



FPT UNIVERSITY

Capstone Project Document

RFID LIBRARY MANAGEMENT SYSTEM

Group GSP21SE16	
Group members	Phan Hoàng Trâm - SE130038 Phạm Minh Hoàng - SE130054 Nguyễn Trung Kiên - SE130154 Nguyễn Đỗ Nhật Khang - SE130148
Supervisor	Nguyễn Thế Hoàng
Capstone Project Code	RLMS

- Ho Chi Minh, 13 January 2021 -

ACKNOWLEDGEMENT

We would like to express sincere appreciation to our lecturers and the FPT University for facilitating such a positive learning environment for us.

This project could not have been completed without the support of Mr. Nguyễn Thế Hoàng – our supervisor. His immense knowledge, plentiful experience and passion have encouraged us during our project.

We would also like to express our gratitude to Mr. Lại Đức Hùng who gave us the opportunity to do this project at the beginning.

Finally, our team also appreciate all the support we received from Mr. Đinh Trường Lâm - Manager at FPTU HCM Library, Mr. Lê Quang Tường - Manager at FPTU Cần Thơ Library, and Ms. Hồ Yên Thực – FPTU HCM Lecturer.

Thank you to everyone and it has been a pleasure to work with you all.

The RLMS Team

Table of Contents

List of Figures.....	6
List of Tables.....	8
I. Project Introduction.....	11
1. Overview.....	11
1.1. Project Information.....	11
1.2. Project Team.....	12
2. Product Background.....	12
3. Existing Systems.....	13
3.1. FPTU Hồ Chí Minh Library System.....	13
3.2. FPTU Cần Thơ Library System.....	13
3.3. Văn Lang University Library System.....	13
4. Problem Identification.....	13
5. Proposed solution.....	14
6. Project Scope & Limitations.....	14
6.1. Major Features.....	14
6.2. Limitations & Exclusions.....	15
7. Terms and Definitions.....	16
II. Project Management Plan.....	18
1. Overview.....	18
1.1. WBS & Estimation.....	19
1.2. Project Objectives.....	20
1.3. Project Risks.....	20
2. Management Approach.....	21
2.1. Project Process.....	21
2.2. Quality Management.....	21
2.3. Training Plan.....	22
3. Project Organization.....	22
3.1. Team & Structures.....	22
3.2. Roles & Responsibilities.....	23
4. Project Communication.....	23
4.1. Communication Plan.....	23
4.2. External Interface.....	24
5. Configuration Management.....	24
5.1. Tools & Infrastructures.....	24

5.2. Document Management.....	25
5.3. Source Code Management.....	25
5.4. Project Tasks Management.....	25
III. Software Requirement Specification.....	26
1. Overall Description.....	26
1.1. User Requirement Overview.....	26
2. User Requirements Specification.....	27
2.1. Use Case.....	27
2.2. Use Case Specifications.....	32
3. Functional Requirements.....	189
3.1. System Functional Overview.....	189
4. Non-Functional Requirements.....	200
4.1. External Interfaces.....	200
4.2. Quality Attributes.....	200
IV. Software Design Document.....	202
1. Overall Description.....	202
1.1. Assumptions.....	202
1.2. Design Constraints.....	202
2. System Architecture Design.....	203
3. Device Description.....	204
3.1. UHF RFID Reader (Book Tag Reader).....	204
3.2. HF RFID Reader Module.....	205
3.3. Stepper Motor.....	206
3.4. Arduino UNO Board.....	207
3.5. Pro Micro Board.....	208
3.6. HF RFID Tags.....	210
3.7. UHF RFID Tags.....	211
3.8. Passive Buzzer.....	213
3.9. LED.....	214
3.10. Mini Breadboard.....	215
3.11. RS232 Converter.....	216
3.12. Stepper Motor Controller Board ULN2003.....	217
4. Component Diagram.....	217
5. System Details Design.....	219
5.1. Class Diagram.....	219

5.2. Class Specification.....	221
5.3. State Machine Diagrams.....	231
5.4. Sequence Diagrams.....	233
6. Data & Database Design.....	236
6.1. Physical Diagram.....	236
6.2. Database Description.....	238
7. Technology Solution.....	247
7.1. RFID Technology.....	247
7.2. Security Gate.....	253
7.3. Book Drop.....	255
7.4 Obstacles and Solutions.....	257
V. Software Testing Documentation.....	259
1. Overall Description.....	259
1.1. Test Model.....	259
1.2. Testing Levels.....	259
2. Test Plan.....	259
2.1. Test Stages.....	259
2.2. Resources.....	259
2.3. Test Milestones.....	260
3. Test Cases.....	260
4. Test Reports.....	260
VI. Release Package & User Guides.....	261
1. Deliverable Package.....	261
1.1. Source codes & documents.....	261
1.2. Known Issues, Limitations & Restriction.....	261
2. Installation Guides.....	261
2.1. System Requirements.....	261
2.2. Setup Files.....	262
2.3. Installation Instruction.....	262
3. User Manual.....	263
3.1. Terms and definitions.....	263
3.2. System Requirements.....	264
3.3. Application Usage.....	265
VII. Appendix.....	345
1. References.....	345

List of Figures

Figure 1 <Fishbone diagram> Major Features.....	15
Figure 2 Scrum Framework.....	21
Figure 3 Team & Structures.....	22
Figure 4 <Use Case Overview> RLMS Use Case Diagram.....	28
Figure 5 <Use Case Overview> Unauthenticated User.....	32
Figure 6 <Use case> Log in.....	32
Figure 7 <Use Case> Return book at kiosk.....	34
Figure 8 <Use Case> Return book at book drop.....	37
Figure 9 <Use Case Overview> Authenticated User.....	40
Figure 10 <Use Case> Log out.....	41
Figure 11 <Use Case> Login with ID card.....	42
Figure 12 <Use Case> View book details.....	45
Figure 13 <Use Case> Change password.....	47
Figure 14 <Use Case> Add book to search list.....	49
Figure 15 <Use Case> Search book by RFID reader.....	50
Figure 16 <Use Case> View profile.....	53
Figure 17 <Use Case> Update profile.....	54
Figure 18 <Use Case Overview> Patron.....	56
Figure 19 <Use Case> Check in with patron card.....	57
Figure 20 <Use Case> Borrow books at kiosk.....	59
Figure 21 <Use Case> Add book to wishlist.....	62
Figure 22 <Use Case> View wishlist books.....	64
Figure 23 <Use Case> View borrow history.....	66
Figure 24 <Use Case> Do self-renew a book.....	68
Figure 25 <Use Case> View renew history.....	70
Figure 26 <Use Case> Report lost book.....	72
Figure 27 <Use Case> View library services.....	74
Figure 28 <Use Case Overview> Librarian.....	76
Figure 29 <Use Case> Borrow books for a patron.....	77
Figure 30 <Use Case> Return books for a patron.....	80
Figure 31 <Use Case> Add book.....	83
Figure 32 <Use Case> Update book.....	85
Figure 33 <Use Case> Tag a book copy.....	88
Figure 34 <Use Case> Create book copies.....	90
Figure 35 <Use Case> Update a book copy.....	92
Figure 36 <Use Case> View patron policy.....	95
Figure 37 <Use Case> View borrow policy.....	97
Figure 38 <Use Case> View fee policy.....	99
Figure 39 <Use Case> View patron checkout information.....	101
Figure 40 <Use Case> View renew history.....	104
Figure 41 <Use Case> Renew a book for a patron.....	106
Figure 42 <Use Case> View fee policy history.....	109

Figure 43 <Use Case> View book copies.....	111
Figure 44 <Use Case> View book copy details.....	113
Figure 45 <Use Case> Add books to wishlist for a patron.....	115
Figure 46 <Use Case> View lost book reports.....	117
Figure 47 <Use Case> Confirm lost book report.....	119
Figure 48 <Use Case> Index book copy location.....	121
Figure 49 <Use Case> Find misplaced book copies.....	124
Figure 50 <Use Case> Print book copies barcodes.....	126
Figure 51 <Use Case Overview> Manager.....	129
Figure 52 <Use Case> Add borrow policy.....	130
Figure 53 <Use Case> Update borrow policy.....	132
Figure 54 <Use Case> Remove borrow policy.....	134
Figure 55 <Use Case> Update patron policy.....	136
Figure 56 <Use Case> Update fee policy.....	138
Figure 57 <Use Case> View librarian list.....	140
Figure 58 <Use Case> Add librarian account.....	142
Figure 59 <Use Case> Update librarian information.....	144
Figure 60 <Use Case> Add patron account.....	146
Figure 61 <Use Case> Update patron account.....	148
Figure 62 <Use Case> View security alarm log.....	150
Figure 63 <Use Case> Import patrons.....	152
Figure 64 <Use Case> View book copy type.....	154
Figure 65 <Use Case> Create book copy type.....	156
Figure 66 <Use Case> Update book copy type.....	158
Figure 67 <Use Case> Delete book copy type.....	160
Figure 68 <Use Case> View patron type.....	162
Figure 69 <Use Case> Create patron type.....	163
Figure 70 <Use Case> Update patron type.....	165
Figure 71 <Use Case> Delete patron type.....	167
Figure 72 <Use Case> View patron list.....	169
Figure 73 <Use Case Overview> System Handler.....	171
Figure 74 <Use Case> Send return receipt.....	172
Figure 75 <Use Case> Send borrow receipt.....	174
Figure 76 <Use Case> Remind due date.....	176
Figure 77 <Use Case> Notify wishlisted book's availability.....	178
Figure 78 <Use Case> Send account information.....	180
Figure 79 <Use Case> Send lost book billing.....	182
Figure 80 <Use Case> Send renew receipt.....	184
Figure 81 <Use Case Overview> Security Scanner.....	186
Figure 82 <Use Case> Detect not checked out books.....	186
Figure 83 <Use Case> Log violated cases.....	188
Figure 84 <Screen Flow> Manager Web Application.....	190
Figure 85 <Screen Flow> Librarian Web Application.....	191
Figure 86 <Screen Flow> Patron Web Application.....	192
Figure 87 <Screen Flow> Self Return and Checkout Kiosk.....	193
Figure 88 <Screen Flow> Book Drop Application.....	193
Figure 89 Entity Relationship Diagram.....	199
Figure 90 Architecture Diagram.....	203

Figure 91 <Device Description> UHF RFID Reader.....	204
Figure 92 <Device Description> HF RFID Reader Module.....	205
Figure 93 <Device Description> Stepper Motor.....	206
Figure 94 <Device Description> Arduino UNO Board.....	207
Figure 95 <Device Description> Sparkfun Pro Micro Board.....	208
Figure 96 <Device Description> HF RFID Tags.....	210
Figure 97 <Device Description> UHF RFID Tags.....	211
Figure 98 <Device Description> Passive Buzzer.....	213
Figure 99 <Device Description> LED.....	214
Figure 100 <Device Description> Mini Breadboard.....	215
Figure 101 <Device Description> RS232 Converter.....	216
Figure 102 <Device Description> Stepper Motor Controller Board ULN2003.....	217
Figure 103 Component Diagram.....	218
Figure 104 Class Diagram.....	221
Figure 105 <State Machine Diagram> Book Status.....	231
Figure 106 <State Machine Diagram> Book Copy Status.....	232
Figure 107 <Sequence Diagram> Return Book at Book Drop.....	233
Figure 108 <Sequence Diagram> Librarian return books.....	234
Figure 109 <Sequence Diagram> Librarian checkout book.....	235
Figure 110 <Sequence Diagram> Librarian renew for a patron.....	236
Figure 111 Physical Diagram.....	238

List of Tables

Table 1 Terms and Definitions.....	18
Table 2 System Actors.....	29
Table 3 Use Case List.....	31
Table 4 <Use case> Log in.....	34
Table 5 <Use Case> Return book at kiosk.....	36
Table 6 <Use Case> Return book at book drop.....	39
Table 7 <Use Case> Log out.....	42
Table 8 <Use Case> Search books.....	44
Table 9 <Use Case> View book details.....	46
Table 10 <Use Case> Change password.....	48
Table 11 <Use Case> Add book to search list.....	50
Table 12 <Use Case> Search book by RFID reader.....	52
Table 13 <Use Case> View profile.....	54
Table 14 <Use Case> Update profile.....	56
Table 15 <Use Case> Check in with patron card.....	59
Table 16 <Use Case> Borrow book at kiosk.....	61
Table 17 <Use Case> Add book to wishlist.....	63
Table 18 <Use Case> View wishlisted books.....	65
Table 19 <Use Case> View borrow history.....	68
Table 20 <Use Case> Do self-renew a book.....	70
Table 21 <Use Case> View renew history.....	72
Table 22 <Use Case> Report lost book.....	73

Table 23 <Use Case> View library services.....	75
Table 24 <Use Case> Borrow books for a patron.....	80
Table 25 <Use Case> Return books for a patron.....	82
Table 26 <Use Case> Add book.....	85
Table 27 <Use Case> Update book.....	87
Table 28 <Use Case> Tag a book copy.....	89
Table 29 <Use Case> Create book copies.....	92
Table 30 <Use Case> Update a book copy.....	94
Table 31 <Use Case> View patron policy.....	96
Table 32 <Use Case> View borrow policy.....	99
Table 33 <Use Case> View patron policy.....	100
Table 34 <Use Case> View patron currently borrowed books.....	103
Table 35 <Use Case> View renew history.....	105
Table 36 <Use Case> Renew a book for a patron.....	108
Table 37 <Use Case> View fee policy history.....	110
Table 38 <Use Case> View book copies.....	112
Table 39 <Use Case> View book copy details.....	114
Table 40 <Use Case> Add books to wishlist for a patron.....	117
Table 41 <Use Case> View lost book reports.....	119
Table 42 <Use Case> Confirm lost book report.....	121
Table 43 <Use Case> Index book copy location.....	123
Table 44 <Use Case> Find misplaced book copies.....	126
Table 45 <Use Case> Print book copies barcodes.....	128
Table 46 <Use Case> Add borrow policy.....	132
Table 47 <Use Case> Update borrow policy.....	134
Table 48 <Use Case> Remove borrow policy.....	135
Table 49 <Use Case> Update patron policy.....	137
Table 50 <Use Case> Update fee policy.....	139
Table 51 <Use Case> View librarian list.....	141
Table 52 <Use Case> Add librarian account.....	143
Table 53 <Use Case> Update librarian information.....	145
Table 54 <Use Case> Add patron account.....	148
Table 55 <Use Case> Update patron account.....	150
Table 56 <Use Case> View security alarm log.....	151
Table 57 <Use Case> Import patrons.....	153
Table 58 <Use Case> View book copy type.....	155
Table 59 <Use Case> Create book copy type.....	157
Table 60 <Use Case> Update book copy type.....	159
Table 61 <Use Case> Delete book copy type.....	161
Table 62 <Use Case> View patron type.....	163
Table 63 <Use Case> Add patron type.....	165
Table 64 <Use Case> Update patron type.....	167
Table 65 <Use Case> Delete patron type.....	168
Table 66 <Use Case> View patron list.....	170
Table 67 <Use Case> Send return receipt.....	173
Table 68 <Use Case> Send borrow receipt.....	175
Table 69 <Use Case> Remind due date.....	177
Table 70 <Use Case> Notify wishlisted book's availability.....	179

Table 71 <Use Case> Send account information.....	181
Table 72 <Use Case> Send lost book billing.....	183
Table 73 <Use Case> Send new receipt.....	185
Table 74 <Use Case> Detect not checked out books.....	187
Table 75 <Use Case> Log violated cases.....	189
Table 76 Screen Details.....	195
Table 77 Screen Authorization.....	197
Table 78 Non-Screen Functions.....	198
Table 79 <Device Description> UHF RFID Reader.....	205
Table 80 <Device Description> HF Reader specification.....	206
Table 81 <Device Description> Stepper motor specification.....	207
Table 82 <Device Description> Arduino UNO Board.....	208
Table 83 <Device Description> Sparkfun Pro Micro Board.....	209
Table 84 <Device Description> HF RFID Tags.....	211
Table 85 <Device Description> UHF RFID Tags.....	212
Table 86 <Device Description> Passive Buzzer.....	213
Table 87 <Device Description> LED.....	214
Table 88 <Device Description> Mini Breadboard.....	216
Table 89 <Device Description> RS232 Converter.....	217
Table 90 <Device Description> Stepper Motor Controller Board ULN2003.....	217
Table 91 <Class Diagram Attributes> Author.....	221
Table 92 <Class Diagram Methods> Author.....	221
Table 93 <Class Diagram Attributes> Borrowing.....	221
Table 94 <Class Diagram Methods> Borrowing.....	221
Table 95 <Class Diagram Attributes> Security Gate Log.....	222
Table 96 <Class Diagram Methods> Security Gate.....	222
Table 97 <Class Diagram Attributes> Fee Policy.....	222
Table 98 <Class Diagram Methods> Fee Policy.....	222
Table 99 <Class Diagram Attributes> Book.....	223
Table 100 <Class Diagram Methods> Book.....	223
Table 101 <Class Diagram Attributes> Profile.....	224
Table 102 <Class Diagram Methods> Profile.....	224
Table 103 <Class Diagram Attributes> Role.....	224
Table 104 <Class Diagram Methods> Role.....	224
Table 105 <Class Diagram Attributes> Account.....	225
Table 106 <Class Diagram Methods> Account.....	225
Table 107 <Class Diagram Attributes> Book Borrowing.....	226
Table 108 <Class Diagram Methods> Book Borrowing.....	226
Table 109 <Class Diagram Attributes> Book Copy.....	226
Table 110 <Class Diagram Methods> Book Copy.....	226
Table 111 <Class Diagram Attributes> Book Copy Type.....	227
Table 112 <Class Diagram Methods> Book Copy Type.....	227
Table 113 <Class Diagram Attributes> Book Lost Report.....	227
Table 114 <Class Diagram Methods> Book Lost Report.....	227
Table 115 <Class Diagram Attributes> Borrow Policy.....	228
Table 116 <Class Diagram Methods> Borrow Policy.....	228
Table 117 <Class Diagram Attributes> Extend History.....	228
Table 118 <Class Diagram Methods> Extend History.....	228

Table 119 <Class Diagram Attributes> Book copy position.....	229
Table 120 <Class Diagram Methods> Book copy position.....	229
Table 121 <Class Diagram Attributes> Patron Type.....	229
Table 122 <Class Diagram Methods> Patron Type.....	229
Table 123 <Class Diagram Attributes> Wishlist Book.....	229
Table 124 <Class Diagram Methods> Wishlist Book.....	230
Table 125 <Class Diagram Attributes> Security Deactivated Copy.....	230
Table 126 <Class Diagram Methods> Security Deactivated Copy.....	230
Table 127 <Physical Diagram> author.....	238
Table 128 <Physical Diagram> book_author.....	238
Table 129 <Physical Diagram> genre.....	239
Table 130 <Physical Diagram> book_genre.....	239
Table 131 <Physical Diagram> book.....	240
Table 132 <Physical Diagram> profile.....	240
Table 133 <Physical Diagram> role.....	240
Table 134 <Physical Diagram> account.....	241
Table 135 <Physical Diagram> book_borrowing.....	242
Table 136 <Physical Diagram> book_copy.....	243
Table 137 <Physical Diagram> book_copy_type.....	243
Table 138 <Physical Diagram> book_lost_report.....	244
Table 139 <Physical Diagram> borrow_policy.....	244
Table 140 <Physical Diagram> extend_history.....	245
Table 141 <Physical Diagram> book_copy_position.....	245
Table 142 <Physical Diagram> patron_type.....	245
Table 143 <Physical Diagram> patron_type.....	246
Table 144 <Physical Diagram> security_deactivated_copy.....	246
Table 145 <Physical Diagram> fee_policy.....	247
Table 146 <Physical Diagram> borrowing.....	247

I. Project Introduction

1. Overview

1.1. Project Information

- Project name: RFID Library Management System
- Project code: RLMS
- Group name: GSP21SE18
- Software type: Web application, Windows form

1.2. Project Team

1.2.1. Supervisor

Full name	Email	Phone Number	Title
Nguyễn Thế Hoàng	hoangnt2@fpt.edu.vn	0986628525	Lecturer

1.2.2. Team Members

Full name	Email	Phone Number	Role
Phan Hoàng Trâm	tramphse130038@fpt.edu.vn	0853899403	Leader
Nguyễn Trung Kiên	kienntse130154@fpt.edu.vn	0909317688	Member
Phạm Minh Hoàng	hoangpmse130054@fpt.edu.vn	0938072401	Member
Nguyễn Đỗ Nhật Khang	khangndnse130148@fpt.edu.vn	0857664974	Member

2. Product Background

The library is very important in schools. Libraries provide a quiet place for the students to study by themselves and do homework. All the documents inside the library can also be research materials for students. With many kinds of books available, libraries also encourage students to read more. Libraries are especially important for university students because besides attending lectures most of their time will be spent on self-study. That is why libraries should be available whenever a student needs access to it.

Many school libraries in Viet Nam still depend only on the librarians for borrowing and returning documents. Students and teachers (patrons) normally visit the library during their break times and lunch breaks, thus making those periods busy and patrons need to stand in a queue to borrow books at the librarian counter. Sometimes new classes start before it is their turn at the librarian counter, and they have to come back again later to borrow the documents they wanted. Furthermore, the working hours of school libraries also depend on the working hours of librarians. So librarians have to work early in the morning until late in the evening or patrons will not be able to use the library before or after school hours (which is when they need to do homework and research).

Currently, to checkout or return books, librarians are using barcodes to identify the patron and books. This process is not fast enough to meet the demand for the library's busy hours. And because barcode labels are usually just a piece of paper applied outside of a book it is easy to be scratched and damaged. Damaged labels cannot be scanned and must be input manually by the librarian, making the process taxing for the librarian.

3. Existing Systems

3.1. FPTU Hồ Chí Minh Library System

The current library of FPT University Hồ Chí Minh, to function, relies on 2 separate systems, which are the electromagnetic (EM) system and the barcode system. The EM system - which consists of EM security gates, EM-labels, and an EM-labels activate/deactivate machine - is for security measures. And the barcode system - which consists of barcode labels, barcode labels printer, and barcode readers- is for identifying and managing library documents. Every time new books are added to the library, the librarian has to print and apply a new barcode on the book cover, then an EM label is also applied to a random page of the book.

Currently, to checkout or return books, librarians would have to scan the barcode on the patron's ID card (only when checkout) and then by each book, scan the book barcode and deactivate/activate the electromagnetic label inside the book.

In this system librarians have to keep track of both EM labels and barcode labels. As mentioned earlier, the borrowing and returning process is slow, making it harder for patrons to borrow and return books if the library is constantly busy. And students and lecturers also cannot borrow or return books after school hours because by then the library is already closed.

3.2. FPTU Cần Thơ Library System

The current library of FPT University Cần Thơ has already replaced the EM system and the barcode system with the RFID system, which consists of an RFID reader and RFID tags. All of their documents are tagged with an RFID tag. RFID technology allows the librarians to scan books easily without having to make sure the RFID tag is in line of sight of the reader, unlike the barcode label. Moreover, an RFID tag can also be used to detect documents thievery.

RFID technology allows users to scan multiple tags at the same time instead of scanning one at a time. But the current management software of their library only supports scanning one book at a time. And like FPTU Hồ Chí Minh Library, students and lecturers can only use the library service (like borrow and return books) through the librarians.

3.3. Văn Lang University Library System

Like FPTU Hồ Chí Minh Library, the Văn Lang University library system also uses the barcode system and electromagnetic system. Each book is put on the shelf alphabetically by the book's call number. The book shelves are arranged into sections, shelves in each section will have similar subjects. For example, there are the Social Science section, Computer Science section, Literature section, etc. This helps patrons to look for books that are related to the subjects that they are studying.

But if the patron is searching for a specific book, they still have to spend time looking for it. Because even when the books are sorted by their call number, they are not always put into their right place. So sometimes the patrons still need help from the library staff to get the books for them.

4. Problem Identification

- Patrons always have to depend on librarians to borrow and return books. This makes the library only usable during working hours of librarians. And working hours of librarians can overlap class timetables.

- Librarians can only scan one book at a time, which makes checkout and return of books inefficient.
- Librarians have to remind the patrons to return books manually by phone calls or emails.
- Security system and management system are separated.
- Barcode labels are applied outside of books which makes them vulnerable to damage and become blurred or torn. A damaged barcode label cannot be scanned by the scanner.
- Books are sometimes misplaced from their shelves, making them difficult to find. And keeping track of the book position frequently is taxing for the library staff.

5. Proposed solution

To solve these problems, our team proposes a system called RLMS (RFID Library Management System) that will be integrated with the current library management system. RLMS aims to support the librarians and make the library more accessible for students and teachers (patrons). The system will allow the library to be open even when there are no librarians available. Patrons will be able to borrow and return books directly by themselves. The system will also aid the librarians by making the book scanning process more efficient and remind patrons to return books automatically. Our team has chosen UHF RFID technology because it can help to reduce the time and effort of scanning and identifying books while replacing both, electromagnetic system and barcode system. This technology has already been used by many libraries in the world and it is becoming more and more affordable. Using RFID technology also makes it easier for our system to add new functions in the future.

Such system will provide the following main features:

- Tagging of Books
- Self-checkout of books for patrons
- Self-return of books for patrons
- Book drop box
- Anti-theft protection
- Automatic mailing
- Remind patrons of due date
- Manage book and book copies
- Find book using RFID reader
- Find misplaced books on a shelf using RFID reader

6. Project Scope & Limitations

6.1. Major Features

FE-01: Allow patrons to self-checkout books using the kiosk.

FE-02: Allow patrons to self-return books using the kiosk and the book-drop.

FE-03: Allow patrons receive notifications over email.

FE-04: Allow patrons to renew their borrowing.

FE-05: Allow patrons to search for book information.

FE-06: Allow patrons to add a book to wishlist when it is not available.

- FE-07: Allow manager to manage accounts.
- FE-08: Allow managers to view and manage policies.
- FE-09: Allow librarians and managers to checkout and return books using RFID technology.
- FE-10: Allow librarians and managers to tag a pre-existing book with new RFID technology.
- FE-11: Allow librarians and managers to manage books and book copies.
- FE-12: Allow librarians and managers to renew.
- FE-13: Allow librarians and managers to add a book to wishlist when it is not available.
- FE-14: Allow the library using the system to assure security (RFID technology security gate).

