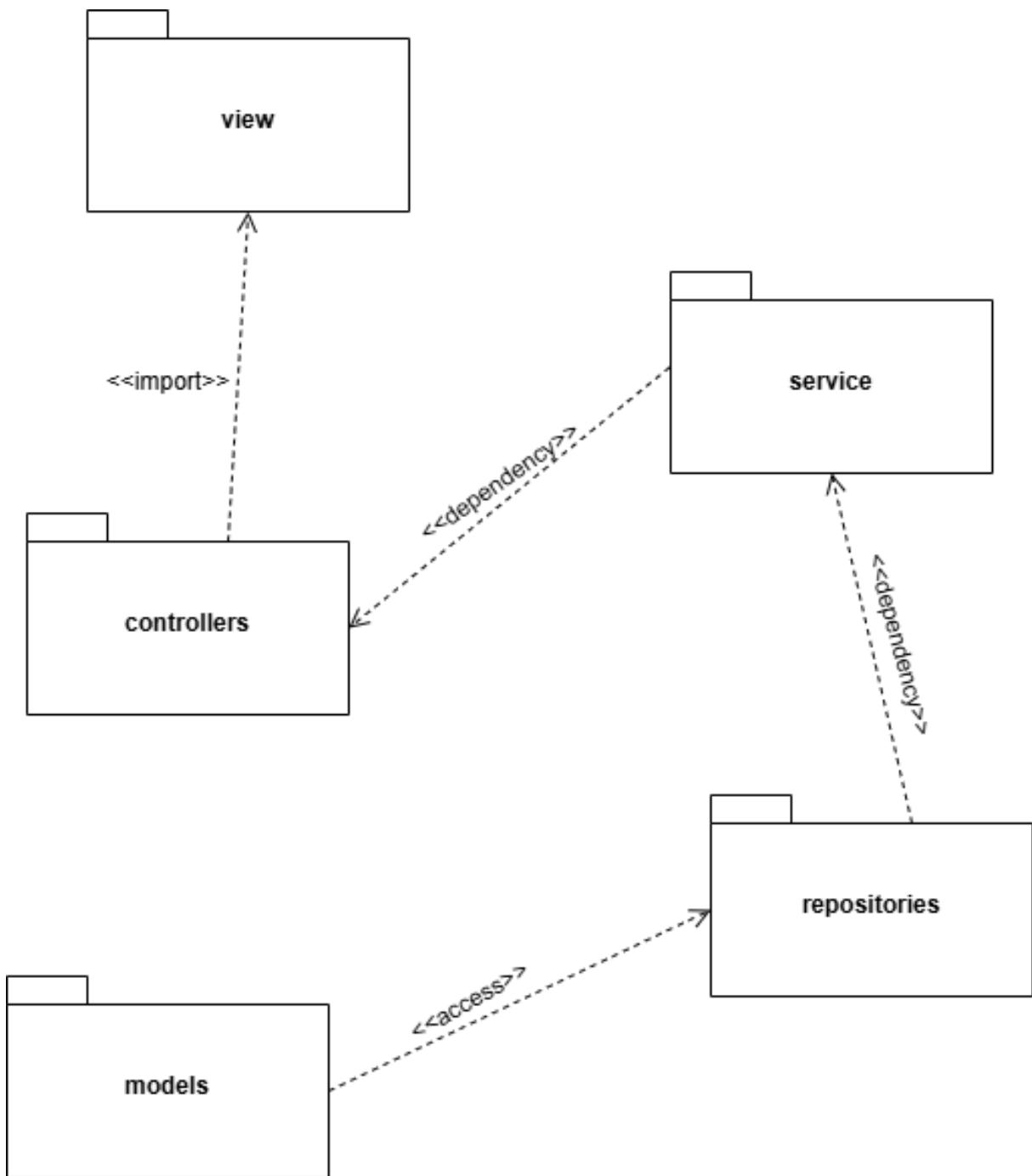


Figure 55. Package Diagram

4 Software Testing

4.1 Overall Description

4.1.1 Testing Levels

- Testing Level: System Test, User Acceptance Test
- Testing Types: Functional Testing, user Interface Testing

4.1.2 Test Plan

Type of Test	State of Test		
	Unit	System	Acceptance
Functional Testing		x	
User interface testing		x	

4.1.3 Test Scope

The scope of the Library Management Software Project encompasses the design, development, and deployment of a comprehensive system to manage all library operations. This includes modules for cataloging books, tracking inventory, managing member accounts, handling loans and returns, and generating reports. The project will integrate a user-friendly interface for both library staff and patrons, ensuring ease of use and accessibility. Additionally, the software will be scalable to accommodate future growth and adaptable to various types of libraries. The project will also include the testing and quality assurance phases to ensure the software's reliability and functionality before full deployment.

