Software Requirements Specification

for

IIT Library

Prepared by

Prosanto Deb ASH1925005M
Abdullah Alif ASH1925009M
Ratna Kumer Tripura ASH1825042M

Institute of Information Technology
Noakhali Science and Technology University

22.08.2022

Table of Contents

List of Figures

List of Tables

1. Introduction

The Software Requirements Specification (SRS) introduction contains the SRS's policy, scope, references, and summary. This document's goal is to gather information about proposed system as we name it as "IIT Library" and is to give readers a greater understanding of it by outlining the issue statement in great detail. While defining the qualities of a high-quality product, it also emphasizes the advantages and requirements of the participants. Details on the "IIT Library" can be found on this document.

1.1 Problem Statement

A library in an educational institution has a collection of books that are accessible for both students and teachers. We can borrow books and have to return them within a specific period. In our institute, we also have a seminar library. But it is a matter of regret that we don't have any automated and online library management system. Our existing library performs all operations manually and can't keep track of the books accurately. It's too difficult to find a book from the library. The solution is either we have to buy paid software or develop a web-based library system of our own. For our team, it is the best time to utilize the period of SPL II and build a web-based library system that can save the external costs to buy expensive software and protect national resources. We will also implement a feature where a user can make a request for a book remotely. Our web-based system will show who is the last person to take a specific book and when will return the book. The system will also store the URL of downloadable electronic copies that are permitted by the publisher.

1.2 Purpose

The main purpose of our project "IIT Library" is to provide a hassle-free system. Using this system, we can easily borrow books and return them within a specific period. To manage membership, the system has a unique registration process for each individual authentic member of the IIT Library. The system can trace easily the misplaced and missing books. To add new books and add them digitally, to delete irrelevant and outdated books. Using this system, the users can easily search the books in the library. Moreover, the system calculates the fine due for non-return or lost or damaged books and the members will be notified about the fines by the system.

1.3 Project Scope

The project is developed to provide book Information, online viewing of books, and many other facilities.

- Users can search for their desired book in a single instant.
- Users can borrow a book, access electronic copy of the book and place a request for a book remotely.
- Users can view their profile to check any dues against their accounts.
- Users can extend the time period of the borrowed book.

- User can send a request for adding some new books to the library.

1.4 Glossary

This section provides definitions for all document names, acronyms, and abbreviations. The application domain's terms and concepts are defined.

HTML – Hyper Text Markup Language
JS – JavaScript
CSS – Cascading Style Sheets
PHP – Hypertext Preprocessor
XML – Extensible Markup Language
SRS – Software Requirement Specification
UI – User Interface
API – Application Programming Interface
MB – Megabytes

1.5 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

1.6 Overview

The primary purpose of a library is to provide resources and services in a variety of media to meet the needs of individuals and groups for education, information and personal development including recreation and leisure. Similarly, The Seminar Library is a very important institution for every single student because they provide reading materials. A seminar library is set up to quench the thirst for knowledge of the departmental students.

Almost all of our library's operations will be managed by this system. It will assist librarians in keeping a database of newly released books as well as books that users have borrowed and their due dates.

Each and every activity in our library will be fully automated by this system. Implementing our system is the greatest approach to keep many books organized, maintained, and handled properly. It will keep track of the statistics regarding the quantity of books in the library, how many are given, how many are returned or updated, how many late fines are assessed, etc. With the help of this system, we will be able to immediately issue and reissue books as well as efficiently and effectively handle all the data. This system's main goal is to save a lot of time and effort by instantly and accurately providing information on any type of book.

2. Stakeholders and Characteristics

2.1 Acquirer

- Customers who request, purchase, and/or pay for the software product – IIT, NSTU
- End-users who actually use the product directly or indirectly – Students, faculty members, officers, staff.

2.2 Supplier

- Individuals and teams that are part of the organization that develops the software product – Team Twins, IIT, NSTU

2.3 Change Control Board

- Responsible for reviewing or proposing changes, approving/disapproving changes, ensuring that approved changes are implemented and validated – Faculty Members, IIT, NSTU

3. Design and Implementation Constraints

In order to ensure the project's success, we used design and implementation limitations. It can also refer to a tool that enables testers and developers to view and interact with the user interface (UI) components of an application.

3.1 Language

User interface Design, usually known as UI Design, is the visual organization of the parts of a website or technological product that a user could interact with. In other words, it is the visual layout of a website. On the other hand, the code that enables a computer program or application to run and cannot be viewed by a user is referred to as the back end. The back end of a computer system is where the majority of data and operating syntax are kept and accessed. Typically, the code is comprised of one or more programming languages.

3.1.1 HTML

HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. Precisely, the coding that organizes a web page's content is called HTML (Hypertext Markup Language). With the help of HTML, you can tell a web page whether it should be recognized as a paragraph, list, heading, link, image, multimedia player, form, or any other of the many other components that are now supported, or even a new element that you design. It is the programming language for formatting web pages that is widely accepted. Small and medium-sized businesses are the main users, as they do not actually require extensive functionality on their

websites. The option to utilize HTML to design the structure of my web pages was made since it is free, works with all browsers on the client's machine, and is simple to use and understand.

3.1.2 CSS

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS specifies how items should be shown in various media, including speech, paper, screens, and other media. One of the fundamental languages of the open web, CSS is defined by the W3C (World Wide Web Consortium) specification and is supported by all major browsers.

3.1.3 Bootstrap (Front-end framework)

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It includes optional JavaScript extensions along with HTML and CSS-based design templates for navigation, buttons, forms, and other interface elements. It only addresses front-end development, unlike many web frameworks. Along with CSS, Bootstrap would be utilized to create the application's styling. Bootstrap is important in the application for the following reasons:

- **Easy to use:** Anyone can begin using Bootstrap with just a basic Knowledge of HTML and CSS.
- **Responsive features:** The responsive CSS in Bootstrap adapts to mobile devices, tablets, and desktops.
- **Mobile-first approach:** The fundamental Bootstrap framework provides mobile-first styling.
- **Browser compatibility:** All current browsers are compatible with Bootstrap (Chrome, Firefox, Internet Explorer, Safari, and Opera).

3.1.4 JavaScript

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. JavaScript adds interactive elements to online pages that keep users engaged, whereas HTML and CSS are languages that give web pages' structure and style.

The prototype is a built-in property that every JavaScript object has. The prototype is itself an object, so the prototype will have its own prototype, making what's called a prototype chain. When we get to a prototype that contains null for its own prototype, the chain comes to an end.

3.1.5 PHP

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It was one of the first server-side languages that HTML could incorporate, making it simpler to add functionality without having to call outside files for information. One of the best things about PHP is how beginner-friendly it is while still offering many advanced features to professional programmers.

3.2 Server-Side Technology

When an application is used, server-side development refers to the processes that happen in the background. Databases, scripting, website architecture, backend logic, APIs, and servers are the main topics covered.

3.2.1 Database Server

MySQL is an open-source relational database management system (RDBMS). A relational database arranges data into one or more tables where it is possible for the data to be connected to one another. Programmers use the SQL language to create, change, and extract data from relational databases and to manage user access to the databases.

4. Requirement Specification

4.1 Functional Requirement

Functional requirements are those that serve as examples for the system's internal operation, its description, and an explanation of each subsystem. It comprises of the task that the system should complete, the associated processes, the data that the system should store, and the user interfaces.

4.1.1 User registration and login

FR-1	User (student, faculty member, officer, staff) registration and login a registered account of IIT Library.		
Description	User should register his/her account for the first time and be able to login to the account which was registered once. Already registered users will not face this stage. At first the registration request is checked by the librarian and then it is finally approved by the director or IIT, NSTU.		
Stakeholders	User(student, faculty member, officer, staff), System	**Priority**	High

4.1.2 Borrow Book

FR-2	User(student, faculty member, officer, staff) can send a borrow request to IIT Library.
Description	To borrow a book, the user first needs to send a request to the IIT Library. The system will provide a receipt with all required information. This receipt will be valid for 72 hours. After showing the receipts to the librarian, he will provide the book and remove the book from the system which means the book is not in the library.

Stakeholders	User(student, faculty member, officer, staff), Librarian	**Priority**	High

4.1.3 Extend Book Period

FR-3	User (student, faculty member, officer, staff) can extend the time period of the borrowed book.		
Description	The user has to log in and goes to his/her profile. There will be a list of borrowed books. From there a user can select and extend the borrowing period for 7days. But it will be valid only two times for each book.		
Stakeholders	User(student, faculty member, officer, staff), System	**Priority**	High

4.1.4 Access Electronic Copy

FR-4	User (student, faculty member, officer, staff) can access the electronic copy of the book if it is available.		
Description	The user has to login and go to his/her profile. There will be an option for searching books and it will show a list of available books. From there a user can select and access the electronic copy of the book is it is available.		
Stakeholders	User(student, faculty member, officer, staff), System	**Priority**	High

4.1.5 Calculate and Check Fine

FR-5	The system can calculate the fine and the user will be able to see it.		
Description	The user can see the fine after entering his/her profile and the system will provide a detailed information of the fine.		
Stakeholders	User(student, faculty member, officer, staff), System	**Priority**	High

4.1.6 Report Generate

FR-6	The system will generate a daily report with all required information.

Description	Sometimes, it is required to see a daily report of which book is returned and which book is borrowed in a single day. That's why the system will perform this operation.		
Stakeholders	System	**Priority**	High

4.1.7 Send Email

FR-7	The system will send an email with all required information as a response for some specific operations.		
Description	Operations like sending registration request, book borrowing requests, extending borrow period, updating passwords, and requesting new book needs to be notified by an email. It increases user satisfaction and leaves a footprint for any critical operation.		
Stakeholders	System	**Priority**	High

4.2 Data Requirement

In our institution, every session has around 35 students. Provide hard copy of books most of them is not possible because of huge amount of cost. To mitigate this liquid money costing we will provide electronic copy books. The user search for a book using the of the book or the name of author or the name of course-title.

4.2.1 Searching Books

DR-1	Searching books using the keyword or book name or author name.		
Description	If Any authentic user of our system wants to search a for viewing or downloading. He can search the book by the keyword or book name or author name.		
Stakeholders	User, Librarian, Director	**Priority**	High

4.3 Performance Requirement

It is important that maintain the performance of the system. To ensure the best performance of the system we must maintain the following steps:

4.3.1 Speed and Latency Requirements

PR-1	Faster searching for books.		
Description	When any authentic user of our system wants to search for a book then the user can feel the fast searching.		
Stakeholders	User, Librarian, Director	**Priority**	High

4.3.2 Safety Critical Requirements

For our project there is no safety critical requirements.

4.4 Maintainability and Supportability

The term "maintenance" describes how simple it is to fix, enhance, and comprehend software code. After the user has received the product, the software maintenance phase of the software development cycle begins.

4.4.1 Maintenance Requirements

MR-1	Develop maintainable code		
Description	Maintainability must be ensured so that it can be modified later and will be readable.		
Stakeholders	Developers (Team Twins)	**Priority**	High

4.4.2 Supportability Requirements

This system satisfies the supportability requirements for testability, maintainability, compatibility, configurability, serviceability, and install ability.

4.5 Security Requirements

Information security is far more crucial for a system to gain user's trust. Here are some security requirements are given below:

4.5.1 Access Requirements

The system will apply some authorization approaches when granting access to information to make sure the right user is using the right data.

4.5.2 Integrity Requirements

Integrity requirements relate to a security system that ensures an expectation of data quality. It also ensures that no data on the system will ever be exposed to malicious modification or accidental deletion.

4.6 Usability and Human Integrity Requirements

Usability in software engineering refers to how well a piece of software may be used by a specific target audience to accomplish goals. A user-friendly environment will be provided by the system.

4.6.1 Ease of Use Requirements

Our system will be easier to use by any type of stakeholder and they don't need any training to use the system.

4.6.2 Accessibility Requirements

The system provides authorization / authentication to get access to it. Numerous modules are used in this system.

SR-1	Safeguards are provided by the system.		
Description	The system is designed in a way that allows all modules to access a mechanism that provides security services.		
Stakeholders	Developers (Team Twins)	**Priority**	High

4.7 Look and Feel Requirements

Look and feel requirements mainly refer to how the system will appear. The "look" of a graphical user interface in software design refers to elements like colors, shapes, layouts, and typefaces. It also refers to the behavior of dynamic elements like buttons, boxes, and menus ("The Feel").

4.7.1 Appearance Requirements

AR-1	Text color and font
Description	Our system has to be different and attractive from other existing library using a better look and feel.

Stakeholders	Developers (Team Twins), User	**Priority**	Medium

4.8 Style Requirements

There are no style requirements in our system.

4.9 Legal Requirements

Legal requirements often refer to an organization's terms and conditions or privacy policy. No third-party software or individual is permitted to use our data for commercial purposes, according to the terms and conditions of our application.

5. Requirement Engineering Process

Software requirements are established using requirements engineering (RE), which takes into account customer wants or requirements. Requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management are all parts of the requirements engineering process.

5.1 Requirement Elicitation Techniques

Requirements elicitation, often known as "requirement gathering," is the process of investigating and discovering system requirements for users, clients, and other stakeholders. Contacting participants directly or conducting research, analysis, and testing are two ways to elicit requirements.

5.1.1 Hold Interviews

We have conversations that can be had alone or with a small group of people. They are a useful approach to accessing services without having to spend a lot of time with participants because we simply meet with them to go through a few key program criteria. Negotiations are useful for getting specific demands from participants in setting up workshops where those program participants gather to address any problems or conflicts. Our interviews are primarily conducted using a set of predetermined standards.

5.1.2 Perform Document Analysis

Existing documentation can assist in demonstrating how systems are being used or what I should do with them. Documents contain textual details regarding existing programs, operational procedures, required specifications, and market research on competitors. Once again, textual analysis can be useful. Determine which features should be removed and which performances should stay by consulting the Software Requirements Specification. The previous document in our investigation, we discovered a number of issues with the current system.

- Existing system cannot trace the book.
- Very time-consuming to find a book.
- Anyone cannot place a request remotely for a book.
- No option for accessing the electronic version of the book.

5.1.3 Distribute Questionnaires

The questionnaire is a helpful tool for examining user satisfaction with priorities and preferences, changes in attitudes and ideas, and styles. We tried to keep our question lists to a minimum. The response can be worn out or angry. A fundamental justification for each question and grouped the subject areas together for the respondent to concentrate on. The primary benefit of the survey responses was that they were gathered in the expected manner. Many others provided summaries of the information.

5.1.4 Sample of Requirement Collection

Table 1 Requirement Collection 01

Requirement Elicitation Techniques	Interview, Field Observation, System Archeology
Collected From	Office Staff of IIT, NSTU
Findings	Access and Register 1) Serial No 2) Fiscal Year 3) Accession No 4) Author 5) Source 6) Name of the Book 7) Price 8) Voucher No. 9) Remarks Circulation 1) Serial No. 2) Students Name 3) Book Name 4) Book Number 5) Issue Date 6) Receiver Signature 7) Roll Number 8) Return Date 9) Signature 10) Year-Term

Advised Topics for Implementation	Make Book Search Easier.

Table 2 Requirement Collection 02

Requirement Elicitation Techniques	Interview Field Observation
Collected From	Helal Uddin, Documentation Officer, Central Library, NSTU
Findings	a) Access and Register I. Accession Number (6 Digit) II. Tittle III. Author IV. Edition V. Year of Publication VI. Publication Place VII. Publisher VIII. Page Number IX. Price X. Source XI. Image Field XII. Classification Number XIII. ISBN XIV. Keyword or Tropical Term XV. Remarks b) Circulation I. Checkout II. Renew (3 Times) III. Overdue IV. Overdue with fees c) Patrons (4 Type) I. Student II. Officer III. Staff IV. Teacher d) Advanced Search with I. Keyword or Phrase II. Personal Name III. Author IV. Title V. Subject

	VI. Journal VII. Magazine VIII. Reference Book IX. Textbook e) Lists f) Cataloging g) Acquisitions h) Reports
Advised Topics for Study	Decimal Classification Number (DDC)

Table 3 Requirement Collection 03

Requirement Elicitation Techniques	Interview
Collected From	Dr. Mohammad Salim Hossain, Director, IIT, NSTU
Suggestions	1. An option for printing a daily transaction for books (borrowing and returning). 2. The System should be adapted to barcode features.

5.2 Requirement Validation

Requirement validation criteria make sure they are accurate and match the standard you desire from this program. Our requirements initially appeared to be good, but after reading them and attempting to implement them, we discovered that they contained gaps and ambiguities.

5.2.1 Review the Requirements

Among the techniques that produce the highest quality software now accessible, negative peer review, particularly the rigorous type known as evaluation, is exceptional. We carefully looked at documented needs, analysis models, and related disability information with a team of reviewers from various viewpoints.

5.2.2 Simulate the Requirements

We can use trading tools to simulate a suggested system in place or to add specifics to textual specifications in order to stimulate requirements. The simulation advances the concept of prototyping.

6. Use Case Diagram

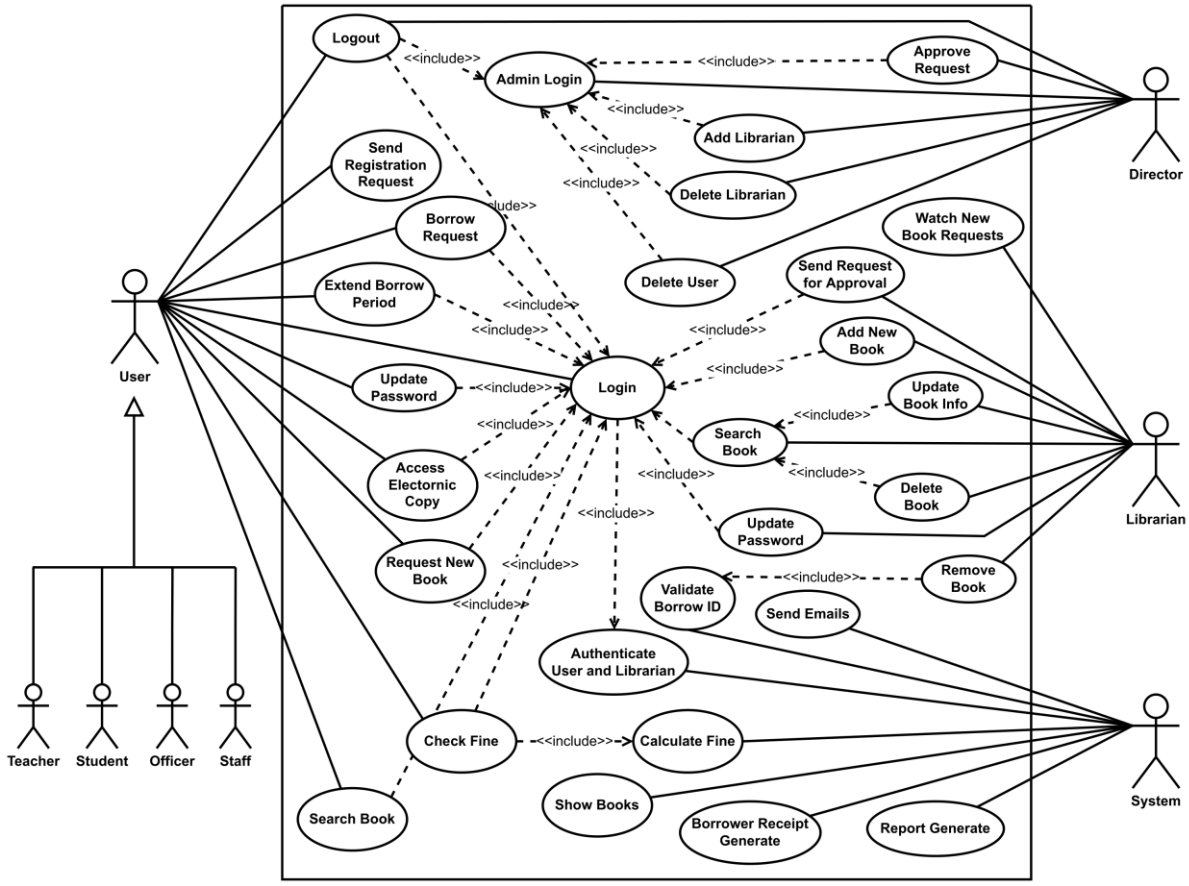

Figure 1 Use Case Diagram

7. Use Case Descriptions

Table 04 Send Registration Request

Use Case No.	01	
Use Case	Send Registration Request	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff) issues a request using IIT Library, expects the request will be approved and will be notified of any response via mail.	
Preconditions <what we expect is already the state of the world>	Must be a student or faculty member or officer or staff of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	"User Registration Request" sent successfully.	
Failed End Condition <the state of the world if goal abandoned>	"User Registration request" sending process failed.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff) System	
Trigger <the action upon the system that starts use case>	The user clicks the "Register" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the first page of the IIT Library.
	2	The user clicks the "Register" button.
	3.1 3.2 3.3 3.4	The user selects the "Register as Student" option. The user selects the "Register as Faculty Member" option. The user selects the "Register as Officer" option. The user selects the "Register as Staff" option.
	4	The user provides the required information.
	5	The user clicks the "Send Registration Request" button
	6	The system sends an email as a response to the user with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The user did not select any option.
	3a1	The registration page remains unchanged.
	4a	The user provided any invalid data.
	4a1	The webpage shows "Invalid Data".
	4a2	The specific field gets cleared in which the user has given the wrong data.
Quality Requirements	**Step**	**Requirement**
	4	The user should give valid information.