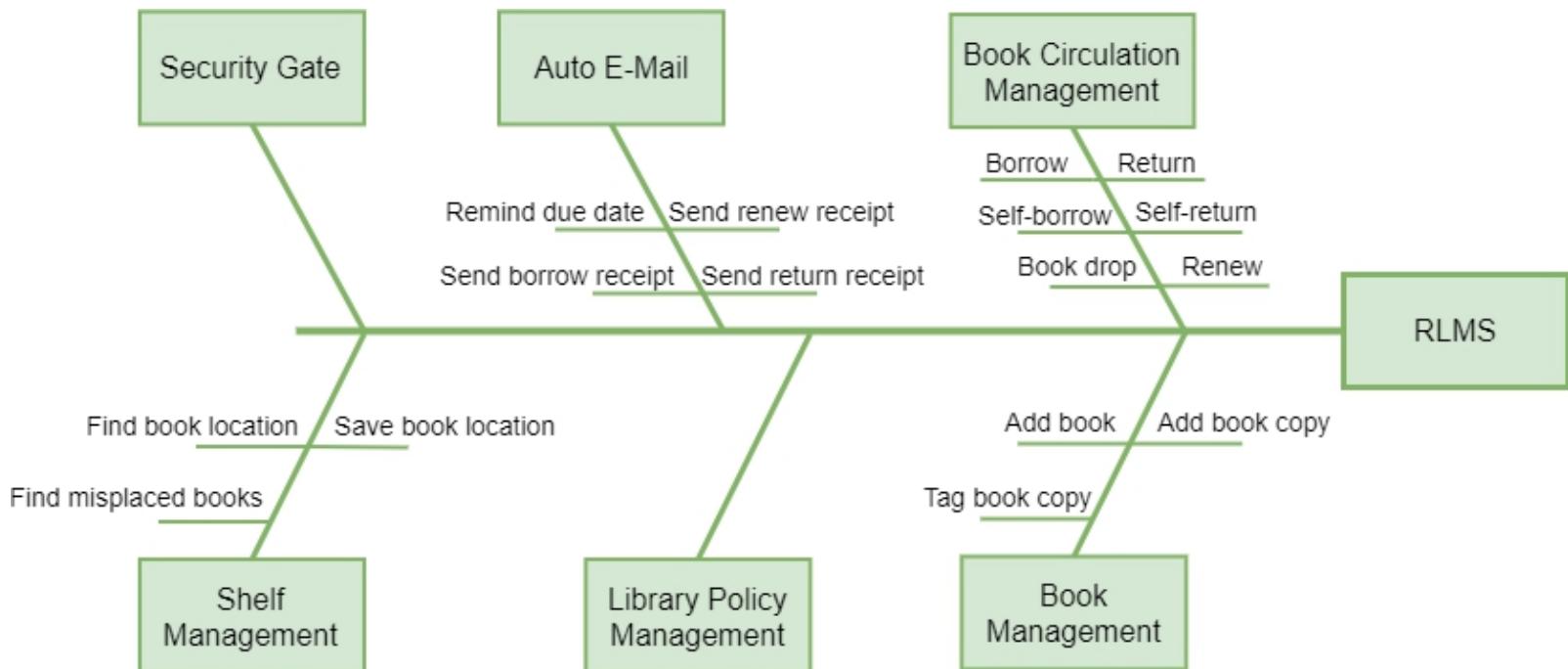


Figure 1 <Fishbone diagram> Major Features

6.2. Limitations & Exclusions

- LI-01: RLMS can only send notifications via Emails. It cannot send push notifications.
- LI-03: RLMS assumes that patrons' id cards use RFID technology.
- LI-04: RLMS assumes that there is a previous working library management system, a legacy system and RLMS only integrates to that system.
- LI-05: RLMS only works with physical book copies.
- LI-06: RLMS assumes that the library systems use LAN to connect to each other.
- LI-07: RLMS assumes that all of the library books are on shelves.
- LI-08: RLMS only manages user accounts in the system.
- LI-09: RLMS is using a customized third party UHF RFID scanner software.

- EX-01: RLMS does not provide communicative features between librarians and patrons.
- EX-02: RLMS does not provide payment features or financial management features.
- EX-03: RLMS does not provide features to check books' physical state such as damages or book sizes.
- EX-04: RLMS does not provide features to manage the bookshelves inside the library.
- EX-05: RLMS does not provide features to manage the library expenses (total lost book price, etc.)
- EX-06: RLMS does not provide features to do accounting or make an inventory.
- EX-07: RLMS does not provide features to check if the book is available on the market or not.
- EX-08: RLMS does not provide features to manage damaged books.
- EX-09: RLMS does not provide features to manage authors.

7. Terms and Definitions

Term	Definition
RLMS	Stands for "RFID Library Management System"
Book copy	Is a physical copy of a book inside the library
Unauthenticated User	Is someone who uses the RLMS without logging in or checking in.
Patron	Is a person who borrows books from the library. There are different types of patrons (for example: students, lecturers, etc.). A patron has a patron account in the RLMS.
Librarian	Is a person who works at the library. A librarian has a librarian account in the RLMS.
Manager	Is a librarian who also manages other librarians and the library. A manager has a manager account in the RLMS.
Borrow Policy	<p>A policy that determines:</p> <ul style="list-style-type: none"> • Which patron type can borrow which book copy type. For example, Students can only borrow Textbooks, but Teachers can borrow Textbooks and Reference books. • How many copies of a copy type can be borrowed by a patron type. For example, a student can borrow 3 reference books but can borrow 10 textbooks. • How many times a book copy can be renewed • How long a book copy can be borrowed by the patron.

	<ul style="list-style-type: none"> • How long a book can be borrowed after it is renewed.
Patron Policy	A policy that determines the maximum books that a patron type can borrow at a time. For example, a student can borrow 20 book copies in total across all book copy types.
Fee Policy	A policy that determines the parameters in the fine calculating formula.
Fine (VND)	The amount of money a patron has to pay when returning overdue book copies. For example if the fine rate is 2 000 VND, and the patron is returning a book copy 3 days late, then the patron will have to pay 6 000 VND for the fine.
Fine rate (VND)	Fine per overdue days. Is determined in Fee Policy
Call number	Is an address for a book. A call number also represents some of the book information like what is the content of the book, etc. Books in the library are arranged in call number order (numerical and alphabetical). The two common methods for classifying call numbers are DDC and LC. For this project, the call number format will be: DDC + Author information + publish year. For example "100 JOH 2007"
DDC	Dewey Decimal Classification System (Is normally used for libraries with a smaller collection size). List of DDC numbers:
LC	Library of Congress Classification System (Is normally used for libraries with a larger collection size)
Borrow Period (days)	The time period that a patron is allowed to keep a borrowed book copy
Renew Period (days)	The time period that will be added to the current due date of a renewing book copy.
Checkouts Allowed (count)	The number of book copies from a specific a patron is allowed to borrow
Renewals Allowed (times)	The number of times a patron is allowed to renew a book copy
Patron Type	Different types of patrons, for example: students, lecturers, etc. Each patron type will follow different policies.

Book Copy Type	Different types of book copies, for example: reference books, textbooks, etc. Each book copy type will have different borrow policies.
Books Tagging	Book tagging is a process where the librarian labels the book with the RFID tag to save and identify that book in our system. Every book in the library has a unique RFID tag.
Barcode	Barcode is used to identify the book in case RFID tag is broken. The barcode in our project has 14 digits: <ul style="list-style-type: none"> • 1st to 2nd digits are Book copy type • 3rd to 6th digits are library's ID • 7th to 15th digits are book copy's ID Every book in the library has a unique barcode.
Security Gate	Security gate is an RFID enabled device, used to detect, warn and log violated cases where not checked out books exit the library.

Table 1 Terms and Definitions

II. Project Management Plan

1. Overview

1.1. WBS & Estimation

#	WBS Item	Complexity	Est. Effort (man-days)
I	Analysis	Complex	21
II	Documentation	Medium	50
III	Design	Complex	20
IV	Prototyping	Medium	15
V	Implementation		200
1	Self-Services Application in Library		15
1.1	Self-checkout at kiosk	Complex	10
1.2	Self-return at kiosk	Medium	5
2	Book Drop Application		20
2.1	Book Drop Model	Complex	10
2.2	Self-return at book drop	Complex	10
3	Patron Web Application		35
3.1	View profile	Simple	1
3.2	Update profile	Simple	1
3.3	View borrowing history	Medium	5
3.4	Renew book	Simple	2
3.5	Search books	Complex	10
3.6	Search book by UHF reader	Complex	10
3.7	View book details	Simple	2
3.8	Add to wishlist	Simple	2
3.9	View policies	Simple	2
4	Librarian Web Application		93
4.1	Manage Book	Complex	20
4.2	Manage Book Copy	Complex	20
4.3	Tag book	Simple	2
4.4	Check out books	Medium	8
4.5	Return books	Medium	8
4.6	Manage patron's borrowing	Complex	15
4.7	Manage patron's account	Medium	5
4.8	Manage lost books	Simple	3

4.9	Find misplace/missing copies	Medium	5
4.11	Add position for book copies	Medium	5
4.12	View policies	Simple	2
5	Manager Web Application		21
5.1	Manage librarian's account	Medium	5
5.2	Manage patron's account	Medium	5
5.3	Manage book copy type	Simple	3
5.4	Manager patron type	Simple	3
5.5	Manage policies	Medium	5
6	System Handler		4
6.1	Send emails	Simple	4
7	Security Gate		12
7.1	Detect theft	Complex	10
7.2	Save log	Simple	2
VI	Testing	Simple	10

Total Estimated Effort (man-days) **320**

1.2. Project Objectives

- All major requirements and business rules are confirmed, and team members are trained before the 3rd week.
- By the end of the 10th week, major functions must be completed and ready for testing.
- By the end of the 14th week, the team must complete fixing bugs for major functions and complete other functions.
- By the end of the 15th week, documentation and testing must be completed.
- From the 16th week, review documentation and get ready for thesis defense.

1.3. Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	Team member drops out	Critical	Medium	Reduce scope
2	Team member might be quarantined due to COVID	Medium	Medium	Meeting online
3	UHF reader gets damage	Critical	Low	Purchase another one but still take time due to international shipping

2. Management Approach

2.1. Project Process

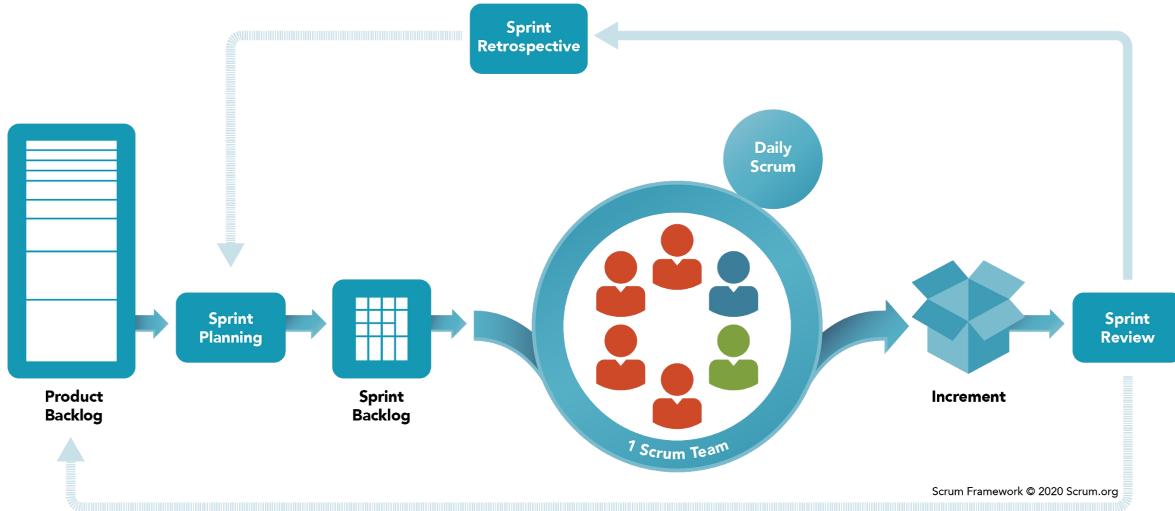


Figure 2 <Reference> Scrum Framework Scrum.org. "What is Scrum". Retrieved 2021, from <https://www.scrum.org/resources/what-is-scrum>

Our team uses Scrum Framework for developing the RLMS project as following reasons:

- Because each library has their own management system, our team is not able to determine the best approach for the project immediately. With Scrum the requirements can be updated overtime to better match our vision flexibly through each sprint.
- Because of some COVID-19 cases in Hồ Chí Minh city at the beginning of our project, it is difficult for our team to go to other libraries to interview the librarians and study their working process. So the requirement was unclear. With Scrum, we can show the increments and get early feedbacks from the stakeholders in each sprint so we can make the requirements clearer.
- Our team has to do research and experiment on different hardware and devices for the project, so we need to have the increments available as soon as possible for us to evaluate if the hardware works well with our system or not.
- Our team members have different working hours because some of them work part-time. With Scrum we can still communicate daily to reduce development conflicts and problems through the flexible daily meeting schedules and sprint reviews.

Our project will last 5 months and will be divided into 18 one-week-sprints.

2.2. Quality Management

In order to improve the project quality, our team will apply the following methods:

- Asking FPTU library's manager for any concerned requirement.
- Review each increment weekly in sprint reviews strictly.

- Our team defined and follow the team coding conventions.
- The source code is stored and managed on GitHub. Code is merged and resolve conflicts regularly.
- Tasks are managed assigned in Trello.

2.3. Training Plan

Training Area	Participants	When, Duration	Waiver Criteria
Java Spring Boot	Everyone	04-01-2021, 1 week	Mandatory
React	Everyone	04-01-2021, 1 week	Mandatory
Arduino	Everyone	04-01-2021, 1 week	Mandatory
RFID	Everyone	04-01-2021, 1 week	Mandatory
Git	Everyone	04-01-2021, 1 week	Mandatory

3. Project Organization

3.1. Team & Structures

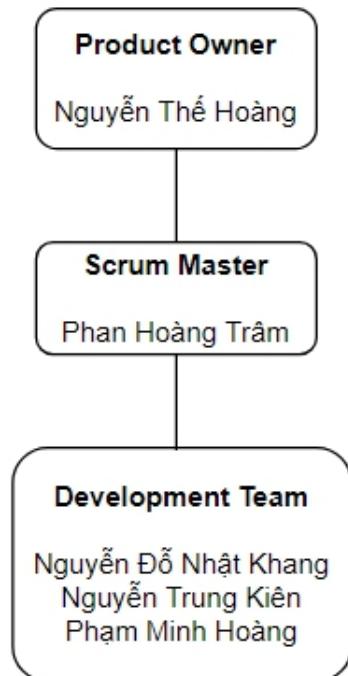


Figure 3 Team & Structures

3.2. Roles & Responsibilities

Role	Responsibility
Product Owner	<ul style="list-style-type: none"> Defining the vision. Managing the product backlog. Prioritizing needs. Overseeing development stages. Anticipating business needs. Acting as primary liaison. Evaluating product progress at each iteration.
Scrum Master	<ul style="list-style-type: none"> Facilitate daily scrum, sprint planning, sprint demo and retrospective meetings. Help product owners to make the product backlogs in good shape and make them ready for the next sprint. Responsible for conducting retrospective meetings. Act as safeguard for the team. Facilitate Scrum events as requested or needed. Help the development team to create high-value products.
Development Team Member	<ul style="list-style-type: none"> Select tasks from the Product Backlog Team ordered by the Product Owner. Plan their work on a daily basis within the time-boxed Sprint to optimize the team's output. Deliver a potentially releasable increment (integrated with increments of other teams)

4. Project Communication

4.1. Communication Plan

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Computer	Team members	Internal Communication, Review task progress	Daily	Skype
In person	Team members	Review sprint progress	Weekly	Offline

Computer	Team members & Mr. Nguyễn Thế Hoàng	Get everyone on the same page	Weekly	Google Meet
In person	Team members & Mr. Nguyễn Thế Hoàng	Review project business and document	Weekly	Offline, FPTU Campus

4.2. External Interface

Function	Contact Person (name, position)	Contact address (email, telephone)	Responsibility
Supervisor	Nguyễn Thế Hoàng	hoangnt2@fpt.edu.vn 0986628525	- Provide document template - Give instruction to project team - Review deliverables - Supervise project status
Supporter	Đinh Trường Lâm, Head of FPT HCM Library	lamdt@fpt.edu.vn	- Provide end user's perspective (librarian)
Supporter	Lê Quang Tường, Head of FPT Cần Thơ Library	tuonglq@fe.edu.vn	- Provide end user's perspective (librarian)

5. Configuration Management

5.1. Tools & Infrastructures

Programming languages	JavaScript, Java, C++, C#
Framework/Library	ReactJS, Spring Boot
DBMS	MySQL
IDEs/Editors	Visual Studio, Visual Studio Code, IntelliJ, Arduino
UML tools	StarUML, Draw.io
Version control system	GitHub
Deployment server	Local server
Project management tool	Trello, Google Sheets

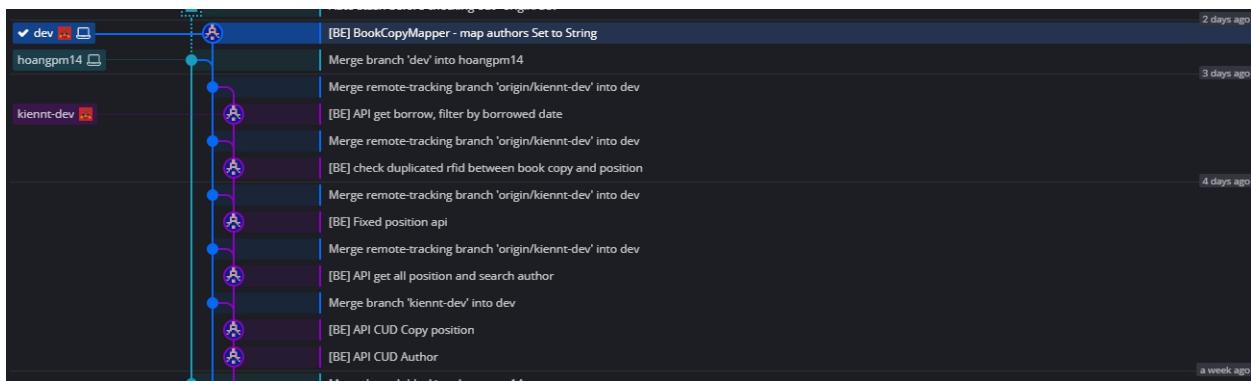
5.2. Document Management

The project documents will be managed using Google Drive.

Link: <https://docs.google.com/document/d/1rKU7j-jYZZmVvS7KepWCMCwPJEghZpazDXosoJcunOk/edit?pli=1#heading=h.w7lieshee0ie>

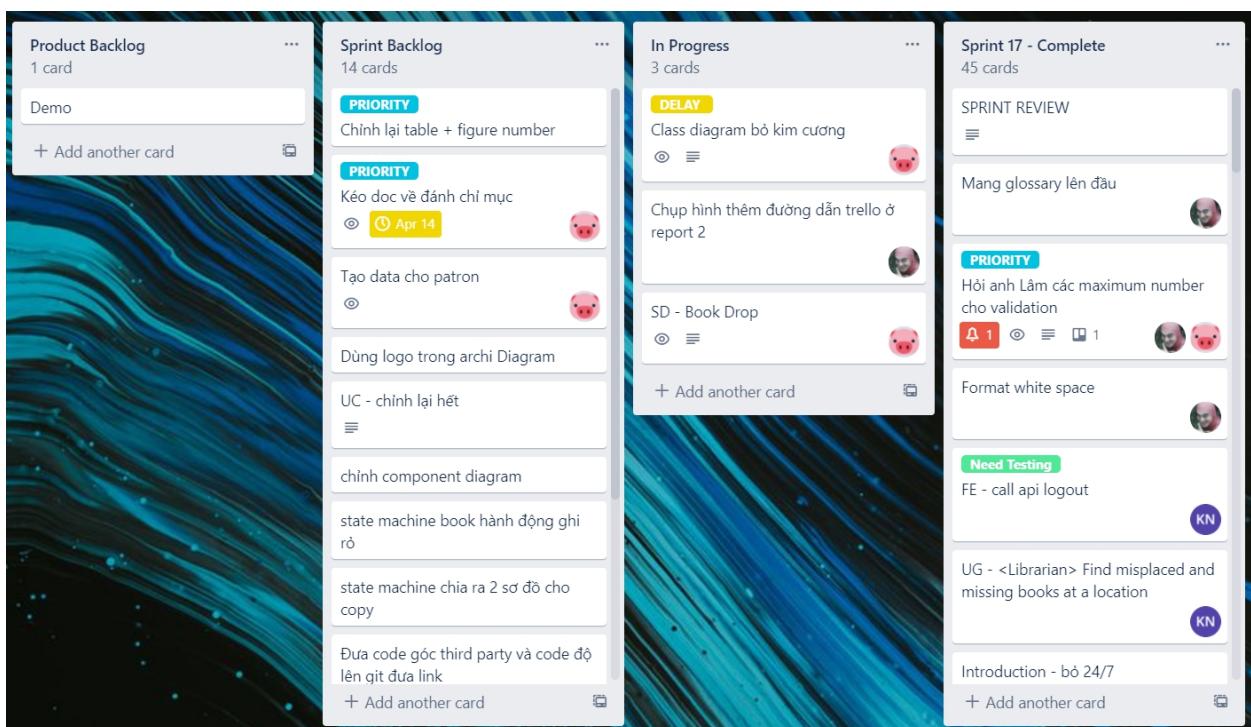
5.3. Source Code Management

The project source code will be managed using GitHub.



5.4. Project Tasks Management

Our team will follow SCRUM methodology and manage our tasks in Trello (<https://trello.com/b/Uha7qXGW/capstone-agile-sprint-board>)



Some of our early tasks (from sprint 1 to sprint 7) were managed in Google Spreadsheet (https://docs.google.com/spreadsheets/d/1TeyZk-06MU6EGz9Zln25RwDXYVGcFdLQ1_ZkI1PbBlk/edit?usp=sharing)

III. Software Requirement Specification

1. Overall Description

1.1. User Requirement Overview

1.1.1. Unauthenticated User Requirements

Unauthenticated Users are people who have not been identified by the system. They can use the following features:

- Login
- Check in
- Return at the kiosk inside library
- Return at the book drop outside library

1.1.2. Authenticated User Requirements

Authenticated Users are people who have identified themselves by the system. They can use the following feature:

- Logout
- Search books
- View book details
- Search book in the library by UHF reader

1.1.3. Patron Requirements

Patrons are authenticated users with a “Patron” role. They can use the following features:

- Check out books at the kiosk inside library
- View checkout information: returned books, overdue books, borrowing books
- Renew a holding book
- View renew history of a borrowed book
- Add an available book to wishlist
- View and update profile
- Receive receipt after checkout, return via email
- Receive notification if a book in the wishlist is available via email
- Receive account information to log into the system via email
- View policies

1.1.4. Librarian Requirements

Librarians are authenticated users with a “Librarian” role. They can use the following features:

- Check out books for patron
- Return books for patron

- Manage books
- Manage book copies
- View a patron's checkout information
- Renew a book for patron
- View renew history
- Add an available book to patron's wishlist
- Report lost book
- Manager patron account (excludes add, update)
- Check misplaced and missing books
- Print multiple barcodes
- View policies

1.1.5. Manager Requirements

Librarians are authenticated users with a “Librarian” role. They can use all the features that librarian can do and the following features:

- Manage patron account
- Manage librarian account
- Manage policies
- Manage patron type
- Manage book copy type

2. User Requirements Specification

2.1. Use Case

2.1.1. Use Case Diagram

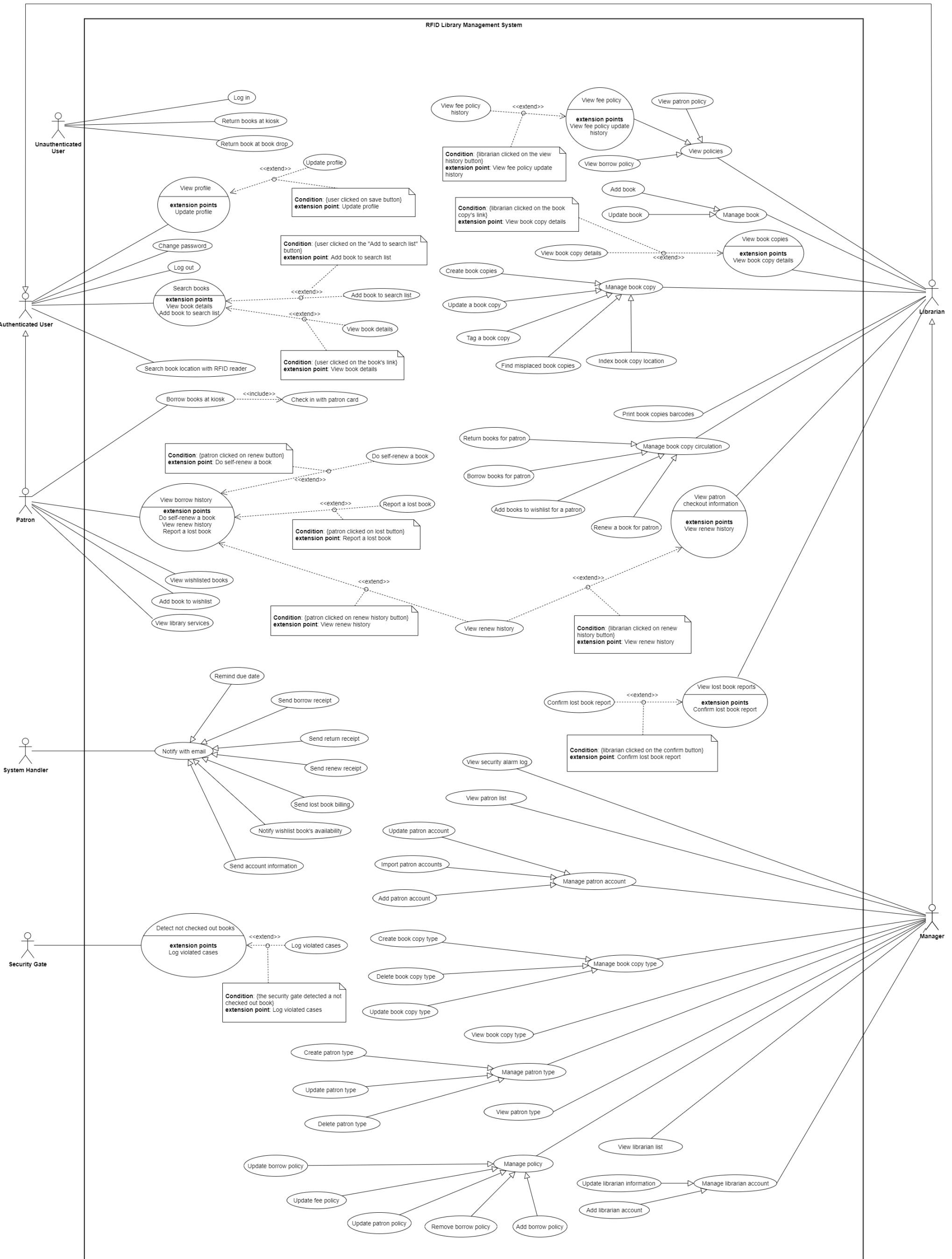


Figure 4 <Use Case Overview> RLMS Use Case Diagram

2.1.2. System Actors

#	Actor	Description
1	Unauthenticated User	Users have not logged into RLMS
2	Authenticated User	Users have logged into RLMS
3	System Handler	RLMS system
4	Patron	Users have logged into RLMS as patron role
5	Librarian	Users have logged into RLMS as librarian role
6	Manager	Users have logged into RLMS as manager role
7	Security Gate	Security gate in library

Table 2 System Actors

2.1.3. Use Case List

ID	Use Case	Primary Actor
01	Log in	Unauthenticated User
02	Return book at kiosk	Unauthenticated User
03	Return book at book drop	Unauthenticated User
04	Log out	Authenticated User
05	Search books	Authenticated User
06	View book details	Authenticated User
07	Change password	Authenticated User
08	Add book to search list	Authenticated User
09	Search book location by RFID reader	Authenticated User
10	View profile	Authenticated User
11	Update profile	Authenticated User
12	Check in with patron card	Patron
13	Borrow books at kiosk	Patron
14	Add book to wishlist	Patron
15	View wishlisted books	Patron
16	View borrow history	Patron

17	Do self-renew a book	Patron
18	View renew history	Patron, Librarian
19	Report lost book	Patron
20	View library services	Patron
21	Borrow books for patron	Librarian
22	Return books for patron	Librarian
23	Add book	Librarian
24	Update book	Librarian
25	Tag a book copy	Librarian
26	Create book copies	Librarian
27	Update a book copy	Librarian
28	View patron policy	Librarian
29	View borrow policy	Librarian
30	View fee policy	Librarian
31	View patron checkout information	Librarian
32	Renew a book for patron	Librarian
33	View fee policy history	Librarian
34	View book copies	Librarian
35	View book copy details	Librarian
36	Add books to wishlist for a patron	Librarian
37	View lost book reports	Librarian
38	Confirm lost book report	Librarian
39	Index book copy location	Librarian
40	Find misplaced book copies	Librarian
41	Print book copies barcodes	Librarian
42	Add borrow policy	Manager
43	Update borrow policy	Manager
44	Remove borrow policy	Manager
45	Update patron policy	Manager
46	Update fee policy	Manager

47	View librarian list	Manager
48	Add librarian account	Manager
49	Update librarian information	Manager
50	Add patron account	Manager
51	Update patron account	Manager
52	View security alarm log	Manager
53	Import patron accounts	Manager
54	View book copy type	Manager
55	Create book copy type	Manager
56	Update book copy type	Manager
57	Delete book copy type	Manager
58	View patron type	Manager
59	Create patron type	Manager
60	Update patron type	Manager
61	Delete patron type	Manager
62	View patron list	Manager
63	Send return receipt	System Handler
64	Send borrow receipt	System Handler
65	Remind due date	System Handler
66	Notify wishlist book's availability	System Handler
67	Send account information	System Handler
68	Send lost book billing	System Handler
69	Send renew receipt	System Handler
70	Detect not checked out books	Security Gate
71	Log violated cases	Security Gate