4.1.4 Test item

Reader features:

Feature	Parent Component	Overview
Log in	Log in UI	The user logs in to the system by a reader account
Create new account	Log in UI	The user creates a new reader account
Get new password	Log in UI	The user gets a new password for existing account via email
Update information	Information UI	The user updating information of themselves

Search book	Available book UI	The user uses name to search all available books
Requesting to borrow a book	Available book UI	The user makes a request to the admin for borrowing a book
View the history of borrowed books	History borrow UI	The user views their history of borrowing
View current book borrow request	Request borrow UI	

Librarian features:

Log in	Log in UI	The librarian logs in to the system with the librarian account
Update information	Information UI	The librarian update information about themselves
Add book	Book management UI	The librarian adds new books to the system
Update book	Book management UI	The librarian updates the book new information to the system
Delete book	Book management UI	The librarian deletes a book from the system
Add reader	Reader management UI	The librarian adds a reader to the system
Delete reader	Reader management UI	The librarian deletes a reader from the system
Update reader	Reader management UI	The librarian changes a reader's information in the system
Ban reader	Reader management UI	The librarian bans a reader from logging in the system
Approve borrowing request	Request management UI	The librarian approves a reader's borrowing books request
View statistical information	Statistical UI	The librarian view statistics in the system generated by users

4.1.5 Testing Risk Register

Risk ID No.	Sum mary	Probability of Occurrence	Customer Impact	Trigger	Mitigation Action	Contingency Action

R001	Inexperienced users interact with new tools	Medium	High	Users and testers struggle with unfamiliar testing tools	Provide training sessions and hands-on practice	Assign experienced mentors to assist the team with tool usage
R002	Legacy modules documented	Low	High	Difficulty understanding and testing legacy code	Allocate time for code review and documentation improvement	Collaborate with original developers or maintainers to resolve issues

4.1.6 Test Approach (Strategy)

- Type of Testing
 - Unit Testing: Will be conducted by the developers on individual components such as UI elements and database interactions. The focus will be on verifying the correctness of each component.
 - Integration Testing: Ensures that the JavaFX UI correctly interacts with the MySQL database. This phase will be crucial to verify data flow and the integrity of database transactions.
 - System Testing: Comprehensive testing of the entire system, covering all user interactions and data operations, to ensure that the software meets all specified requirements.
 - User Acceptance Testing (UAT): Involves library staff and key members to ensure the system meets practical needs and usability standards.
- Testing Prioritization
 - Must-Have Features: Core functionalities like user log in, book management, and borrow/return processes will be tested first.
 - Should-Have Features: Features like user management and advanced search will follow once core functionalities are stable.
 - Could-Have Features: Lower-priority features such as user interface themes or non-critical settings will be tested last.
- Remote Testing and Non-Functional Testing
 - Remote Testing: No remote testing is anticipated for this project as the team is localized.

- Non-Functional Testing: Basic performance tests, such as load and compatibility testing, will be run towards the end of the testing cycle to ensure the software performs well under expected conditions.
- Test Tools
 - Database Management Tools: MySQL Workbench for verifying database integrity and executing test queries.
 - Communication Tools: Microsoft Teams for team collaboration and updates.
- Test Data
 - **Test Data Sample:** Test data will be provided by the test team with support from the developers for complex scenarios. Data will be anonymized if sourced from live systems.
 - **Automatic Test Data Generators:** Simple scripts may be developed to reset the database to a known state before each testing cycle.
- Test Environment
 - **Software Setup:** The environment includes the JavaFX application, MySQL database, and necessary Java development tools.
 - **Hardware Requirements:** Standard development machines capable of running the JavaFX UI and MySQL will be used. No special hardware is anticipated.
 - **Technical Publications:** Documentation will be available for all team members, including system architecture, data models, and test cases.

4.1.7 Management and Metrics

- Metrics Collection:
 - **Test Coverage:** Percentage of requirements covered by test cases.
 - **Defect Density:** Number of defects found per module.
 - **Defect Resolution Time:** Average time taken to resolve defects.

4.1.8 Test Estimation

- Test Planning: Estimated at 10% of the overall testing effort, involving the creation of test plans and test cases.
- Test Execution: Estimated at 60% of the testing effort, covering the execution of test cases across all phases.
- Test Reporting: Estimated at 90% of the effort, focusing on documenting test results, creating reports, and communicating with other members of the teams.

These estimates were agreed upon through a team consensus, with input from developers, testers, and the project manager. They feed into the overall project schedule.