Table 05 Borrow Request

Use Case No.	02	
Use Case	Borrow Request	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff) issues a borrow request using IIT Library, expects the book will be reserved for him/her, gets a system-generated receipt, and will be notified of any response via mail.	
Preconditions <what we expect is already the state of the world>	Log in to the system successfully and must be a student or faculty member or officer or staff of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The user has sent borrow requests successfully.	
Failed End Condition <the state of the world if goal abandoned>	The user could not send a borrow request.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff) System	
Trigger	The user clicks the “Borrow Book” button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the IIT Library home page after login.
	2	The user clicks the “Borrow Book” button.
	3	The system shows a list of available books along with a search option.
	4	The user searched for the book from the list.
	5	The desired book is found.
	6	The user clicks the book and the system shows book details.
	7	The user clicks the “Borrow” button.
	8	The user provides the time period.
	9	The user clicks the “Send Borrow Request” button.
	10	The system generates a borrow receipt with all required information.
	11	The system sends an email as a response with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The user searched using the search option.
	4a1 4a2 4a3	The user selects the “Search by Keyword” option. The user selects the “Search by Book Name” option. The user selects the “Search by Author Name” option.
	5a	The desired book is not found.
	5a1	The system prints a “Sorry” message and shows a button called “Request New Book”.
	5a2	Use case “Request New Book”.
Quality Requirements	**Step**	**Requirement**
	4	Searching available books should be fast and accurate
	10	Auto-generated borrow receipt must be downloadable and will be valid for 72 hours.

Table 06 Extend Borrow Period

Use Case No.	03	
Use Case	Extend Borrow Period	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff) extends the borrowing period for 7 days using IIT Library and will be notified any response via mail.	
Preconditions <what we expect is already the state of the world>	Log in to the system successfully and must have borrowed any books earlier.	
Success End Condition <the state of the world upon successful completion>	The user has extended the borrowing period successfully.	
Failed End Condition <the state of the world if goal abandoned>	The user could not extend the borrowing period.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff) System	
Trigger <the action upon the system that starts use case>	The user clicks the “Extend Borrow Period” button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the IIT Library home page after login.
	2	The user enters into his/her profile.
	3	The user clicks the “Borrowed Books” button.
	4	The system shows a list of borrowed books by the user.
	5	The user searched for the book from the list.
	6	The desired book is found.
	7	The user clicks the book and the system shows book details.
	8	The user clicks the “Extend Borrow Period” button.
	9	The system checks of the validation.
	10	The system shows a “Successful” message.
	11	The system updates all required information about that user.
	12	The system sends an email as a response with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The book list is empty.
	4a1	The user did not borrow any book earlier.
	9a	The user is not valid to extend the borrowing period.
	9a1	The system shows an “Error” message.
	9a2	The user can not extend the borrowing period.
Quality Requirements	**Step**	**Requirement**
	9	A user who has already extended for 2 times, cannot extend further.
	11	The system will mark that book as unavailable for 7 days.

Table 07 Access Electronic Copy

Use Case No.	04	
Use Case	Access Electronic Copy	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff) can access (view and download) an electronic copy of the book if available.	
Preconditions <what we expect is already the state of the world>	Log in to the system successfully and must be a student or faculty member or officer or staff of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The user has accessed (viewed and downloaded) the book successfully.	
Failed End Condition <the state of the world if goal abandoned>	The user could not access (view and download) the book.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff) System	
Trigger <the action upon the system that starts use case>	The user clicks the "Open Electronic Copy" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the IIT Library home page after login.
	2	The user clicks the "Borrow Book" button.
	3	The system shows a list of available books along with search option.
	4	The user searched for the book from the list.
	5	The desired book is found.
	6	The user clicks the book and the system shows book details.
	7	The user clicks the "Open Electronic Copy" button.
	8	The system opens the electronic copy of the book in a new window.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The user searched using the search option
	4a1 4a2 4a3	The user selects the "Search by Keyword" option. The user selects the "Search by Book Name" option. The user selects the "Search by Author Name" option.
	5a	The desired book is not found.
	5a1	The system prints a "Sorry" message and shows a button called "Request New Book".
	5a2	Use case "Request New Book".
	7a	The "Open Electronic Copy" button is not available.
	7a1	An Electronic copy of that particular book is not available.
Quality Requirements	**Step**	**Requirement**
	4	Searching available books should be fast and accurate.
	7	An Electronic copy can be viewed and downloaded only.

Table 08 Request New Book

Use Case No.	05	
Use Case	Request New Book	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff) can request for a new book to be available in the IIT Library.	
Preconditions <what we expect is already the state of the world>	Log in to the system successfully and must be a student or faculty member or officer or staff of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The user has requested the new book successfully.	
Failed End Condition <the state of the world if goal abandoned>	The user could not request any new book.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff) System, Librarian	
Trigger <the action upon the system that starts use case>	The user clicks the "Request New Book" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the IIT Library home page after login.
	2	The user clicks the "Request New Book" button.
	3	The system shows some fields to enter information about the book.
	4	The user provides information about the book.
	5	The user clicks the "Send Request for New Book" button.
	6	The system sends an email as a response with all required information.
	7	Wait until the librarian performs any operation.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The user has provided any invalid data.
	4a1	The webpage shows "Invalid Data".
	4a2	The specific field gets cleared in which the user has given the wrong data.
	7a	Librarian checks for new book requests.
	7a1	Use Case "Watch New Book Requests"
Quality Requirements	**Step**	**Requirement**
	4	The user should give valid information.
	7	The system should send an email to the librarian as a notification regarding the new book requests.

Table 09 Check Fine

Use Case No.	06	
Use Case	Check Fine	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff) can check how much fine he/she has to pay.	
Preconditions <what we expect is already the state of the world>	Log in to the system successfully, must borrow a book, and exceed the return date.	
Success End Condition <the state of the world upon successful completion>	The user has checked the fine successfully.	
Failed End Condition <the state of the world if goal abandoned>	The user could not check the fine.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff) System	
Trigger <the action upon the system that starts use case>	The user clicks the “Check Fine” button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the IIT Library home page after login.
	2	The user enters into his/her profile.
	3	The user clicks the “Check Fine” button.
	4	The system calculates the fine.
	5	Use Case “Calculate Fine”.
	6	The system shows fine with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The user has not borrowed any book that the return date exceeded.
	4a1	The system shows the “No Fine To Pay” message.
Quality Requirements	**Step**	**Requirement**
	5	The fine calculation must be valid.

Table 10 Search Book (By User)

Use Case No.	07	
Use Case	Search Book (By User)	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff) can search for any book.	
Preconditions <what we expect is already the state of the world>	Must be a student or faculty member or officer or staff of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The user has searched for any book successfully.	
Failed End Condition <the state of the world if goal abandoned>	The user could not search for any book.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff) System	
Trigger <the action upon the system that starts use case>	The user clicks the "Search" icon on the top of a button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the first page of the IIT Library.
	2	The user enter some text in the input field
	3	The user clicks the "Search" icon on the top of a button.
	4	The system shows a list of all available books.
	5	The user searches for a book.
	6	The desired book is found.
	7	The system shows all required information regarding the book.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The user wants to search by a specific option.
	3a1 3a2 3a3	The user selects the "Search by Keyword" option. The user selects the "Search by Book Name" option. The user selects the "Search by Author Name" option.
	6a	The desired book is not found.
	6a1	The system shows a "Sorry" message
Quality Requirements	**Step**	**Requirement**
	3	Searching available books should be fast and accurate
	7	The user can see only some basic and required information.

Table 11 Login

Use Case No.	08	
Use Case	Login	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff) and a librarian can log in to the system.	
Preconditions <what we expect is already the state of the world>	Must be a student or faculty member or officer or staff or librarian of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The user or librarian has successfully logged in to the system.	
Failed End Condition <the state of the world if goal abandoned>	The user or librarian could not log in to the system.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff), Librarian System	
Trigger <the action upon the system that starts use case>	The primary actor clicks the "Login" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The primary actor enters the first page of the IIT Library.
	2.1 2.1 2.3 2.4 2.5	The primary actor clicks "Login as Student" option. The primary actor clicks "Login as Faculty Member" option. The primary actor clicks "Login as Officer" option. The primary actor clicks "Login as Staff" option. The primary actor clicks "Login as Librarian" option.
	3	The primary actor enters username and password.
	4	The system checks for validation. Use Case "Authenticate User".
	5	The primary actor successfully logs in to the system.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The primary actor provides invalid data.
	4a1	The system prints a "Error" message.
	4a2	The primary actor cannot log in to the system.
Quality Requirements	**Step**	**Requirement**
	3	The system will not allow incorrect passwords more than 5 times and if happens, keeps waiting for 15 minutes.
	4	The primary actor authentication must be accurate.

Table 12 Admin Login

Use Case No.	09	
Use Case	Admin Login	
Goal <a longer statement of the goal in context if needed>	An admin (Director) can log in to the system.	
Preconditions <what we expect is already the state of the world>	Must be the Director of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The admin has successfully logged in to the system.	
Failed End Condition <the state of the world if goal abandoned>	The admin could not log in to the system.	
Primary Actors: **Secondary Actors:**	Admin (Director) System	
Trigger <the action upon the system that starts use case>	The admin clicks the "Login" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The admin enters the first page of the IIT Library.
	2	The admin clicks "Login as Admin" option.
	3	The admin enters username and password.
	4	The system checks for validation. Use Case "Authenticate User".
	5	The admin successfully logs in to the system.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The admin provides invalid data.
	4a1	The system prints an "Error" message.
	4a2	The admin cannot log in to the system.
Quality Requirements	**Step**	**Requirement**
	3	The system will not allow incorrect passwords more than 3 times and if happens, keeps waiting for 10 minutes.
	4	The admin authentication must be accurate.

Table 13 Logout

Use Case No.	10	
Use Case	Logout	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff), a librarian, and the admin (Director) can log out from the system.	
Preconditions <what we expect is already the state of the world>	Log in to the system successfully and must be a user (student, faculty member, officer, staff) or a librarian, or the Director of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The user (student, faculty member, officer, staff) or the librarian or the admin has successfully logged out from the system.	
Failed End Condition <the state of the world if goal abandoned>	The admin could not log out from the system.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff), Librarian, Admin (Director) System	
Trigger <the action upon the system that starts use case>	The admin clicks the "Logout" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The primary actor enters the home page of IIT the Library after login.
	2	The primary actor clicks "Logout" button.
	3	The primary actor has successfully logged out from to the system.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2a	The primary actor does not click "Log out" button.
	2a1	The system performs according to the instructions of primary actor.
	4a2	The primary actor cannot log in to the system.
Quality Requirements	**Step**	**Requirement**
	2	The system should ask "Are you Sure to Logout?" after clicking the logout button.

Table 14 Send Emails

Use Case No.	11	
Use Case	Send Emails	
Goal <a longer statement of the goal in context if needed>	A user (student, faculty member, officer, staff), a librarian, and the admin (Director) will be notified by email from the system for any specific purpose.	
Preconditions <what we expect is already the state of the world>	There should be any operation that requires a response via email. Use Cases (Send Registration Request, Borrow Request, Extend Borrow Period, Request New Book, Approve Request)	
Success End Condition <the state of the world upon successful completion>	A system-generated email has been sent to a specific destination with all the required information.	
Failed End Condition <the state of the world if goal abandoned>	A system-generated email has not been sent to a specific destination.	
Primary Actors: **Secondary Actors:**	System User (student, faculty member, officer, staff), Librarian, Admin (Director)	
Trigger <the action upon the system that starts use case>	An email request comes in.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The system sets the destination email address.
	2	The system sets the subject of the email.
	3	The system generates the body of the email.
	4	The system sends the email to the destination address.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	1a	The destination email address is not valid.
	1a1	The system shows an “Error” message.
	1a2	Email is not sent.
Quality Requirements	**Step**	**Requirement**
	3	The email body should be accurate and contain all the required information.

Table 15 Validate Borrow ID

Use Case No.	12	
Use Case	Validate Borrow ID	
Goal <a longer statement of the goal in context if needed>	A user's (student, faculty member, officer, staff) borrower ID is checked whether it is valid or not.	
Preconditions <what we expect is already the state of the world>	A user must borrow a book.	
Success End Condition <the state of the world upon successful completion>	A user's borrower ID is checked successfully.	
Failed End Condition <the state of the world if goal abandoned>	A user's borrower ID is not checked.	
Primary Actors: **Secondary Actors:**	System Librarian	
Trigger <the action upon the system that starts use case>	A borrower ID validation request comes in.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The librarian enters the home page of IIT Library after login.
	2	The librarian clicks the "Check Borrower ID" button.
	3	The system checks the borrower ID.
	4	The borrower ID is valid.
	5	The system shows all information against the borrower ID.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The borrower ID is not valid.
	4a1	The system shows an "Error" message.
Quality Requirements	**Step**	**Requirement**
	4	The borrower ID validation must be accurate and the system shows all the required information against the borrower ID.

Table 16 Authenticate User and Librarian

Use Case No.	13	
Use Case	Authenticate User and Librarian	
Goal <a longer statement of the goal in context if needed>	A user's (student, faculty member, officer, staff) and librarian's login information borrower ID is checked whether it is authenticate or not.	
Preconditions <what we expect is already the state of the world>	A user's (student, faculty member, officer, staff) registration request must be approved by the Director of IIT, NSTU or a librarian must be added by the Director of IIT, NSTU	
Success End Condition <the state of the world upon successful completion>	A user's or librarian's login information is authenticated successfully.	
Failed End Condition <the state of the world if goal abandoned>	A user's or librarian's login information is not authenticated.	
Primary Actors: **Secondary Actors:**	System User (student, faculty member, officer, staff), Librarian	
Trigger <the action upon the system that starts use case>	An authentication request comes in while login.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user or librarian clicks the button required for login.
	2	The system checks the login information.
	3	The login information is valid.
	4	The system allows the secondary actors to log in and opens the home page of IIT, NSTU.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The login information is not valid or authenticated by the system.
	3a1	The system shows an "Error" message.
Quality Requirements	**Step**	**Requirement**
	2	The authentication must be accurate.

Table 17 Calculate Fine

Use Case No.	14	
Use Case	Calculate Fine	
Goal <a longer statement of the goal in context if needed>	The system calculates a user's (student, faculty member, officer, staff) fine.	
Preconditions <what we expect is already the state of the world>	A user (student, faculty member, officer, staff) must borrow a book, and exceed the return date.	
Success End Condition <the state of the world upon successful completion>	A user's fine is calculated successfully.	
Failed End Condition <the state of the world if goal abandoned>	A user's fine is not calculated.	
Primary Actors: **Secondary Actors:**	System User (student, faculty member, officer, staff)	
Trigger <the action upon the system that starts use case>	An fine calculation request comes in while checking the fine by the user.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user clicks the "Check Fine" button.
	2	The system checks whether the user has borrowed any book.
	3	The system checks whether the user has exceeded the return date.
	4	The system checks whether the user has any previous fine.
	5	The system calculates the fine.
	6	The system shows fine with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2a	The user did not borrow any books.
	2a1	The system shows the "No Fine To Pay" message.
	3a	The user did not exceed the return date.
	3a1	The system shows the "No Fine To Pay" message.
	4a	The user does not have any previous fine.
	4a1	The system shows the current fine only.
Quality Requirements	**Step**	**Requirement**
	5	A fine of Taka 5 will be added for each day delay for a single book.

Table 18 Show Books

Use Case No.	15	
Use Case	Show Books	
Goal <a longer statement of the goal in context if needed>	The system shows the book's information to the user (student, faculty member, officer, staff), librarian, and Director of IIT, NSTU.	
Preconditions <what we expect is already the state of the world>	The librarian must have added at least one book information.	
Success End Condition <the state of the world upon successful completion>	The system shows all the required information of books.	
Failed End Condition <the state of the world if goal abandoned>	The system is unable to show all the required information of books.	
Primary Actors: **Secondary Actors:**	System User (student, faculty member, officer, staff), Librarian, Director	
Trigger <the action upon the system that starts use case>	Search book request comes in.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.1 1.2	The secondary actor search for a book. The secondary actor clicks a book.
	2	The system finds the book.
	3	The system shows the book with all the required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2a	The system does not find the book.
	2a1	The system shows the "Sorry" message.
Quality Requirements	**Step**	**Requirement**
	2	Book searching should take a keyword or book name or author name.

Table 19 Borrower Receipt Generate

Use Case No.	16	
Use Case	Borrower Receipt Generate	
Goal <a longer statement of the goal in context if needed>	The system generates a receipt with all required information of the borrower or the user (student, faculty member, officer, staff) along with book information and return date.	
Preconditions <what we expect is already the state of the world>	The user (student, faculty member, officer, staff) must borrow a book.	
Success End Condition <the state of the world upon successful completion>	The system has generated a borrower receipt successfully.	
Failed End Condition <the state of the world if goal abandoned>	The system could not generate a borrower receipt.	
Primary Actors: **Secondary Actors:**	System User (student, faculty member, officer, staff)	
Trigger <the action upon the system that starts use case>	Borrower receipt generate request comes in.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user clicks the “Send Borrow Request” button.
	2	The system collects all required information.
	3	The system generates the receipt.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2a	The system cannot collect all information.
	2a1	The system shows the “Unable to generate receipt” message.
Quality Requirements	**Step**	**Requirement**
	2	The receipt must contain the user information, book information and return date.

Table 20 Report Generate

Use Case No.	17	
Use Case	Report Generate	
Goal <a longer statement of the goal in context if needed>	The system generates a report with all required information of the transactions (borrowed and returned books) made in a single day.	
Preconditions <what we expect is already the state of the world>	There must have been at least one borrowed or returned book in a single day.	
Success End Condition <the state of the world upon successful completion>	The system has generated a report successfully.	
Failed End Condition <the state of the world if goal abandoned>	The system could not generate a report.	
Primary Actors: **Secondary Actors:**	System Librarian, Director	
Trigger <the action upon the system that starts use case>	Report generate request comes in.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.1 1.2	The librarian clicks the “Report Generate” button. The director clicks the “Report Generate” button
	2	The system collects all required information.
	3	The system generates the report.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2a	The system cannot collect all information.
	2a1	The system shows the “Unable to generate report” message.
Quality Requirements	**Step**	**Requirement**
	2	The receipt must contain the book information and user information related to the borrowing and the returning books.