Table 3 Use Case List

2.2. Use Case Specifications

2.2.1. Unauthenticated User

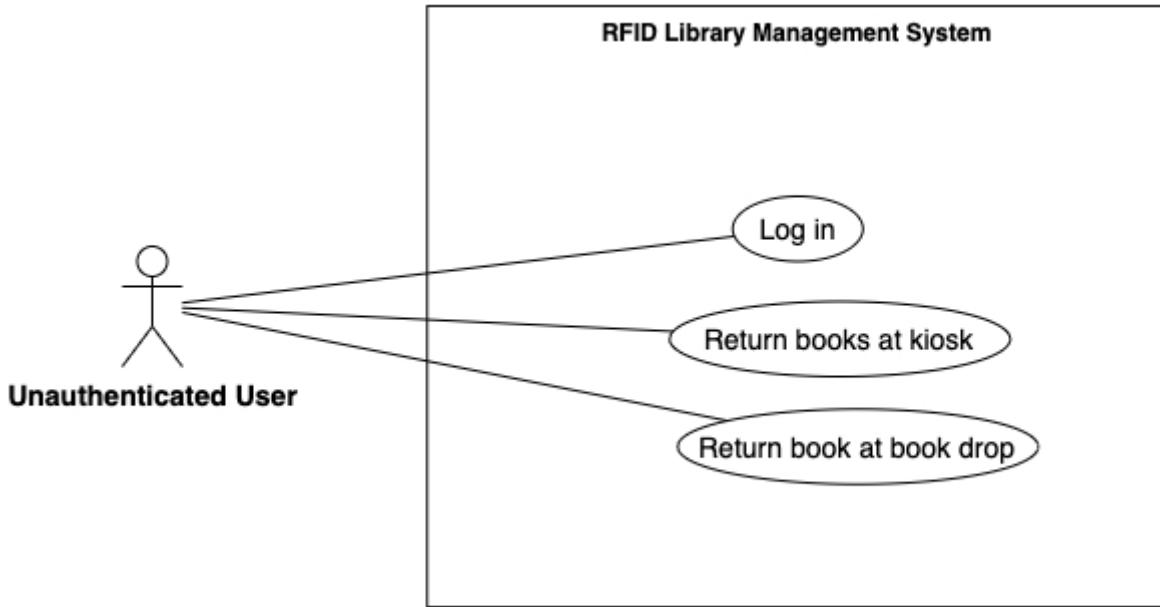


Figure 5 <Use Case Overview> Unauthenticated User

2.2.1.1. Log in

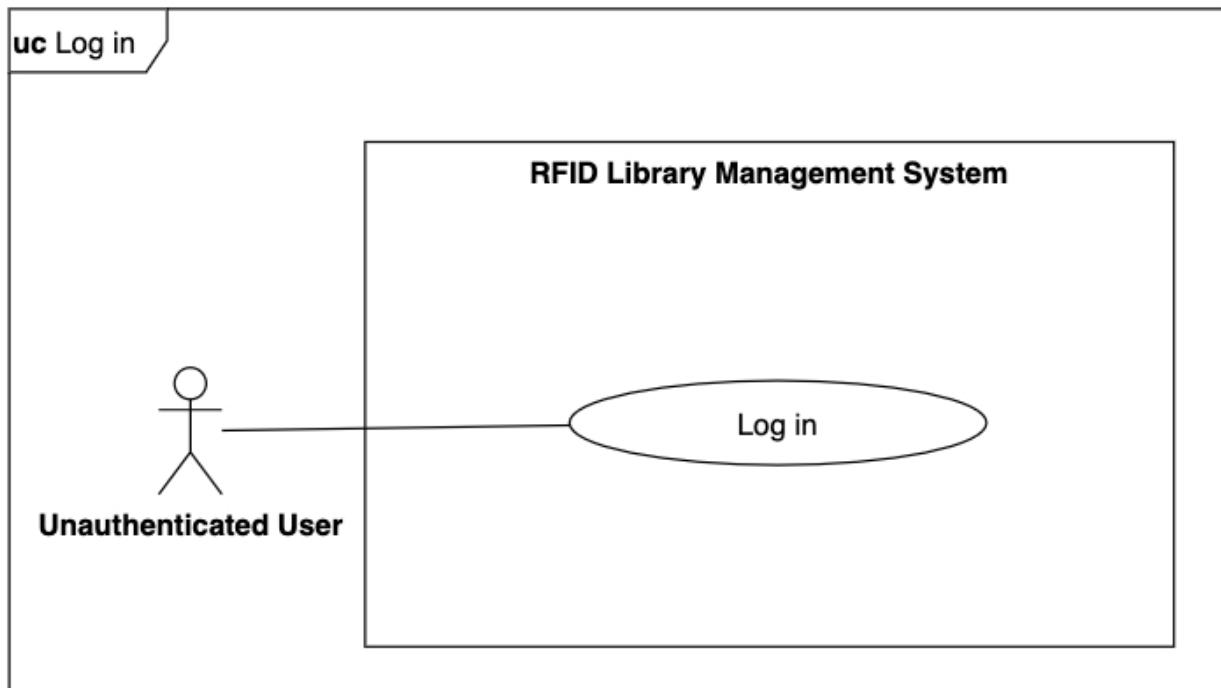


Figure 6 <Use case> Log in

ID and Name:	UC_01 Log in		
Created By:	HoangPM	Date Created:	13/01/2021
Primary Actor:	Unauthenticated user	Secondary Actors:	N/A
Description:	Authenticate and authorize users, to give them access to the system.		
Trigger:	Unauthorized user sends request to login		
Preconditions:	PRE-1: User is not yet authenticated		
Postconditions:	POST-1: If login successfully, the user is authenticated and authorized to use corresponding features in the systems. POST-2: If login unsuccessfully, The user is notified with an error message.		
Normal Flow:	Step	Actor action	System response
	1	The user inputs username and password, then clicks the “Sign in” button.	The system validates user’s login information. Then the system opens a session for the user and redirects the user to the web site’s home page. [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user inputs wrong username or password	RLMS shows an error message.
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. Users must use the university/school email. 2. Patrons and librarians accounts are created by the manager. Passwords are auto generated and sent to the patrons and librarians by their emails. 3. Each login session lasts 1 days. After the session has ended, the user is automatically logged out. 		

Other Information	N/A
Assumptions:	<ol style="list-style-type: none"> 1. The manager's account is in the system by default. 2. All users have already received their accounts.

Table 4 <Use case> Log in

2.2.1.2. Return book at kiosk

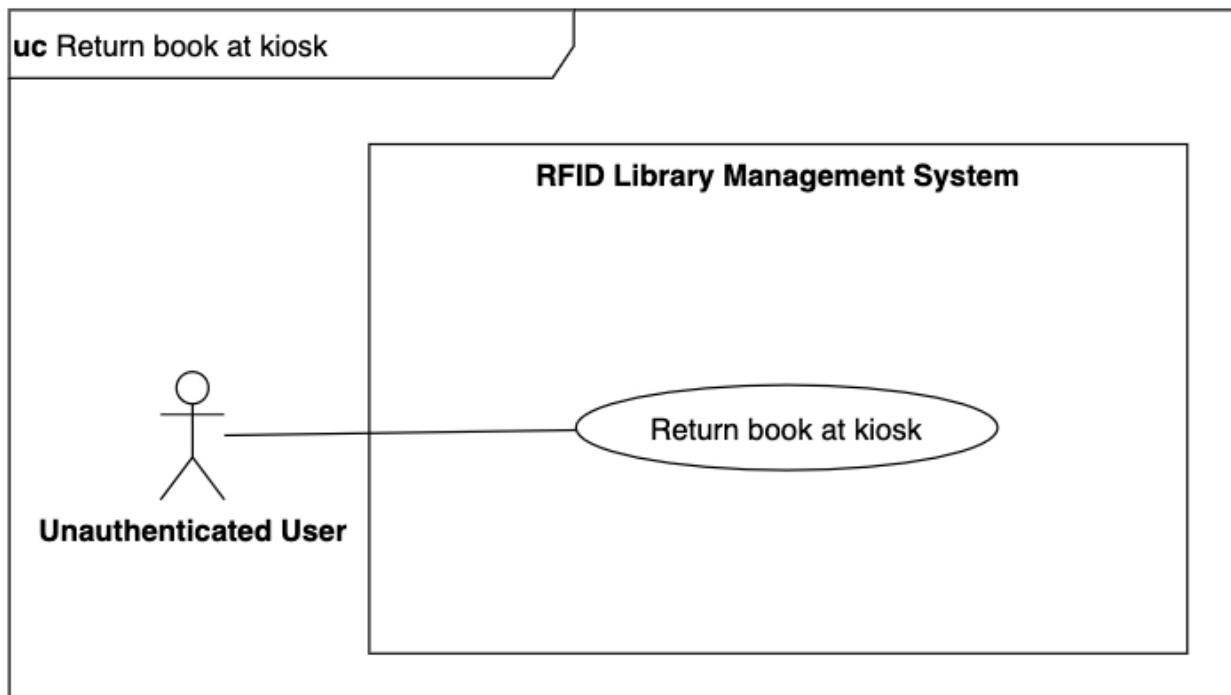


Figure 7 <Use Case> Return book at kiosk

ID and Name:	UC_02 Return book at kiosk		
Created By:	TramPH	Date Created:	13/01/2021
Primary Actor:	Unauthenticated user	Secondary Actor:	N/A
Description:	Let the users return borrowed books by themselves inside the library at self-service kiosks.		
Trigger:	Unauthenticated users request to return books.		

Preconditions:	PRE-1: Users have borrowed books.		
Postconditions:	POST-1: The system shows a list of returned books. POST-2: If return failed, show error message.		
Normal Flow:	Step	Actor action	System response
	1	The users click the "Return" button.	RLMS redirects to the return page and the session counts down.
	2	The users put borrowed books on the UHF scanner.	RLMS returns the list of scanned books. [Exception 1]
	3	The users click the "Confirm" button. [Alternative 1] [Alternative 2]	RLMS return the list of returned books also books cannot return with reason.
Alternative Flows:	Step	Actor action	System response
	1.1	The users use the return service for up to 4 minutes.	Session is expired and the RLMS closes the return page.
	2.1	The users click the "Cancel" button.	RLMS show cancel confirm dialog.
	2.2	The users click the "Yes" button.	RLMS closes the return screen.
Exceptions:	No.	Cause	System response
	1	RLMS cannot find the scanned book in the system.	RLMS shows error messages and asks the user to scan books again.
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	1. The users cannot return overdue books at the kiosks, they must return at the librarian counter.		

	<ol style="list-style-type: none"> 2. The users can return multiple copies at the same time. 3. After scanning books, the system shows: <ul style="list-style-type: none"> • Book information: image, title, subtitle, edition, authors, group 4. After returning books, the system shows: <ul style="list-style-type: none"> • Book information: image, title, subtitle, edition, authors, group • Return information: borrower, overdue day(s), fine (VND), returned time (if returned) or message (if couldn't return) 5. Overdue days excludes Saturdays and Sundays 6. Fine is calculated based on fee policy at the time users borrowed those books <ul style="list-style-type: none"> • Fine = overdue days * fine rate • Max fine = book copy's price * max overdue fine percentage / 100 7. Book copy's status will be updated depends on Book's status: <ul style="list-style-type: none"> • If Book's status is "IN_CIRCULATION", book copy's status is "AVAILABLE" • If Book's status is "OUT_OF_CIRCULATION", book copy's status is "OUT_OF_CIRCULATION" • If Book's status is "LIB_USE_ONLY", book copy's status is "LIB_USE_ONLY" 8. Each session lasts 4 minutes by default, after that the system will close the return process. That makes sure the users do not use the kiosk for a long time.
Other Information	<ol style="list-style-type: none"> 1. The users shall be able the cancel book-returned process. 2. Session shall be paused when the system's making a request to the service. 3. If there is no internet connection, the system cannot load the image of book
Assumptions:	N/A

Table 5 <Use Case> Return book at kiosk

2.2.1.3. Return book at book drop

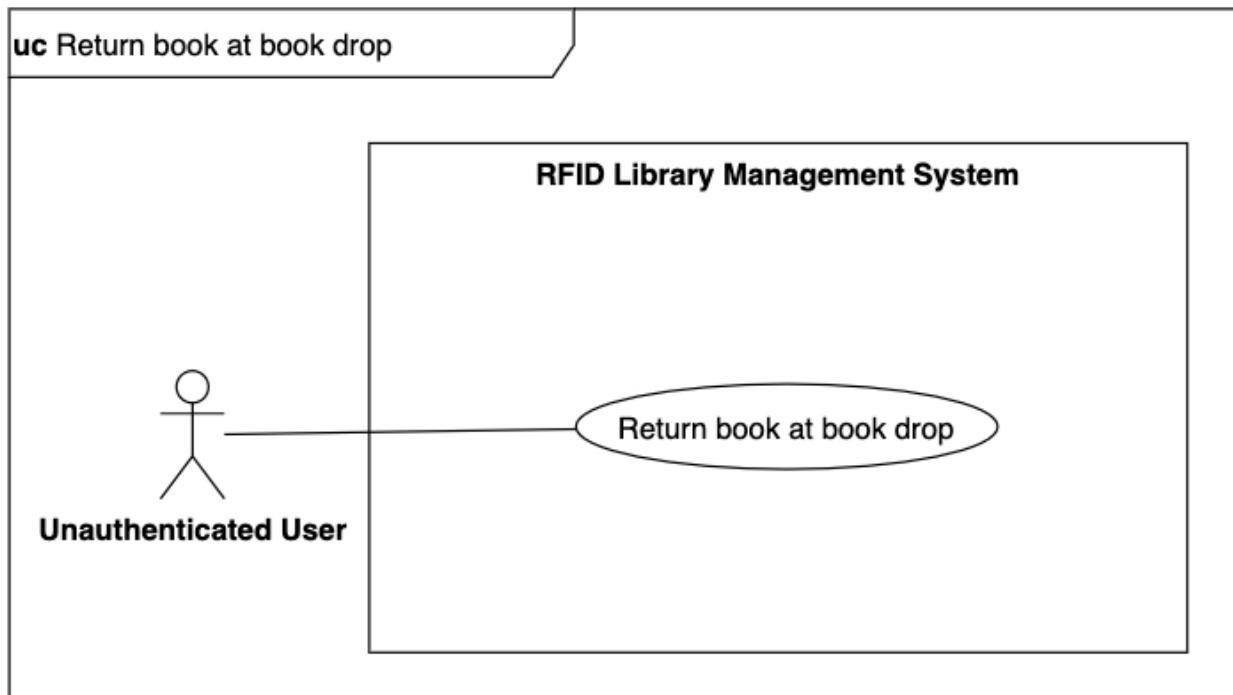


Figure 8 <Use Case> Return book at book drop

ID and Name:	UC_03 Return book at book drop		
Created By:	TramPH	Date Created:	13/01/2021
Primary Actor:	Unauthenticated user	Secondary Actors:	N/A
Description:	Let users return borrowing books by themselves outside the library.		
Trigger:	Unauthenticated users request to return books.		
Preconditions:	PRE-1: Users have borrowed books.		
Postconditions:	POST-1: The system shows the returned book successfully. POST-2: If return failed, show error message.		
Normal Flow:	Step	Actor action	System response
	1	The users click the "Start" button.	RLMS shows session counting down. Front door opens and asks the user to put a book into the box.

	2	The users insert a book into the box.	The front door closes and the system starts scanning the book.
	3		RLMS opens the back door, lets the book drop into library storage, then closes the back door. The system showed the return successfully. [Exception 1] [Exception 2] [Exception 3] [Exception 4] [Exception 5]
Alternative Flows:	Step	Actor action	System response
	1.1	The users use the return service for up to 4 minutes.	Session is expired and the RLMS reset the return page.
Exceptions:	No.	Cause	System response
	1	The users do not insert any book.	RLMS shows error messages and reset the return process.
	2	The users insert more than 1 book.	RLMS shows error messages and opens the front door after 5s then closes the front door.
	3	The users insert the overdue book.	RLMS shows error messages and opens the front door after 5s then closes the front door.
	4	The users insert a book that cannot be found in the system.	RLMS shows error messages and opens the front door after 5s then closes the front door.
	5	The users insert a book that has not been borrowed.	RLMS shows error messages and opens the front door after 5s then closes the front door.
Priority:	High		
Frequency of Use:	Usually		

Business Rules:	<ol style="list-style-type: none"> 1. The users cannot return overdue books at the kiosks, they must return at the librarian counter. 2. The users can only return 1 copy at the time. 3. After returning books, the system shows: <ul style="list-style-type: none"> • Book information: image, title, subtitle, edition, authors, group • Return information: borrower, overdue day(s), fine (VND), returned time (if returned) or message (if couldn't return) 4. Overdue days excludes Saturday's and Sundays 5. Fine is calculated based on fee policy at the time users borrowed those books <ul style="list-style-type: none"> • Fine = overdue days * fine rate • Max fine = book copy's price * max overdue fine percentage / 100 6. Book copy's status will be updated depends on Book's status: <ul style="list-style-type: none"> • If Book's status is "IN_CIRCULATION", book copy's status is "AVAILABLE" • If Book's status is "OUT_OF_CIRCULATION", book copy's status is "OUT_OF_CIRCULATION" • If Book's status is "LIB_USE_ONLY", book copy's status is "LIB_USE_ONLY" 7. Each session lasts 4 minutes by default, after that the system will reset the return process. That makes sure the users do not use the book drop for a long time. 8. Back door only opens when returning successfully.
Other Information	<ol style="list-style-type: none"> 1. Session shall be paused when the system's making a request to the service. 2. Once the door is closing, the users cannot interrupt the closing process. 3. If there is no internet connection, the system cannot load the image of book
Assumptions:	<ol style="list-style-type: none"> 1. The book drop can only contain lightweight and medium sized books.

Table 6 <Use Case> Return book at book drop

2.2.2. Authenticated user

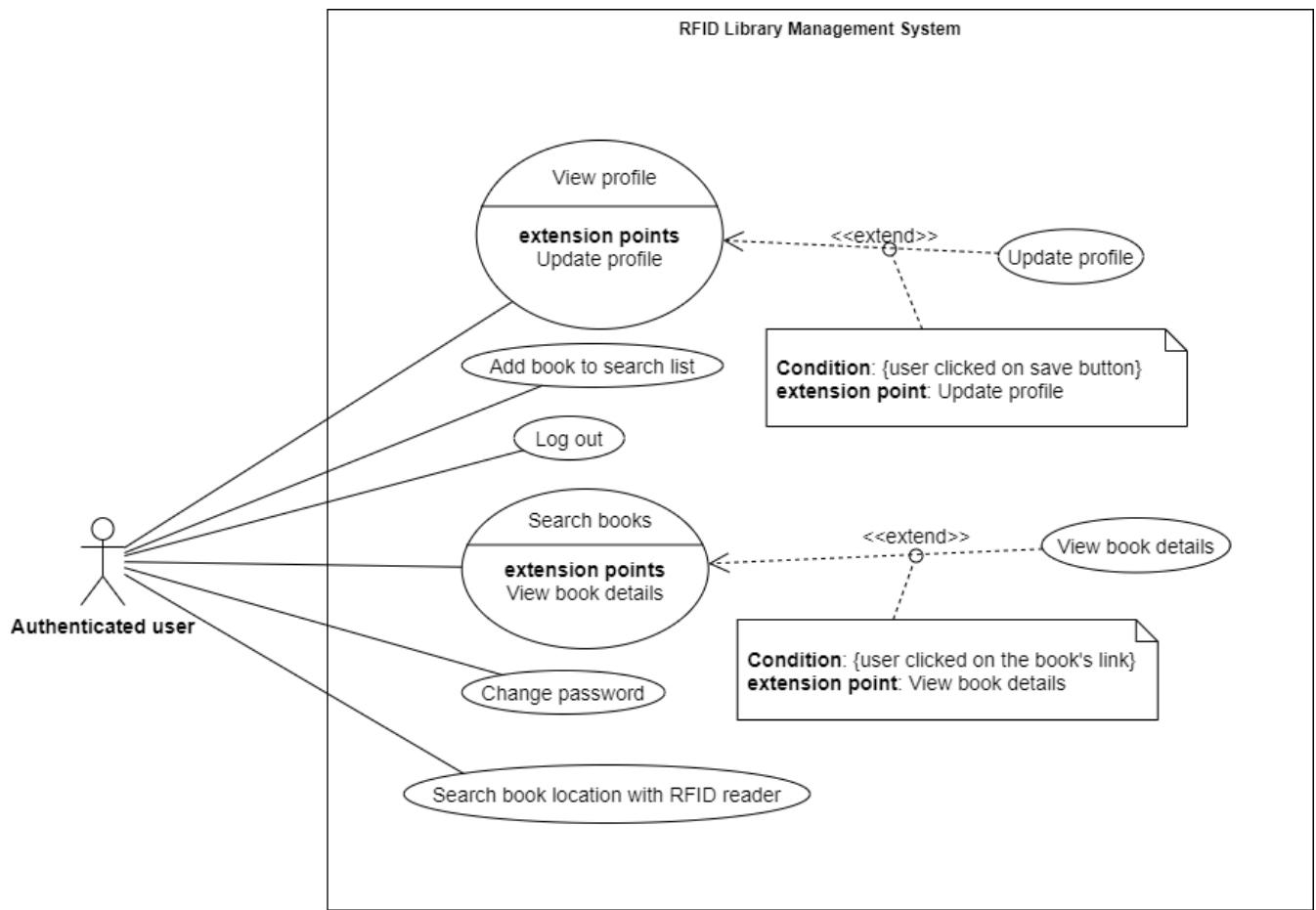


Figure 9 <Use Case Overview> Authenticated User

2.2.2.1. Log out

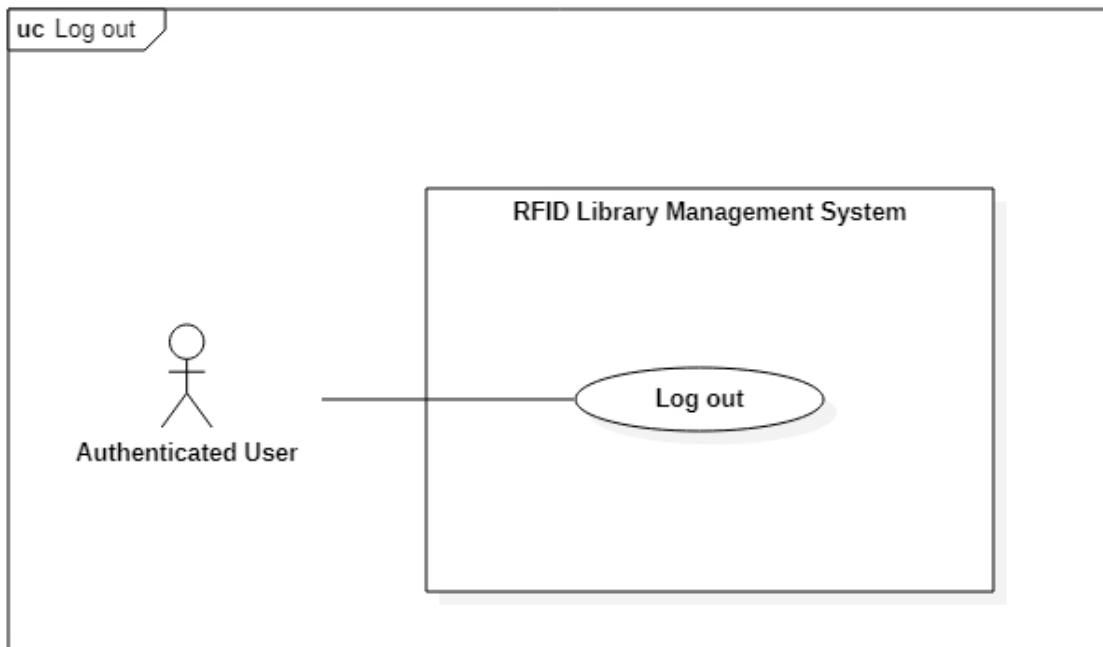


Figure 10 <Use Case> Log out

ID and Name:	UC_04 Logout		
Created By:	HoangPM	Date Created:	13/01/2021
Primary Actor:	Authenticated user	Secondary Actors:	N/A
Description:	Enable authenticated users to log out of the system on the current machine.		
Trigger:	Authorized user sends a request to logout of the system.		
Preconditions:	PRE-1: User has already logged into the system.		
Postconditions:	POST-1: User is redirected to login page. POST-2: The system terminates the login session		
Normal Flow:	Step	Actor action	System response
	1	The user clicks on the “Logout” tab. [Alternative 1]	The system removes the user’s JWT token and redirects the user to the login page.