4.1.9 Test Deliverables

ID	Test case name	Description	Steps	Expected result	Actual result	Status
TC001	Register a new reader account successfully	The user to create a new reader account.	1. The user clicks on “Create new account” 2. The user fills out the given form correctly 3. The user clicks “sign up”	1. The system displays the message “Sign up successfully” 2. A new account is created in the database	✓	Pass
TC002	Register a new reader account with not enough info	The user to create a new reader account.	1. The user clicks on “Create new account” 2. The user fills out the given form with missing field(s) 3. The user clicks “sign up”	1. The system displays the message “All fields are required” 2. The system prevents the click of “sign up” if DOB field is not filled	✓	Pass
TC003	Register a new reader account with not existing info	The user to create a new reader account.	1. The user clicks on “Create new account” 2. The user fills out the given form with existing information available in the database in that field(s) 3. The user clicks “sign up”	The system displays the message “Username already existed”, “Email already existed “or “Phone Number already existed “. The system then prompts the user to re-enter the form.	✓	Pass
TC004	Logs in to an existing account successfully	The user (librarian and reader) logs in to an existing	1. The user chose their role 2. The user fills out the correct	The system directs the user to their respective dashboard based on their role.	✓	Pass

		account successfully	username and password 3. The user clicks “login”			
TC005	Logs in to an existing account failed	The user (librarian and reader) logs in to an existing account failed	1. The user chose their role 2. The user fills out the wrong username and password 3. The user clicks “login”	The system displays an error message indicating the credentials are invalid and prompts the user to try again.	✓	Pass
TC006	Reset password successfully	The user reset the password of a created account via email in the log in UI.	1. The user clicks on “Forgot your password?” 2. The user enters their email 3. Click “submit”	The system sends a new 8-characters-long password to this email	✓	Pass
TC006	Reset password successfully	The user reset the password of a created account via email in the log-in UI.	1. The user clicks on “Forgot your password?” 2. The user enters their email 3. Click “submit”	The system sends a new 8-characters-long password to this email	✓	Pass
TC007	Reset password failed	The user reset the password of a created account via email in the log-in UI.	1. The user clicks on “Forgot your password?” 2. The user enters a wrong email 3. Click “submit”	The system displays an error message and prompts the user to enter valid email.	✓	Pass
TC008	Change password successfully	The user changes their password successfully	1. The user clicks on “Change password” 2. Enter valid passwords	The user's password is successfully updated, and they can log in with the new password.	✓	Pass
TC009	Change password failed	The user enters the wrong current password	1. The user clicks on “Change password” 2. Enter the wrong existing passwords	The user is shown message “Incorrect current password”	✓	Pass

TC010	Change password failed	The user enters an invalid new password	1. The user clicks on “Change password” 2. Enter an invalid new password	The user is prompted to enter a valid password.	✓	Pass
TC011	Change password failed	The user re-enters new password failed	1. The user clicks on “Change password” 2. Re-enter new password doesn’t match	The user shows the error message and prompts the user to re-enter the field “Re-entering new password”	✓	Pass
TC012	Log out	The user logs out	1. The user selects the "Log Out" option from the “System” section	The system redirects the user to the login page or displays a log-out confirmation page.	✓	Pass
TC013	View profile	The user views their own profile	1. The user selects the "Information" option on the homepage.	The system retrieves the user’s profile information from the database. Then the system displays the user’s profile details, including Full name, ID, Birth, Address, email, phone number	✓	Pass
TC014	Edit profile successfully	The user changes their own profile	1. The user selects the "Information" option on the homepage. 2. The user clicks on “Update information” 3. The user edits necessary fields in the form 4. The user submits the form by clicking on “Save”	The system validates the input and updates the profile information in the database.	✓	Pass

TC014	Edit profile failed	The user changes their own profile failed	<ol style="list-style-type: none"> 1. The user selects the "Information" option on the homepage. 2. The user clicks on "Update information" 3. The user filled wrong information (e.g., incorrect email format) 4. The user submits the form by clicking on "Save" 	The system displays an error message, and then asks the user to correct the input and resubmit.	✓	Pass
TC015	View notifications	The user views their own profile notifications	<ol style="list-style-type: none"> 1. The user successfully logs in to the system 	The system displays necessary notifications	✓	Pass
TC016	Ask for borrowing book	The user asks for borrowing book	<ol style="list-style-type: none"> 1. The user clicks on "Available book" section 2. The user searches and selects the desired books. 3. The user enters the date to return the book 4. The user clicks on "Request borrow" 	The borrowing request is successfully sent and awaits approval by the librarian.	✓	Pass
TC017	Ask for borrowing book failed	The user asks for borrowing book failed	<ol style="list-style-type: none"> 1. The user clicks on "Available book" section 2. The user searches and selects the desired books. 3. The user enters the date to return the book (invalid date) 	The system displays an error message highlighting the issues and prompts the user to correct them.	✓	Pass