Table 21 Watch New Book Requests

Use Case No.	18	
Use Case	Watch new book requests	
Goal <a longer statement of the goal in context if needed>	The librarian watches how many requests of a new book arrived in the system by the users (student, faculty member, officer, staff).	
Preconditions <what we expect is already the state of the world>	Log in to the system by the librarian and there must have at least one request for a new book.	
Success End Condition <the state of the world upon successful completion>	The librarian watches for new book requests successfully.	
Failed End Condition <the state of the world if goal abandoned>	The librarian could not watch for new book requests.	
Primary Actors: **Secondary Actors:**	Librarian System, User(student, faculty member, officer, staff)	
Trigger <the action upon the system that starts use case>	Report generate request comes in.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The librarian enters the home page of IIT Library after login.
	2	The librarian clicks the “Watch New Book Requests” button.
	3	The system shows the information about the requests.
	4	The librarian analyses the validity and feasibility of the requested books.
	5	The librarian discuss with the director and take necessary steps.
	6	The librarian provides a response via email about the actions regarding the new book request.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The system cannot find any new book requests.
	3a1	The system shows the “No New Book Request” message.
	4a	The book request is not valid or feasible.
	4a1	The librarian provides a response via email that the request cannot be implemented.
Quality Requirements	**Step**	**Requirement**
	3	The system should give an email to the librarian’s account about the new book request.

Table 22 Send Request for Approval

Use Case No.	19	
Use Case	Send Request for approval	
Goal <a longer statement of the goal in context if needed>	The librarian sends registration requests made by the users (student, faculty member, officer, staff) to the director for approval.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the librarian and there must have at least one registration request by the users (student, faculty member, officer, staff)	
Success End Condition <the state of the world upon successful completion>	The librarian sends the registration requests to the director for approval successfully.	
Failed End Condition <the state of the world if goal abandoned>	The librarian could not send the registration requests to the director for approval.	
Primary Actors: **Secondary Actors:**	Librarian User (student, faculty member, officer, staff), Director	
Trigger <the action upon the system that starts use case>	Registration request comes in to the system.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The librarian enters the home page of IIT Library after login.
	2	The librarian clicks the “Watch Requests for Approval” button.
	3	The system shows the information about available the requests.
	4	The librarian checks the validity of the registration requests.
	5	The librarian clicks the “Send Request for Approval” button. (Only the Valid Requests)
	6	The system sends an email as a response with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The system cannot find any new registration requests.
	3a1	The system shows the “No New Registration Request” message.
	4a	The registration request is not valid.
	4a1	The librarian provides a response via email that the request cannot be granted.
Quality Requirements	**Step**	**Requirement**
	3	The system should give an email to the librarian’s account about the new registration request.

Table 23 Search Book (By Librarian)

Use Case No.	20	
Use Case	Search Book (By Librarian)	
Goal <a longer statement of the goal in context if needed>	The librarian can search for any book.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the librarian of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The librarian has searched for any book successfully.	
Failed End Condition <the state of the world if goal abandoned>	The librarian could not search for any book.	
Primary Actors: **Secondary Actors:**	Librarian System	
Trigger <the action upon the system that starts use case>	The librarian clicks the "Search" icon on the top of a button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The librarian enters the home page of IIT Library after login.
	2	The librarian enter some text in the input field.
	3	The librarian clicks the "Search" icon on the top of a button.
	4	The system shows a list of all available books.
	5	The librarian searches for a book.
	6	The desired book is found.
	7	The system shows all required information regarding the book.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The librarian wants to search by a specific option.
	3a1 3a2 3a3	The librarian selects the "Search by Keyword" option. The librarian selects the "Search by Book Name" option. The librarian selects the "Search by Author Name" option.
	6a	The desired book is not found.
	6a1	The system shows a "Sorry" message.
Quality Requirements	**Step**	**Requirement**
	3	Searching available books should be fast and accurate.

Table 24 Add New Book

Use Case No.	21	
Use Case	Add New Book	
Goal <a longer statement of the goal in context if needed>	The librarian can add all information of a new book.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the librarian of IIT, NSTU and the information of the book must be unique.	
Success End Condition <the state of the world upon successful completion>	The librarian has added all information of a new book successfully.	
Failed End Condition <the state of the world if goal abandoned>	The librarian could not add information of a new book.	
Primary Actors: **Secondary Actors:**	Librarian System	
Trigger <the action upon the system that starts use case>	The librarian clicks the "Add New Book" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The librarian enters the home page of IIT Library after login.
	2	The librarian clicks the "Add New Book" button.
	3	The librarian provides all necessary information of a new book.
	4	The librarian clicks the "Add Book" Button".
	5	The system checks whether the information is valid.
	6	The system checks whether the information is unique.
	7	The system updates the database.
	8	The system shows a "Successful" message.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	5a	The information is not valid.
	5a1	The system shows an "Error" message.
	6a	The information is not unique.
	6a1	The system shows an "Error" message.
Quality Requirements	**Step**	**Requirement**
	7	The database must be updated after every operation performed by the librarian.

Table 25 Update Book Information

Use Case No.	22	
Use Case	Update Book Information	
Goal <a longer statement of the goal in context if needed>	The librarian can update the information of an existing book.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the librarian of IIT, NSTU and there must be at least one book in the system to update information.	
Success End Condition <the state of the world upon successful completion>	The librarian has updated the information of an existing book successfully.	
Failed End Condition <the state of the world if goal abandoned>	The librarian could not update the information of an existing book.	
Primary Actors: **Secondary Actors:**	Librarian System	
Trigger <the action upon the system that starts use case>	The librarian clicks the "Update Book Information" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The librarian enters the home page of IIT Library after login.
	2	Use Case "Search Book".
	3	The librarian clicks the "Update Book Information" button.
	4	The librarian edits the information where required.
	5	The librarian clicks the "Update" button.
	6	The system checks whether the information is valid.
	7	The system checks whether the information is unique.
	8	The system updates the database.
	9	The system shows a "Successful" message.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	6a	The information is not valid.
	6a1	The system shows an "Error" message.
	7a	The information is not unique.
	7a1	The system shows an "Error" message.
Quality Requirements	**Step**	**Requirement**
	8	The database must be updated after every operation performed by the librarian.

Table 26 Delete Book

Use Case No.	23	
Use Case	Delete Book	
Goal <a longer statement of the goal in context if needed>	The librarian can delete the information of an existing book.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the librarian of IIT, NSTU and there must be at least one book in the system to delete information.	
Success End Condition <the state of the world upon successful completion>	The librarian has deleted the information of an existing book successfully.	
Failed End Condition <the state of the world if goal abandoned>	The librarian could not delete the information of an existing book.	
Primary Actors: **Secondary Actors:**	Librarian System	
Trigger <the action upon the system that starts use case>	The librarian clicks the "Delete Book Information" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The librarian enters the home page of IIT Library after login.
	2	Use Case "Search Book".
	3	The librarian clicks the "Delete Book" button.
	4	The system updates the database.
	5	The system shows a "Successful" message.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2a	The book is not found.
	2a1	The system shows a "Sorry" message.
Quality Requirements	**Step**	**Requirement**
	4	The database must be updated after every operation performed by the librarian.

Table 27 Remove Book

Use Case No.	24	
Use Case	Remove Book	
Goal <a longer statement of the goal in context if needed>	The librarian can remove the book from the system which is borrowed by any user.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the librarian of IIT, NSTU and there must be at least one book in the system to remove.	
Success End Condition <the state of the world upon successful completion>	The librarian has removed the borrowed book successfully.	
Failed End Condition <the state of the world if goal abandoned>	The librarian could not remove the borrowed book.	
Primary Actors: **Secondary Actors:**	Librarian System, User (student, faculty member, officer, staff)	
Trigger <the action upon the system that starts use case>	The librarian clicks the "Remove Book" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The librarian enters the home page of IIT Library after login.
	2	The librarian clicks the "Remove Book" button.
	3	The librarian enters the borrower ID.
	4	Use Case "Validate Borrow ID".
	5	The librarian clicks the "Remove" button.
	6	The system updates the database.
	7	The system shows a "Successful" message.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	The borrow id is not valid.
	4a1 4a2	The system shows an "Error" message. The book cannot be removed.
Quality Requirements	**Step**	**Requirement**
	6	The database must be updated after every operation performed by the librarian.

Table 28 Approve Request

Use Case No.	25	
Use Case	Approve Request	
Goal <a longer statement of the goal in context if needed>	The director can approve a registration request.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the Director of IIT, NSTU and there must be at least one registration request.	
Success End Condition <the state of the world upon successful completion>	The director has approved the registration request successfully.	
Failed End Condition <the state of the world if goal abandoned>	The director could not approve the registration request.	
Primary Actors: **Secondary Actors:**	Director System, Librarian	
Trigger <the action upon the system that starts use case>	The director clicks the "Pending Requests" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	The director enters the home page of IIT Library after login.
	2	The director clicks the "Pending Requests" button. (Send from Librarian)
	3	The director clicks the "Approve" button.
	4	The system updates the database.
	5	The system shows a "Successful" message.
	6	The system sends an email as a response with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The director clicks the "Reject" button.
	3a1	The system sends an email of rejection as a response to the user's email account.
Quality Requirements	**Step**	**Requirement**
	4	The database must be updated after every operation is performed.

Table 29 Add Librarian

Use Case No.	26	
Use Case	Add Librarian	
Goal <a longer statement of the goal in context if needed>	The director can add a librarian.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the Director of IIT, NSTU.	
Success End Condition <the state of the world upon successful completion>	The director has added a librarian successfully.	
Failed End Condition <the state of the world if goal abandoned>	The director could not add a librarian.	
Primary Actors: **Secondary Actors:**	Director System, Librarian	
Trigger <the action upon the system that starts use case>	The director clicks the "Add Librarian" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	The director enters the home page of IIT Library after login.
	2	The director clicks the "Add Librarian" button.
	3	The director enters all required information of a librarian.
	4	The director clicks the "Add" button.
	5	The information is valid and unique.
	6	The system updates the database.
	7	The system shows a "Successful" message.
	8	The system sends an email as a response to the librarian with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	5a	The information is not valid or unique.
	5a1	The system shows an "Error" message.
Quality Requirements	**Step**	**Requirement**
	6	The database must be updated after every operation is performed.

Table 30 Delete Librarian

Use Case No.	27	
Use Case	Delete Librarian	
Goal <a longer statement of the goal in context if needed>	The director can delete all information of an existing librarian.	
Preconditions <what we expect is already the state of the world>	Log in to the system by the Director of IIT, NSTU and there must be at least one librarian to delete.	
Success End Condition <the state of the world upon successful completion>	The director has deleted all the information of an existing librarian successfully.	
Failed End Condition <the state of the world if goal abandoned>	The director could not delete the information of an existing librarian.	
Primary Actors: **Secondary Actors:**	Director System	
Trigger <the action upon the system that starts use case>	The director clicks the "Delete Librarian" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	The director enters the home page of IIT Library after login.
	2	The director clicks the "Delete Librarian" button.
	3	The system shows the information of the librarian.
	4	The director clicks the "Delete" button.
	5	The system updates the database.
	6	The system shows a "Successful" message.
	7	The system sends an email as a response to the librarian with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The system does not find any information of a librarian.
	3a1	The system shows the "No Librarian to Delete" message.
Quality Requirements	**Step**	**Requirement**
	5	The database must be updated after every operation is performed.

Table 31 Delete User

Use Case No.	28	
Use Case	Delete User	
Goal <a longer statement of the goal in context if needed>	The director can delete all information of an existing user (student, faculty member, officer, staff).	
Preconditions <what we expect is already the state of the world>	Log in to the system by the Director of IIT, NSTU and there must be at least one user to delete.	
Success End Condition <the state of the world upon successful completion>	The director has deleted all the information of an existing user (student, faculty member, officer, staff) successfully.	
Failed End Condition <the state of the world if goal abandoned>	The director could not delete the information of an existing user (student, faculty member, officer, staff).	
Primary Actors: **Secondary Actors:**	Director System	
Trigger <the action upon the system that starts use case>	The director clicks the "Delete User" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	The director enters the home page of IIT Library after login.
	2	The director clicks the "Delete User" button.
	3	The system shows a list of the information of the users.
	4	The director clicks the "Delete" button
	5	The system updates the database.
	6	The system shows a "Successful" message.
	7	The system sends an email as a response to the user with all required information.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	The system does not find any information of a user.
	3a1	The system shows the "No User to Delete" message
Quality Requirements	**Step**	**Requirement**
	7	The database must be updated after every operation is performed.

Table 32 Update Password

Use Case No.	29	
Use Case	Update Password	
Goal <a longer statement of the goal in context if needed>	The user (student, faculty member, officer, staff), and the librarian can update his/ her password.	
Preconditions <what we expect is already the state of the world>	Log in to the system successfully by the user (student, faculty member, officer, staff) and the librarian.	
Success End Condition <the state of the world upon successful completion>	The user and librarian has updated the password successfully.	
Failed End Condition <the state of the world if goal abandoned>	The user and librarian could not update the password.	
Primary Actors: **Secondary Actors:**	User (student, faculty member, officer, staff), Librarian System	
Trigger <the action upon the system that starts use case>	The primary actor clicks the “Update Password” button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean-up after>	**Step**	**Action**
	1	The primary actor enters the home page of IIT Library after login.
	2	The primary actor clicks the “Update Password” button.
	3	The primary actor enters the old password.
	4	The primary actor enters the new password.
	5	The primary actor clicks the “Update” button.
	6	The system finds that the old password is matched.
	7	The system updates the database.
	8	The system shows a “Successful” message.
	9	The system sends an email as a response to the primary actor that the password is changed.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	6a	The old password is not matched.
	6a1	The system shows a “Error” message.
Quality Requirements	**Step**	**Requirement**
	7	The database must be updated after every operation is performed.