Alternative Flows:	Step	Actor action	System response
	1.1	The user logged into the system until the session expires.	The system removes the user's JWT token and redirects the user to the login page.
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> After having logged out, an authenticated user becomes an unauthenticated user Authorization cookies on the client side will be deleted after logout 		
Other Information	N/A		
Assumptions:	N/A		

Table 7 <Use Case> Log out

2.2.2.2. Search books

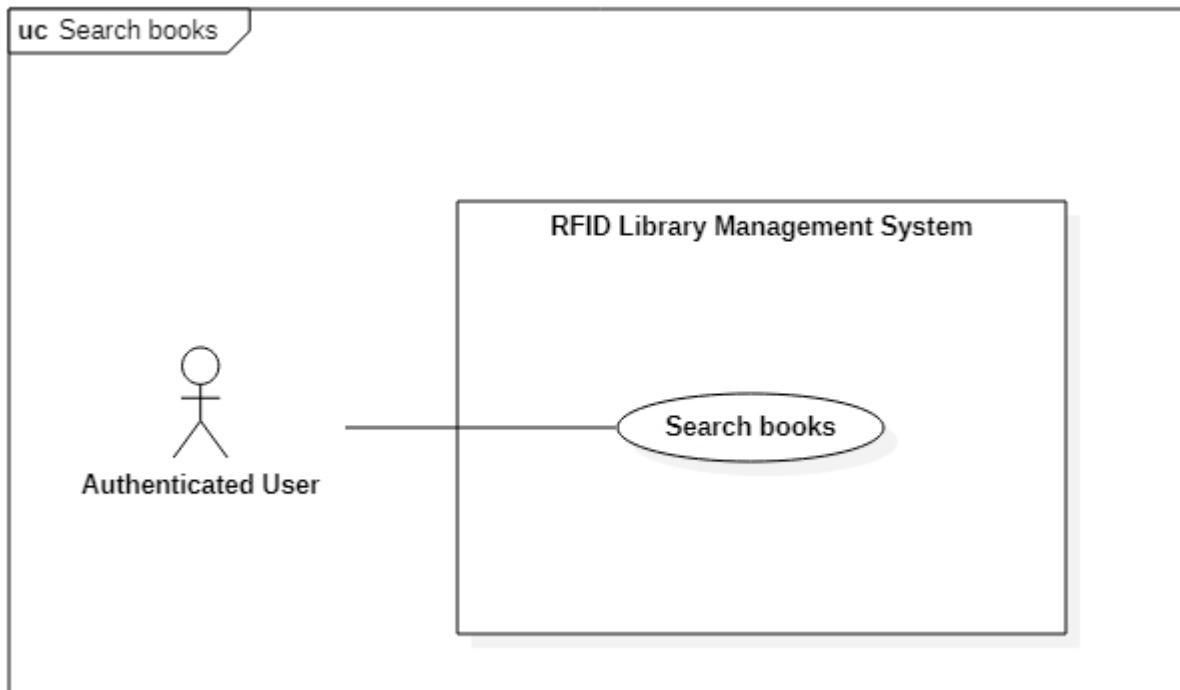


Figure 11 <Use Case> Login with ID card

ID and Name:	UC_05 Search books		
Created By:	HoangPM	Date Created:	13/01/2021
Primary Actor:	Authenticated user	Secondary Actors:	N/A
Description:	Users can search for book information using ISBN, book title, author, etc.		
Trigger:	The user requests to search books by ISBN or book title.		
Preconditions:	PRE-1: User has already logged into the system.		
Postconditions:	POST-1: Show a list of book based on the user's search POST-2: If there is no result, show message		
Normal Flow:	Step	Actor action	System response
	1	The user enters ISBN or book title into the search textbox and presses search. [Exception 1]	The system searches for books based on the user's condition and displays the results to screen.
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Medium		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> Patrons can view books that are in circulation or can be read in the library only. Librarians can view every books in the system The result list should show which books are available to borrow If a book is not available, there should be a "wishlist" button next to that book Only authorized users can add book to wishlist Books can be searched by: <ul style="list-style-type: none"> Like title 		

	<ul style="list-style-type: none"> ● Like subtitle ● Equals ISBN <p>6. If the authenticated role is Patron, then each items in the search result will contains:</p> <ul style="list-style-type: none"> ● Book image ● Title ● Subtitle ● Authors ● Edition ● ISBN ● Genres ● Total available copies ● Available at: (Call number) ● Add to wishlist button (in case the book is not available) <p>7. If the authenticated role is not Patron, then each items in the search result will contains:</p> <ul style="list-style-type: none"> ● Book image ● Title ● Subtitle ● Authors ● Edition ● ISBN ● Genres ● Total available copies ● Call number ● Add to wishlist button (in case the book is not available) ● Create copies button ● Edit button
Other Information	<p>1. If there is no internet connection, the system cannot load the image of book</p>
Assumptions:	N/A

Table 8 <Use Case> Search books

2.2.2.3. View book details

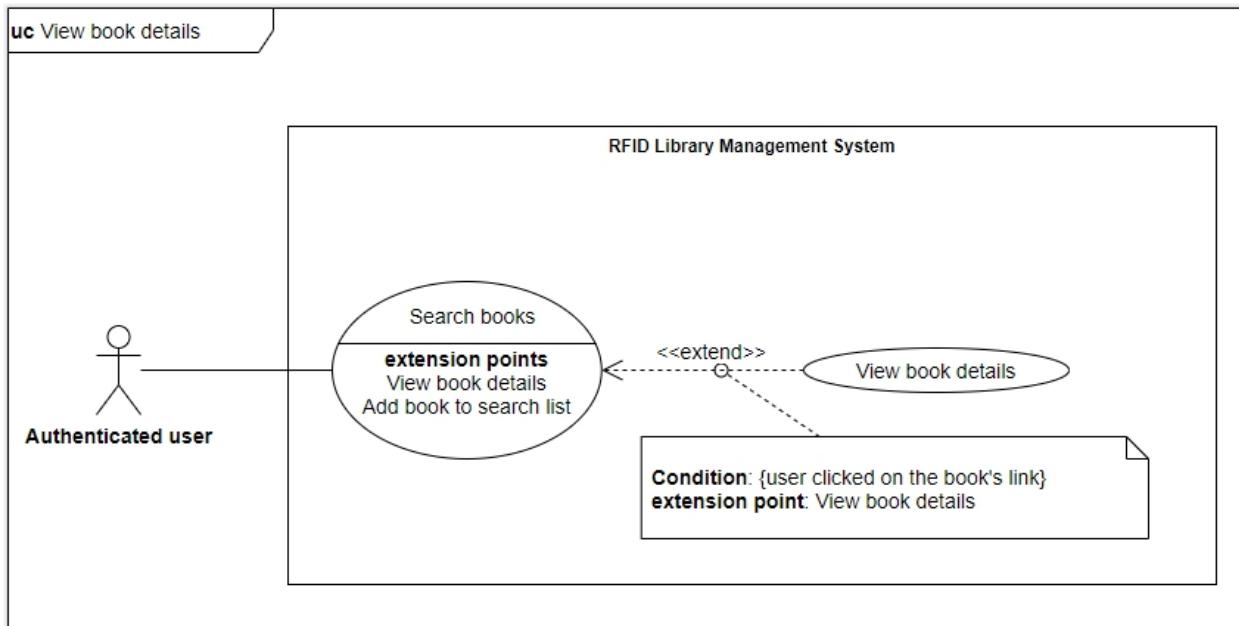


Figure 12 <Use Case> View book details

ID and Name:	UC_06 View book details		
Created By:	TramPH	Date Created:	26/01/2021
Primary Actor:	Authenticated user	Secondary Actors:	N/A
Description:	Let authenticated users view a book's details.		
Trigger:	The authenticated user requests to view a book's details.		
Preconditions:	PRE-1: User has already logged into the system.		
Postconditions:	POST-1: The system shows a book's details.		
Normal Flow:	Step	Actor action	System response
	1	The user clicks on the title of a book.	RLMS redirects to book details screen.
Alternative Flows:	N/A		

Exceptions:	N/A
Priority:	High
Frequency of Use:	Usually
Business Rules:	<p>1. The book detail will contain:</p> <ul style="list-style-type: none"> • General book information: image, title, subtitle, authors, ISBN, edition, publisher, published year, page number, language, genres and book's location • Total available: number of available copy • Call number • Status: <ul style="list-style-type: none"> ○ For patron role: <ul style="list-style-type: none"> ■ AVAILABLE (can borrow and available) ■ NOT AVAILABLE (can borrow but unavailable) ■ LIBRARY USE ONLY (cannot borrow) ○ For librarian and manager role: show all status in the system • List of locations includes: shelf, row • Add to wishlist button: is shown when the book's total available copies is 0 and the book has the status "AVAILABLE" or "LIB_USE_ONLY".
Other Information	<p>1. If there is no internet connection, the system cannot load the image of book</p>
Assumptions:	<p>1. The DDC book classification is added to the system</p>

Table 9 <Use Case> View book details

2.2.2.4. Change password

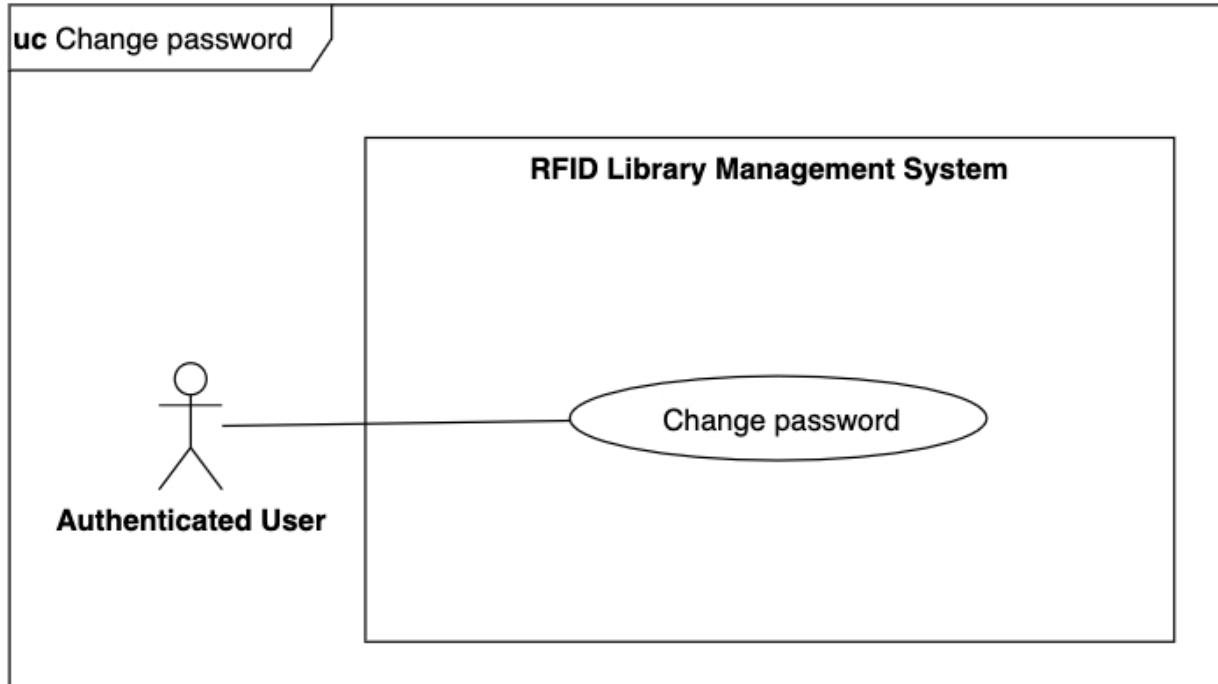


Figure 13 <Use Case> Change password

ID and Name:	UC_07 Change password		
Created By:	KhangNDN	Date Created:	4/03/2021
Primary Actor:	Authenticated user	Secondary Actors:	N/A
Description:	Let authenticated users change their current password.		
Trigger:	The authenticated user requests to change password.		
Preconditions:	PRE-1: User has already logged into the system.		
Postconditions:	POST-1: User's password is updated, the system displays success message and log the user out POST-2: If change password fail, show error message		
Normal Flow:	Step	Actor action	System response
	1	User requests to change password by pressing "Change"	RLMS redirects to Change password Screen

		password” link	
	2	User fills require fields then click “Save” button	The RLMS processes the change password transaction, then shows the success message. [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	Data is invalidated	The system shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Change password form requires following fields: <ul style="list-style-type: none"> • Current password: required • New password: required, 6-50 characters • Confirm password: required, must matches new password 2. After the user has changed their password, they will be required to log in again to continue using the system. 		
Other Information	N/A		
Assumptions:	N/A		

Table 10 <Use Case> Change password

2.2.2.5. Add book to search list

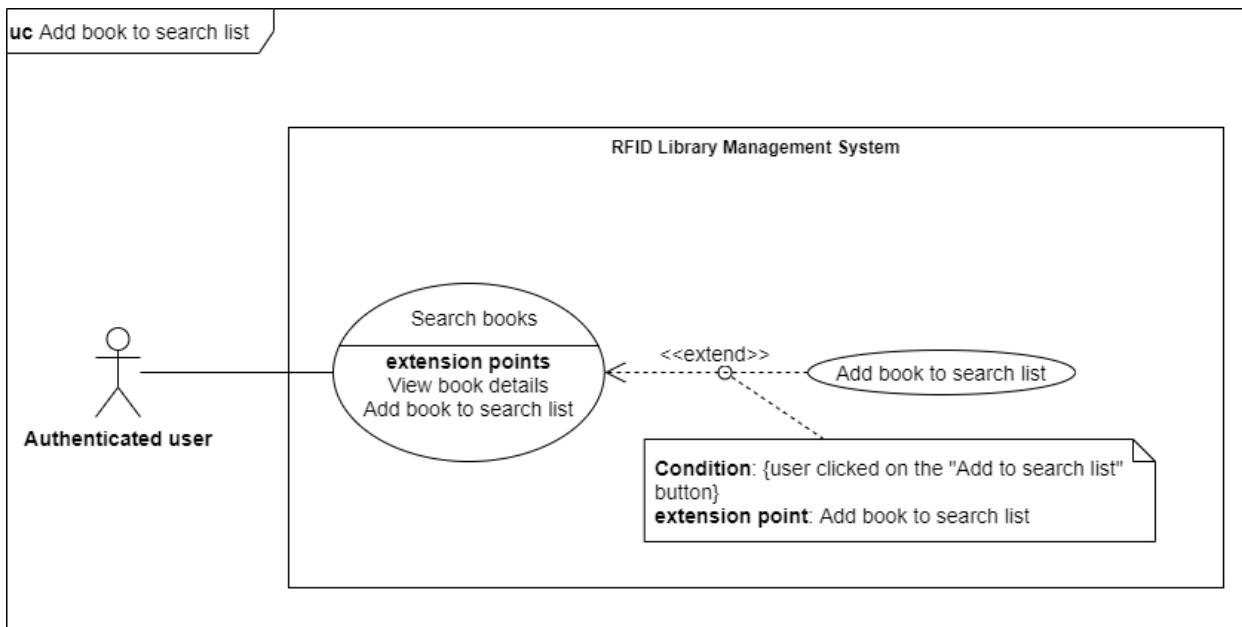


Figure 14 <Use Case> Add book to search list

ID and Name:	UC_08 Add book to search list		
Created By:	TramPH	Date Created:	23/03/2021
Primary Actor:	Authenticated user	Secondary Actors:	N/A
Description:	Let authenticated users add books to a search list in order to search these books by UHF reader later		
Trigger:	The authenticated user requests to add books to a search list		
Preconditions:	PRE-1: User has already logged into the system.		
Postconditions:	POST-1: Books are added to a search list and show success message POST-2: If add fail, show error message		
Normal Flow:	Step	Actor action	System response
	1	User ticks on the checkbox in a book record in the list.	RLMS enables the “Add To Search List” button
	2	User clicks on “Add To Search	RLMS shows confirmation modal

		List” button	
	3	User clicks on “OK” button [Alternative 1]	RLMS adds books to a search list for the user then show success message
Alternative Flows:	Step	Actor action	System response
	1.1	User clicks on “Close” button	RLMS close the confirmation modal
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. User can only add book that are available in the library to the search list 2. The users can access the search if they switch to another device 		
Other Information	N/A		
Assumptions:	N/A		

Table 11 <Use Case> Add book to search list

2.2.2.6. Search book by RFID reader

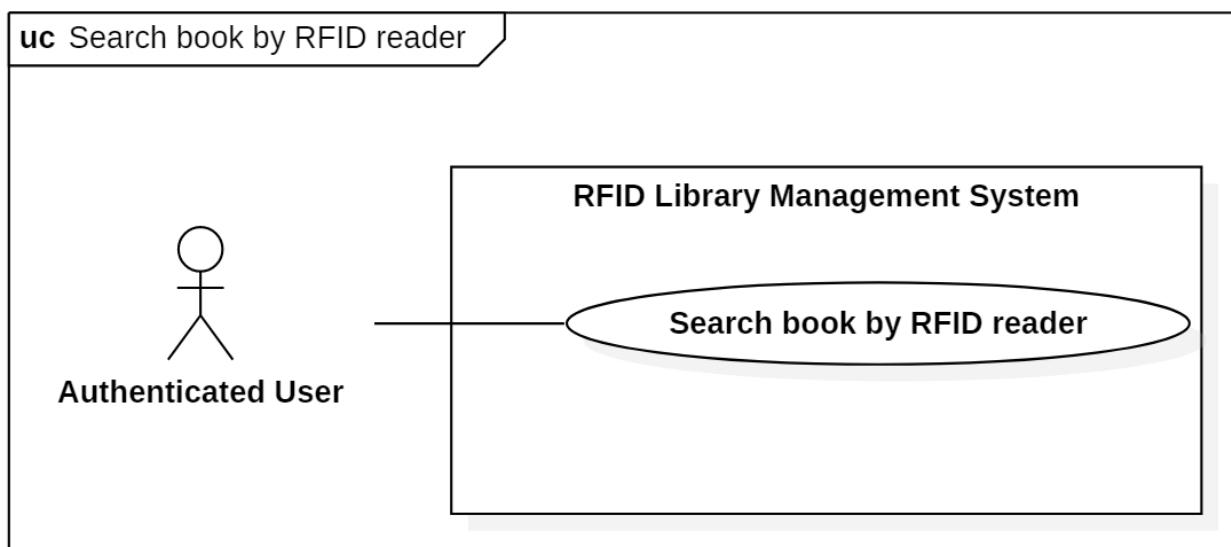


Figure 15 <Use Case> Search book by RFID reader

ID and Name:	UC_09 Search book by RFID reader		
Created By:	TramPH	Date Created:	23/03/2021
Primary Actor:	Authenticated user	Secondary Actors:	N/A
Description:	Let authenticated users search books in the search list by RFID reader		
Trigger:	The authenticated user requests to search books by RFID reader		
Preconditions:	PRE-1: User has already logged into the system.		
Postconditions:	POST-1: If the user move the RFID reader in front of the searching books, the system will alert the user		
Normal Flow:	Step	Actor action	System response
	1	The user clicks on search book menu	RLMS shows the search list of the user
	2	The user clicks on “Start” button [Alternative 1]	RLMS scans books around the reader
	3	The user moves the RFID reader along each shelf’s row [Alternative 1] [Alternative 2]	RLMS alerts the user if it detects a book that the user is searching for.
	4	User clicks “Finish” button	RLMS shows the confirmation modal
	5	User clicks “OK” button [Alternative 3]	RLMS delete the search list and finish searching process
Alternative Flows:	Step	Actor action	System response
	1.1	User clicks on “Clear” button	RLMS shows the confirmation modal

	1.2	User clicks on “OK” button	RLMS return to step 1 of normal flow
	2.1	User ticks on checkbox in a book container	RLMS will understand the user found that book and will not alert the user if it detects the same book again.
	3.1	Users clicks “Close” button	RLMS close the confirmation modal and return to step 2 of normal flow
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. The system show searching book information: <ul style="list-style-type: none"> • Title • Subtitle • Edition • Image • Cal number • ISBN • Locations: if the book is no longer available in the library show message “No longer available” • Checkbox: if the book is no longer available the checkbox will be hide 2. The RLMS will not check for unavailable books in the searching list. 3. When the user clicks the clear or finish button the system will delete the search list. 4. When the RLMS detects a book in the search list, it will alert the user by making a sound and highlight the corresponding book in the list on the screen 5. The users can tick/check books in the search list that they have already found. The RLMS will not alert if it detects these books. 		
Other Information	<ol style="list-style-type: none"> 1. The system should allow users to notice the system if they find the target books or not manually by tick on the checkbox 		
Assumptions:	N/A		

Table 12 <Use Case> Search book by RFID reader

2.2.2.7. View profile

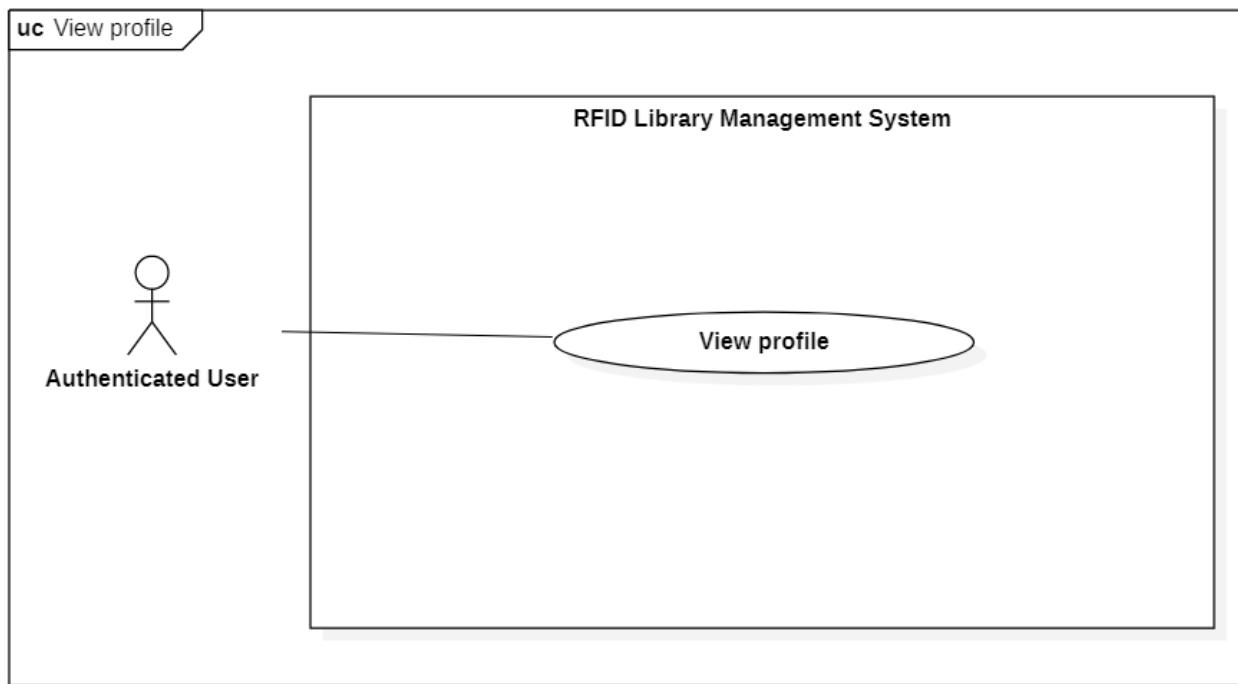


Figure 16 <Use Case> View profile

ID and Name:	UC_10 View profile		
Created By:	TramPH	Date Created:	25/02/2021
Primary Actor:	Authenticated user	Secondary Actors:	N/A
Description:	Let the user view his/her profile.		
Trigger:	The user sends a request to get his/her profile.		
Preconditions:	PRE-1: User has already logged into the system.		
Postconditions:	POST-1: The system shows the user's profile.		
Normal Flow:	Step	Actor action	System response
	1	The user clicks on the profile tab.	RLMS shows the user's profile.
Alternative Flows:	N/A		

Exceptions:	N/A
Priority:	Low
Frequency of Use:	Rarely
Business Rules:	<p>1. The system shows the user profile information:</p> <ul style="list-style-type: none"> • User's avatar • Full name • Email • Phone (editable)
Other Information	<p>1. If there is no internet connection, the system cannot load the avatar</p>
Assumptions:	N/A

Table 13 <Use Case> View profile

2.2.2.8. Update profile

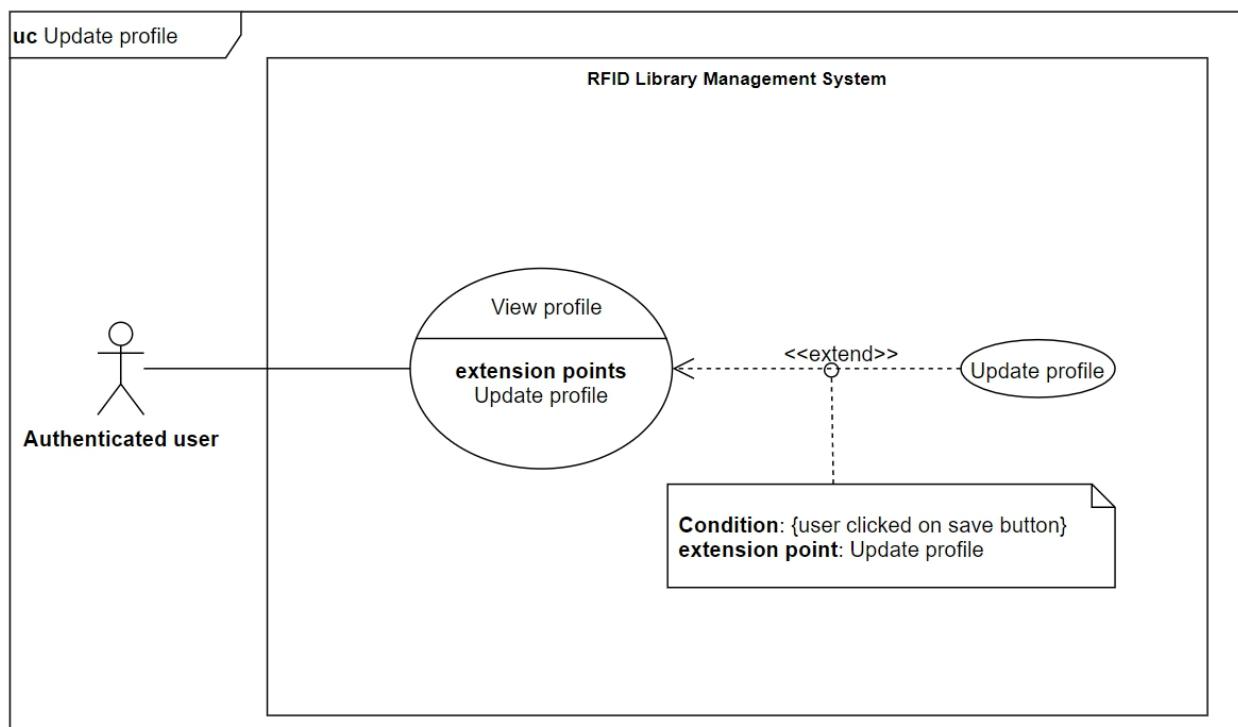


Figure 17 <Use Case> Update profile

ID and Name:	UC_11 Update profile
--------------	-----------------------------

Created By:	TramPH		Date Created:	25/02/2021		
Primary Actor:	Authenticated user		Secondary Actors:	N/A		
Description:	Let the user update his/her profile.					
Trigger:	The user sends a request to update his/her profile.					
Preconditions:	PRE-1: User has already logged into the system.					
Postconditions:	POST-1: If successful, the system updates the profile and shows a successful message. POST-2: If failed, the system shows an error message.					
Normal Flow:	Step	Actor action	System response			
	1	The user changes editable fields and clicks the “Save” button.	RLMS checks validation, updates the profile and shows a successful message. [Exception 1]			
Alternative Flows:	N/A					
Exceptions:	No.	Cause	System response			
	1	The user inputs invalid fields.	RLMS shows error messages.			
Priority:	Low					
Frequency of Use:	Rarely					
Business Rules:	<ol style="list-style-type: none"> 1. The system only allows the user to edit the phone number (10 digits) as updating contact information 2. The user cannot update the avatar by themselves because the avatar is used to verify the user's face only the manager can update the user's avatar 					
Other Information	N/A					
Assumptions:	N/A					