			4. The user clicks on “Request borrow”			
TC018	View all personal requests	The user views all personal requests	The student clicks on “Manage personal requests” on the homepage	The system displays the history of requests to borrow the book	✓	Pass
TC019	Search personal requests	The user searches a personal request	1. The student clicks on “Manage personal requests” on the homepage 2. The student enters a keyword	The history of related requests is shown.	✓	Pass
TC020	Search personal requests failed	The user searches a personal request	1. The student clicks on “Manage personal requests” on the homepage 2. The student enters a keyword	The system displays the messages “No request found”, and prompts the user to enter another keyword.	✓	Pass
TC021	Cancel personal requests	The user cancels personal requests	1. The student clicks on “Manage personal requests” on the homepage 2. The student selects one or more than one desired request to cancel 3. The student clicks on “Cancel request” 4. The student clicks on “Ok”	The system removes information of the selected requests from the database	✓	Pass
TC021	Cancel personal requests	The user cancels personal requests	1. The student clicks on “Manage personal requests” on the homepage 2. The student selects one or more than one desired request to cancel	The system removes information of the selected requests from the database	✓	Pass

			3. The student clicks on “Cancel request” 4. The student clicks on “Ok”			
TC021	Add new book	The librarian adds a new book	1. The librarian clicks on “Book management” in the homepage 2. The librarian fills out the given form 3. The librarian clicks on “Add book”	The system validates the submitted form, and then the system informs “Added new book”	✓	Pass
TC022	Add new book failed	The librarian adds a new book failed	1. The librarian clicks on “Book management” in the homepage 2. The librarian fills out the given form 3. The librarian clicks on “Add book”	The system displays an error message highlighting the issues and prompts the librarian to correct them.	✓	Pass
TC023	Remove book	The librarian removes a book	1. The librarian clicks on “Book management” in the homepage 2. The librarian searches and selects for the book in the 3. The librarian clicks on “Remove book”. 4. The librarian confirms the removal.	The books are successfully removed from the system	✓	Pass
TC024	Update book details	The librarian updates a book detail	1. The librarian clicks on “Book management” in the homepage	The system validates the submitted form, then the system updates the book	✓	Pass

			<p>2. The librarian searches and selects for the book in the</p> <p>3. The librarian updates the necessary fields by filling out the given form.</p> <p>4. The librarian clicks on “Update book”</p> <p>5. The librarian confirms updating the book by clicking on “Ok”</p>	details in the database, and then sends the message “Updated book”.		
TC024	Update book details failed	The librarian enters incomplete or invalid details while updating the book.	<p>1. The librarian clicks on “Book management” in the homepage</p> <p>2. The librarian searches and selects for the book in the</p> <p>3. The librarian updates the necessary fields by filling out the given form.</p> <p>4. The librarian clicks on “Update book”</p> <p>5. The librarian confirms updating the book by clicking on “Ok”</p>	The system displays an error message highlighting the issues and prompts the librarian to correct them.	✓	Pass
TC025	Update book details failed	The librarian enters incomplete or invalid details while updating the book.	<p>1. The librarian clicks on “Book management” in the homepage</p> <p>2. The librarian searches and selects for the book in the</p>	The system displays an error message highlighting the issues and prompts the librarian to correct them.	✓	Pass

			<p>3. The librarian updates the necessary fields by filling out the given form.</p> <p>4. The librarian clicks on “Update book”</p> <p>5. The librarian confirms updating the book by clicking on “Ok”</p>			
TC026	Search book	Users (librarians, students) search for books within the library system.	<p>1. The librarian clicks on “Book management” / The student clicks on “Available book” in the home page in the homepage</p> <p>2. The user enters the desired search criteria</p> <p>3. The user submits the search request.</p>	The system searches the library's database for books that match the criteria. If found, then the system shows the search results, including details like title, author, availability, etc. If not found, the system shows message “not found”.	✓	Pass
TC027	Search book failed	The user enters invalid or incomplete search criteria.	<p>1. The librarian clicks on “Book management” / The student clicks on “Available book” in the home page in the homepage</p> <p>2. The user enters the desired search criteria</p> <p>3. The user submits the search request.</p>	The system prompts the user to correct the input and resubmit the search.	✓	Pass

TC028	Add reader	The librarian adds a new reader (Hust Student) to the system	1. The librarian clicks on “Reader management” 2. The librarian uses the information of a new reader to fill out the given form. 3 The librarian clicks on “Add reader”	The system creates a new account for this new reader, based on the input Username (password is generated randomly) The system shows message “Add reader”	✓	Pass
TC029	Add reader failed	The librarian entered information that already existed	1. The librarian clicks on “Reader management” 2. The librarian uses the information of a new reader to fill out the given form. 3. The librarian clicks on “Add reader”	The system shows the message “The username existed” The system shows the message “The email existed” The system shows the message “The phone number existed”	✓	Pass
TC030	Remove reader	The librarian removes a reader (Hust Student) from the system	1. The librarian clicks on “Reader management” 2. The librarian selects one reader to remove 3. The librarian clicks on “Remove reader” 4. The librarian clicks on “Ok”	The system removes everything about this reader from the database	✓	Pass
TC031	Update reader	The librarian updates the reader's details	1. The librarian clicks on “Reader management” 2. The librarian selects a reader 3. The librarian edits necessary fields in the given form	The system updates the database The system shows the message “Updated reader”	✓	Pass