8. Activity Diagram

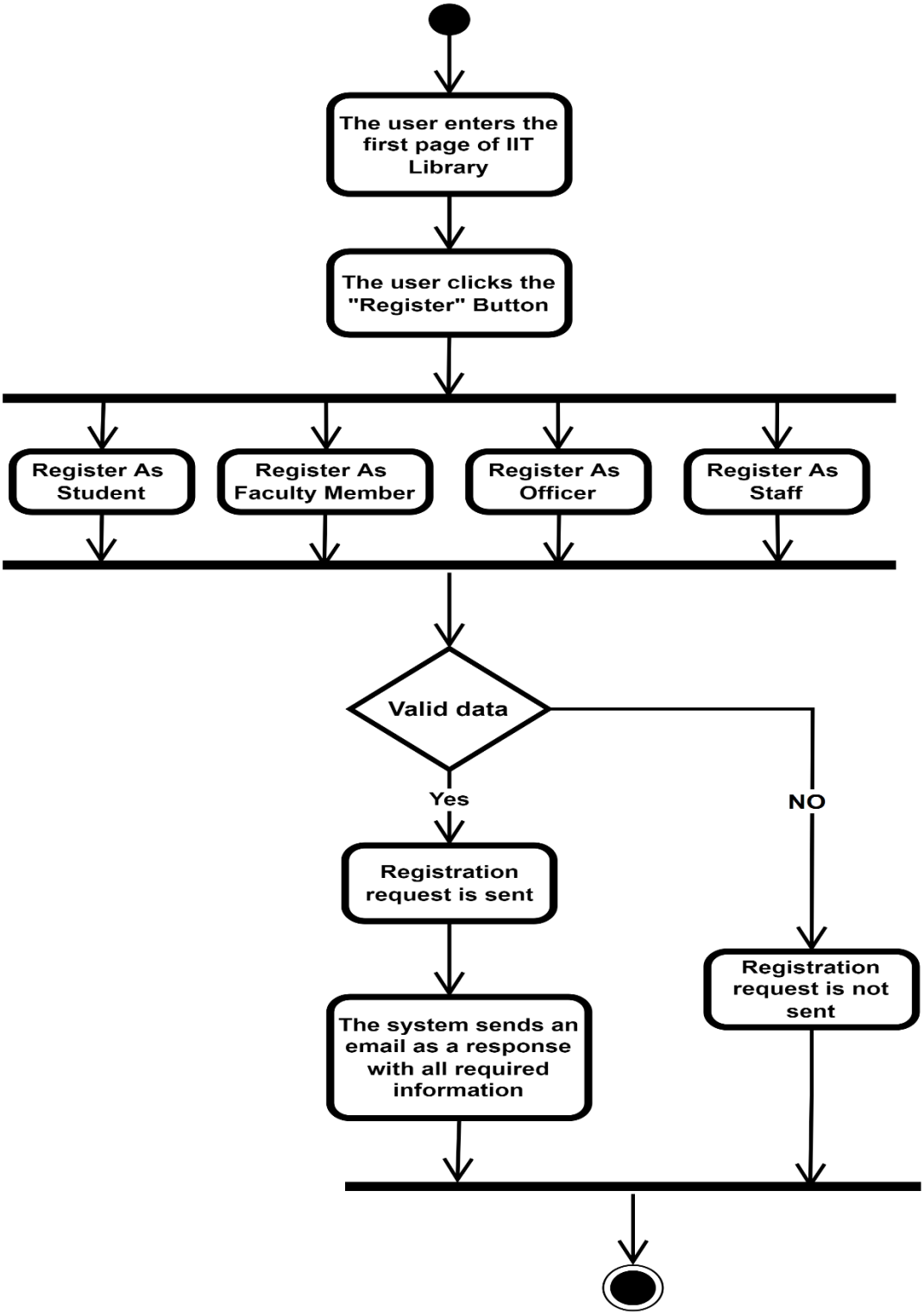

Figure 2 Send Registration Request

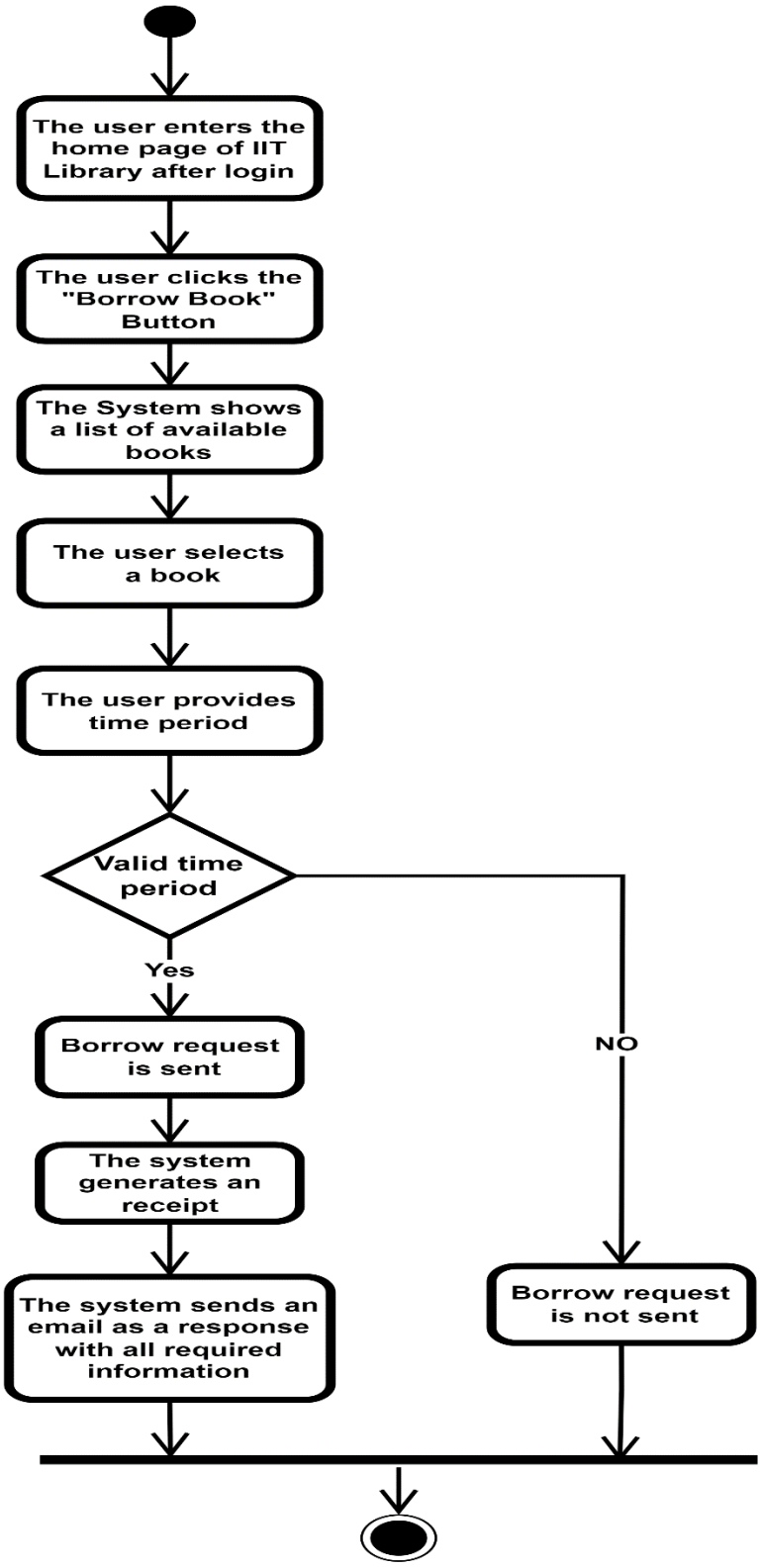

Figure 3 Borrow Request

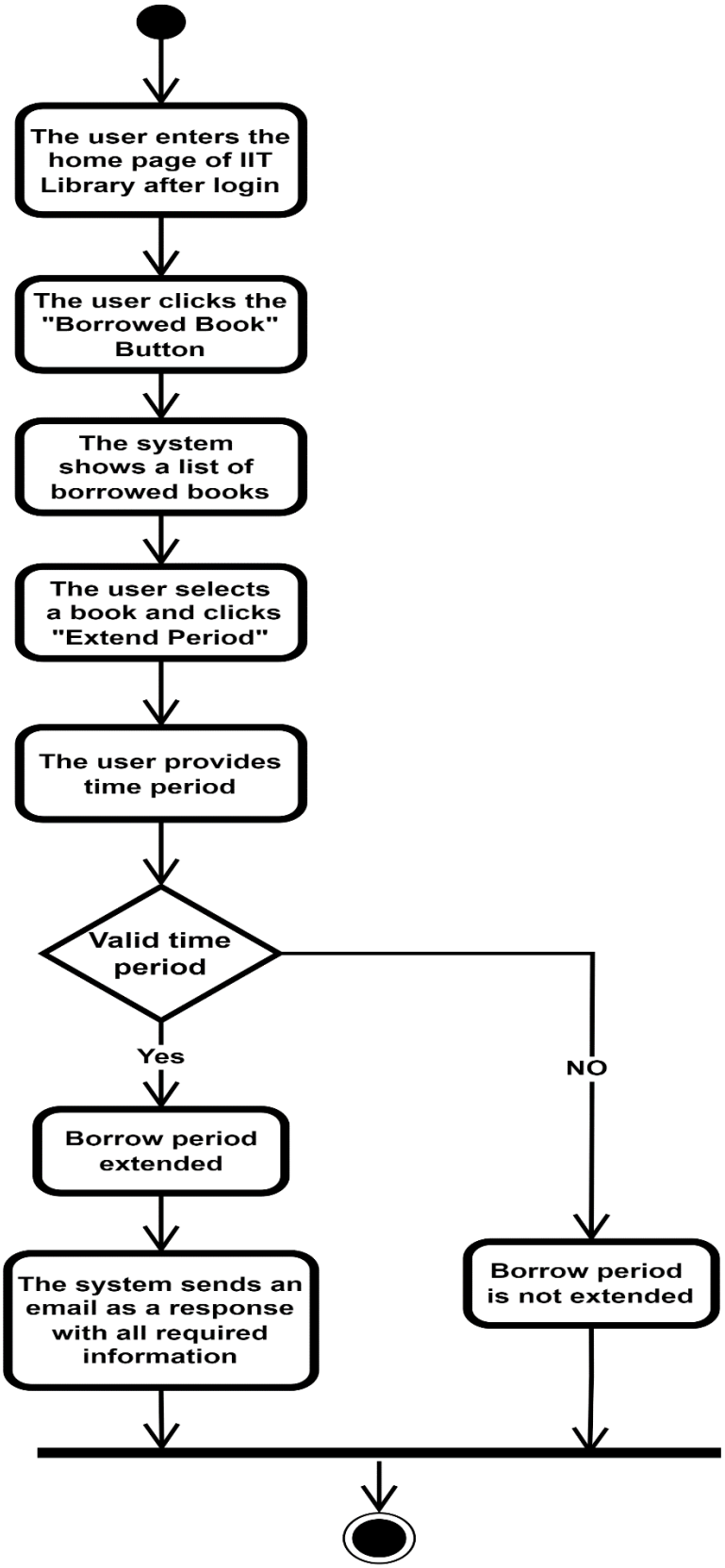

Figure 4 Extend Borrow Period

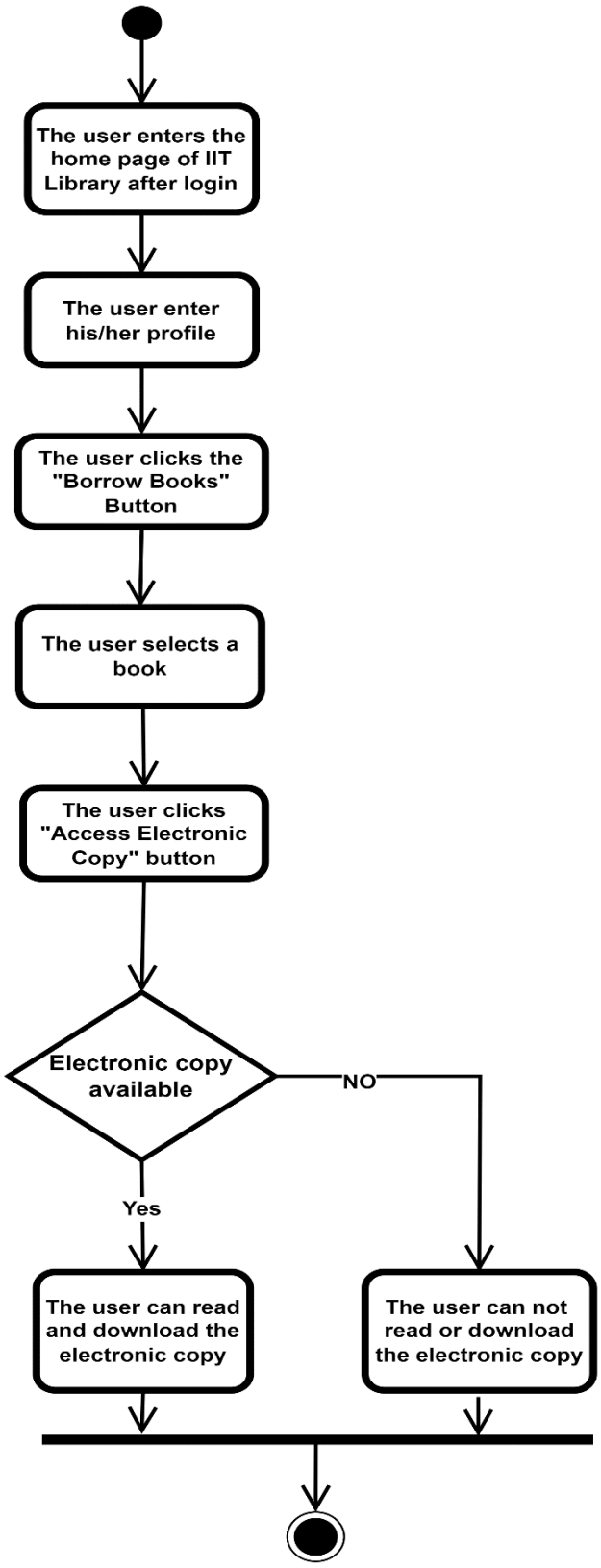

Figure 5 Access Electronic Copy

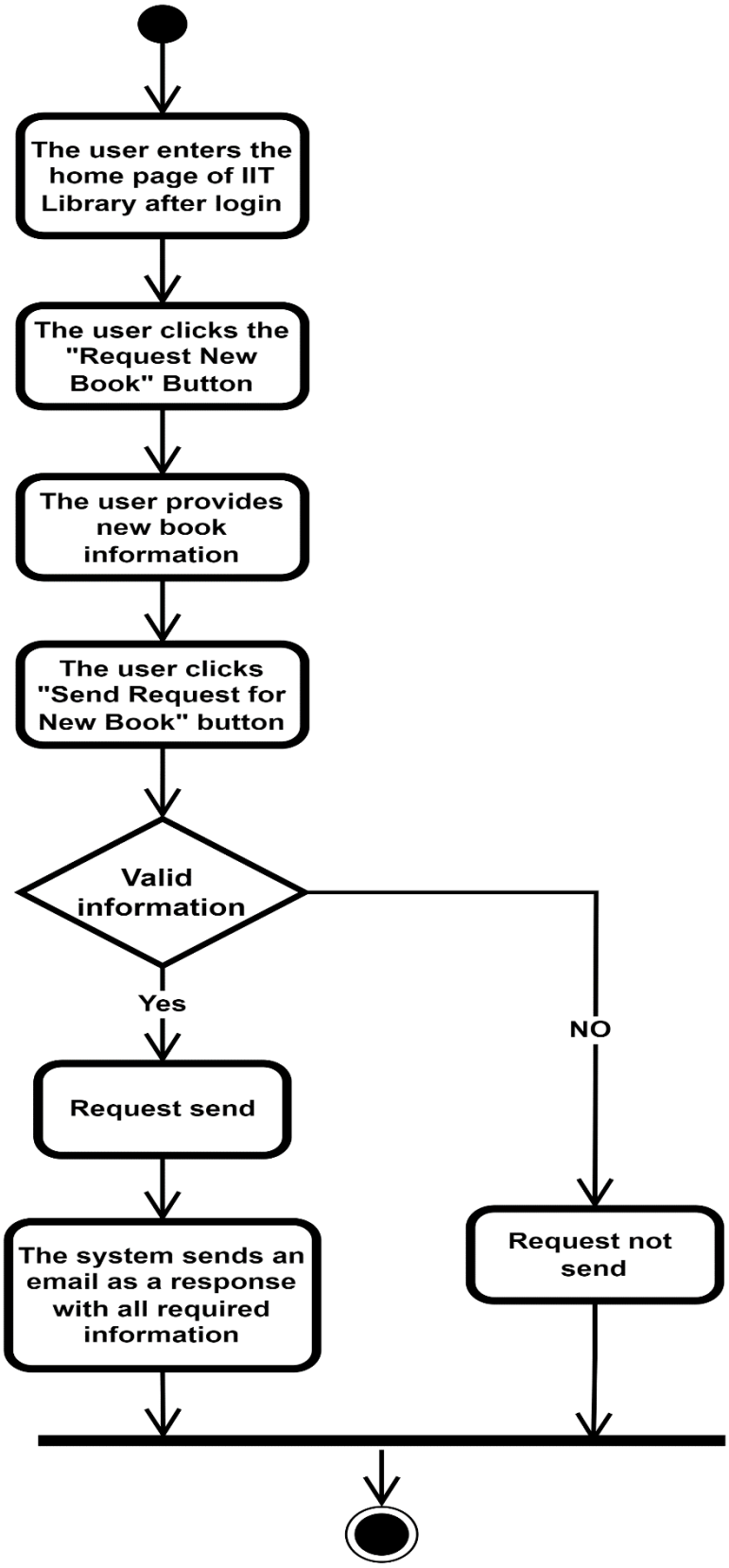

Figure 6 Request New Book

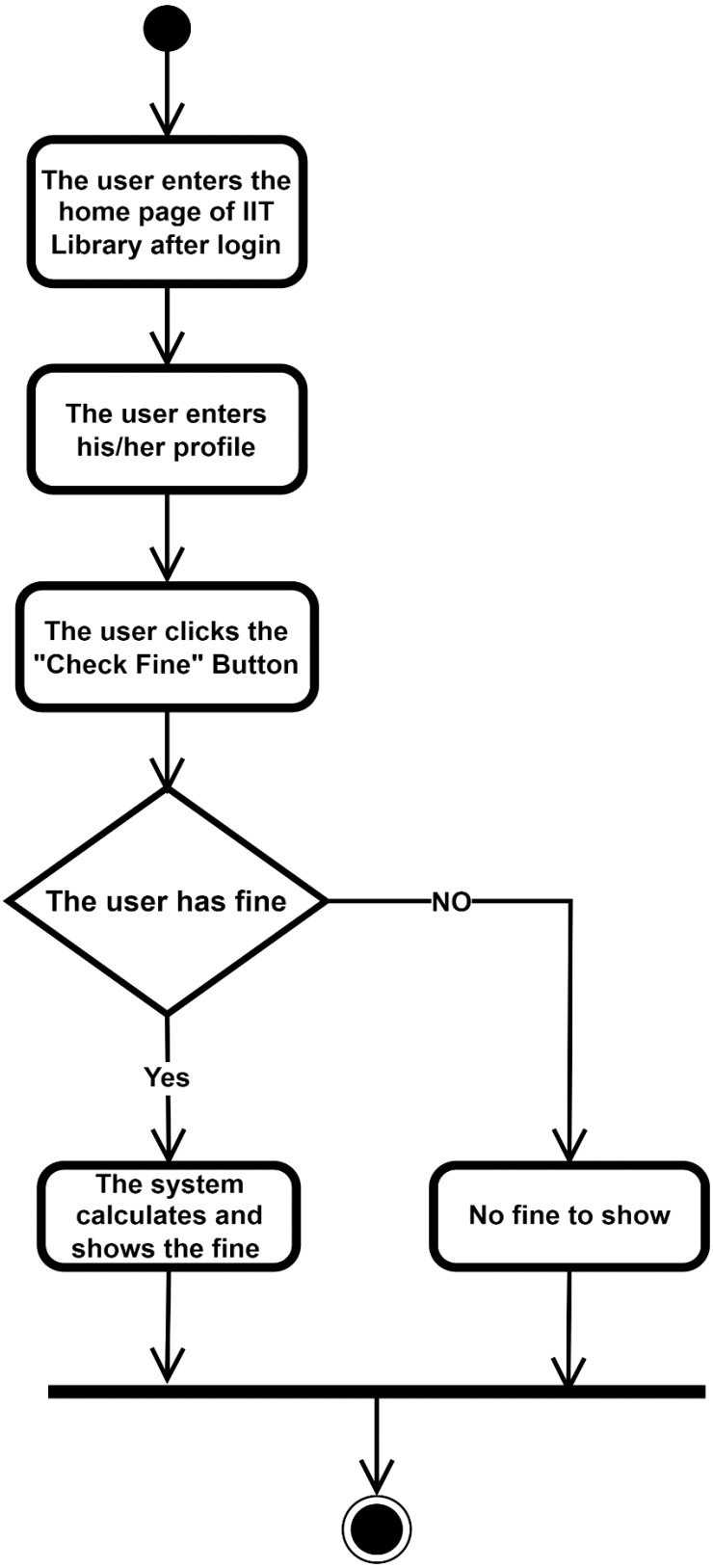

Figure 7 Check Fine

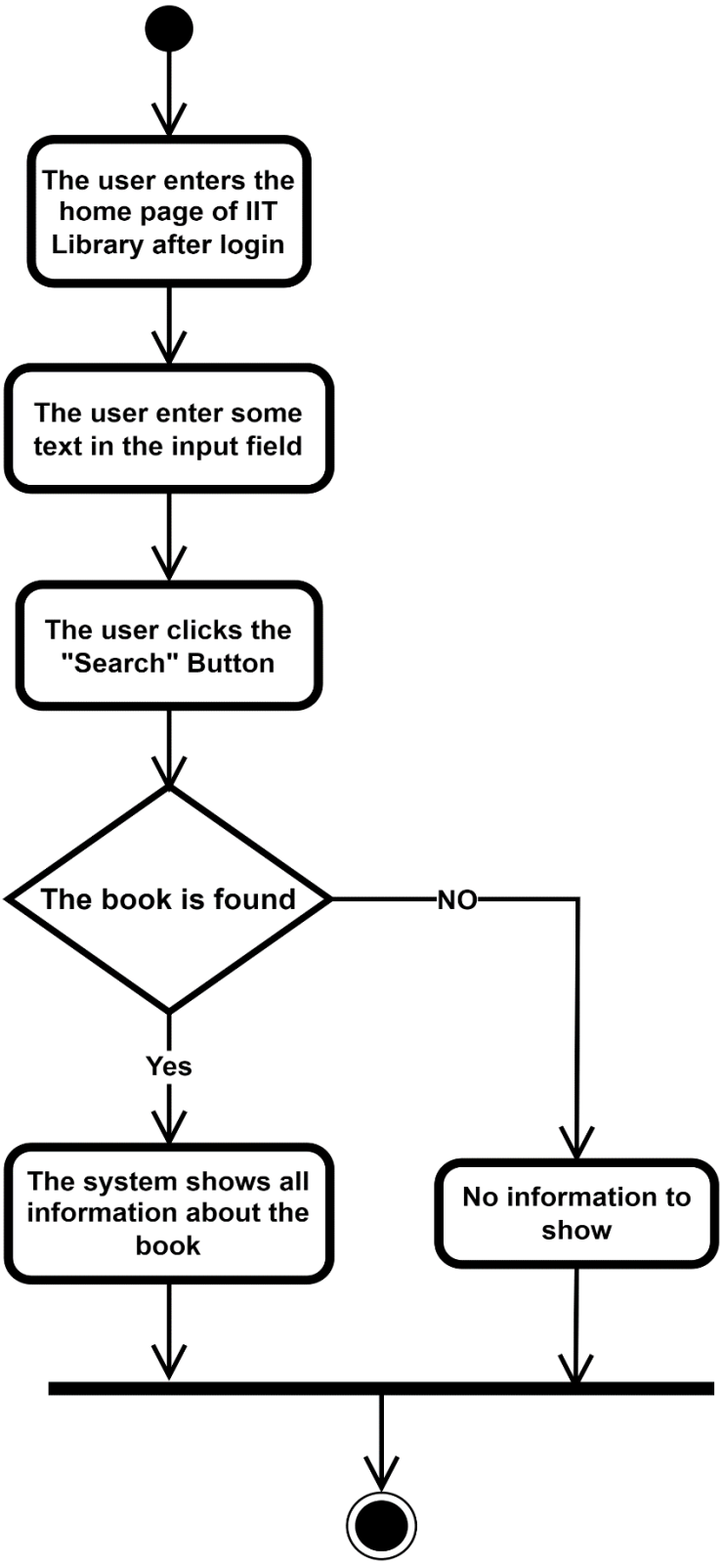

Figure 8 Search Book (By User)