Table 14 <Use Case> Update profile

2.2.3. Patron

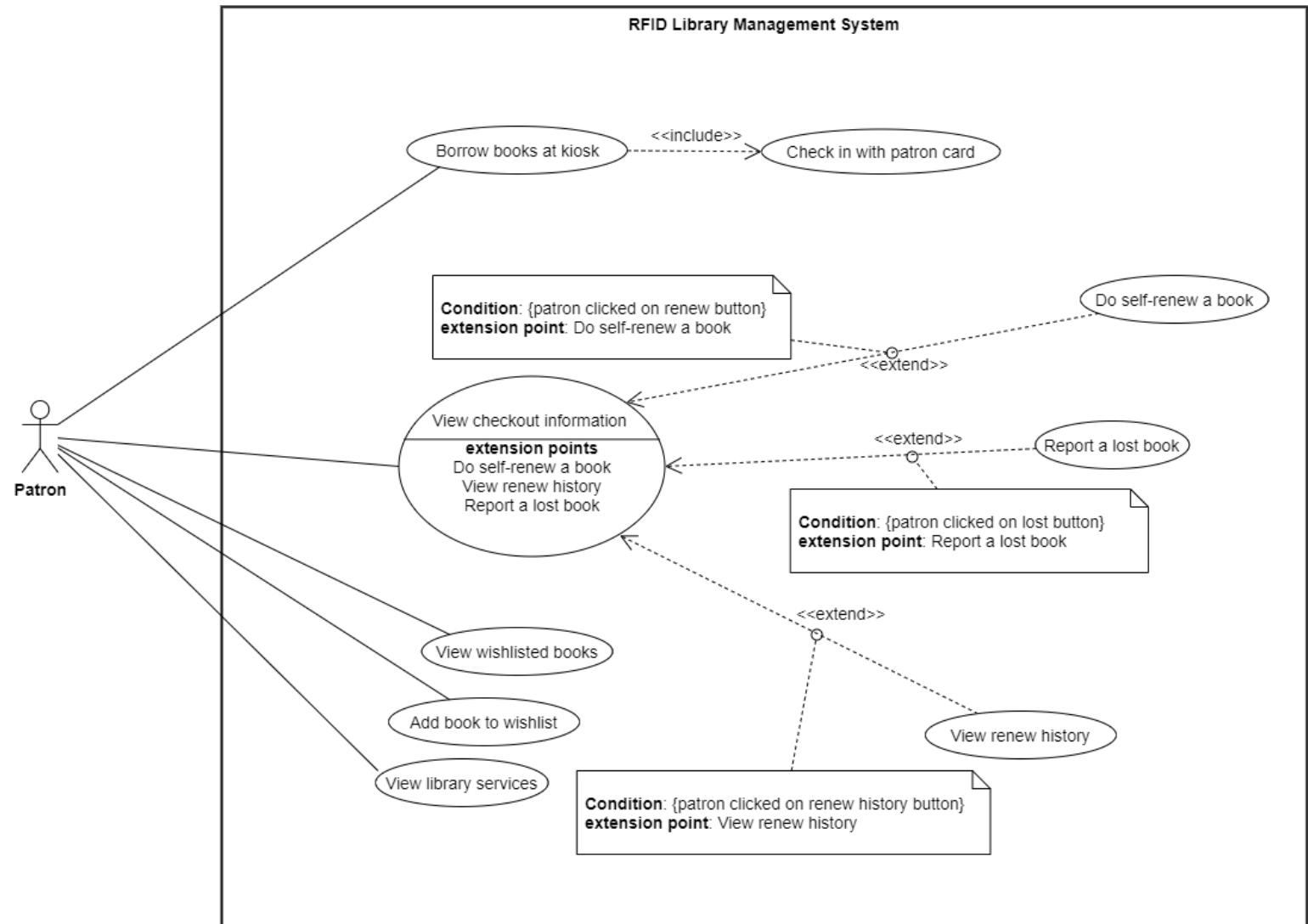


Figure 18 <Use Case Overview> Patron

2.2.3.1. Check in with patron card

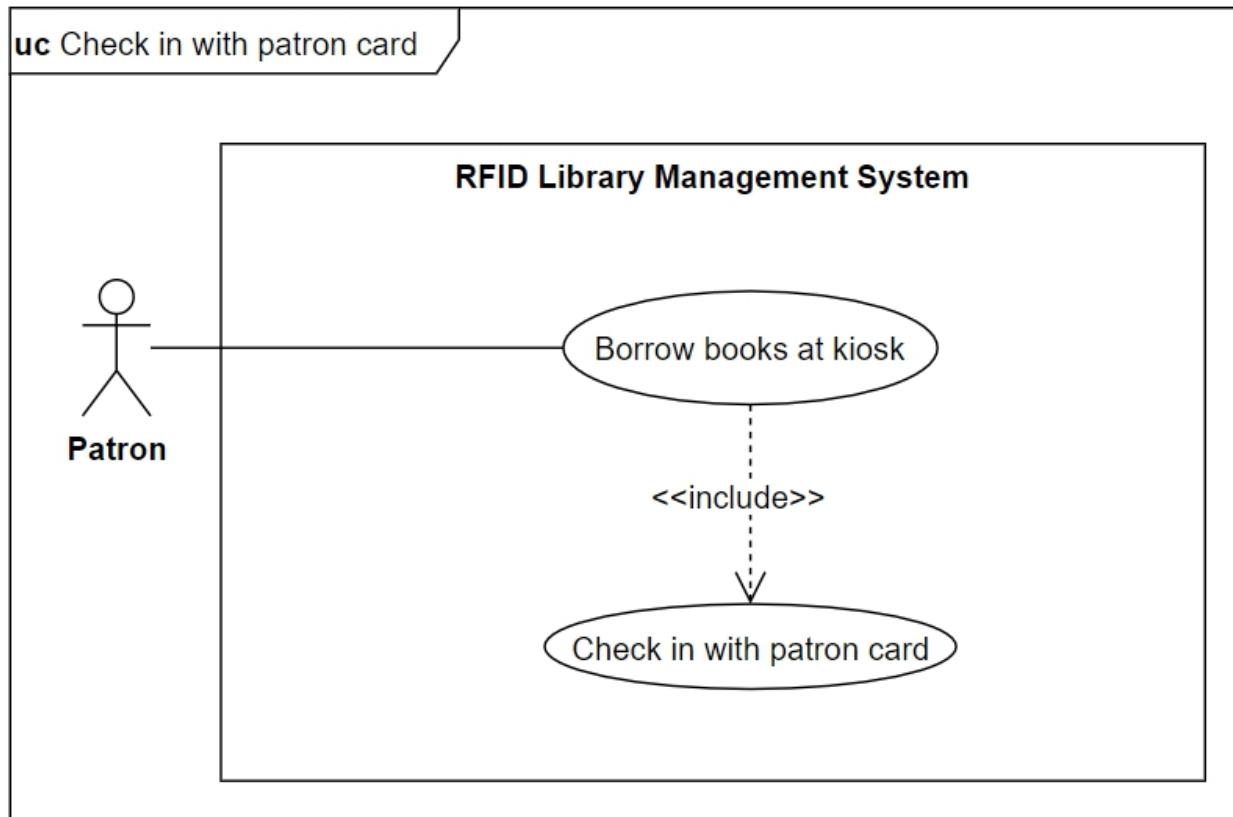


Figure 19 <Use Case> Check in with patron card

ID and Name:	UC_12 Check in with patron card		
Created By:	TramPH	Date Created:	13/01/2021
Primary Actor:	Patron	Secondary Actors:	N/A
Description:	Let the patron get access to the self-checkout function at the kiosk		
Trigger:	The patron scan the patron card at the self-service kiosk in the library		
Preconditions:	PRE-1: User has an patron card		
Postconditions:	POST-1: User is authenticated and authorized to check out books. POST-2: If check in failed, show error message.		
Normal Flow:	Step	Actor action	System response

	1	The users click on “Borrow” button	RLMS redirects to the check in page and the session counts down.
	2	The users put their ID card on the ID card reader at the library self-checkout and return machine. [Alternative 1]	The system validates user’s ID card information. Then the system opens a session for the user and redirects the user to the checkout screen. [Exception 1] [Exception 2] [Exception 3]
Alternative Flows:	Step	Actor action	System response
	1.1	The users use the check-in service for up to 2 minutes.	Session is expired and the RLMS closes the login page.
Exceptions:	No.	Cause	System response
	1	The user’s ID card is not valid as a patron role or inactivated.	RLMS shows the user error message.
	2	The user has overdue books.	RLMS shows the message and closes the login page.
	3	The user scans the invalid card at the third time.	RLMS shows the message and closes the login page.
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. The patron has to check in by their patron card before checkout books 2. The users can only check in at the kiosks as a patron role and be activated. 3. The patron has any overdue book cannot check in at the kiosks 4. The patrons are allowed to scan the invalid ID card 3 times. After the third invalid time, the system will automatically close the check-in process 5. Each check-in session lasts 2 minutes by default, after that the system will close the check-in process. That makes sure the users do not use the kiosk for a long time. 		

Other Information	1. Session shall be paused when the system's making a request to the service.
Assumptions:	1. The patron had an ID card from school. They can be student cards, lecturer cards, etc. They must be RFID cards.

Table 15 <Use Case> Check in with patron card

2.2.3.2. Borrow books at kiosk

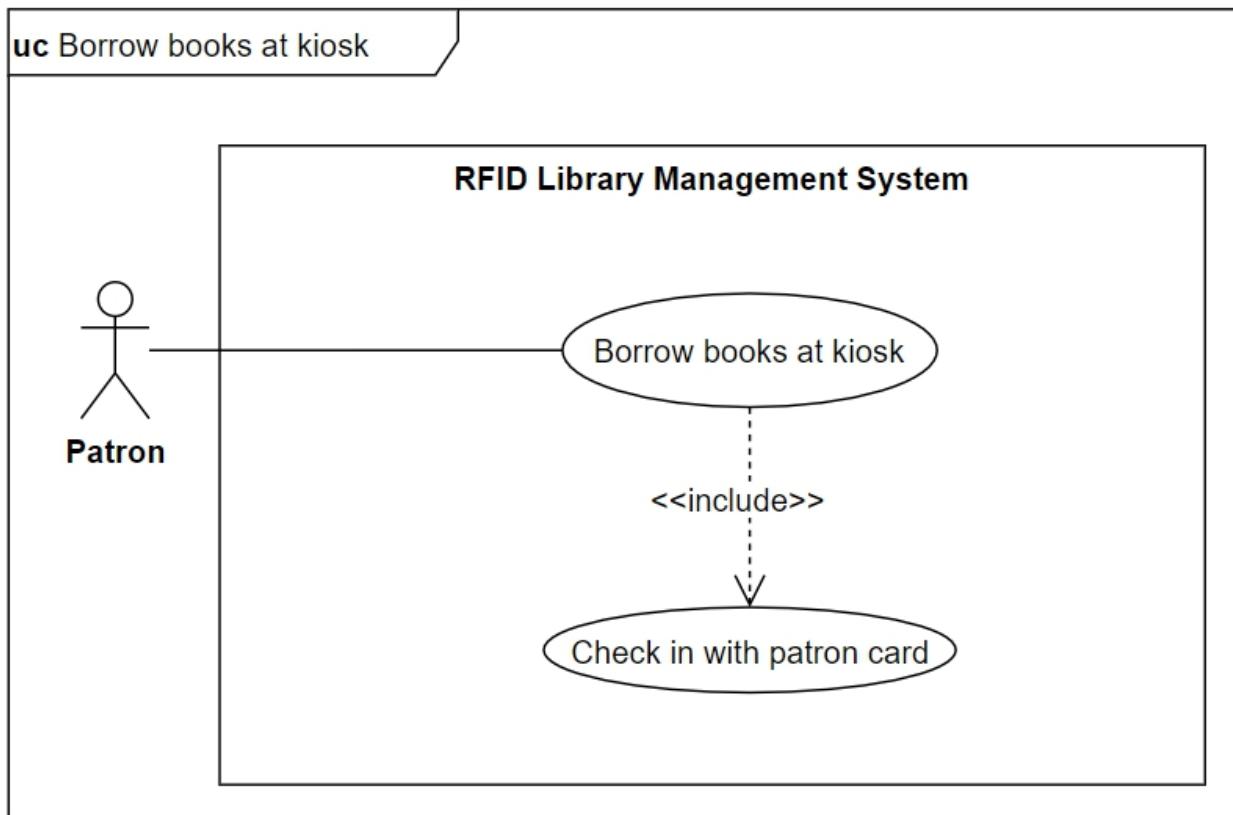


Figure 20 <Use Case> Borrow books at kiosk

ID and Name:	UC_13 Borrow books at kiosk		
Created By:	TramPH	Date Created:	15/01/2021
Primary Actor:	Patron	Secondary Actors:	N/A
Description:	Let patrons borrow books by themselves inside the library.		
Trigger:	The patrons send requests to check out books.		

Preconditions:	PRE-1: The patrons have checked in successfully as a patron role PRE-2: The patrons do not keep any overdue book.		
Postconditions:	POST-1: The system shows a list of borrowed books. POST-2: If checkout failed, show error message.		
Normal Flow:	Step	Actor action	System response
	1	The patrons check in successfully.	RLMS redirects to the checkout page and the session counts down.
	2	The patrons put books on the UHF scanner.	RLMS returns the list of scanned books. [Exception 1]
	3	The patrons click the “Confirm” button. [Alternative 1] [Alternative 2]	RLMS returns the list of borrowed or cannot borrow books then system send receipt via email to the patron [Exception 2] [Exception 3]
Alternative Flows:	Step	Actor action	System response
	1.1	The patrons use the check-out service for up to 4 minutes.	Session is expired and the RLMS closes the check-out screen.
	2.1	The patrons click the “Cancel” button.	RLMS show cancel confirm dialog.
	2.2	The patrons click the “Yes” button.	RLMS closes the check-out screen.
Exceptions:	No.	Cause	System response
	1	RLMS cannot find the scanned book in the system.	RLMS shows error messages and asks the patrons to scan books again.
	2	The patrons borrow over the	RLMS shows error messages and asks the patrons to scan books

		limit allowed.	again.
	3	The patrons borrow duplicated books.	RLMS shows error messages and asks the patrons to scan books again.
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. The patrons can borrow multiple books at the same time by using UHF reader 2. The patrons can only borrow book copy with status “AVAILABLE” and must comply with borrow policy and patron policy. 3. The patrons cannot borrow duplicated books: <ul style="list-style-type: none"> • Duplicated with books scanning at the kiosk. • Duplicated with books the patrons are keeping 4. After scanning books, the system shows: <ul style="list-style-type: none"> • Book information: image, title, subtitle, edition, authors, group 5. After borrowing books, the system shows <ul style="list-style-type: none"> • Book information: image, title, subtitle, edition, authors, group • Checkout information: due date, borrowed time (if borrowed) or message (if couldn't borrow) 6. Due date is calculated based on borrow policy: <ul style="list-style-type: none"> • Due date = borrowed date + maximum borrow days • If the due date is Saturday or Sunday, move the due date to the next Monday. 7. Checked out copy status is updated to “BORROWED” after being borrowed 8. The patron cannot borrow book if they are inactivated or keeping any overdue book 9. The self-checkout session will last 4 minutes by default, after that the system will close the borrowing process. That makes sure the users do not use the kiosk for a long time. 		
Other Information	<ol style="list-style-type: none"> 1. Session shall be paused when the system's making a request to the service. 		
Assumptions:	N/A		

Table 16 <Use Case> Borrow book at kiosk

2.2.3.3. Add book to wishlist

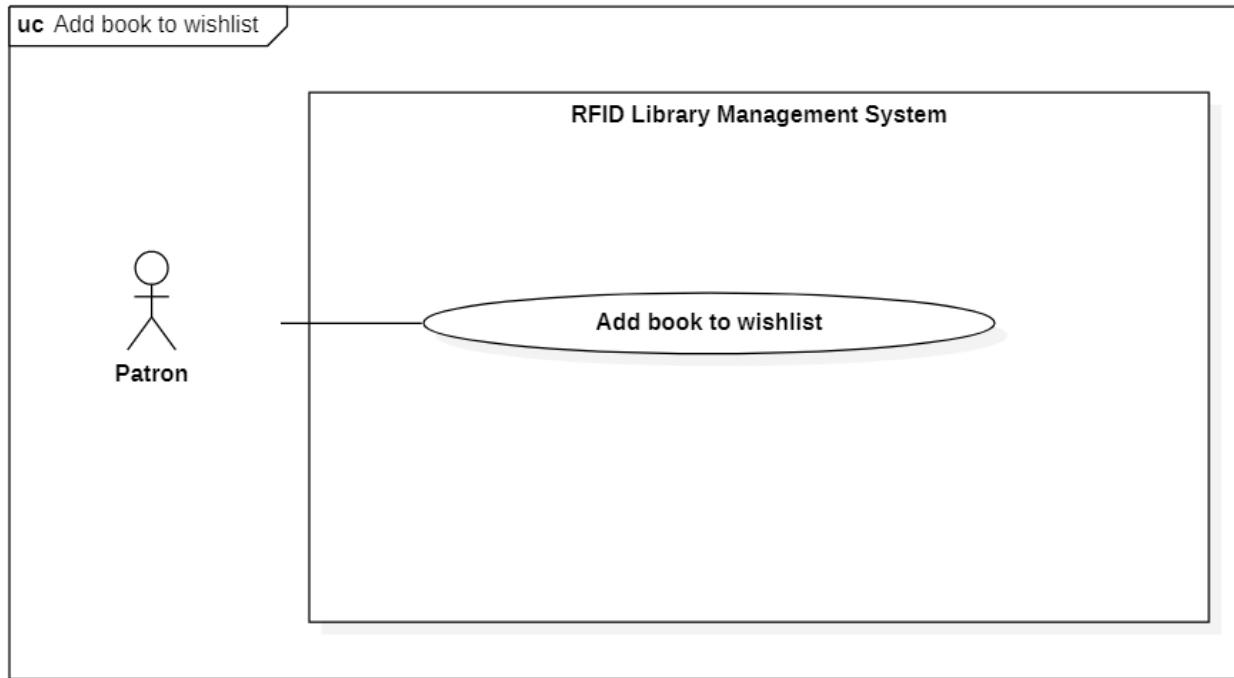


Figure 21 <Use Case> Add book to wishlist

ID and Name:	UC_14 Add book to wishlist		
Created By:	TramPH	Date Created:	25/01/2021
Primary Actor:	Patron	Secondary Actors:	N/A
Description:	Let patrons add an unavailable book to their wishlist.		
Trigger:	The patrons send requests to add a book to the wishlist.		
Preconditions:	PRE-1: The patrons have logged in successfully as a patron role PRE-2: The book is unavailable in the library.		
Postconditions:	POST-1 The book is added to the patron's wishlist and shows a success message. POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response

	1	The patron clicks on the “Add to wishlist button next to an available book from the books screen. [Alternative 1]	RLMS adds that book to the patron’s wishlist and shows a success message. [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	The patron clicks on the “Add to wishlist” button next to an available book from the book details screen.	RLMS behaves as step 1 in normal flow.
Exceptions:	No.	Cause	System response
	1	The patron has added that book to his/her wish list already but not email yet.	RLMS shows an error message.
Priority:	Medium		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. The patron can only add the following books into wish list: Books with status “IN_CIRCULATION” or “LIB_USE_ONLY” with stock is 0 2. The patron cannot have the same book in the wishlist. 3. The system will notify the patron by email when the book is available in the library and remove the available books from the patron’s wish list 		
Other Information	N/A		
Assumptions:	N/A		

Table 17 <Use Case> Add book to wishlist

2.2.3.4. View wishlisted books

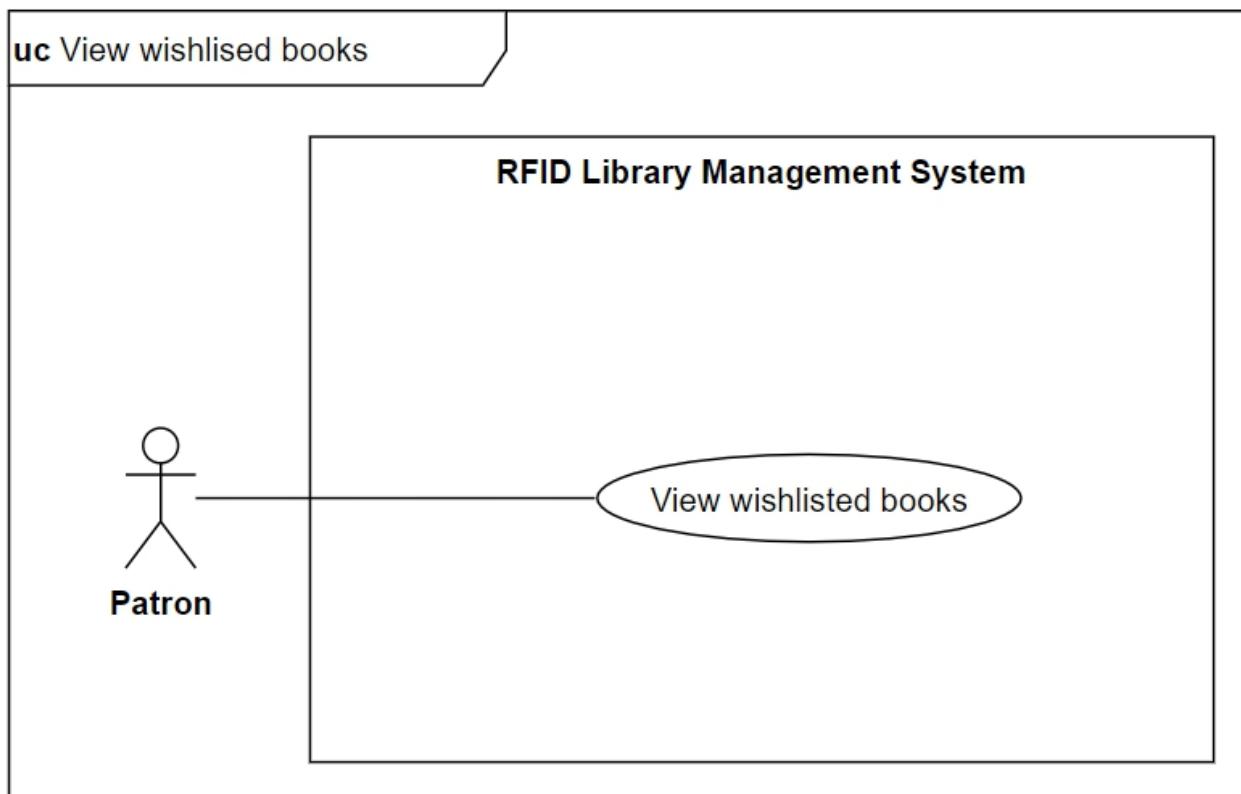


Figure 22 <Use Case> View wishlisted books

ID and Name:	UC_15 View wishlisted books		
Created By:	TramPH	Date Created:	25/02/2021
Primary Actor:	Patron	Secondary Actors:	N/A
Description:	Let the patron view current wishlisted books		
Trigger:	The patron sends a request to get their current wishlist.		
Preconditions:	PRE-1: The user has logged in as Patron role		
Postconditions:	POST-1: The system shows a list of wishlist books POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response

	1	The patron clicks on the “My Wishlist” button on the books screen. [Exception 1]	RLMS shows a list of current wishlist books.
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. The patron can only view the current wishlist books - wishlist books have not been emailed to notify if available yet. 2. The system shows a list of wishlist book includes: <ul style="list-style-type: none"> ● Book image ● Title - subtitle ● Authors ● Edition ● ISBN 		
Other Information	<ol style="list-style-type: none"> 1. If there is no internet connection, the system cannot load the book image 		
Assumptions:	N/A		

Table 18 <Use Case> View wishlisted books

2.2.3.5. View borrow history

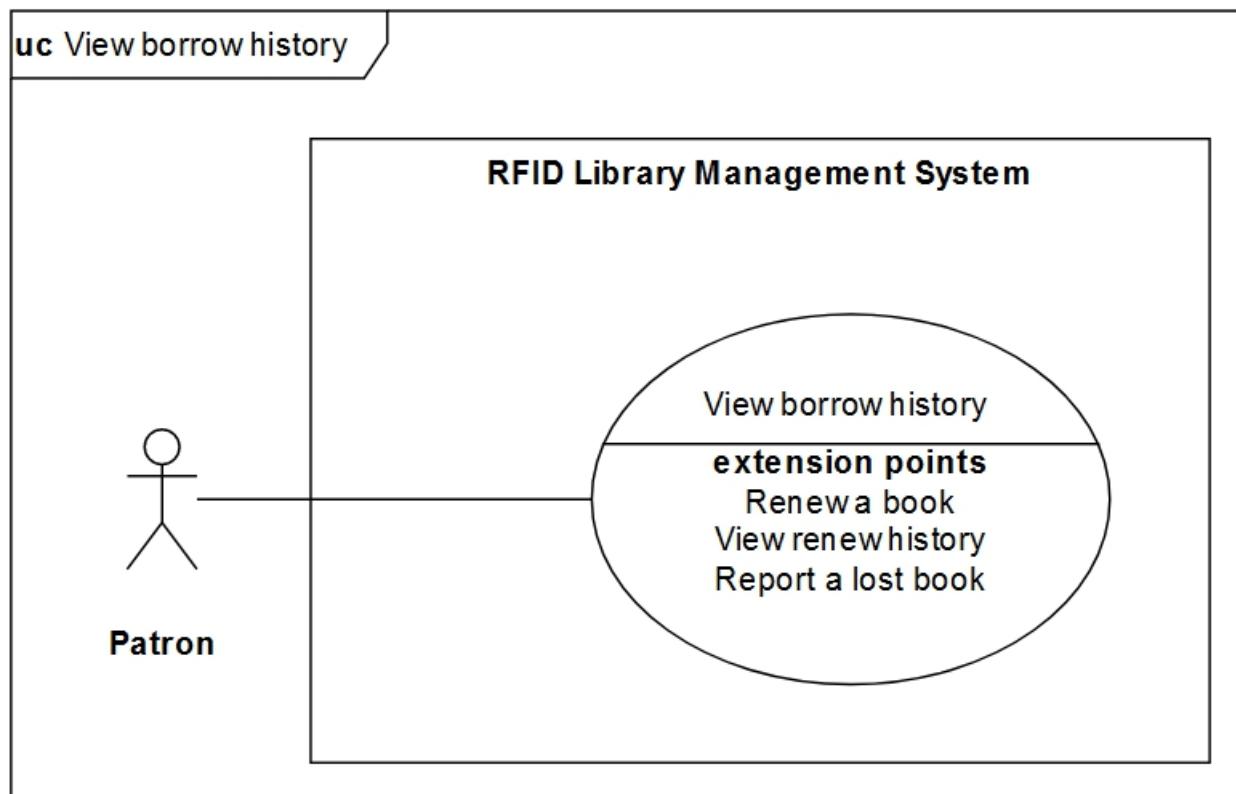


Figure 23 <Use Case> View borrow history

ID and Name:	UC_16 View borrow history		
Created By:	TramPH	Date Created:	25/02/2021
Primary Actor:	Patron	Secondary Actors:	N/A
Description:	Let the patron view their borrow history		
Trigger:	The patron sends a request to get their borrow history		
Preconditions:	PRE-1: The user has logged in as Patron role		
Postconditions:	POST-1: The system shows the patrons borrow history POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response

	1	The patron clicks on “Checkout information” in the sidebar.	RLMS shows list of borrowing books
	2	The patrons clicks on “Returned Books”, “Lost Books”, “Overdue Books”	RLMS shows corresponding books.
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. The system allows the patron to keep track of all their borrowing records: borrowing books, returned books, overdue books and lost books. 2. The returned book list and overdue book list include: <ul style="list-style-type: none"> • Book information: title, subtitle, edition • Borrowed time • Returned time • Overdue days • Fine • Renew history button 3. The borrowing book list includes: <ul style="list-style-type: none"> • Book information: title, subtitle, edition • Borrowed time • Due date • Renew button • Renew history button 4. The lost book list includes: <ul style="list-style-type: none"> • Book information: title, subtitle, edition • Borrowed time • Lost time • Fine • Status (CONFIRMED or PENDING) 5. Overdue days will exclude Saturday and Sunday 6. Overdue fine is calculated based on fee policy at the time users borrowed those books <ul style="list-style-type: none"> • Overdue fine = overdue days * fine rate • Max overdue fine = book copy's price * max overdue fine percentage / 100 7. Lost fine is calculated based on the lost fine the librarians input when they confirm the lost record plus the overdue fine. 		

	8. The table will be sorted by when the book copy was borrowed. First table record is the latest borrowed book copy.
Other Information	N/A
Assumptions:	N/A

Table 19 <Use Case> View borrow history

2.2.3.6. Do self-renew a book

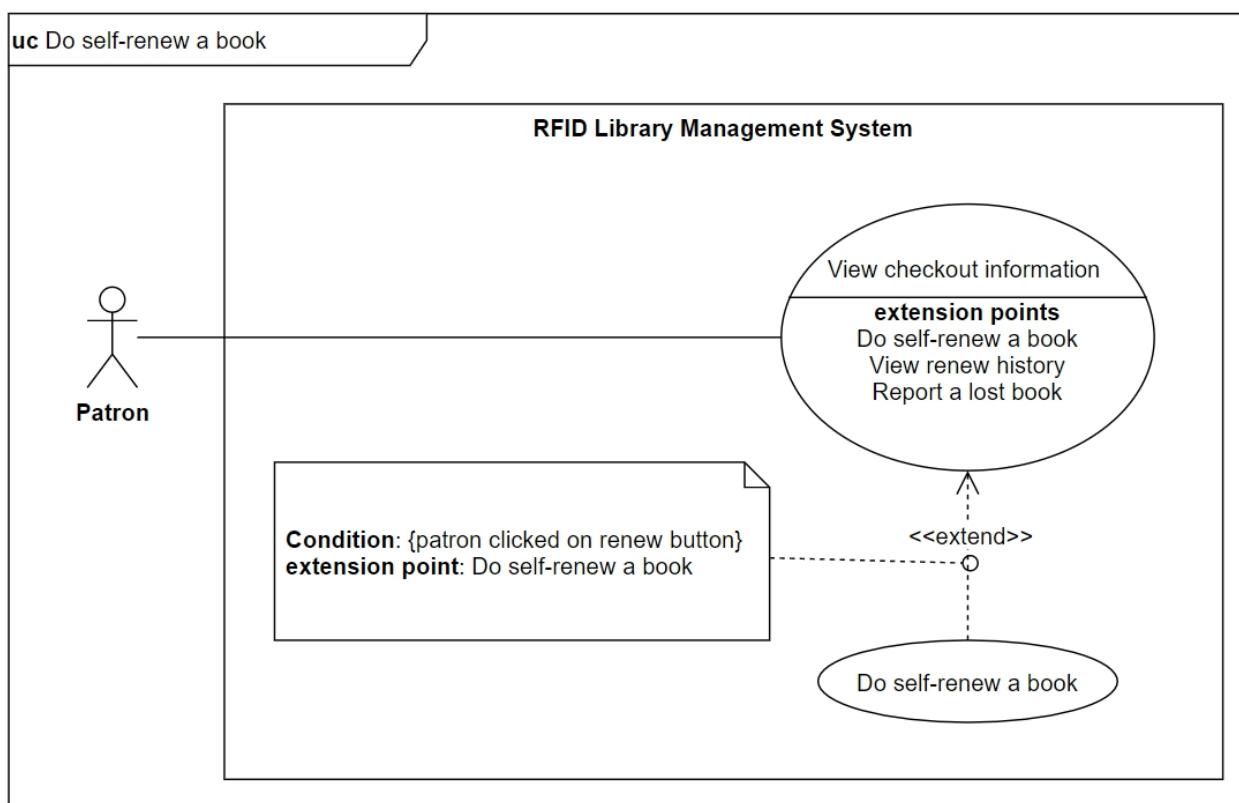


Figure 24 <Use Case> Do self-renew a book

ID and Name:	UC_17 Do self-renew a book		
Created By:	TramPH	Date Created:	25/02/2021
Primary Actor:	Patron	Secondary Actors:	N/A
Description:	Let the patron renew a borrowing book.		