			4. The librarian clicks on “Update reader”			
TC032	Update reader failed	The librarian updates the reader's details with invalid information	1. The librarian clicks on “Reader management” 2. The librarian selects a reader 3. The librarian edits necessary fields in the given form 4. The librarian clicks on “Update reader”	The system shows the message “The username existed” The system shows the message “The email existed” The system shows the message “The phone number existed”	✓	Pass
TC033	Search reader	The librarian to search readers	1. The librarian clicks on “Reader management” 2. The librarian enters a keyword to search for a reader's details”	The system shows all related readers The system shows the message “No result found”	✓	Pass
TC034	Ban reader	The librarian bans a reader	1. The librarian clicks on “Reader management” 2. The librarian selects one reader to ban 3. The librarian clicks on “Ban reader” 4. The librarian clicks on “Ok”	The system does not allow the account of this banned user to access the system anymore. The system shows the message “Banned reader”	✓	Pass
TC035	Unban reader	The librarian unbans a reader	1. The librarian clicks on “Reader management” 2. The librarian selects a banned reader to unban 3. The librarian clicks on “Ban reader” 4. The librarian clicks on “Ok”	The system does not allow the account of this banned user to access the system anymore. The system shows the message “The reader is unbanned”	✓	Pass

TC036	View history of borrowing and returning books	The librarian to view the history of borrowing and returning books of readers	<ol style="list-style-type: none"> 1. The librarian clicks on “Reader management” 2. The librarian selects one reader to view his/her history of borrowing and returning book. 3. The librarian clicks on “History borrow” 	The system shows the history of borrowing and returning books of the selected reader	✓	Pass
TC036A	View history of borrowing and returning books	Students view their history of borrowing and returning books	<ol style="list-style-type: none"> 1. The student clicks on “History borrow” 	The system shows the history of borrowing and returning books	✓	Pass
TC037	Issue book	The librarian issues books for corresponding readers, based on their requests to borrow books.	<ol style="list-style-type: none"> 1. The librarian clicks on “Issue book” on the homepage 2. The librarian selects one request 3. The librarian selects one request 	<p>The reader borrows the book successfully</p> <p>The history of borrowing books is recorded</p> <p>The system shows the message “Issued book successfully”</p>	✓	Pass
TC037	Issue book	The librarian issues books for corresponding readers, based on their requests to borrow books.	<ol style="list-style-type: none"> 1. The librarian clicks on “Issue book” on the homepage 2. The librarian selects one request 3. The librarian clicks on “Approve” 	<p>The reader borrows the book successfully</p> <p>The history of borrowing books is recorded</p> <p>The system shows the message “Issued book successfully”</p> <p>The system sends a confirmation message to the corresponding reader’s email.</p>	✓	Pass

				For example, “Book Name, ID, has been approved by the librarian. Please go to the librarian to take the book”		
TC038	Reject book	The librarian rejects books for corresponding readers, based on their requests to borrow books.	1. The librarian clicks on “Issue book” on the homepage 2. The librarian selects one request 3. The librarian clicks on “Reject”	The system sends a rejection message to the corresponding reader’s email. For example, “Book Name, ID, has been rejected.”	✓	Pass
TC039	View requests to borrow book	The librarian views all requests to borrow books from the readers	1. The librarian clicks on “Issue book” on the homepage	The system shows all requests to borrow books from readers	✓	Pass
TC040	Seach requests to borrow book	The librarian searches a request to borrow books from the readers	1. The librarian clicks on “Issue book” on the homepage 2. The librarian enters a keyword to search for the desired requests	The system shows all related requests The system shows the message “No corresponding request found”, and then prompts the librarian to enter another keyword.	✓	Pass
TC040	Seach requests to borrow book	The librarian searches a request to borrow books from the readers	1. The librarian clicks on “Issue book” on the homepage 2. The librarian enters a keyword to search for the desired requests	The system shows all related requests The system shows the message “No corresponding request found”, and then prompts the librarian to enter another keyword.	✓	Pass