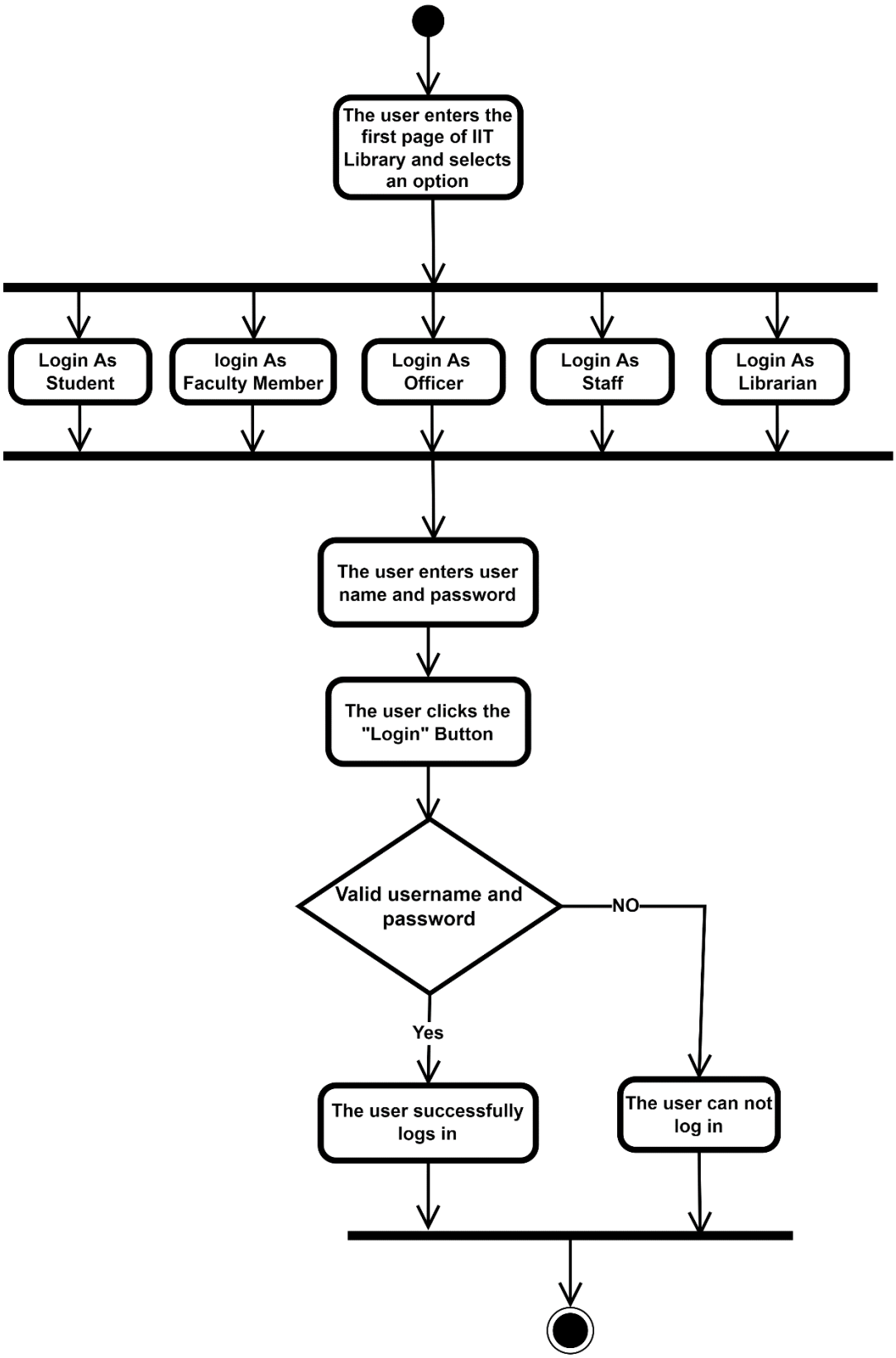

Figure 9 Login

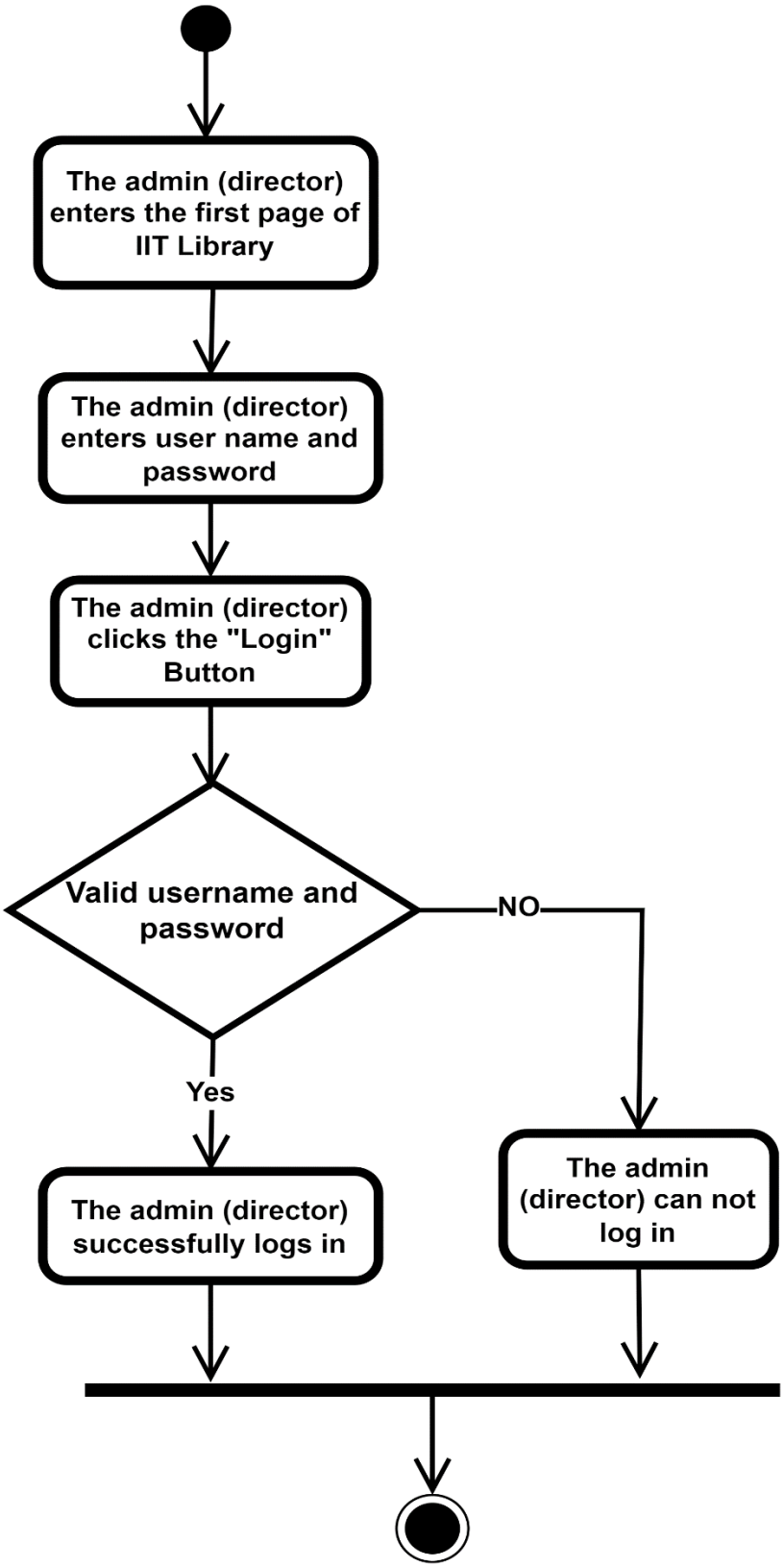

Figure 10 Admin Login

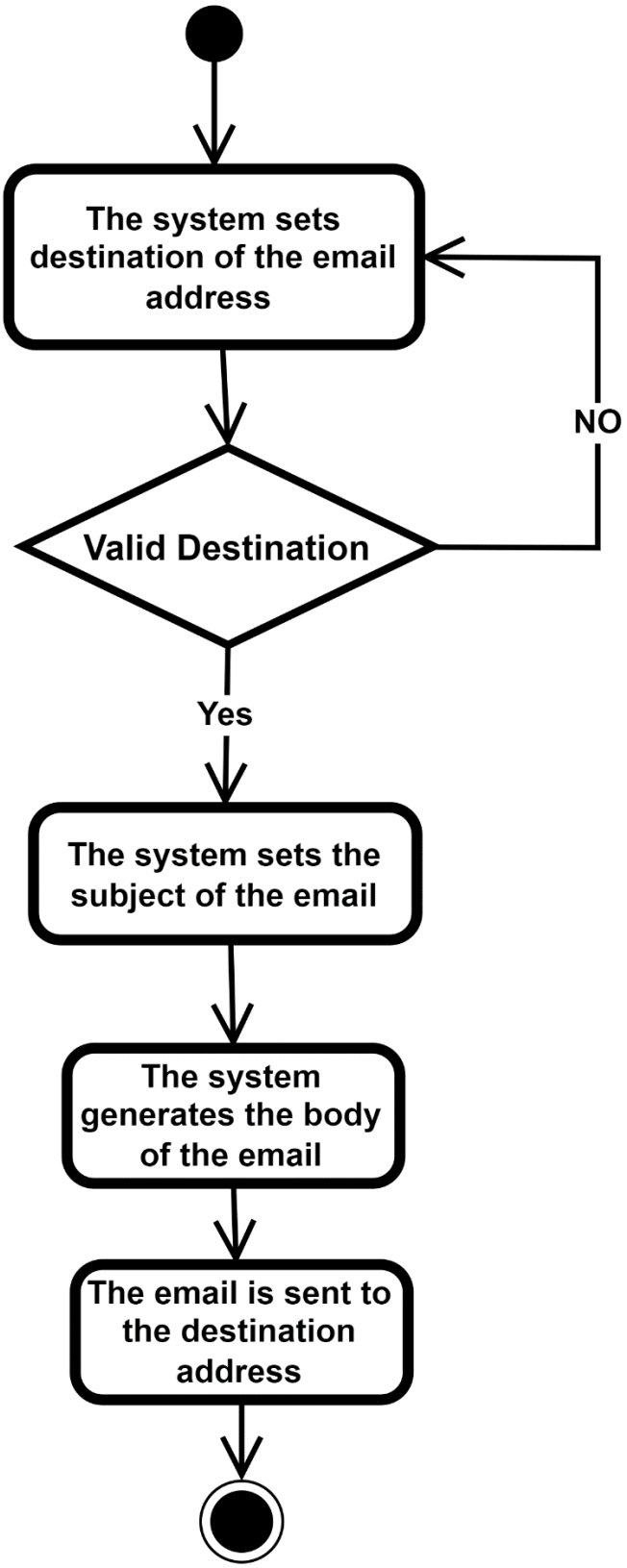

Figure 11 Send Emails

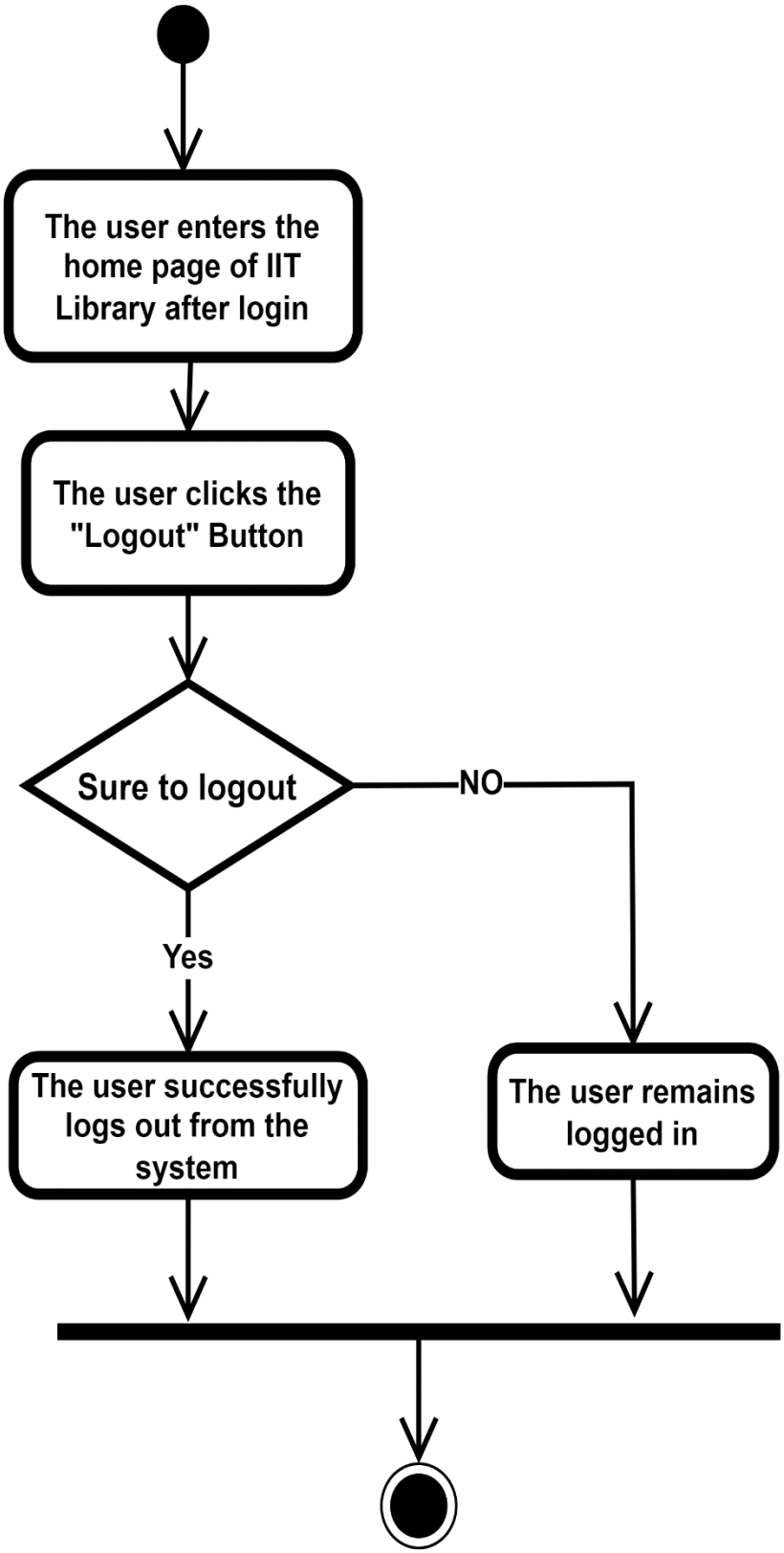

Figure 12 Logout

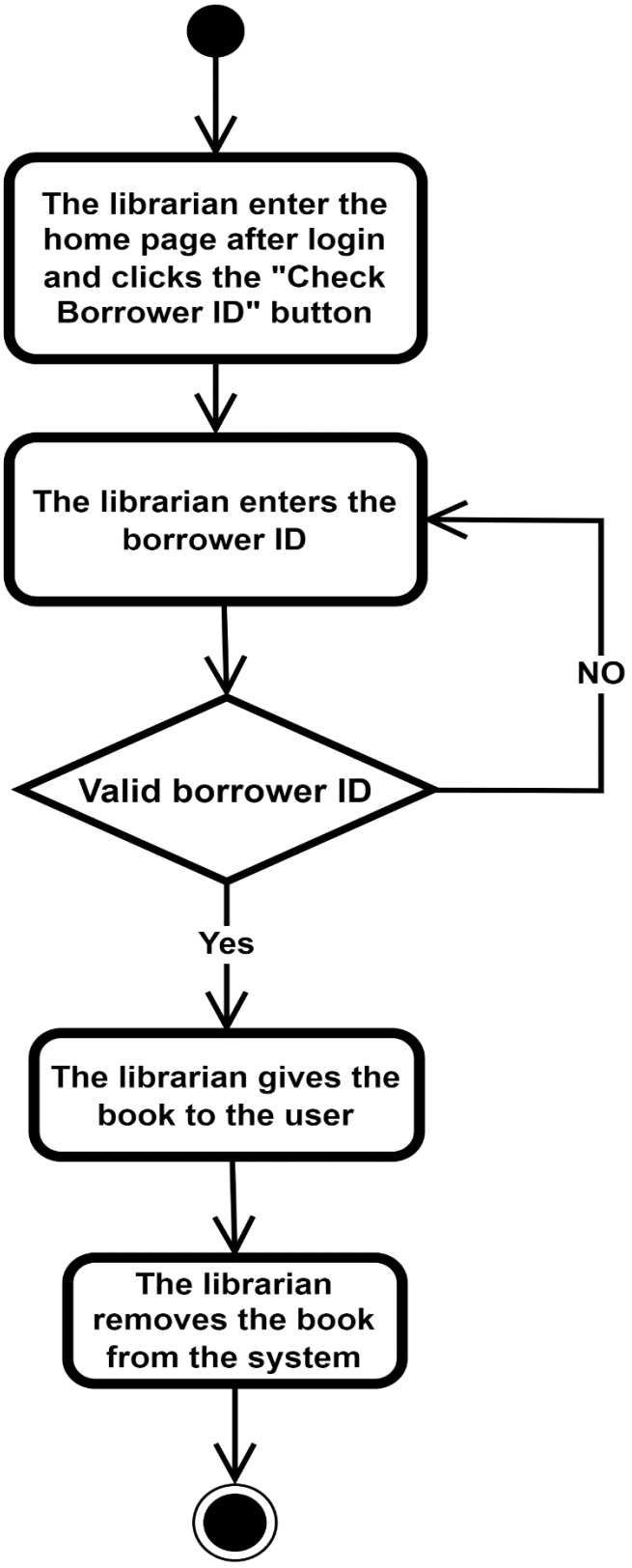

Figure 13 Validate Borrower ID

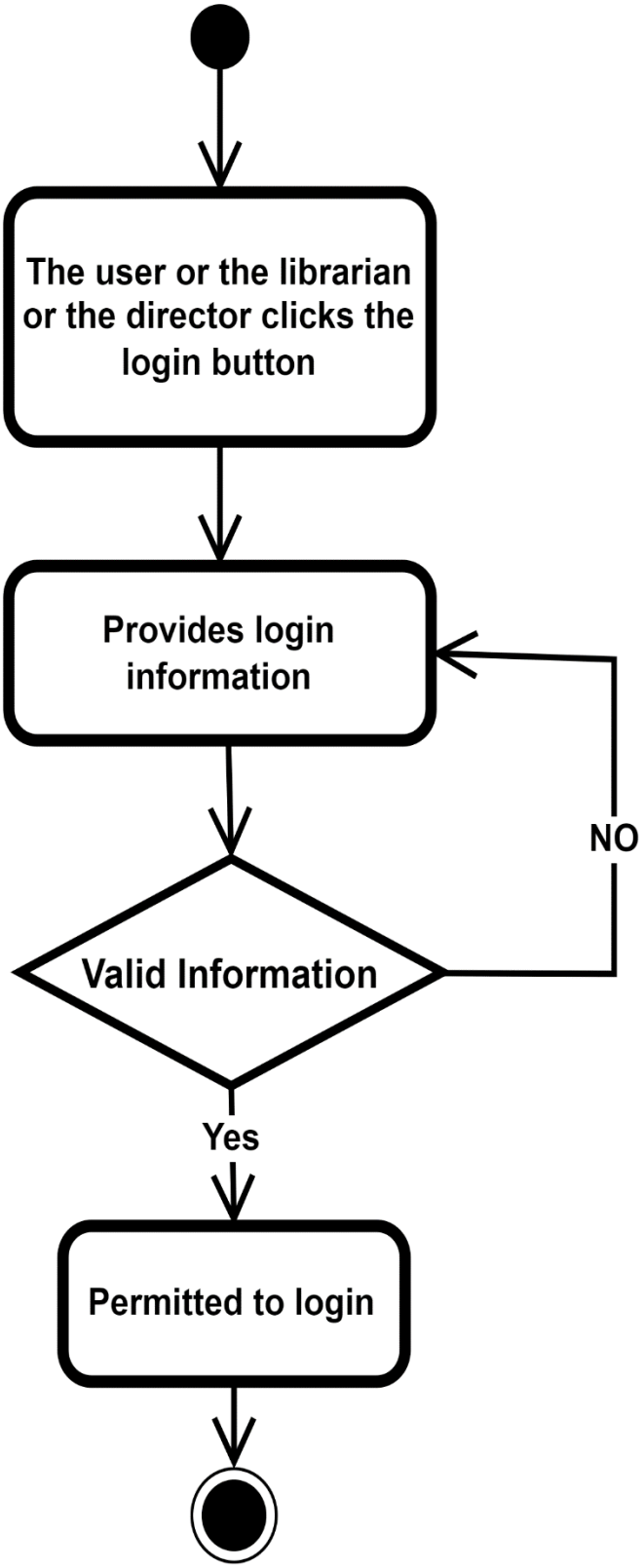

Figure 14 Authenticate User and Librarian

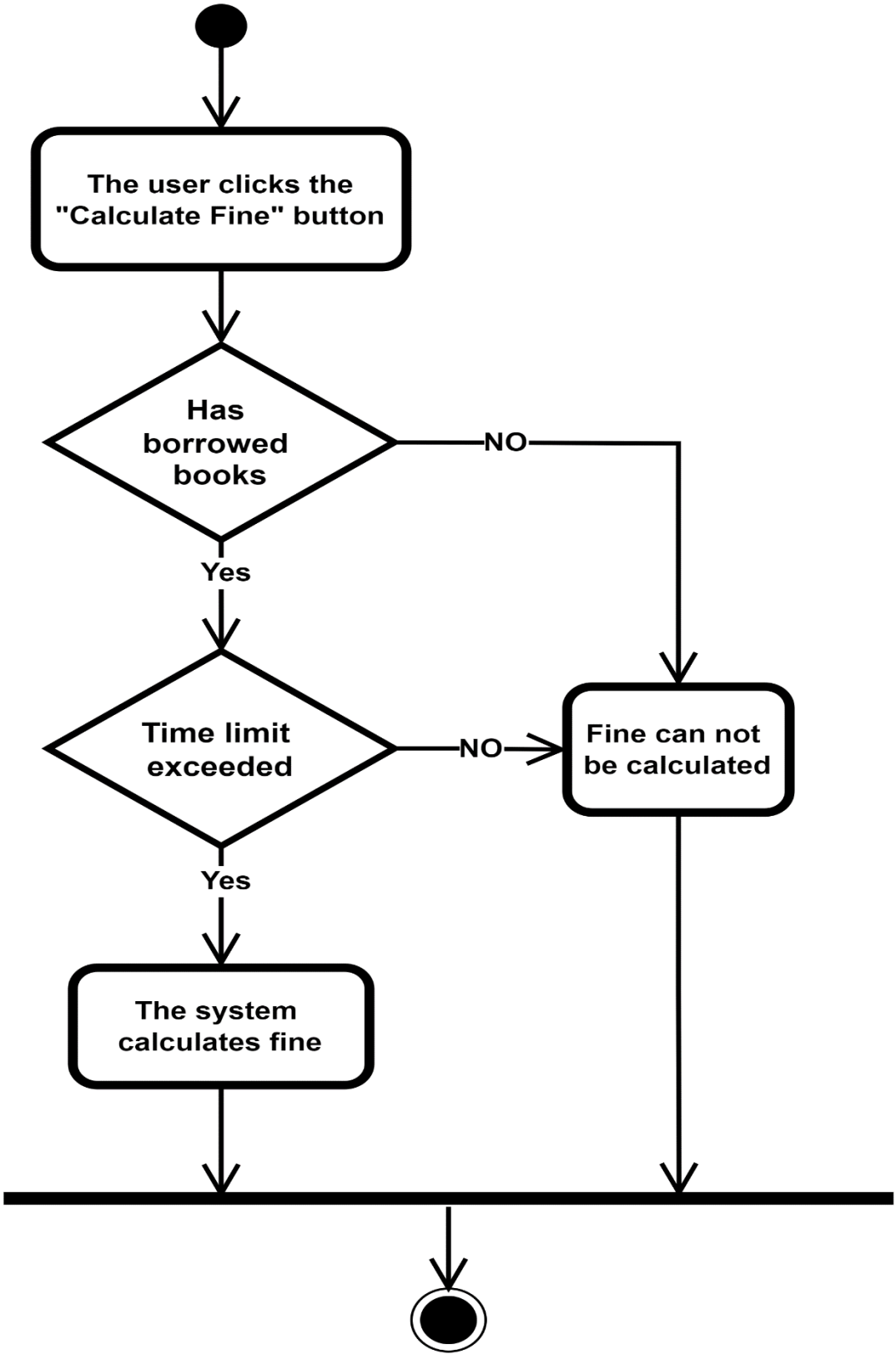