Trigger:	The patron sends a request to renew a borrowing book.		
Preconditions:	PRE-1: The user has logged in as Patron role PRE-2: Borrowing book is not overdue.		
Postconditions:	POST-1: The system renews the borrowing book and shows a successful message. POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The patron clicks on the “Renew” button.	RLMS checks the borrow policy and show confirmation dialog with new due date [Exception 1] [Exception 2] [Exception 3]
	2	The patron clicks on the “Confirm” button [Alternative 1]	RLMS processes renewing the borrowing book and shows a successful message.
Alternative Flows:	Step	Actor action	System response
	1.1	The patron click on the “Back” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The patron is keeping overdue books.	RLMS shows error messages.
	2	The patron violates the borrow policy	RLMS shows error messages.
	3	The patron is not allowed to borrow this book copy type anymore.	RLMS shows error messages.
Priority:	High		

Frequency of Use:	Occasionally
Business Rules:	<ol style="list-style-type: none"> 1. The patron keeping any overdue book copies or violating any borrow policy cannot renew a book. 2. Cannot renew if the patron type is no longer allowed to borrow the book copy type 3. The system shows the new due date first to let the patron decide whether to renew or not 4. New due date = old due date + renew period (renew period is set on the Borrow Policy)
Other Information	N/A
Assumptions:	N/A

Table 20 <Use Case> Do self-renew a book

2.2.3.7. View renew history

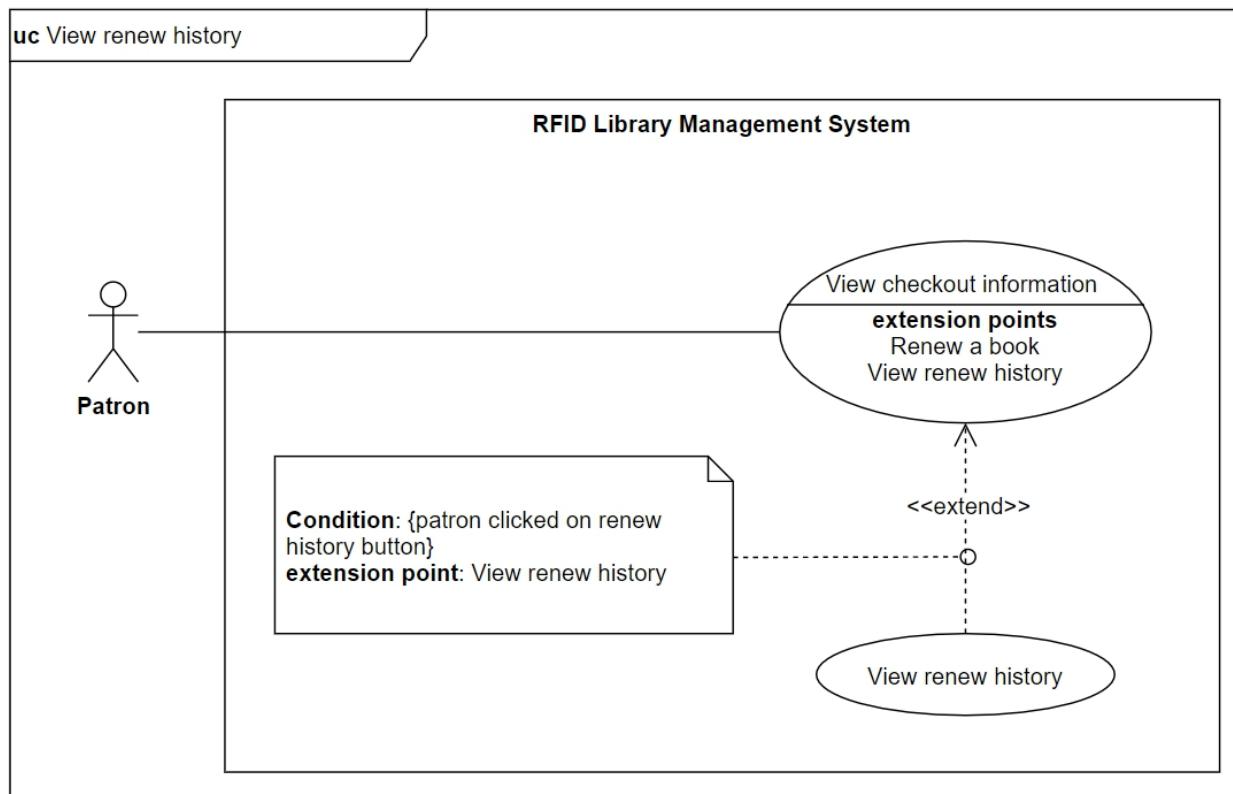


Figure 25 <Use Case> View renew history

ID and Name:	UC_18 View renew history
--------------	--------------------------

Created By:	TramPH		Date Created:	25/02/2021		
Primary Actor:	Patron, Librarian		Secondary Actors:	N/A		
Description:	Provide the patron or librarian with the ability to view the renew history of a borrowed book					
Trigger:	The patron or librarian sends request to view renew history					
Preconditions:	PRE-1: The user has logged in as Patron or Librarian role					
Postconditions:	POST-1: The system shows the renew history POST-2: If failed, show error message					
Normal Flow:	Step	Actor action	System response			
	1	The patron or librarian clicks on the “History” button next to a borrowed book [Exception 1]	The RLMS process the request, then shows a table contains renew history records			
Alternative Flows:	N/A					
Exceptions:	No.	Cause	System response			
	1	The user retrieves an empty list	RLMS shows there is no data			
Priority:	Low					
Frequency of Use:	Rarely					
Business Rules:	<ol style="list-style-type: none"> 1. The system allows the patron and librarian to keep track of all the renew times of a borrowed book. 2. A record in the renew history contains the following: <ul style="list-style-type: none"> • Index number • When was the book copy renewed • When was the book copy due date 3. The records will be sorted by when the book copy was renewed. From the first renewal to the latest renewal. 					

Other Information	N/A
Assumptions:	N/A

Table 21 <Use Case> View renew history

2.2.3.8. Report lost book

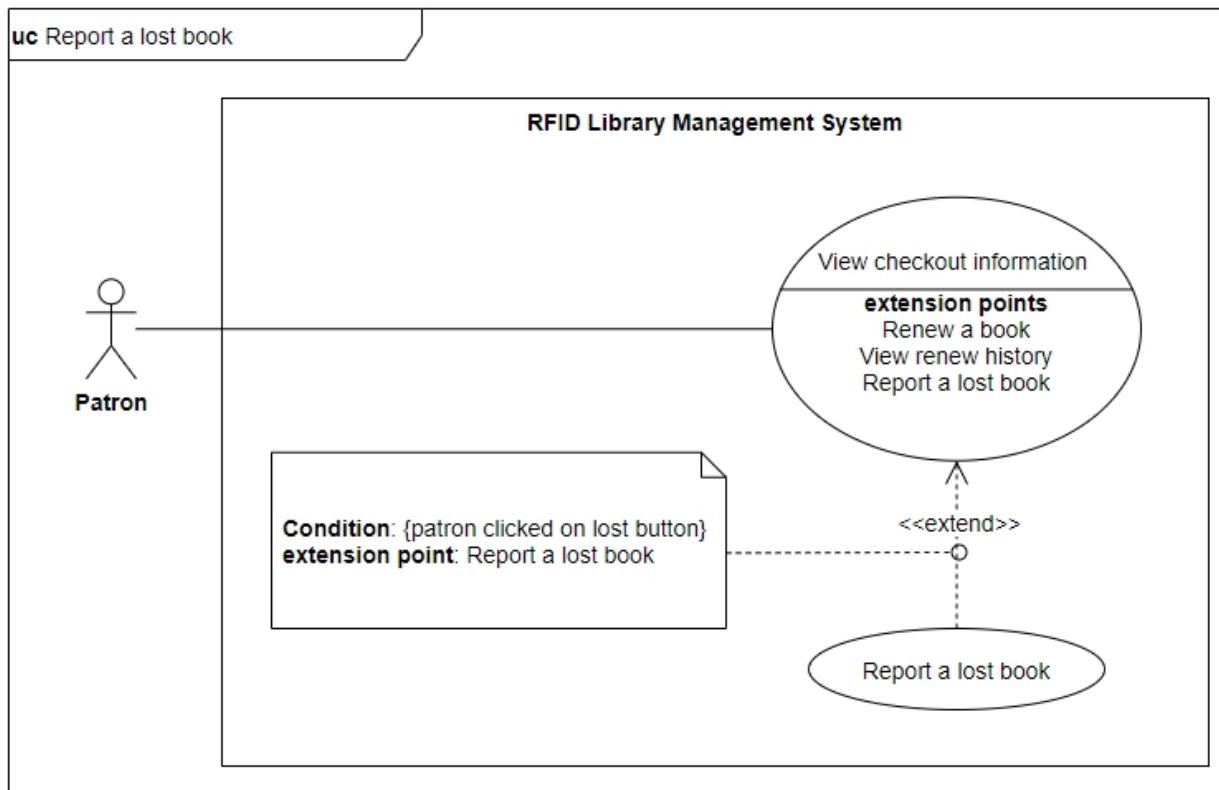


Figure 26 <Use Case> Report lost book

ID and Name:	UC_19 Report lost book		
Created By:	TramPH	Date Created:	25/02/2021
Primary Actor:	Patron	Secondary Actors:	N/A
Description:	Let the patron report the borrowing book that is lost		
Trigger:	The patron sends request to report lost a book		
Preconditions:	PRE-1: The user has logged in as Patron role		

	PRE-2: The book is in borrowing books or overdue books		
Postconditions:	POST-1: The system shows the success message POST-2: If failed, the system show error message		
Normal Flow:	Step	Actor action	System response
	1	The patron clicks on the “Lost” button next to a borrowed book. [Alternative 1]	RLMS shows confirmation modal
	2	The patron clicks on the “OK” button	RLMS process the request and show success message
Alternative Flows:	Step	Actor action	System response
	1.1	The patron click the “Close” button	RLMS close the confirmation modal
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. The system allows the patron to report a borrowing book that is lost. 2. After reporting, the status of that report will be “PENDING”, and the patron will wait for the librarian calculate the lost fine then send the email to notify the patron 3. After reporting, the system will change book copy status to “LOST” 4. When the patron reports a book that was lost, that book is no longer in library’ assets. The ownership of the lost book will transfer to that patron 5. The patron can only report lost for the borrowing books and overdue books in the borrowing history menu. 		
Other Information	N/A		
Assumptions:	<ol style="list-style-type: none"> 1. The system will not manage the finance 		

Table 22 <Use Case> Report lost book

2.2.3.9. View library services

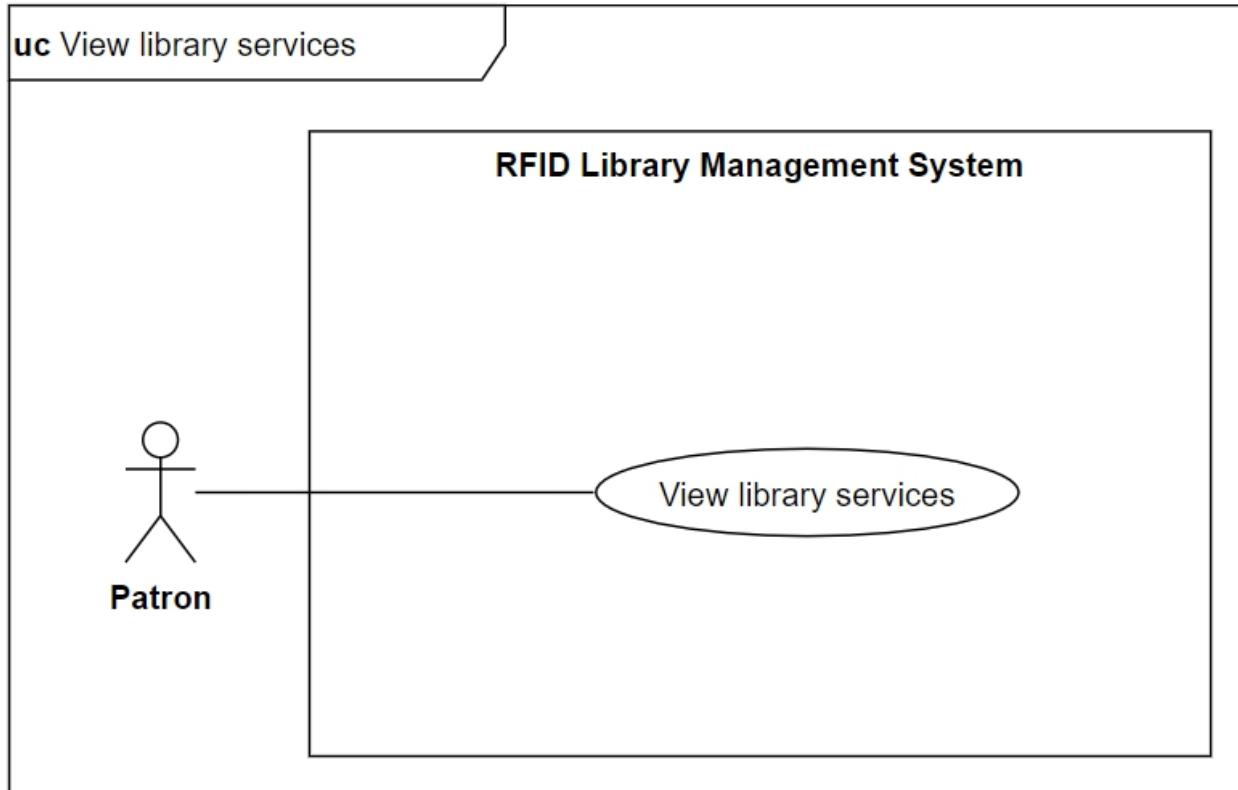


Figure 27 <Use Case> View library services

ID and Name:	UC_20 View library services		
Created By:	KhangNDN	Date Created:	24/03/2021
Primary Actor:	Patron	Secondary Actors:	N/A
Description:	Let the patron view the policies of library		
Trigger:	The patron sends request to view policy		
Preconditions:	PRE-1: The user has logged in as Patron role		
Postconditions:	POST-1: The system shows the library's policy POST-2: If view fail, show error message		
Normal Flow:	Step	Actor action	System response

	1	The patron clicks on the Library services menu	The RLMS process the request, then shows all the library services
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. The system shall public all the current policies to the patron: borrow policy, patron policy and fee policy 2. The system provides the patrons with service information of the library: book drop service, self-service kiosk, search book location using handheld devices 		
Other Information	N/A		
Assumptions:	N/A		

Table 23 <Use Case> View library services

2.2.4. Librarian

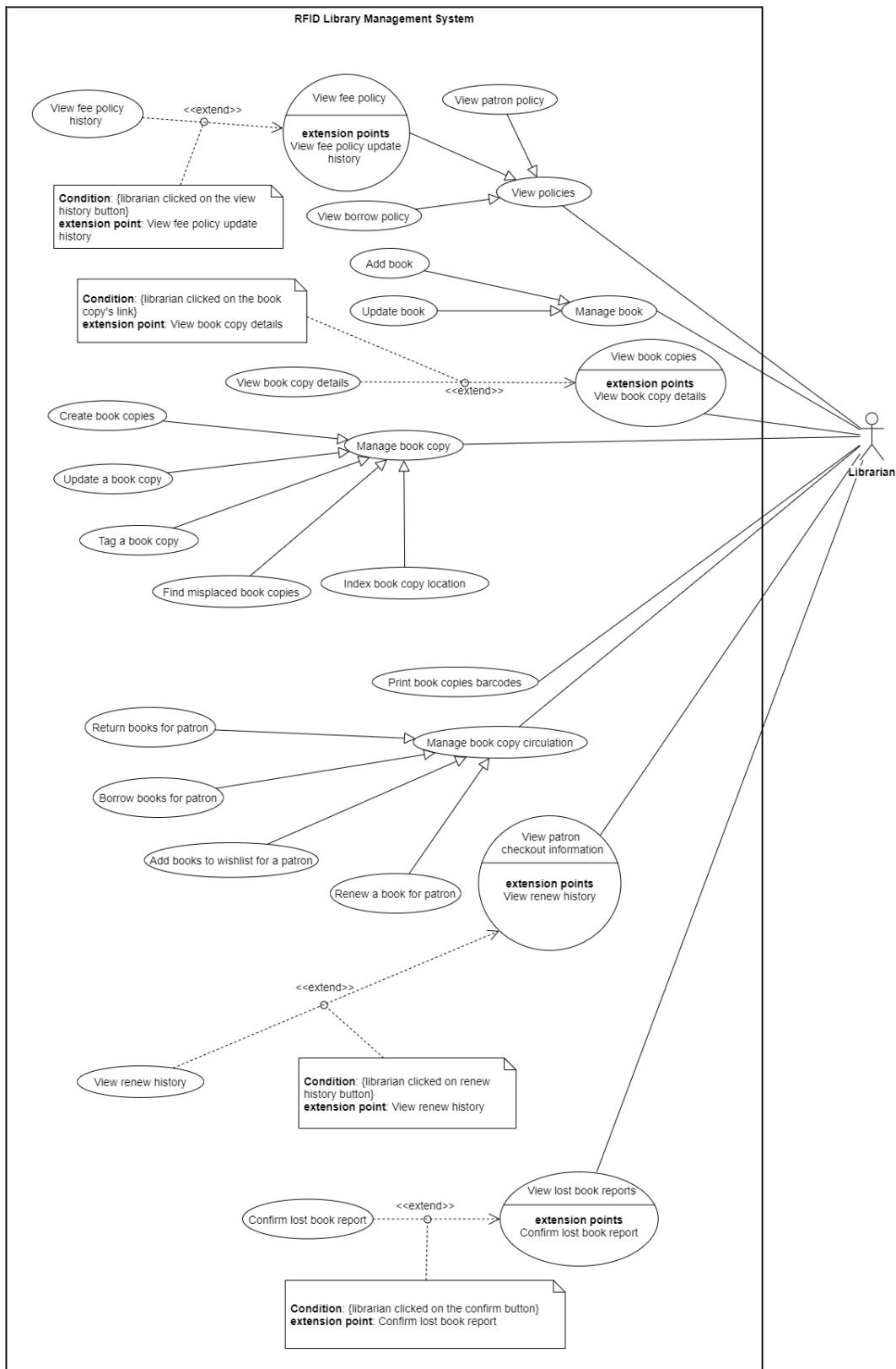


Figure 28 <Use Case Overview> Librarian

2.2.4.1. Borrow books for a patron

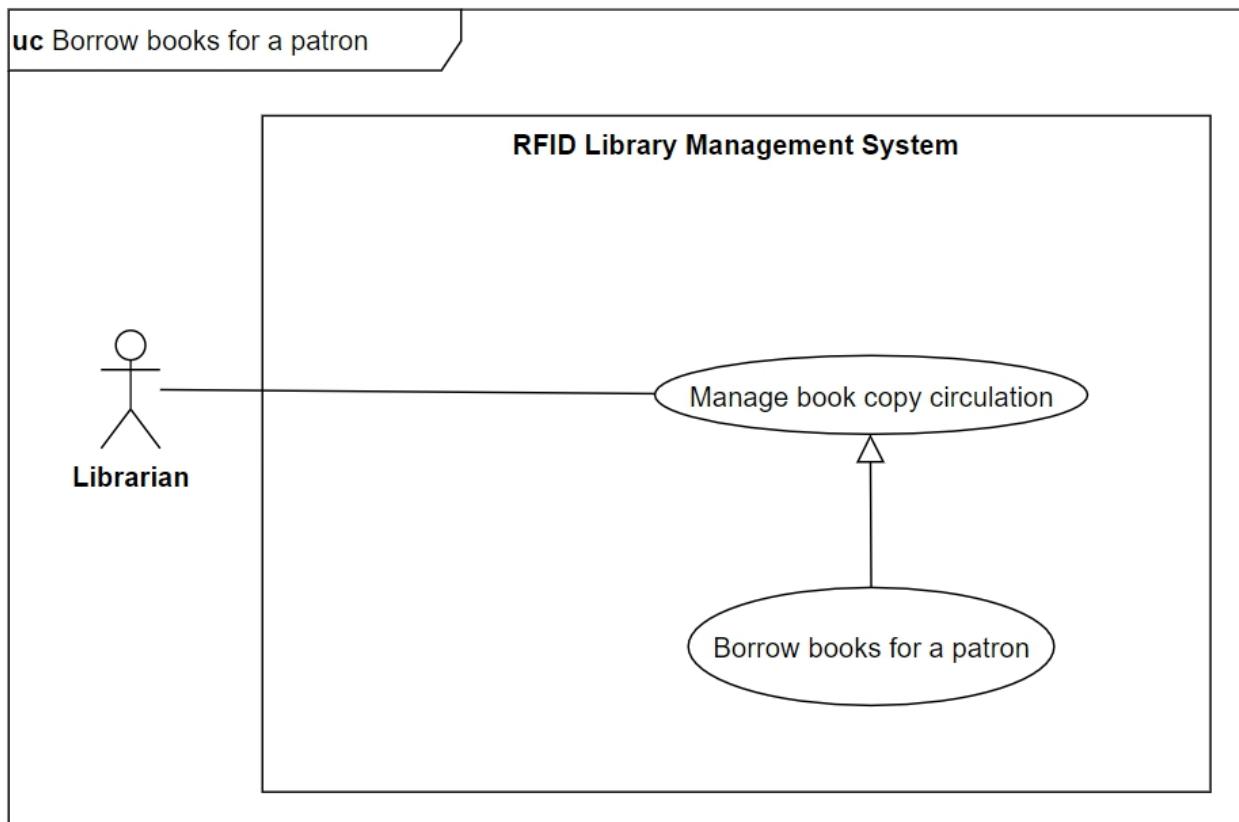


Figure 29 <Use Case> Borrow books for a patron

ID and Name:	UC_21 Borrow books for a patron		
Created By:	KhangNDN	Date Created:	18/02/2021
Primary Actor:	Librarian	Secondary Actors:	System handler, Patron
Description:	Let the librarian check out books for a patron.		
Trigger:	The librarian requests to check out books for a patron.		
Preconditions:	PRE-1: User has logged in as Librarian role		
Postconditions:	POST-1: Show success message, the patron receives a checkout receipt email POST-2: If checkout failed, show error message		
Normal Flow:	Step	Actor action	System response

	1	The librarian clicks on the checkout button on the sidebar	The RLMS displays the checkout UI
	2	Librarian scan patron's ID card or enter the patron's email and click on the search button	The system searches for patron information and displays the results to the screen. [Exception 1]
	3	The librarian scans all the checkout books or scan the book barcode and click on the search button	The system retrieves data from the database and checks borrow policy then shows those books information on the screen. [Exception 2]
	4	The librarian checks if all the books are shown on screen. The librarian can rescan the book(s) that are not shown on screen. The librarian can also click the delete button on the book that the patron does not want to borrow. Then, the librarian clicks Checkout button [Alternative 1]	The system checks patron policy then shows confirmed modal and warning messages (if any).
	5	The librarian clicks Confirm button [Alternative 2] [Alternative 3]	The system processes the checkout transaction, then shows the checkout success message. Then the system sends a receipt email to the patron.
Alternative Flows:	Step	Actor action	System response
	1.1	Librarian clicks Clear button	The system clears all scanned books and patron's information.
	2.1	Incase if the checkout is violating any policy, the librarian must input borrowing reason and then clicks Confirm button	The system processes the checkout transaction, then shows the checkout success message. Then the system sends a receipt email to the patron.

	3.1	Librarian clicks Cancel button	Confirm modal closes and returns to the checkout UI
Exceptions:	No.	Cause	System response
	1	User is not found in the database or the user's role is not Patron.	The system shows an error message.
	2	Book is not found in the database	The system shows an error message.
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. Librarian can only check out for patrons' ID card 2. Book information: Image, title, subtitle, edition, barcode, genre, ISBN and due date. 3. The librarian can only click the Checkout button if the <i>Borrow Policy</i> allows the patron type to borrow all the checkout copies. 4. If scanned books violate the <i>Borrow Policy</i>, error message would be shown. 5. Only book copy with status "AVAILABLE" can be checked out. 6. Before checking out, the RLMS must validate and show warning message when: <ul style="list-style-type: none"> • Borrowing book copies violate <i>Borrow Policy</i> • Borrowing book copies violate <i>Patron Policy</i> • Does not exist a <i>Borrow Policy</i> for the patron type and the book copy type (The patron is not allowed to borrow the book copy type) • Borrowing more than 1 copy of a same book • The patron is keeping any overdue book copy • The borrowing book copy is not "AVAILABLE" 7. Confirm modal includes: Warning message, list of borrowing books and borrowing reason. 8. Librarians can still checkout for patrons (must provide reasons) if: <ul style="list-style-type: none"> • Borrowing book copies violate <i>Patron Policy</i> • Borrowing book copies violate <i>Borrow Policy</i> • The patron is keeping any overdue book copy • Borrowing more than 1 copy of a same book 9. The RLMS will be able to checkout multiple book copies at a time. 10. Checkout book copies can be scanned by UHF RFID scanner or by barcode reader. The librarian can also enter the book barcode manually. 		