TC041	Return book	The librarian returns a book	<ol style="list-style-type: none"> 1. The librarian clicks on “Return book” on the homepage 2. The librarian enters information about the book returned, and the information of the reader who returns the book. 3. The librarian clicks on “Return book” 	The system verifies information of the book returned, and the information of the reader who returns the book.	✓	Pass
TC041	Return book failed	The information of the returner and the information of the book returned do not match	<ol style="list-style-type: none"> 1. The librarian clicks on “Return book” on the homepage 2. The librarian enters information about the book returned, and the information of the reader who returns the book. 3. The librarian clicks on “Return book” 	The system shows the message “Returned book and Returner do not match”, then the system prompts the librarian to enter again.	✓	Pass
TC042	View statistic	Librarian views statistics including the number of readers, available books, books currently borrowed, overdue readers, and return-book-on-time readers.	<p>The librarian clicks on “View statistics” option in the homepage</p>	<p>The system displays the following statistics</p> <ol style="list-style-type: none"> 1. The system shows the number of available books 2. The system shows the number of books currently borrowed 3. The system shows the number of overdue readers 4. The system shows the number 	✓	Pass

				of return-book-on-time readers		
--	--	--	--	--------------------------------	--	--

4.2 User Guides

4.2.1 Deliverables Package

No	Items	Sub - items
1	Code Packages	Source codes and dependency
2	Database	
3	Documentation	Report CS3332.docx

4.2.2 Software Requirements

Component	Name and version	Description
Operating System	Windows 7 SP1/8.1/10	
DBMS	MySQL latest version	Used to manage database
Dependency	Java	The app is Java-based
	Apache Maven	The app uses Maven to manage build
	IDE	Any IDE can be used

4.2.3 Installation Guide

The installation in this manual is going to be on Visual Code Studio

Note that the same step is applied to other IDE

Step 1: Download and open files

```

src > main > java > com > example > library > J Main.java > ...
1 package com.example.library;
2
3 public class Main {
4     Run | Debug
5     public static void main(String[] args) {
6         App.main(args);
7     }
8

```

TERMINAL

```

PS C:\Users\thang\Desktop\Cs3332\project\ktpm>

```

MAVEN

```

> m library com.examplelibrary

```

PROBLEMS (122)

OUTPUT DEBUG CONSOLE TERMINAL PORTS

Ln 8, Col 1 Spaces: 4 UTF-8 CRLF Java

Step 2: Compiling

TERMINAL

```

[INFO] -----[ jar ]-----
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ library ---
[INFO] Deleting c:\Users\thang\Desktop\Cs3332\project\ktpm\target
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.247 s
[INFO] Finished at: 2024-08-19T14:29:40+07:00
[INFO] -----

```

Ln 8, Col 1 Spaces: 4 UTF-8 CRLF Java

- Maven/library/Lifecycle/clean execute button (triangle one on the right)
- Wait for “build success” prompt

```

src > main > java > com > example > library > J Main.java > ...
1 package com.example.library;
2
3 public class Main {
4     Run | Debug
5     public static void main(String[] args) {
6         App.main(args);
7     }
8

```

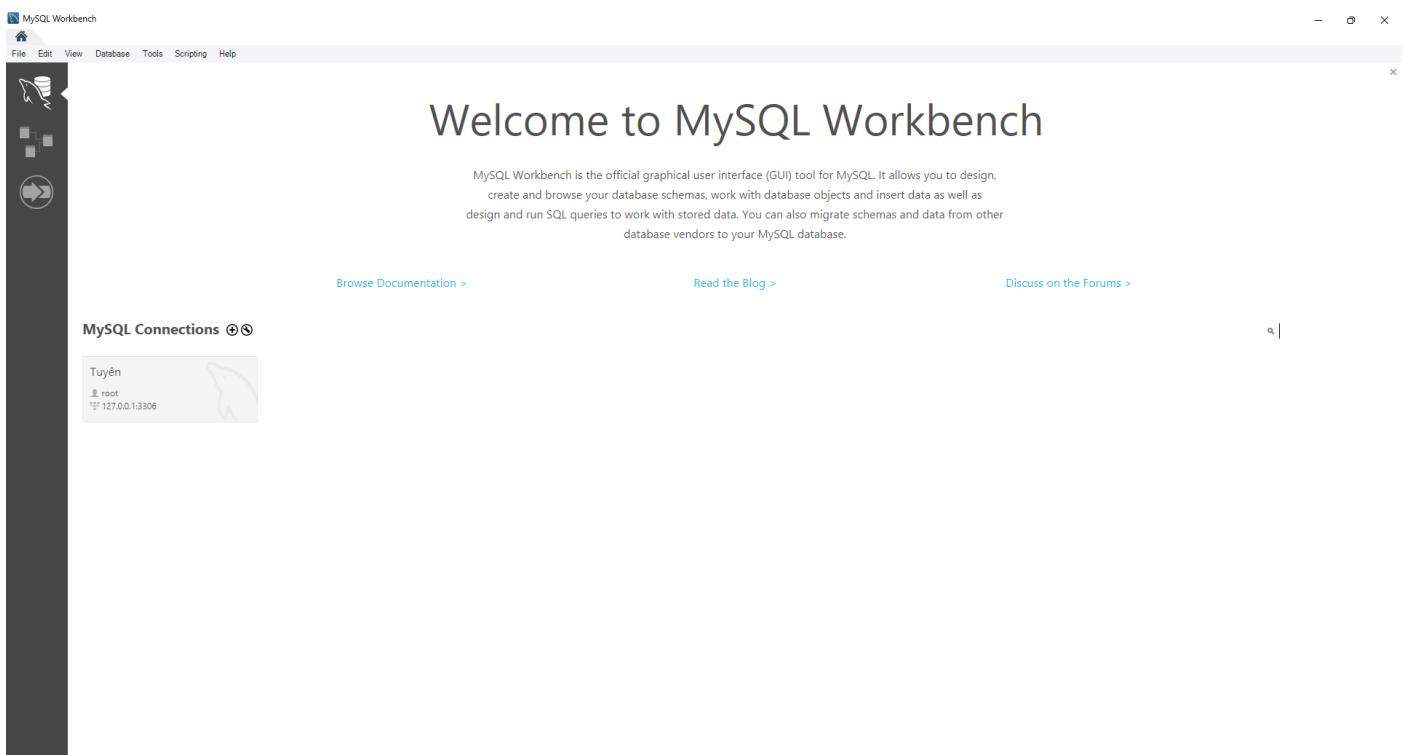
PROBLEMS | 122 OUTPUT DEBUG CONSOLE TERMINAL PORTS