Figure 15 Calculate Fine

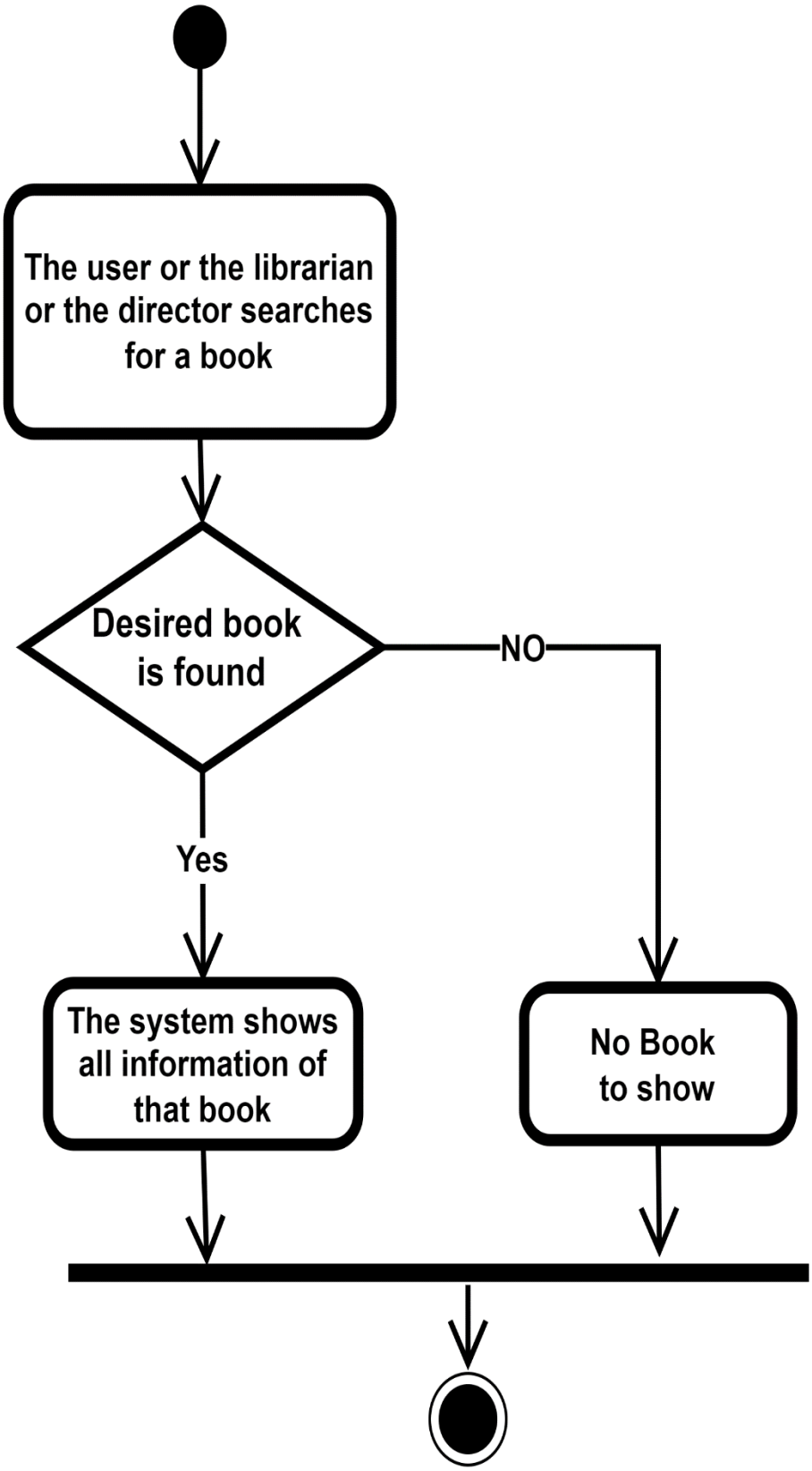

Figure 16 Show Books

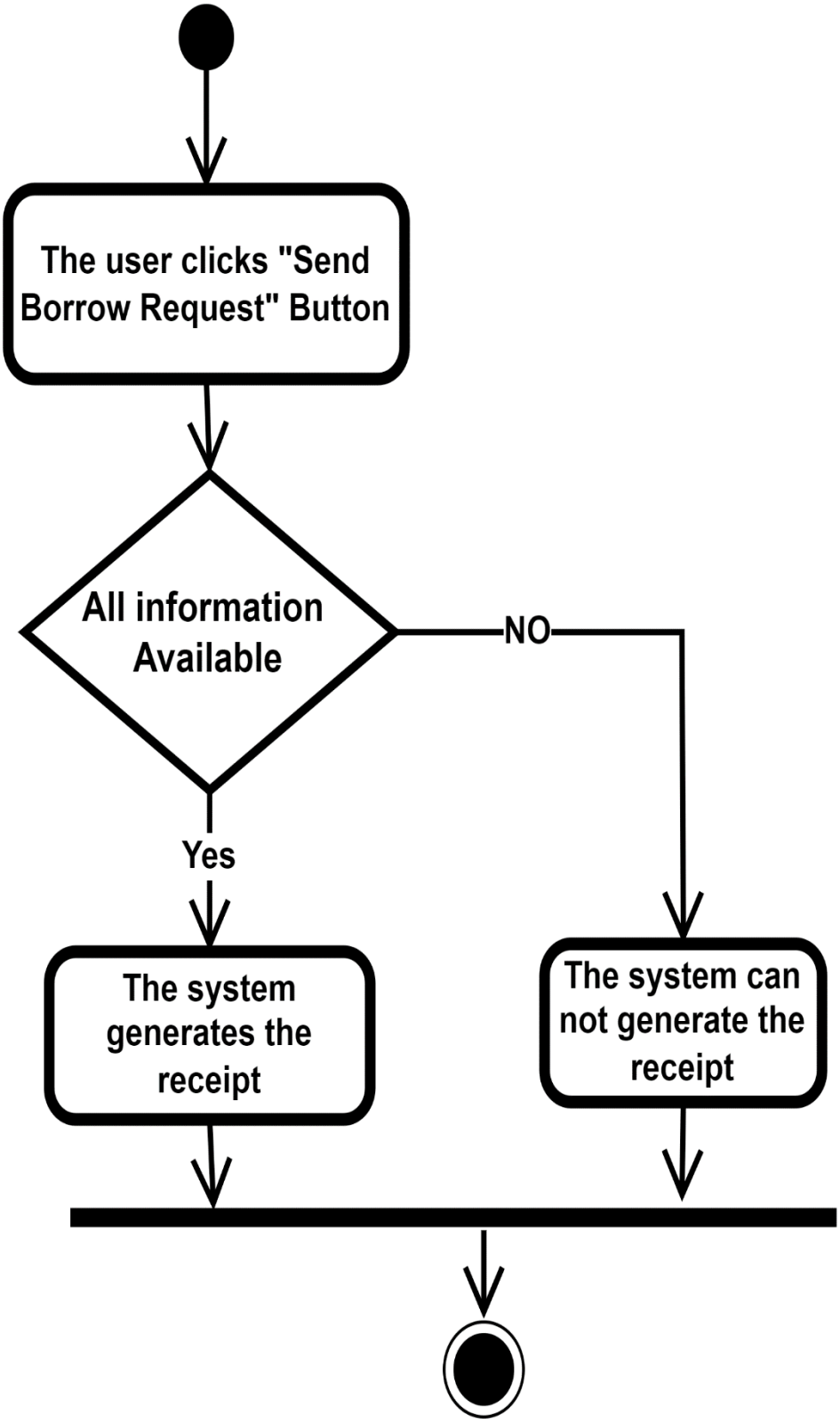

Figure 17 Borrower Receipt Generate

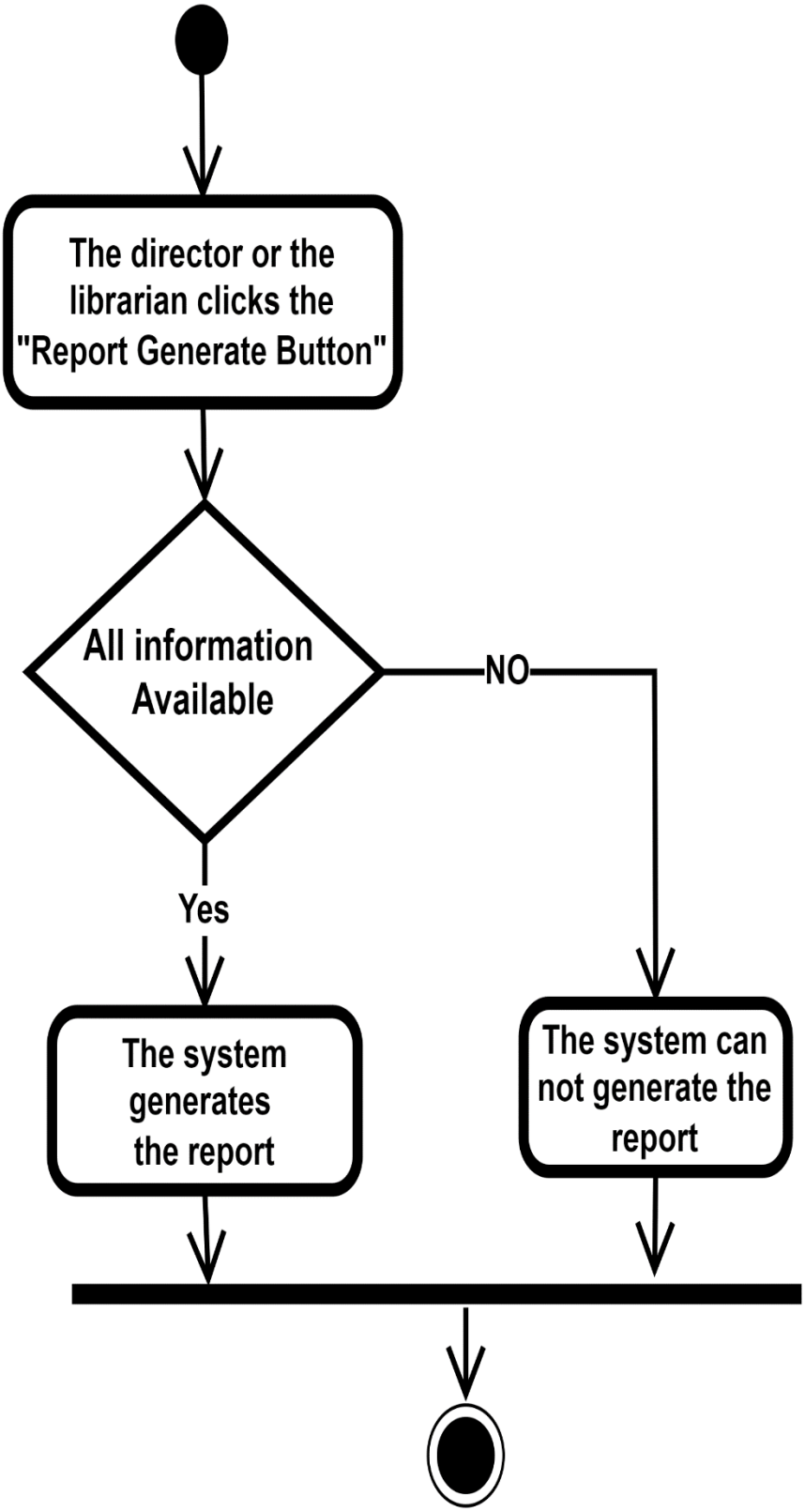

Figure 18 Report Generate

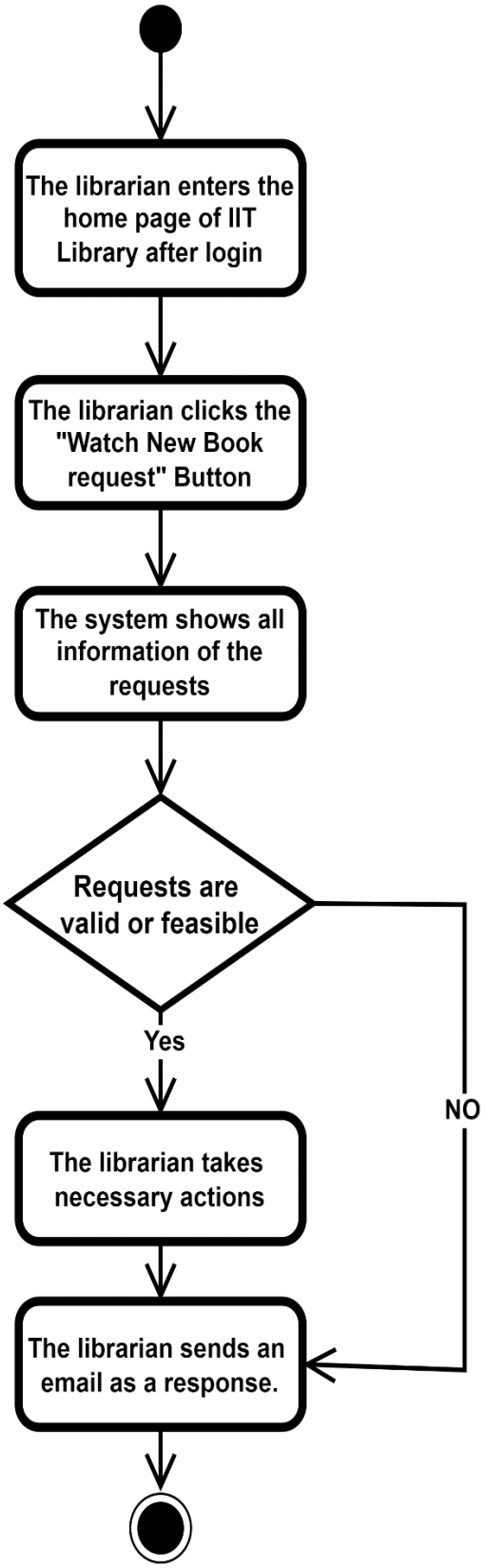

Figure 19 Watch New Book Requests

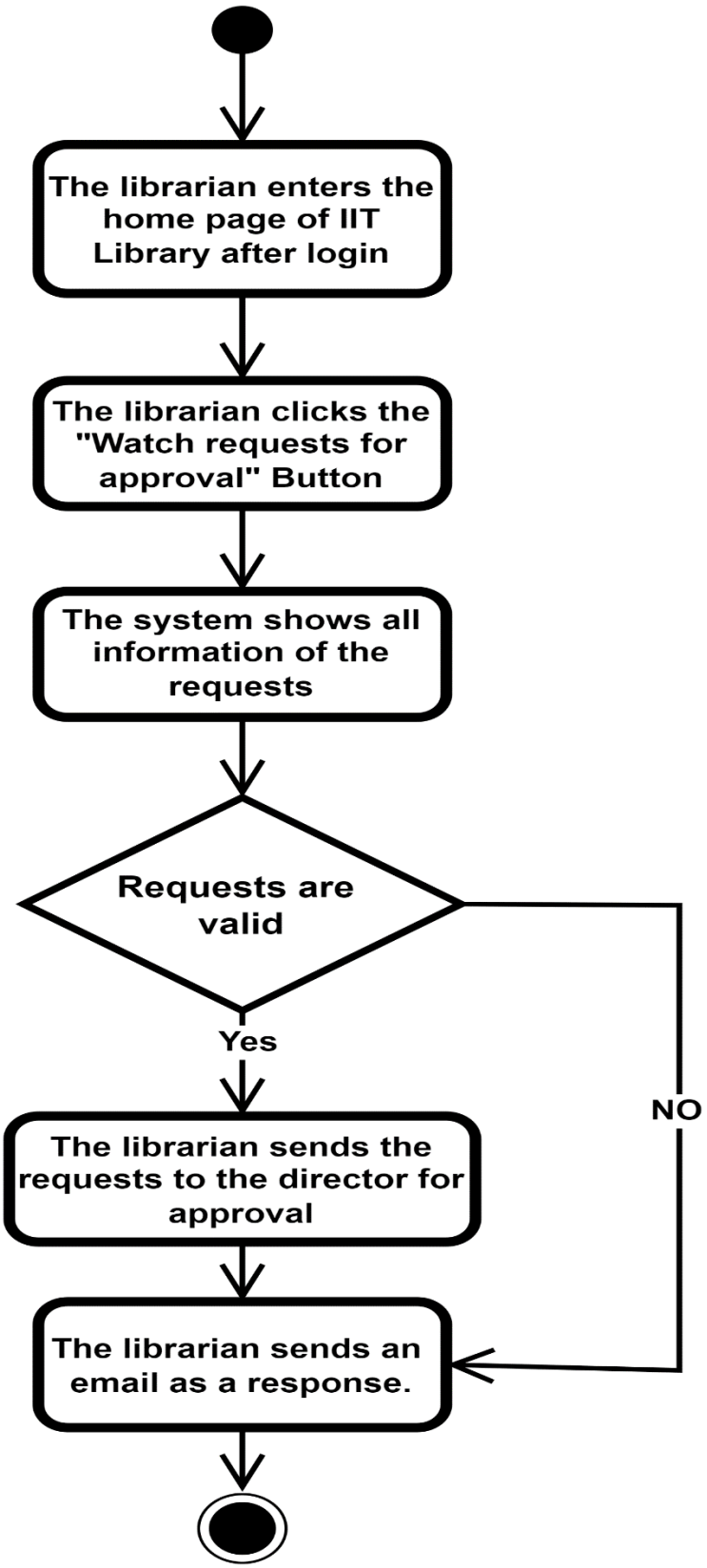

Figure 20 Send Request for Approval

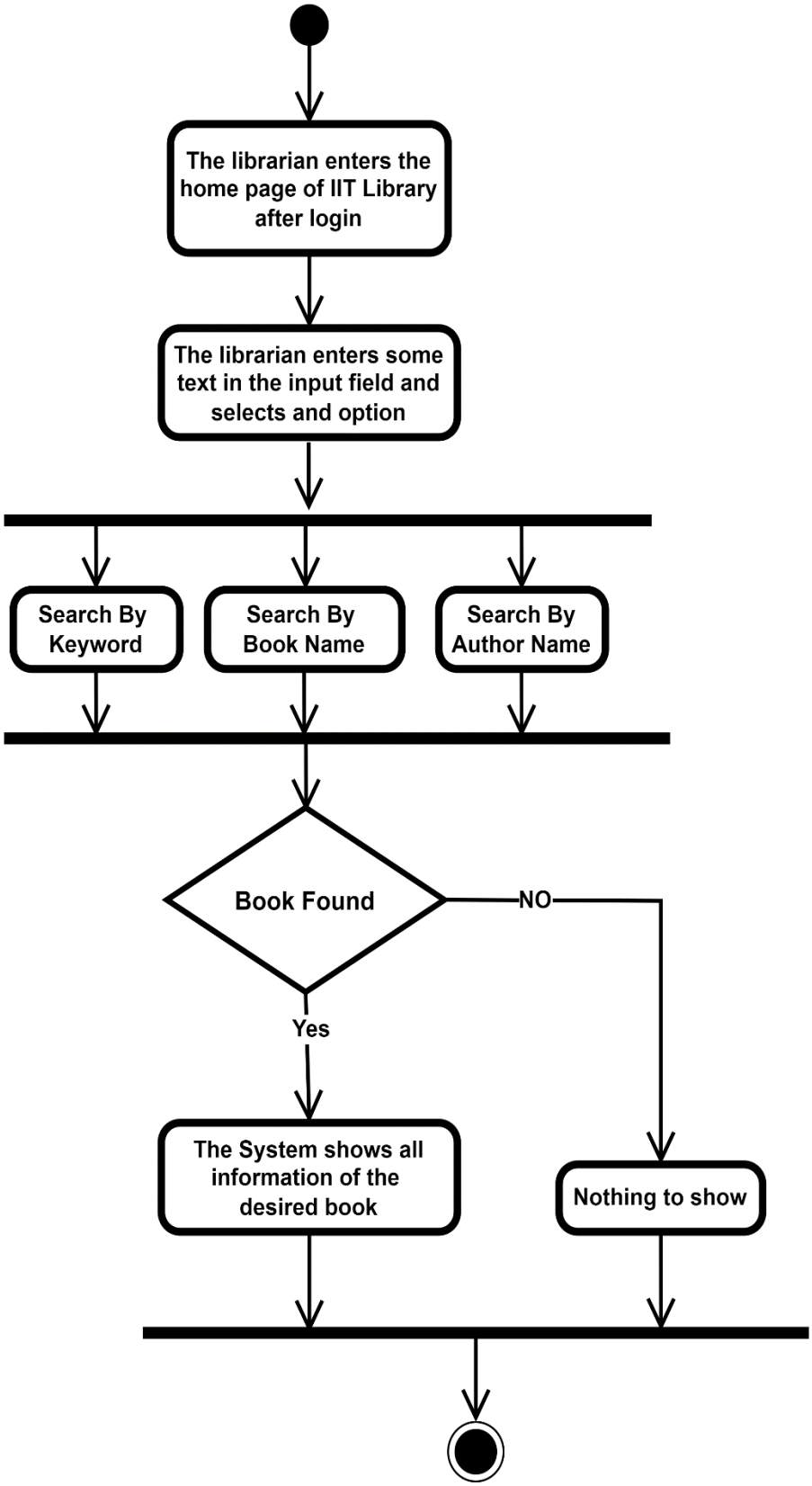

Figure 21 Search Book (By Librarian)