	<ul style="list-style-type: none"> 11. Due date is calculated for each checkout book copy by adding n day(s) to the current date, (n days is taken from <i>Borrow Policy</i>). If the due date is on Saturday or Sunday, then move the due date to Monday. 12. Checked out copy status is updated to “BORROWED” 13. Cannot checkout if patron is inactive 14. Security alarm is deactivated for checked out book copies
Other Information	<ul style="list-style-type: none"> 1. If there is no internet connection, the system cannot load the book image
Assumptions:	N/A

Table 24 <Use Case> Borrow books for a patron

2.2.4.2. Return books for a patron

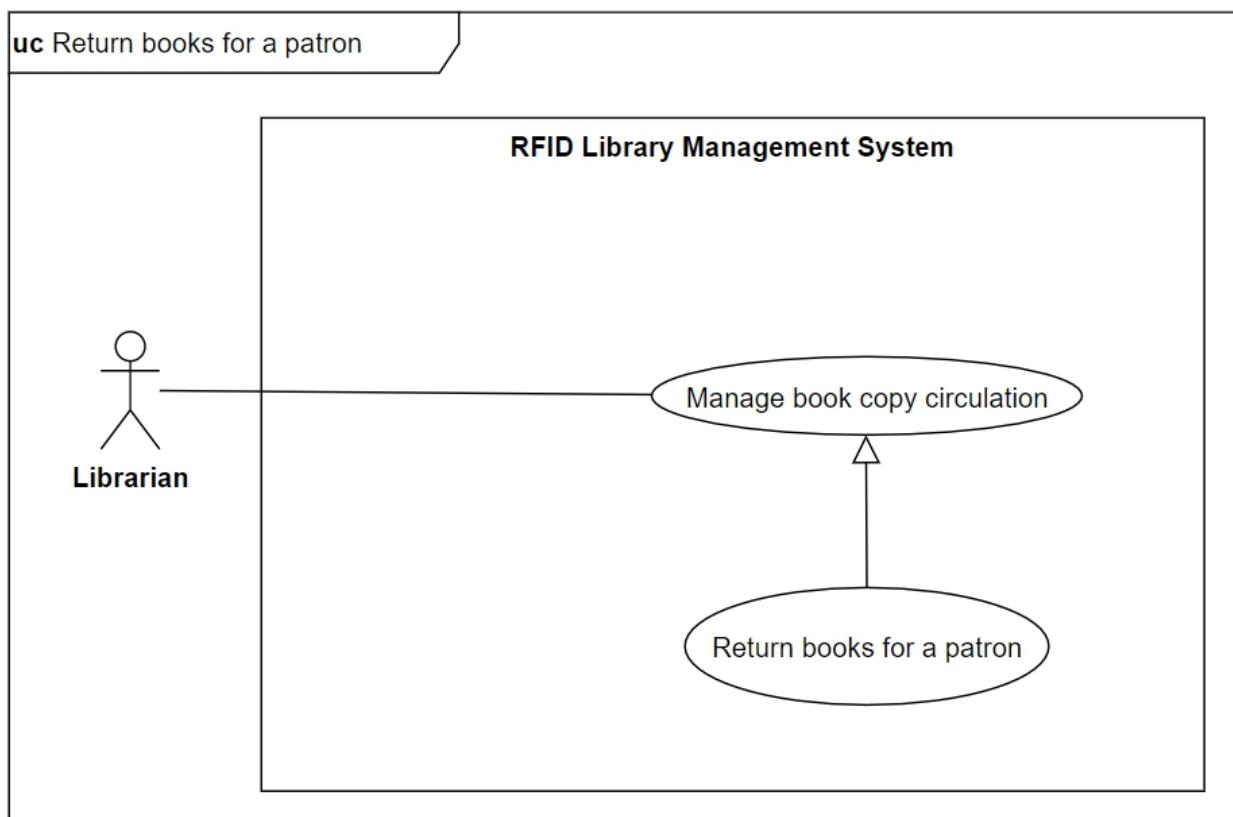


Figure 30 <Use Case> Return books for a patron

ID and Name:	UC_22 Return books for a patron		
Created By:	KhangNDN	Date Created:	18/02/2021

Primary Actor:	Librarian		Secondary Actors:	System Handler
Description:	Librarian can return books for a patron			
Trigger:	The librarian requests to return books for a patron			
Preconditions:	PRE-1: User has logged in as Librarian role			
Postconditions:	POST-1: Show success message, the patron receives a return receipt email POST-2: If return failed, show error message			
Normal Flow:	Step	Actor action	System response	
	1	The librarian clicks on return button on the side bar	The RLMS displays the return UI	
	2	The librarian scans the books with RFID scanner or barcode scanner.	The system retrieves data from the database then shows those books information on the screen. [Exception 1]	
	3	The librarian checks if all the books shown on screen are correct. The librarian can rescan the book(s) that are not shown on screen. The librarian can also click the delete button on the book that the patron does not want to return. Then, the librarian clicks the Return button. [Alternative 1]	The system then shows the confirmed modal.	
	4	The librarian clicks Confirm button [Alternative 2]	The system processes the return transaction, then shows the return success message. Then the system sends a return receipt to the patron via email.	
Alternative Flows:	Step	Actor action	System response	
	1	Librarian clicks Clear button	The system clears all scanned	

			books.
	2	Librarian clicks Cancel button	Confirm modal closes
Exceptions:	No.	Cause	System response
	1.1	Book is not found in the database	The system shows an error message.
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. Librarian scan to show book copy information before clicking Return button 2. Book copy information: Image, title, subtitle, edition, barcode, genre, borrower, due date and overdue day(s). 3. The RLMS allows users to return multiple book copies at a time. 4. Overdue book copies will require the patron to pay a fine <ul style="list-style-type: none"> • Fine = overdue day(s) * fine rate (fine rate is taken from <i>Fee Policy</i>). Overdue days does not include Saturdays and Sundays • Max fine = book copy's price * MAX OVERDUE FINE / 100 (MAX OVERDUE FINE is taken from <i>Fee Policy</i>) 5. A returned book copy status will be updated based on its book status. If the book status is: <ul style="list-style-type: none"> • IN_CIRCULATION update book copy status to AVAILABLE • OUT_OF_CIRCULATION update book copy status to OUT_OF_CIRCULATION • DISCARD update book copy status to DISCARD • LIB_USE_ONLY update book copy status to LIB_USE_ONLY 6. The librarian is not required to scan patron card when returning book copies 7. The version of Fee Policy used when returning is based on the version of the policy when the book copy is borrowed 8. Inactive patrons can still return book copies. 9. Reactivate security alarm for returned book copies 		
Other Information	<ol style="list-style-type: none"> 1. If there is no internet connection, the system cannot load the book image 		
Assumptions:	N/A		

Table 25 <Use Case> Return books for a patron

2.2.4.3. Add book

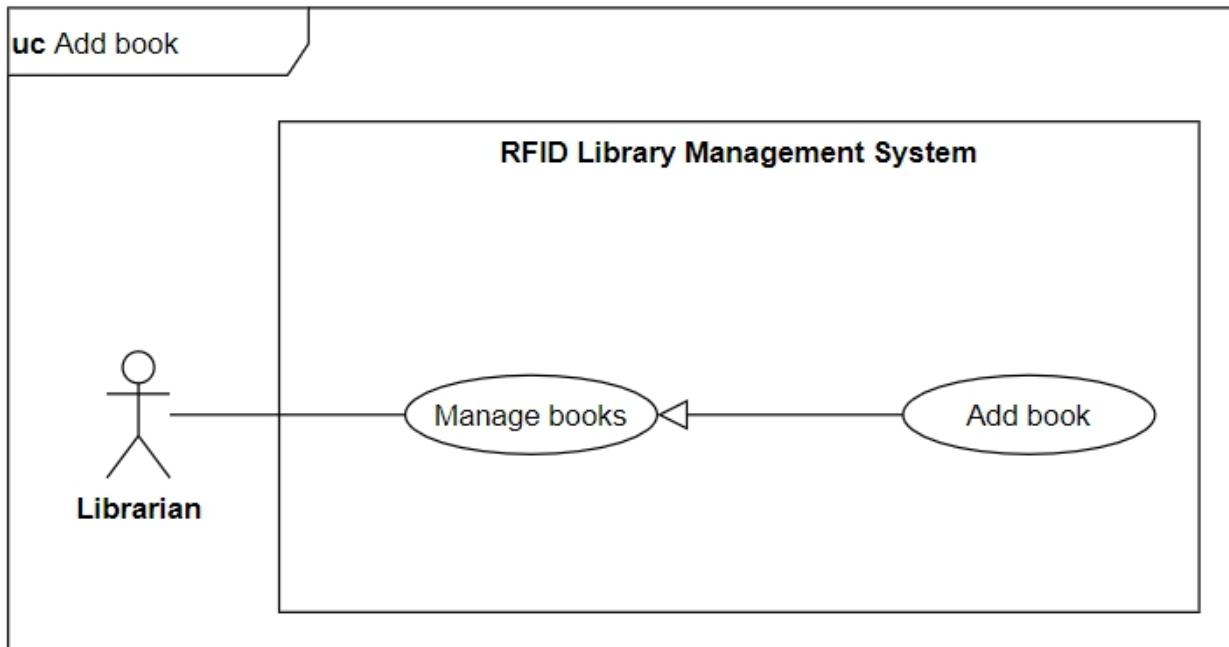


Figure 31 <Use Case> Add book

ID and Name:	UC_23 Add book		
Created By:	KhangNDN	Date Created:	18/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Librarian can add new book to the system		
Trigger:	The Librarian sends request to add new book		
Preconditions:	User has logged in as Librarian role		
Postconditions:	POST-1: Show success message POST-2: If add failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	Librarian requests to create new book by pressing "Add book" button	The RLMS requires librarian to fill book information

	2	The librarian fills require fields and click Save button [Alternative 1]	The RLMS processes the create transaction, then shows the success message. [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	Librarian clicks Cancel button	The add book modal closes
Exceptions:	No.	Cause	System response
	1	Data is invalidated	The system shows an error message.
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. Required to fill all of the following fields: <ul style="list-style-type: none"> • Image: image of the book's cover • ISBN: 10 or 13 digits, is unique • Title: the book title, 1-255 characters • Subtitle: the book subtitle, 1 - 255 characters • Publisher: the book publisher, 1-255 characters • Language: language of the book, 1-30 characters • Page number: book total pages, a positive integer <= 100000 • Call number: 1-50 characters • Edition: the book edition, number less than or equal to 1000 • Publish year: cannot enter years in the future • Author: Select from author list • Status: Out of circulation, In circulation, Library use only 2. Call number format: DDC + first 3 letters of author's last name + publish year (example: "100 ROW 2005"). 3. The RLMS will determine the genre of the book based on the entered DDC 		
Other Information	<ol style="list-style-type: none"> 1. If there is no internet connection, the system cannot save the book image 		

Assumptions:	<ol style="list-style-type: none"> 1. The RLMS does not manage authors, so we assume that there is already a list containing authors information in the database 2. Because call numbers are different from libraries to libraries, for this project we assume that the call number follows the format: "DDC + first 3 letters of author's last name + publish year" (example: "100 ROW 2005"). There can also be some special cases, for example: "100.123 JOH 2010b". (reference: link)
--------------	---

Table 26 <Use Case> Add book

2.2.4.4. Update book

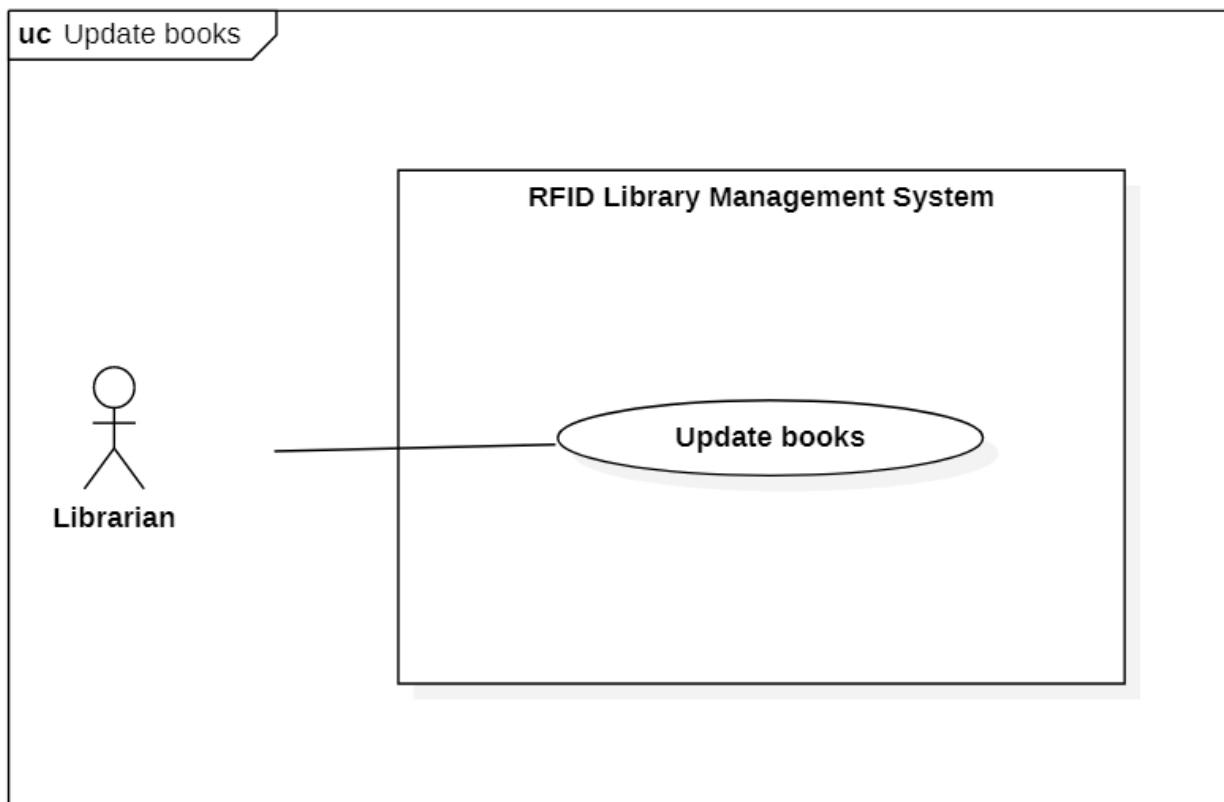


Figure 32 <Use Case> Update book

ID and Name:	UC_24 Update book		
Created By:	KhangNDN	Date Created:	18/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Librarian can update existing book in the system		

Trigger:	The Librarian sends request to update existing book		
Preconditions:	PRE-1: User has logged in as Librarian role		
Postconditions:	POST-1: If update succeeded, show success message POST-2: If update failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	Librarian requests to update existing book by pressing "Update book" button	The RLMS requires librarian to fill book update information
	2	The librarian fills update fields and click Save button [Alternative 1]	The RLMS processes the update transaction, then shows the success message. [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	Librarian clicks Cancel button	The update book modal closes
Exceptions:	No.	Cause	System response
	1	Data is invalidated	The system shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Book information requires following fields: <ul style="list-style-type: none"> • Image: image of the book's cover • ISBN: 10 or 13 digits, is unique • Title: the book title, 1-255 characters • Subtitle: the book subtitle, 1 - 255 characters • Publisher: the book publisher, 1-255 characters • Language: language of the book, 1-30 characters 		

	<ul style="list-style-type: none"> ● Page number: book total pages, a positive integer <= 100000 ● Call number: 1-50 characters ● Edition: the book edition, number less than or equal to 1000 ● Publish year: cannot enter years in the future ● Author: Select from author list ● Status: Out of circulation, In circulation, Library use only <ol style="list-style-type: none"> 2. Call number format: DDC + first 3 letters of author's last name + publish year (example: "100 ROW 2005") 3. The RLMS will determine the genre of the book based on the entered DDC 4. If the book's status is changed then update the book copies: <ul style="list-style-type: none"> ● If the book's status is changed to "IN_CIRCULATION" or "OUT_OF_CIRCULATION" or "LIB_USE_ONLY" then change its copies status, where the copy is "AVAILABLE" or "LIB_USE_ONLY" or "OUT_OF_CIRCULATION", to the corresponding book's status. Copies that are "BORROWED" will be updated to the corresponding book's status when it is returned. ● If the book's status is changed to "DISCARD" then change its copies status, where the copy status is "AVAILABLE" or "LIB_USE_ONLY" or "IN_PROCESS" or "OUT_OF_CIRCULATION", to "DISCARD".
Other Information	<ol style="list-style-type: none"> 1. If there is no internet connection, the system cannot update the book image
Assumptions:	N/A

Table 27 <Use Case> Update book

2.2.4.5. Tag a book copy

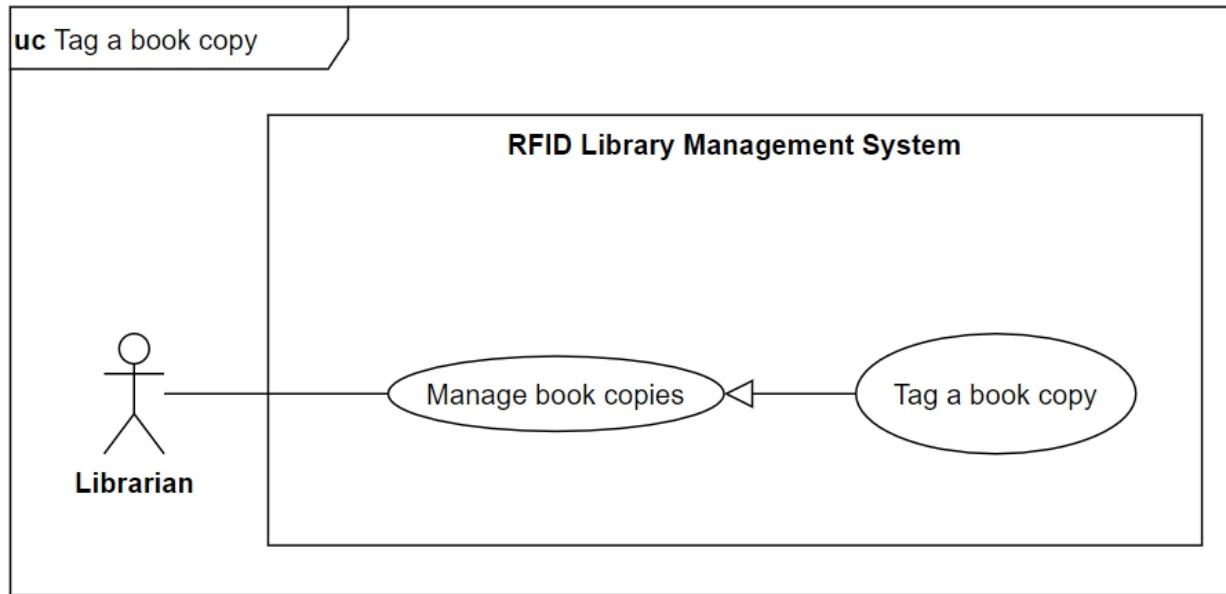


Figure 33 <Use Case> Tag a book copy

ID and Name:	UC_25 Tag a book copy		
Created By:	HoangPM	Date Created:	22/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	In order to be in circulation, a book must be tagged with an RFID tag by the Librarian		
Trigger:	The Librarian sends request to tag a book copy		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: Show success message POST-2: If update failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	Librarian requests to update existing book	The system requires librarian to enter book copy's barcode and new RFID UID

	2	The librarian fills in required fields and click Confirm button	The system processes the request transaction, then shows the success message. [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	There's already a book copy tagged with the same RFID code (Duplicate RFID code)	The RLMS returns a warning message, telling the librarian that the RFID code must be unique
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. Book information must be shown after the librarian has entered the book copy's barcode. 2. Both barcode and RFID fields must be filled. 3. After being tagged, book copy's status will be updated depends on Book's status: <ul style="list-style-type: none"> • If Book's status is "IN_CIRCULATION", book copy's status is "AVAILABLE" • If Book's status is "OUT_OF_CIRCULATION", book copy's status is "OUT_OF_CIRCULATION" • If Book's status is "LIB_USE_ONLY", book copy's status is "LIB_USE_ONLY" 4. Book copy RFID UID is unique 5. Barcode can be inputted by scanning or entering with keyboard manually 6. Can only tag book copy if the copy is "IN_PROCESS" or "AVAILABLE" 		
Other Information	<ol style="list-style-type: none"> 1. The librarian must have the UHF reader to complete the tagging process. 		
Assumptions:	N/A		

Table 28 <Use Case> Tag a book copy

2.2.4.6. Create book copies

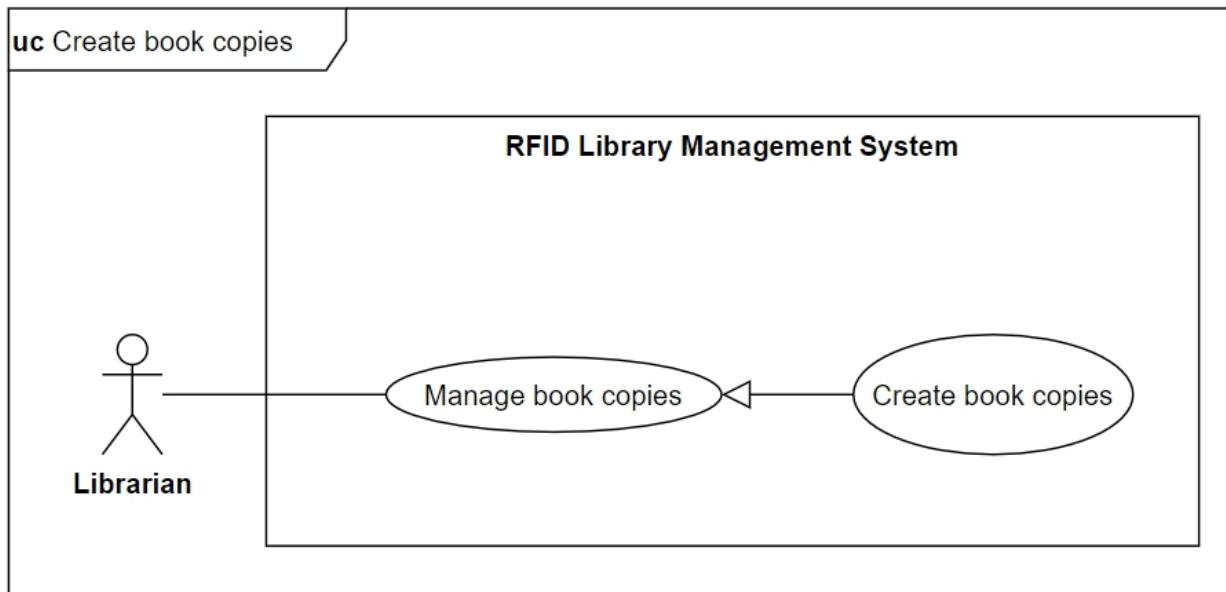


Figure 34 <Use Case> Create book copies

ID and Name:	UC_26 Create book copies		
Created By:	HoangPM	Date Created:	22/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to create multiple book copies for a book		
Trigger:	The Librarian sends request to create copies for a book title		
Preconditions:	PRE-1: The user has logged in as Librarian role or Manager role PRE-2: The book is already in the system PRE-3: The book copy types are already in the system		
Postconditions:	POST-1: Show success message POST-2: If update failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	Librarian request to add copies for a book by pressing "Make copy" button	The system returns a screen requiring the librarian to enter the book copy type, price and total

			number of adding copies.
	2	The librarian fills in required fields and click on Save button	The system generates a barcode for each copy, then shows the generated barcodes to the librarian for confirmation.
	3	The librarian re-check the information then click on Confirm button	The system processes the request transact, auto start to download a generated barcodes file on the user's browser and show success message
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. The barcode is 14 digits: <ul style="list-style-type: none"> • 1st to 2nd digits are Book copy type • 3rd to 6th digits are library's ID • 7th to 15th digits are book copy's ID 2. Book copy barcodes are unique 3. Book copy status after adding is "IN_PROCESS" 4. The function requires the librarian to enter the following fields: <ul style="list-style-type: none"> • Price: required, min = 1 000, max = 1 000 000 000 • Price Note: details about the book copy price (like if the shipping fee is included or not). If the user does not input anything, a default note will be inserted to the RLMS. • Number of copies: required, min = 1, max = 5000 • Copy type: required, must be selected from a list • Price note: required, max = 500 5. The total number of copies of the book will be increased by the number of copies that will be added 6. The system generates barcode base on Code-39 barcode format 7. The pdf file name has format: ISBN-Book Copy Type-Price 		
Other Information	<ol style="list-style-type: none"> 1. The barcode label has the following dimension: 2.25" x 1" 		

	<p>2. While already having RFID tags, we still keep the barcode system as a fail-safe, in case the RFID tag is broken.</p>
Assumptions:	<p>1. Barcodes format is different from libraries to libraries. For this project, the barcode is 14 digits:</p> <ul style="list-style-type: none"> • 1st to 2nd digits are Book copy type • 3rd to 6th digits are library's ID • 7th to 15th digits are book copy's ID <p>2. The RLMS does not manage the barcode printers, so we assume that the library is equipped with printers according to our barcode label dimension. (We recommend this printer: Zebra GX430t Thermal Transfer Desktop Printer, and this barcode label: BETCKEY - 2.25" x 1" label)</p>

Table 29 <Use Case> Create book copies

2.2.4.7. Update a book copy

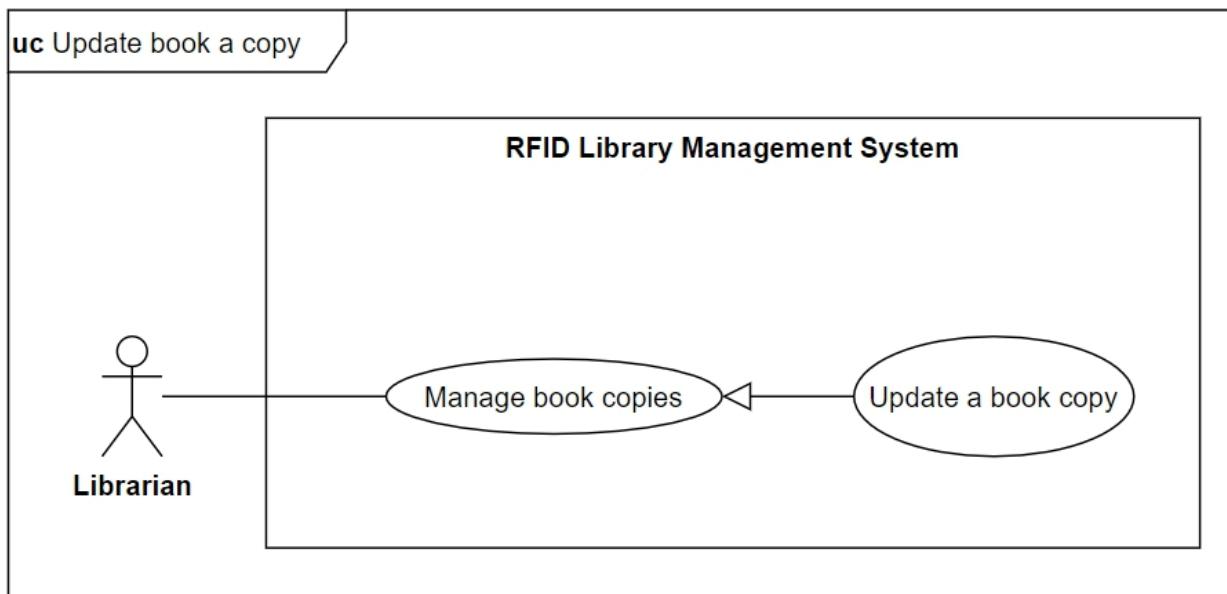


Figure 35 <Use Case> Update a book copy

ID and Name:	UC_27 Update a book copy		
Created By:	HoangPM	Date Created:	22/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to update some of the book copy's		

	information after it has been added		
Trigger:	The Librarian sends request to update information of a book copy		
Preconditions:	PRE-1: The user has logged in as Librarian role PRE-2: The book copy is already in the system PRE-3: The book copy types are already in the system		
Postconditions:	POST-1: Show success message POST-2: If update failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	Librarian request to update a book copy by pressing the “Edit” button	The system returns a dialogue requiring the librarian to enter the book copy's type, price and RFID code.
	2	The librarian fills in required fields and click on Save button [Alternative 1]	The system processes the request transaction, then shows the success message. [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	Librarian clicks Cancel button	The Edit Copy dialogue closes.
Exceptions:	No.	Cause	System response
	1	There's already a book copy tagged with the same RFID code (Duplicate RFID code)	The RLMS returns a warning message, telling the librarian that the RFID code must be unique
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	1. Book copy update requires the librarian to enter the following fields: <ul style="list-style-type: none"> • Price: required • RFID code: required, unique • Copy type: required, must be selected from a list 		

	<ol style="list-style-type: none"> 2. RFID code can be inputted by scanning or entering with keyboard manually 3. The function requires the librarian to enter the following fields: <ul style="list-style-type: none"> • Price: required, min = 1 000, max = 1 000 000 000 • Number of copies: required, min = 1, max = 50 • Copy type: required, must be selected from a list 4. If book copy status is “IN_PROCESS” then update book copy status in correspond with its book status: <ul style="list-style-type: none"> • If Book’s status is “IN_CIRCULATION”, book copy’s status is “AVAILABLE” • If Book’s status is “OUT_OF_CIRCULATION”, book copy’s status is “OUT_OF_CIRCULATION” • If Book’s status is “LIB_USE_ONLY”, book copy’s status is “LIB_USE_ONLY” 5. Cannot update copies whose status is “DISCARD” or “LOST” or “BORROWED”
Other Information	N/A
Assumptions:	N/A