[INFO] ./:/Users/thang/Desktop/Cs3332/project/ktpm/src/main/java/com/example/library/controllers/BorrowManagementController.java: Some input files use unchecked or unsafe operations.
[INFO] ./:/Users/thang/Desktop/Cs3332/project/ktpm/src/main/java/com/example/library/controllers/BorrowManagementController.java: Recompile with -Xlint:unchecked for details.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.676 s
[INFO] Finished at: 2024-08-19T14:31:42+07:00
[INFO] -----

PS C:\Users\thang\Desktop\Cs3332\project\ktpm>

- Maven/library/Lifecycle/compile execute button (triangle one on the right)
- Wait for “build success” prompt

- Step 3: Connect to the SQL database



- Open MySQL workbench
- Establish connection of your account

```

1 package com.example.library.utils;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8
9 public class DbConnect { // singleton pattern
10     private static DbConnect instance;
11
12     private final static String jdbcUrl = "jdbc:mysql://127.0.0.1:3306/library";
13     private final static String username = "root";
14     private final static String password = "kwehfj123";
15
16     private Connection connection;
17
18     private DbConnect() {
19         try {
20             connection = DriverManager.getConnection(jdbcUrl, username, password);
21         } catch (SQLException e) {
22             e.printStackTrace();
23         }
24     }
25
26     public static DbConnect getInstance() {
27         if(instance == null) {