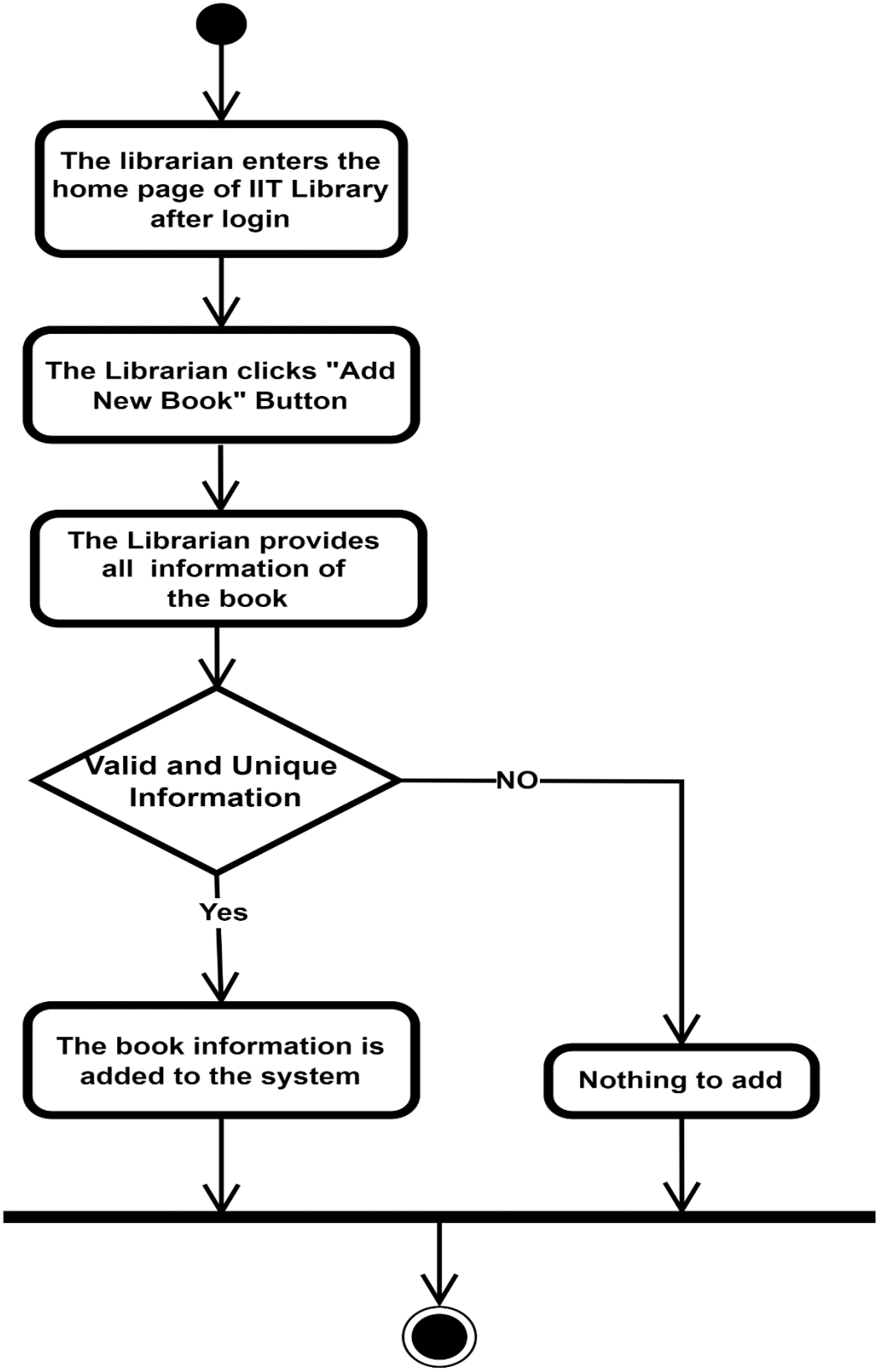

Figure 22 Add New Book

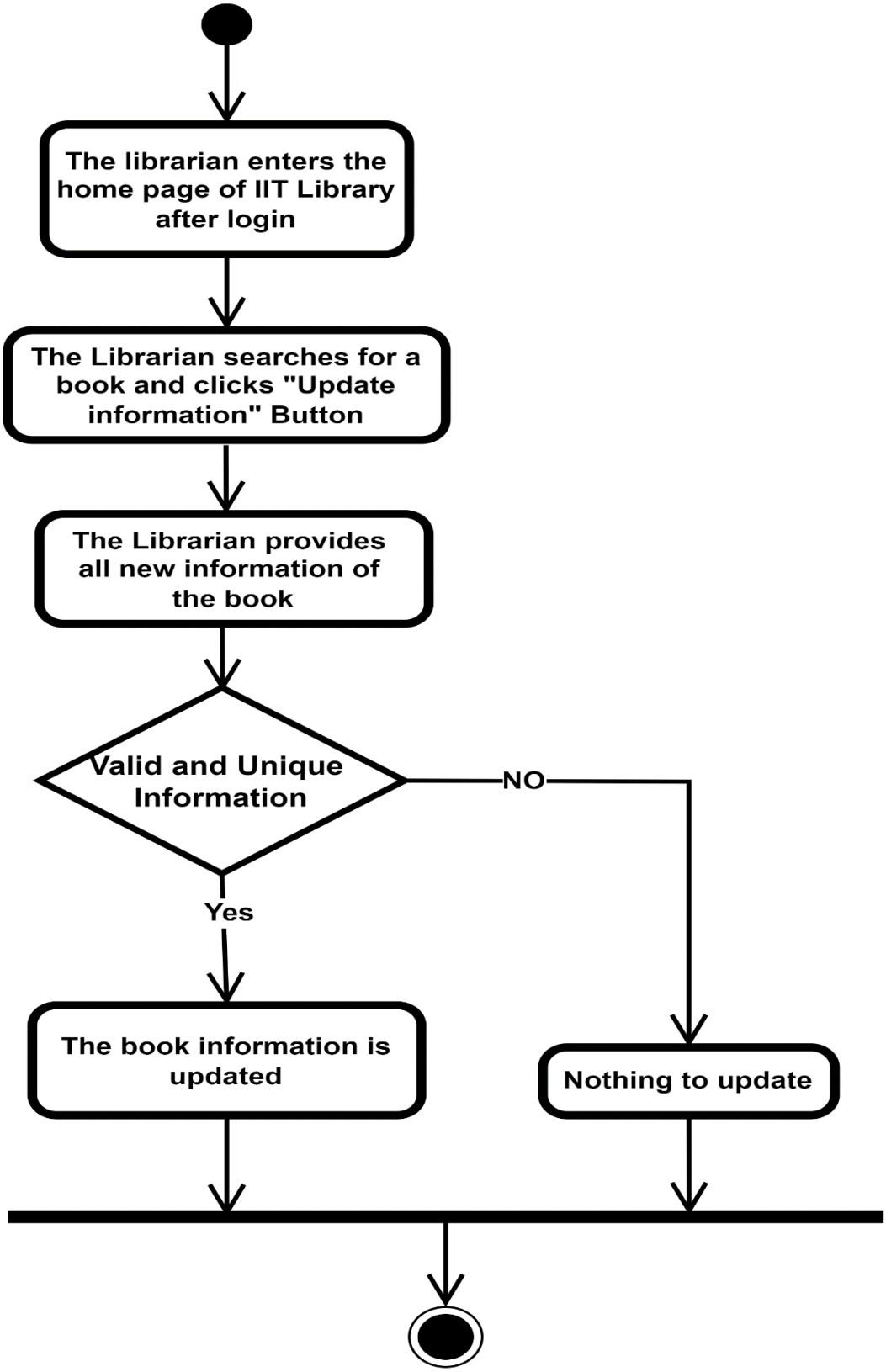

Figure 23 Update Book Information

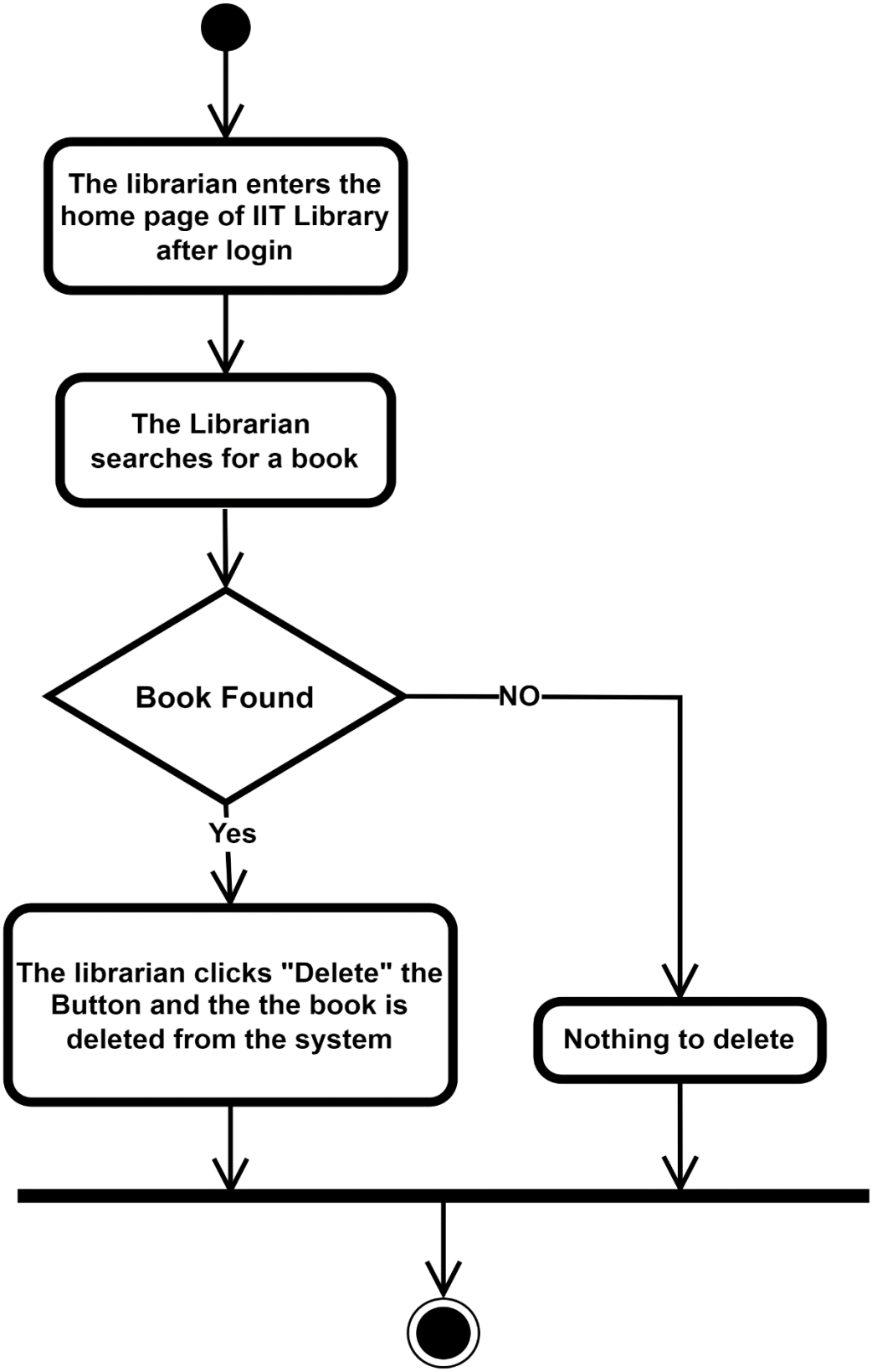

Figure 24 Delete Book

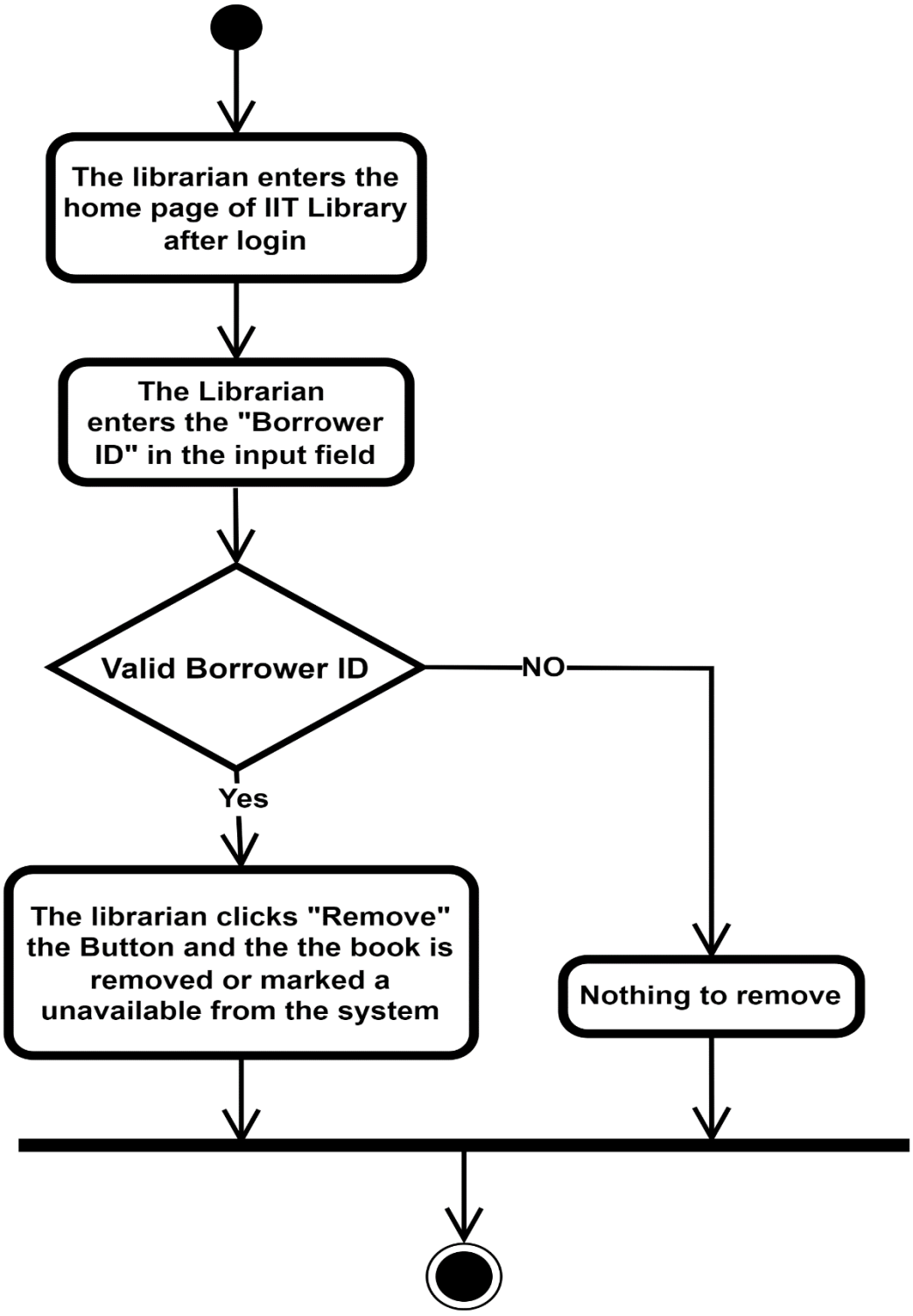

Figure 25 Remove Book

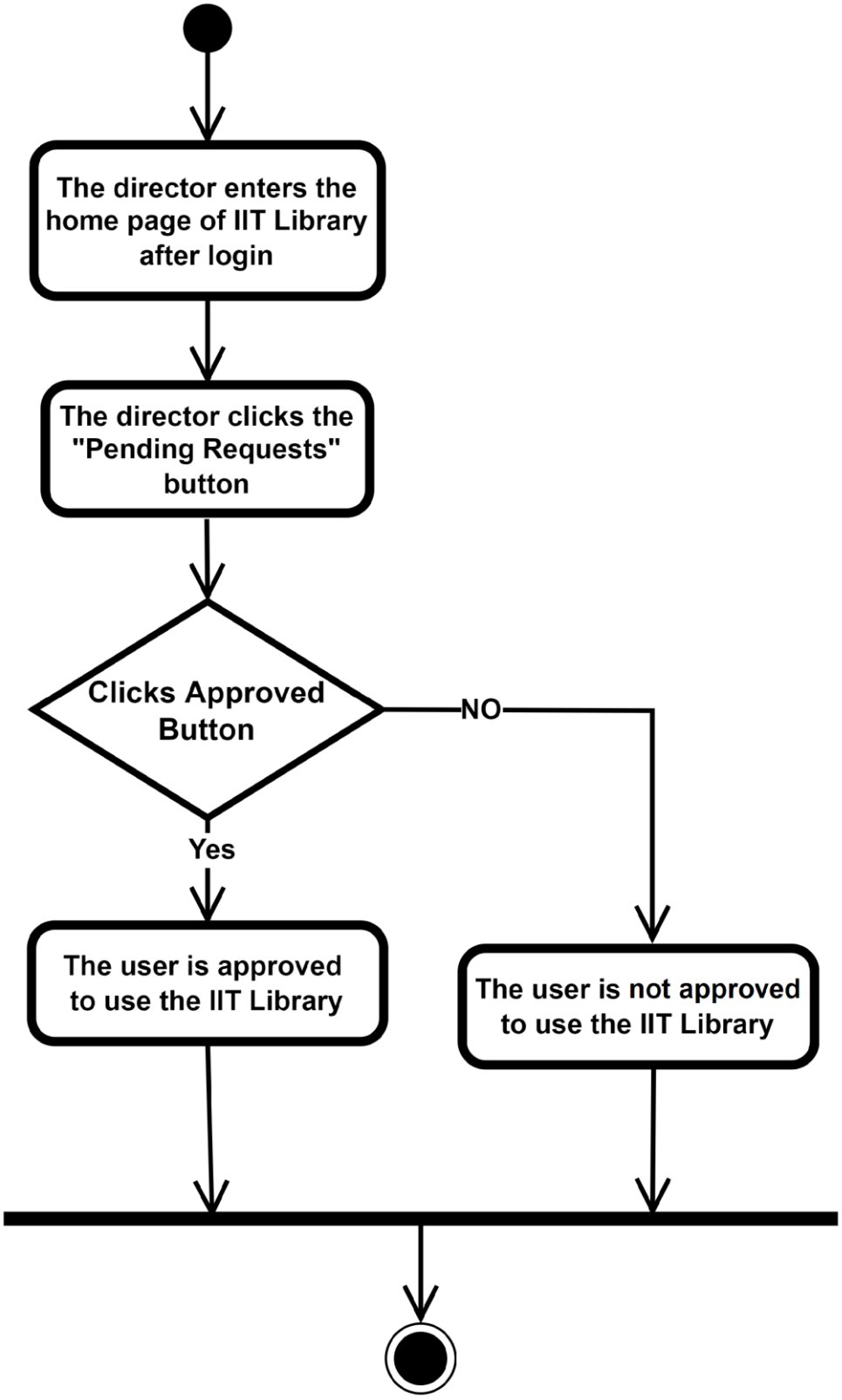

Figure 26 Approve Request

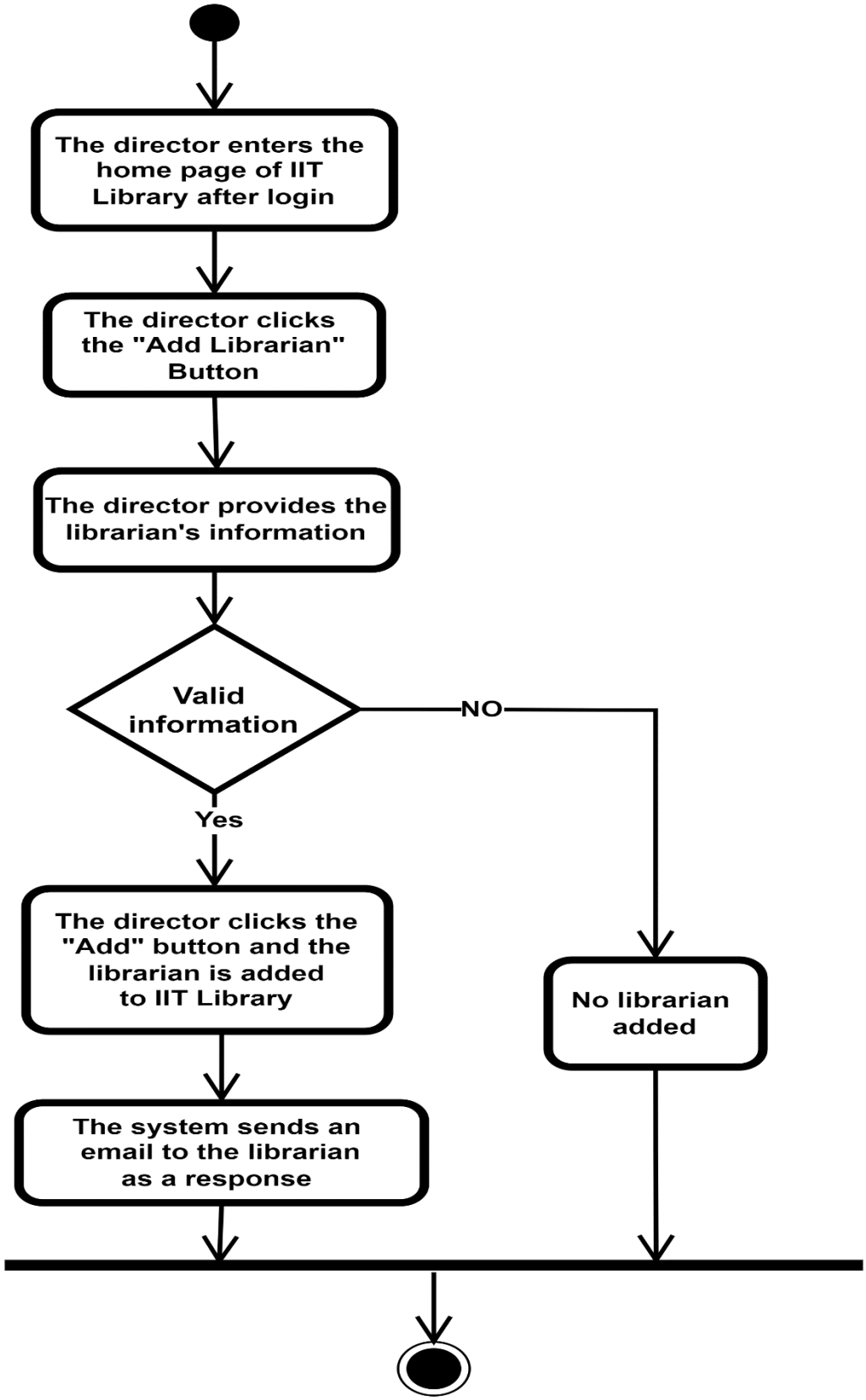

Figure 27 Add Librarian

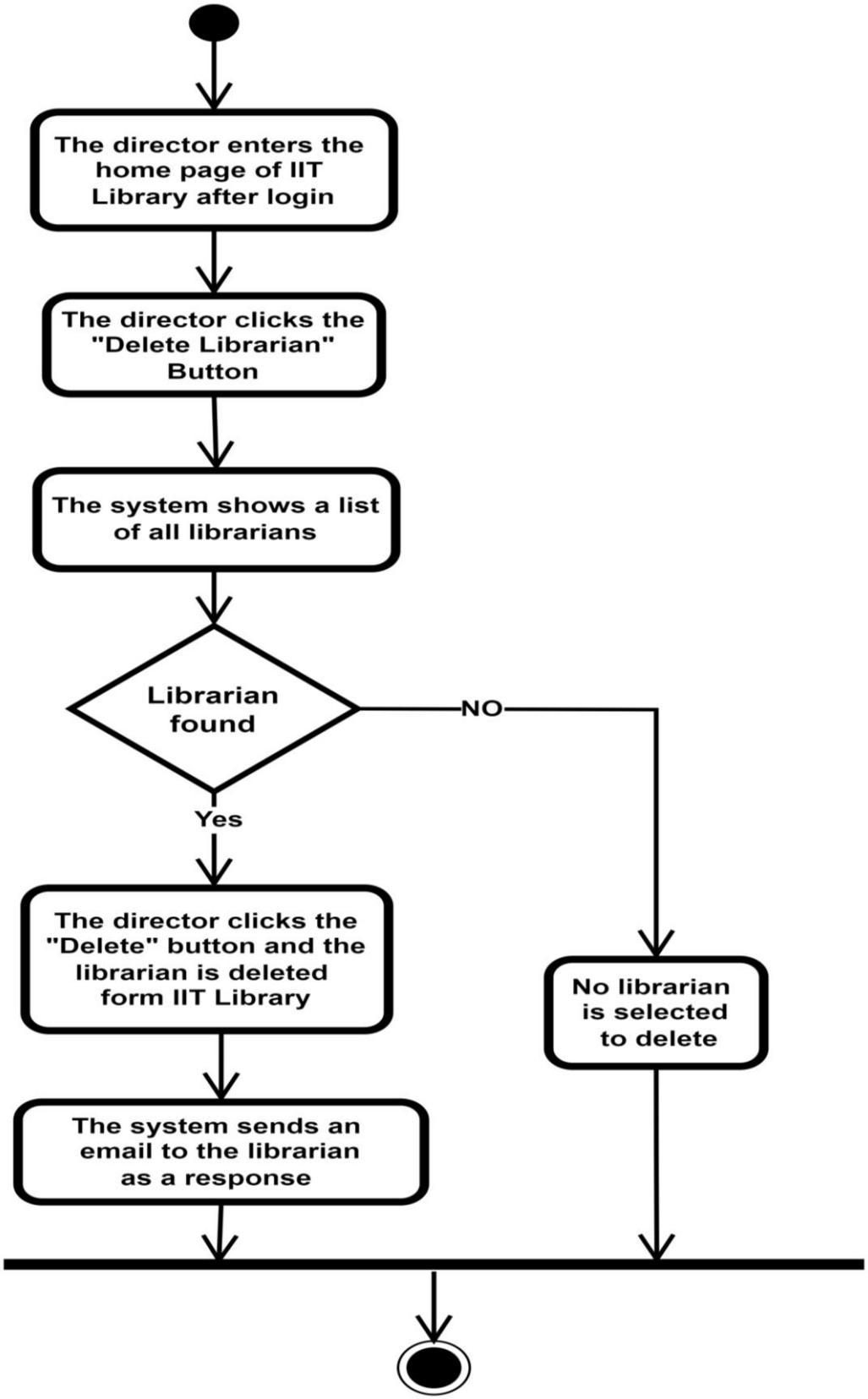

Figure 28 Delete Librarian

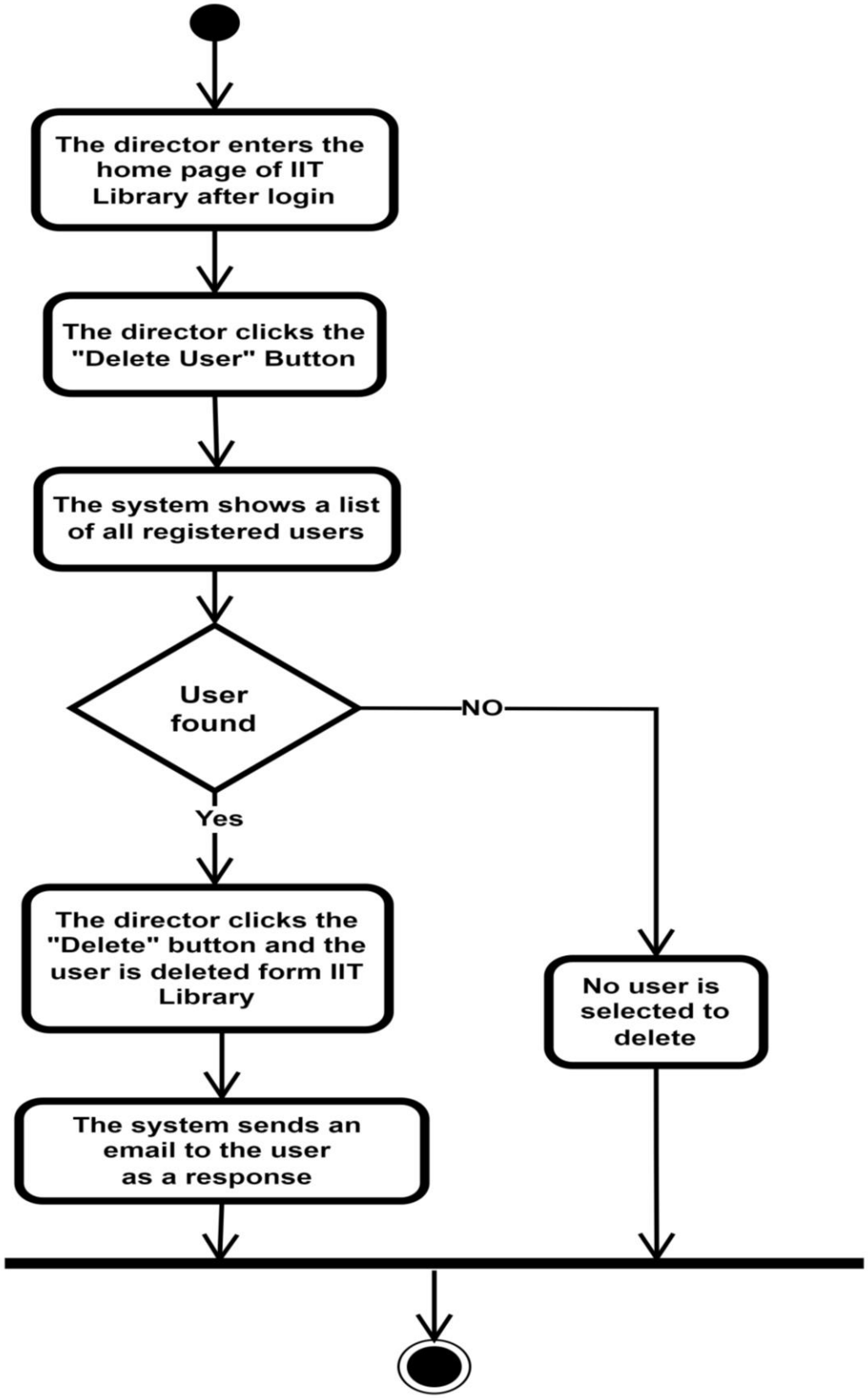

Figure 29 Delete User

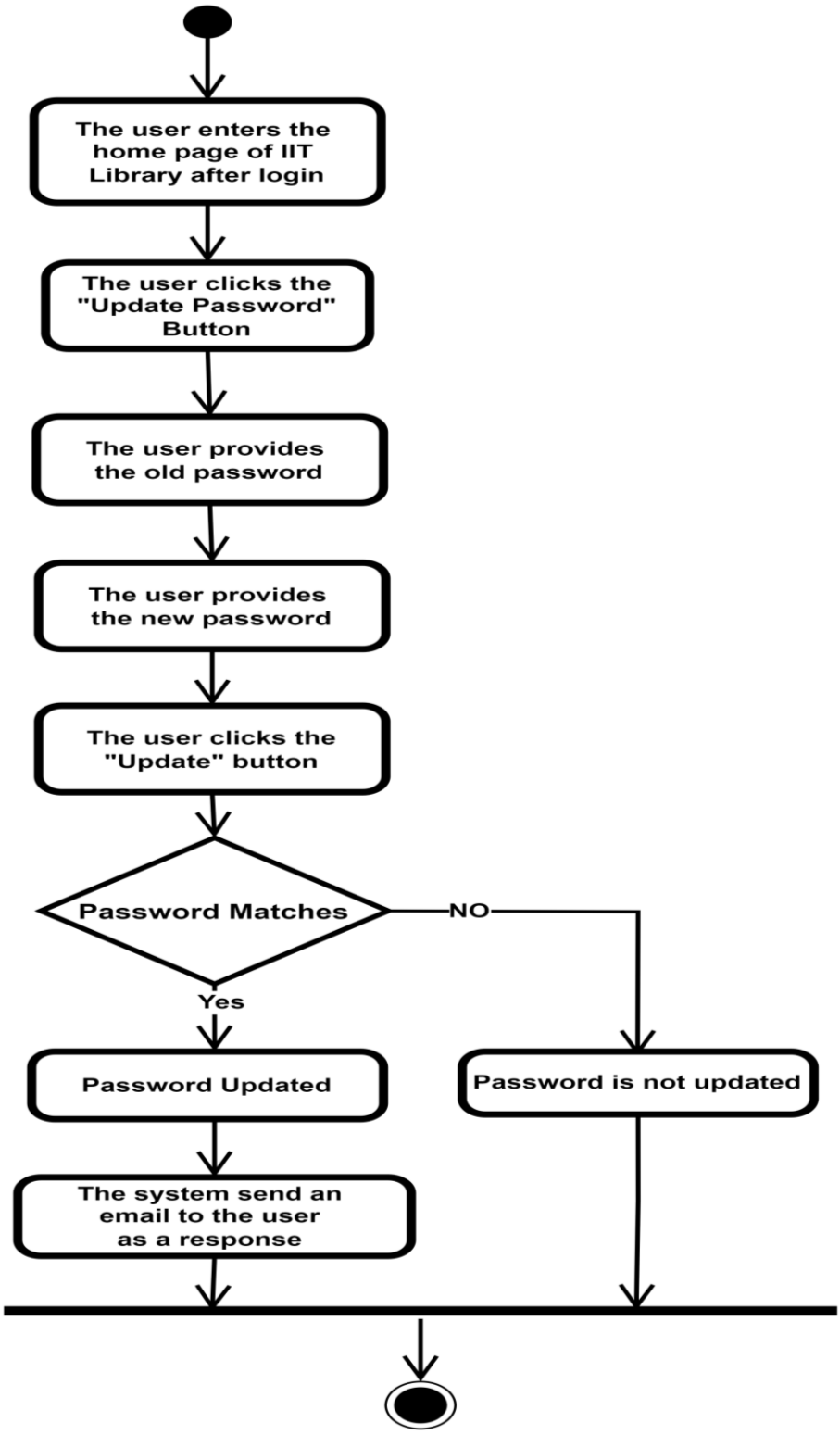

Figure 30 Update Password