```

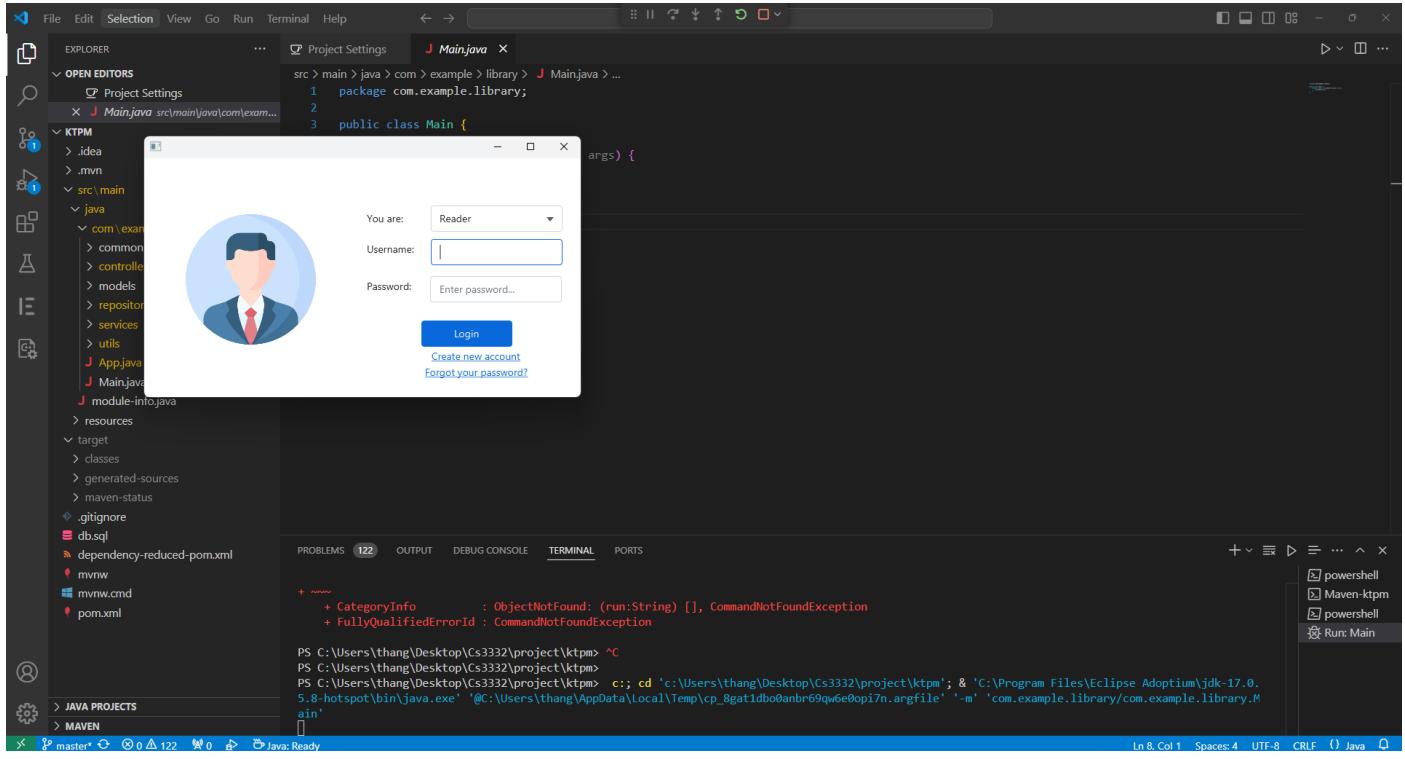
- KTPM/src/java/com/utils/DbConnect.Java
- Change the username to your username (default is “root”)
- Change the passwords to your passwords
- Step 4: Run the app**

```

1 package com.example.library;
2
3 public class Main {
4     public static void main(String[] args) {
5         App.main(args);
6     }
7 }
8

```

- Run the app using run feature in KTPM/src/java/com/main.java or directly by the IDE (VS Code): Run java



5 Evaluation and Results

5.1 Evaluation of Library Management Software

Evaluating the library management software is a key step to ensure it meets all the set requirements and goals. The main criteria used for evaluation include:

5.1.1 Easy of Use

The library management software has a user-friendly and intuitive interface, making it easy for users to access and use. Library staff training goes smoothly thanks to detailed guides and support features within the software. Users can quickly and easily perform tasks like searching for books, borrowing, and returning them.

5.1.2 Effectiveness

The software optimizes the library management process by automating many daily tasks. Tasks like data entry, updating book information, and managing user accounts are done quickly and accurately. This not only reduces the workload for library staff but also significantly improves work performance. In particular, the time to process user requests is greatly reduced, leading to higher satisfaction.

5.1.3 Flexibility

The software allows customization and expansion of features to meet the specific needs of each library. Its ability to integrate with other management systems, such as student management or digital document management systems, enhances compatibility and application. This allows the library to easily adapt to changes in needs and job requirements.

5.2 Results of Implementing Library Management Software

Implementing the library management software has brought positive and noticeable results, improving the quality and efficiency of library operations. Specific results include:

5.2.1 Improved Management Efficiency

The software helps systematize and professionalize the library management process. With centralized management and automation of processes, library staff can easily track and update book and document information. This reduces errors and saves time and effort for staff, allowing them to focus on other important tasks.

5.2.2 Reporting and Performance Analysis

The software provides detailed reports on library activities, including statistics on borrowed, returned books, and document status. These reports help management track performance and make strategic decisions to improve service quality. Additionally, data analysis helps the library identify user trends and needs, allowing for reasonable adjustments to policies and operations.

5.3 Summary

In summary, the implementation of the library management software has provided clear benefits, enhancing management efficiency and service quality. The initial goals have been achieved and even exceeded expectations, highlighting the value and importance of applying technology in library management. The software not only improves internal operations but also enhances the user experience, contributing to the overall development of the library.

6 References

Jamhuri, A. (2023, August 21). *Model-view-controller architecture in programming*. LinkedIn. <https://www.linkedin.com/pulse/model-view-controller-architecture-programming-aditira-jamhuri/>

Viblo. (n.d.). *Repository pattern là gì? Sử dụng repository pattern trong Laravel*. Viblo. <https://viblo.asia/p/repository-pattern-la-gi-su-dung-repository-pattern-trong-laravel-gDVK2jqwKLj>