Table 30 <Use Case> Update a book copy

2.2.4.8. View patron policy

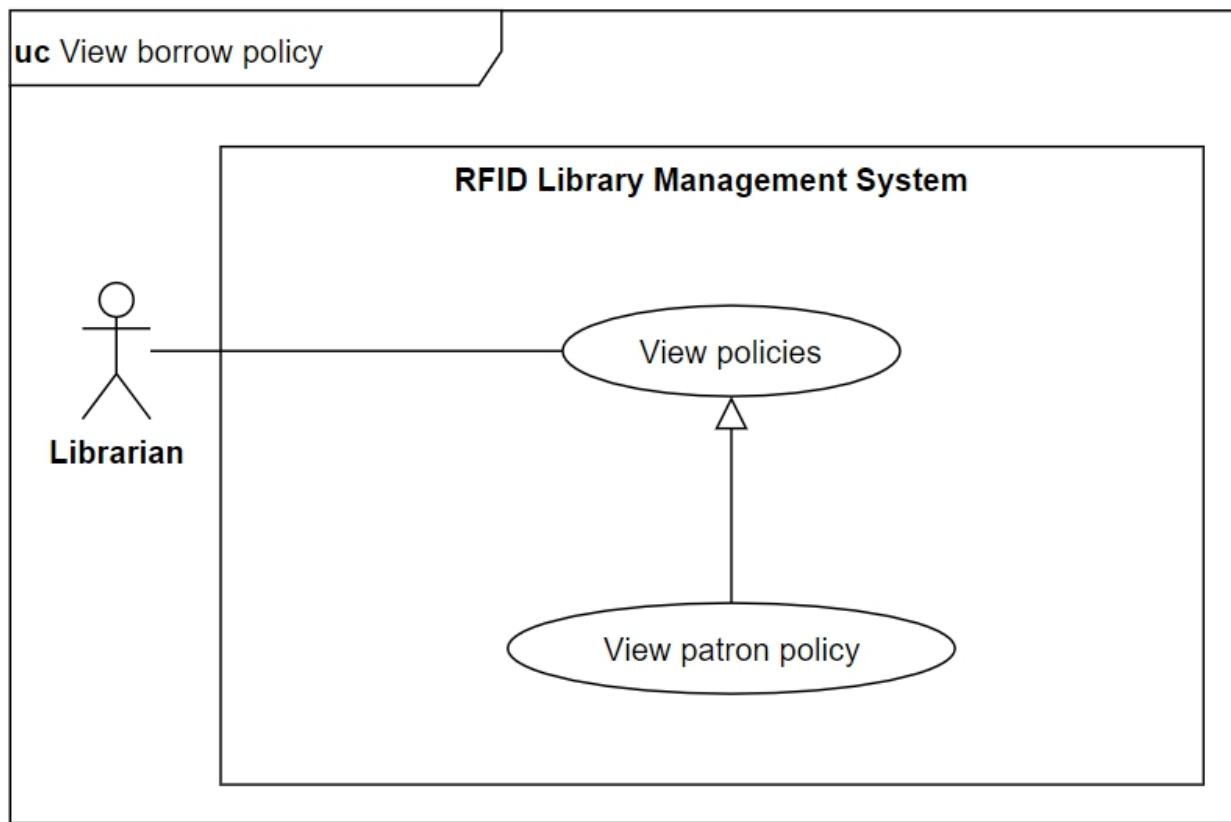


Figure 36 <Use Case> View patron policy

ID and Name:	UC_28 View patron policy		
Created By:	HoangPM	Date Created:	24/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to view Patron Policy		
Trigger:	The Librarian sends request to view the policy		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: RLMS shows a table containing the Patron Policy		
Normal Flow:	Step	Actor action	System response

	1	The Librarian sends request to view the Patron Policy [Exception 1]	The RLMS processes the request, then shows a table containing the Patron Policy
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. Each records in the Patron Policy table contains: <ul style="list-style-type: none"> • Patron Type: Patron type name • Total checkouts allowed: Maximum number book copies can be borrowed by the corresponding patron type (editable) 2. Patron type name is unique 		
Other Information	N/A		
Assumptions:	N/A		

Table 31 <Use Case> View patron policy

2.2.4.9. View borrow policy

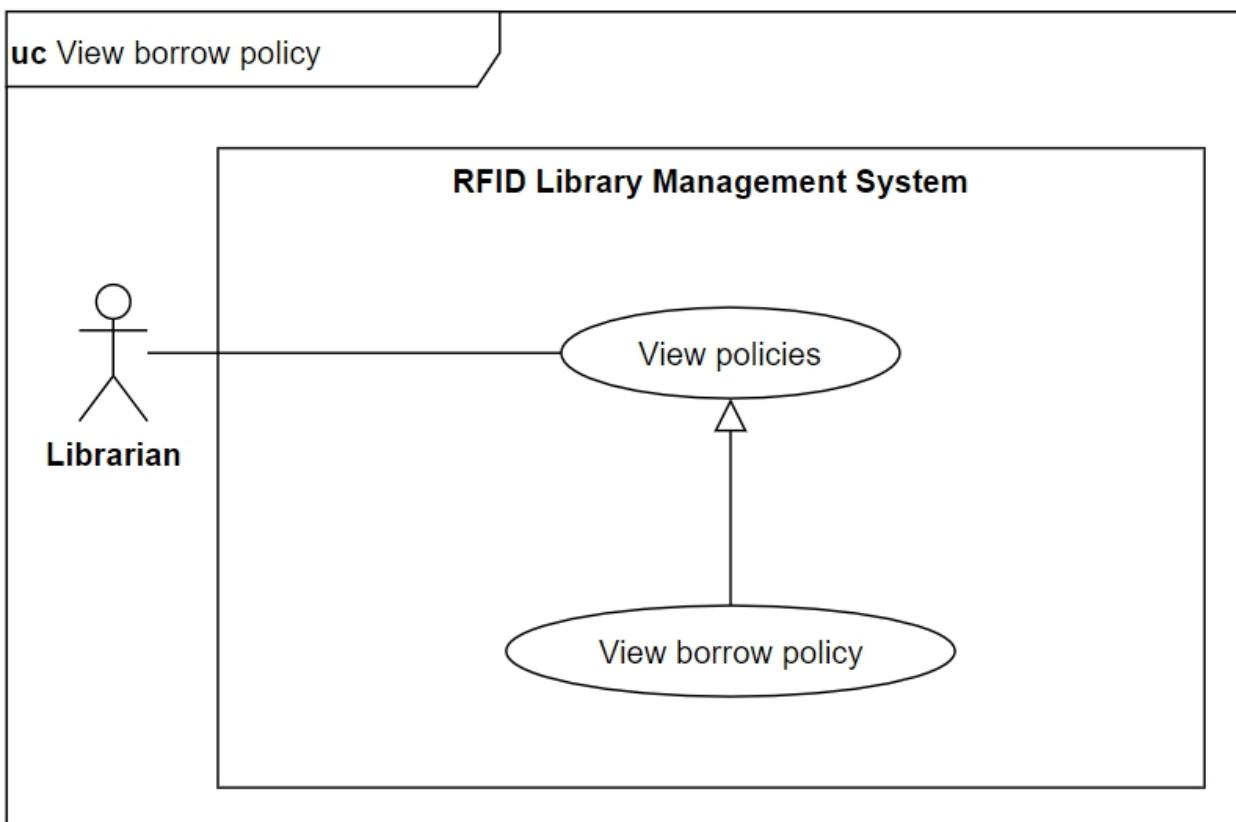


Figure 37 <Use Case> View borrow policy

ID and Name:	UC_29 View borrow policy		
Created By:	TramPH	Date Created:	24/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to view <i>Borrow Policy</i> . It is for showing which patron type can borrow which book copy type.		
Trigger:	The Librarian sends request to view the borrow policy		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: RLMS shows a list of borrow policy.		
Normal Flow:	Step	Actor action	System response

	1	The librarian clicks the “Borrow” tab on the Policy page. [Exception 1]	RLMS shows a list of borrow policies.
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. Borrow policy can be filtered by PATRON TYPE and/or BOOK COPY TYPE 2. Each borrow policy shows with appropriated patron type and book copy type: <ul style="list-style-type: none"> • PATRON TYPE • BOOK COPY TYPE • BORROW PERIOD: how many days a patron can keep a borrowed book • CHECKOUTS ALLOWED: maximum number of copy a patron can borrow at a time • RENEWALS ALLOWED: maximum time a patron can renew a book • RENEWAL PERIOD: the time period that will be added to the current due date of a renewing book copy. • Update button • Remove button 3. Each couple of PATRON TYPE and BOOK COPY TYPE is 1 borrow policy. For example: 1) STUDENT - RARE BOOK, 2) STUDENT - REFERENCE BOOK, 3) LECTURER - REFERENCE BOOK, etc. 4. Every couple of PATRON TYPE and BOOK COPY TYPE is unique. 5. If the system does store a borrow policy for a patron type and a book copy type, then that patron type cannot borrow book copies from that book copy type. 		
Other Information	N/A		
Assumptions:	N/A		

Table 32 <Use Case> View borrow policy

2.2.4.10. View fee policy

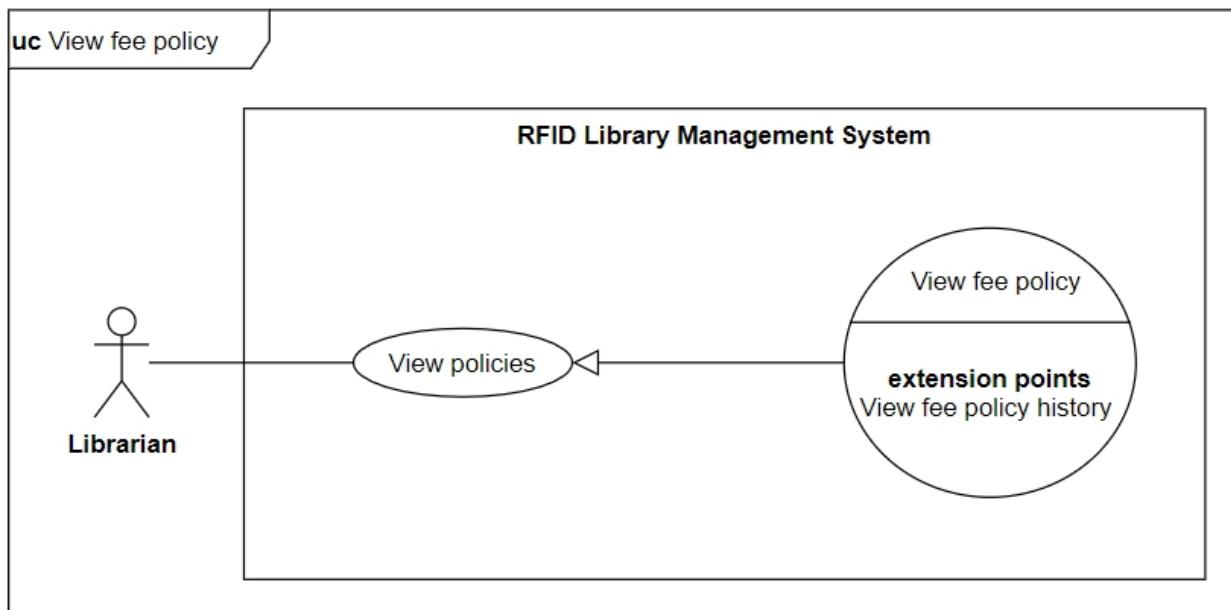


Figure 38 <Use Case> View fee policy

ID and Name:	UC_30 View fee policy		
Created By:	HoangPM	Date Created:	24/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to view Fee Policy		
Trigger:	The Librarian sends request to view the fee policy		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: RLMS shows a table containing the Fee Policy		
Normal Flow:	Step	Actor action	System response
	1	The Librarian sends request to view the Fee Policy	The RLMS processes the request, then shows a table containing the Fee Policy

Alternative Flows:	N/A
Exceptions:	N/A
Priority:	Low
Frequency of Use:	Occasionally
Business Rules:	<ol style="list-style-type: none"> 1. Each records in the Fee Policy table contains: <ul style="list-style-type: none"> • OVERDUE FINE PER DAY: The fine rate (VND) • MAX OVERDUE FINE: The percentage of the book copy price (as recorded in the system) (max = 100%). This is used to calculate the overdue fine • MISSING DOCUMENT MULTIPLIER: Is used to calculate lost book copy fine in case the lost book is no longer available to re-purchase • DOCUMENT PROCESSING FEE: The fee required to process the lost report, replace the lost book, etc. • CREATED AT: the day the Fee Policy was created and applied 2. The Fee Policy is applied right after it is created 3. Max overdue fine = book copy price * MAX OVERDUE FINE 4. Lost book copy fee is calculated as follow: <ul style="list-style-type: none"> • If the book copy is still sold on the market: Fine = The book copy price (as recorded in the system) + DOCUMENT PROCESSING FEE • If the book copy is no longer sold on the market: Fine = The book copy price (as recorded in the system) * MISSING DOCUMENT MULTIPLIER
Other Information	N/A
Assumptions:	N/A

Table 33 <Use Case> View patron policy

2.2.4.11. View patron checkout information

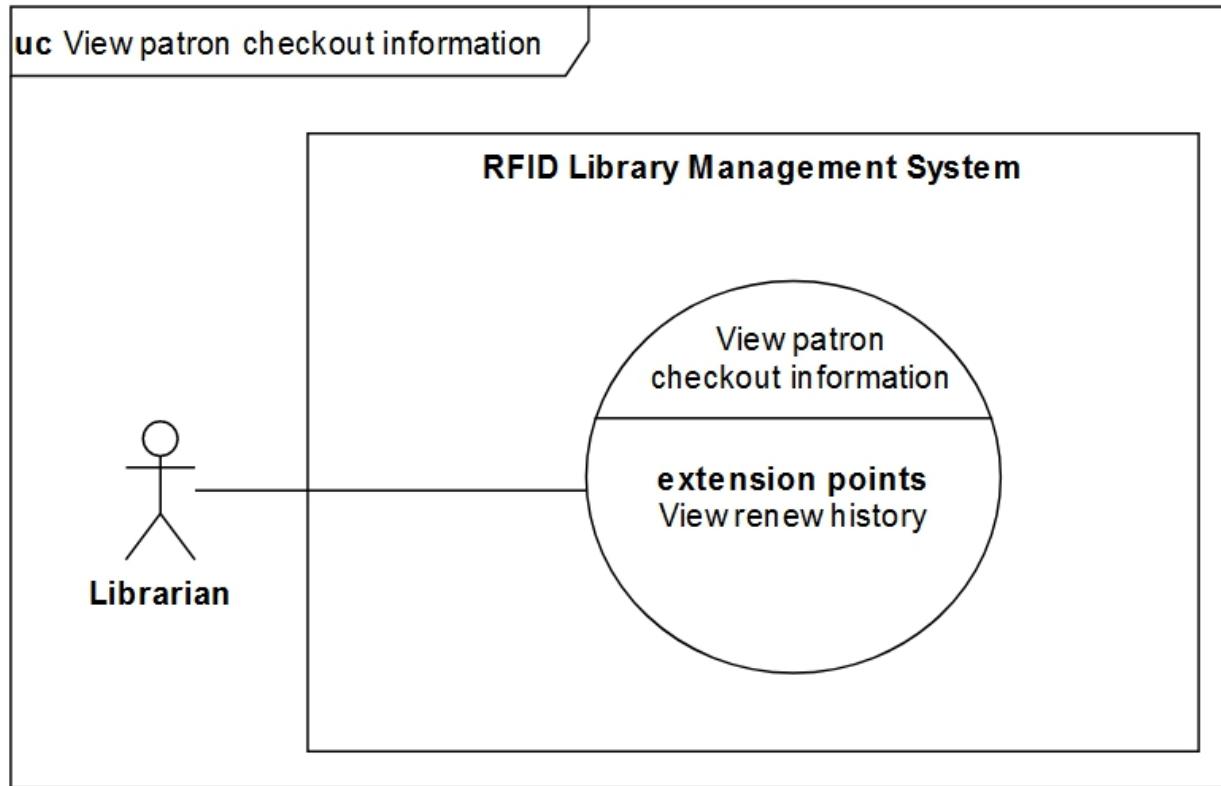


Figure 39 <Use Case> View patron checkout information

ID and Name:	UC_31 View patron checkout information		
Created By:	HoangPM	Date Created:	23/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to view a patron's currently borrowed books		
Trigger:	The Librarian sends request to view patron's currently borrowed books		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: RLMS displays the patron's checkout information		
Normal Flow:	Step	Actor action	System response

	1	The Librarian sends request to view patron's checkout information	The RLMS processes the request, then shows a list of currently borrowed books, a list of overdue books, a list of returned books and a list of lost books of the patron
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. The returned book list and overdue book list contains: <ul style="list-style-type: none"> • Book information: title, subtitle, edition • Borrowed time • Returned time • Overdue days • Fine (VND) • Renew history button 2. The borrowing book list (sorted by when the book copy was borrowed, first record is the latest borrowed book copy.), contains: <ul style="list-style-type: none"> • Book information: title, subtitle, edition • Borrowed time • Due date • Renew button • Renew history button 3. The overdue book list (sorted by when the book copy was overdue, first record is the book copy with the most overdue days), contains: <ul style="list-style-type: none"> • Book information: title, subtitle, edition • Borrowed time • Returned time • Overdue days • Fine (VND) • Renew history button 4. The lost book list (sorted by lost date, most recent first) contains: <ul style="list-style-type: none"> • Book information: title, subtitle, edition • Borrowed at • Lost at • Fine (VND) • Status (Pending or Confirmed) 5. Overdue days excludes Saturdays and Sundays 6. Overdue fine is calculated based on fee policy at the time the user borrowed the books 		

	<ul style="list-style-type: none"> ● Fine = overdue days * fine rate ● Max fine = book copy's price * max overdue fine percentage / 100 <p>7. Lost book fine is calculated based on fee policy at the time the user borrowed the books:</p> <ul style="list-style-type: none"> ● If the book copy is still sold on the market: Fine = The book copy price (as recorded in the system) + DOCUMENT PROCESSING FEE ● If the book copy is no longer sold on the market: Fine = The book copy price (as recorded in the system) * MISSING DOCUMENT MULTIPLIER <p>8. To be in the borrowing list, the book copy must be borrowed by the patron account, and the due date is in the future.</p> <p>9. To be in the overdue list, the book copy must be borrowed by the patron account, and the due date belongs to the past (from the day before the current date backward).</p> <p>10. To be in the returned list, the book copy must be borrowed by the patron account, and has been returned to the library (self-returned or returned at the librarian desk).</p> <p>11. To be in the lost list, the book must be reported by the patron as lost. After being reported by the patron the lost book status is "Pending" until it is confirmed by the manager and updated to "Confirmed".</p>
Other Information	N/A
Assumptions:	N/A

Table 34 <Use Case> View patron currently borrowed books

2.2.4.12. View renew history

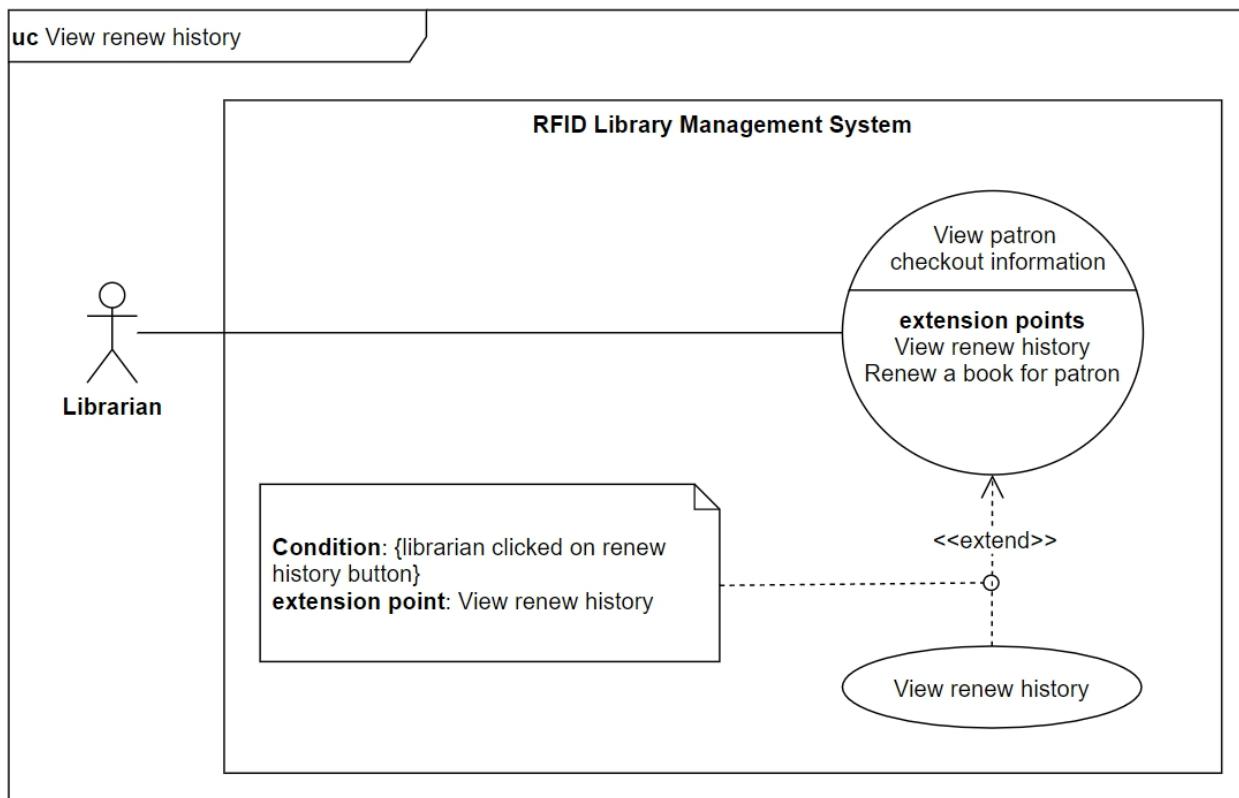


Figure 40 <Use Case> View renew history

ID and Name:	UC_18 View renew history		
Created By:	HoangPM	Date Created:	23/02/2021
Primary Actor:	Patron, Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to view the patron's renew history of a book copy		
Trigger:	The Librarian sends request to view patron's renew history		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: RLMS shows renew history record of a patron for a book copy		
Normal Flow:	Step	Actor action	System response

	1	From the Checkout information of a patron, the librarian sends request to view the patron's renew history of a book	The RLMS processes the request, then shows records of the renew history.
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. A record in the renew history contains the following: <ul style="list-style-type: none"> • Index number • Renew date • New due date 2. The records will be sorted by when the book copy was renewed. From the first renewal to the latest renewal. 3. Only show the history of the current borrow 		
Other Information	N/A		
Assumptions:	N/A		

Table 35 <Use Case> View renew history

2.2.4.13. Renew a book for a patron

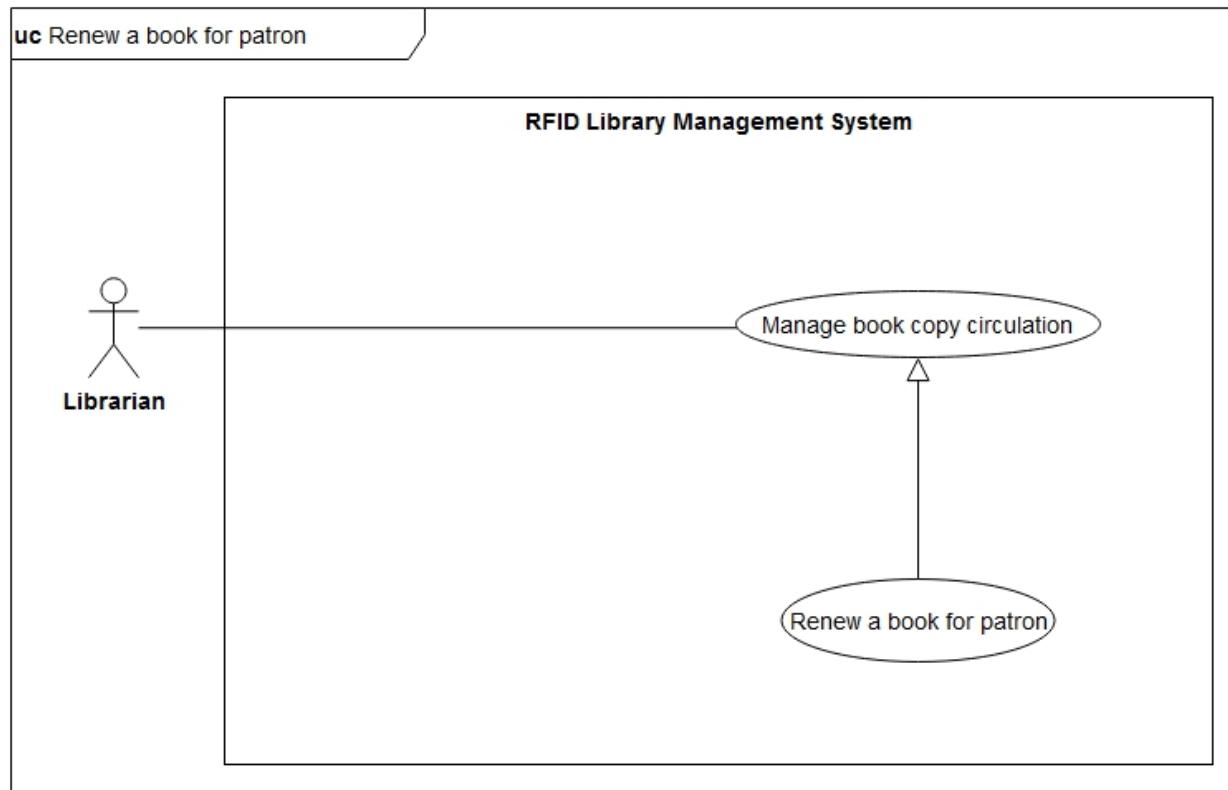


Figure 41 <Use Case> Renew a book for a patron

ID and Name:	UC_32 Renew a book for a patron		
Created By:	HoangPM	Date Created:	23/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to renew a borrowed book copy for the patron		
Trigger:	The Librarian sends request to view patron's renew history		
Preconditions:	PRE-1: The user has logged in as Librarian role PRE-2: The book is in borrowing books		
Postconditions:	POST-1: If the patron has returned any book copies, RLMS will show a table containing the patron's returned book copies		
Normal Flow:	Step	Actor action	System response

	1	The Librarian sends request to view patron's checkout information	The RLMS returns a screen requiring the librarian to enter the patron's email or scan the patron card
	2	Librarian clicks on the Search button	The RLMS processes the request, then shows records of the patron's checkout information. [Exception 1]
	3	Librarian clicks on the Renew button on a record [Alternative 1]	The RLMS processes the request, then shows a confirmation dialog with a new due date. [Exception 2] [Exception 3] [Exception 4]
	4	Librarian clicks on the Confirm button [Alternative 2]	The system processes the request transaction, then shows the success message then returns to the Checkout Information screen.
	Step	Actor action	System response
Alternative Flows:	1.1	Librarian clicks on the Renew button on a record. And the patron is keeping overdue book(s) or the patron is exceeding renewal allowance.	The RLMS processes the request, then shows a confirmation dialog with a new due date, policy violation message and a text box to enter the reason for letting the patron renew.
	1.2	The librarian enters the reason and clicks on the Confirm button [Alternative 2]	The system processes the request transaction, then shows the success message then returns to the Checkout Information screen.
	2.1	The Librarian clicks on Back button	The RLMS closes the confirmation dialog and return to the Checkout Information screen
	No.	Cause	System response
Exceptions:	1	The patron the librarian is searching for is not in the	The RLMS returns a message saying the patron does not exist

		RLMS.	
	2	The patron is keeping overdue books.	RLMS shows error messages.
	3	The patron violates the borrow policy	RLMS shows error messages.
	4	The patron is not allowed to borrow this book copy type anymore.	RLMS shows error messages.
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. If the patron is not keeping any overdue book copies and not violating any Borrow Policy, then the confirmation dialog will contain: 1) New due date. 2) Confirm button. 3) Back button. 2. If the patron is keeping any overdue book copies or violating any Borrow Policy, then the confirmation dialog will contain: 1) New due date. 2) Reason text box. 3) Confirm button. 4) Back button 3. The librarian has to provide a reason if the patron is keeping any overdue book copies or violating any Borrow Policy. 4. Cannot renew if the patron type is no longer allowed to borrow the book copy type 5. New due date = old due date + extend due duration (extend due duration is in the Borrow Policy) 		
Other Information	N/A		
Assumptions:	N/A		

Table 36 <Use Case> Renew a book for a patron

2.2.4.14. View fee policy history

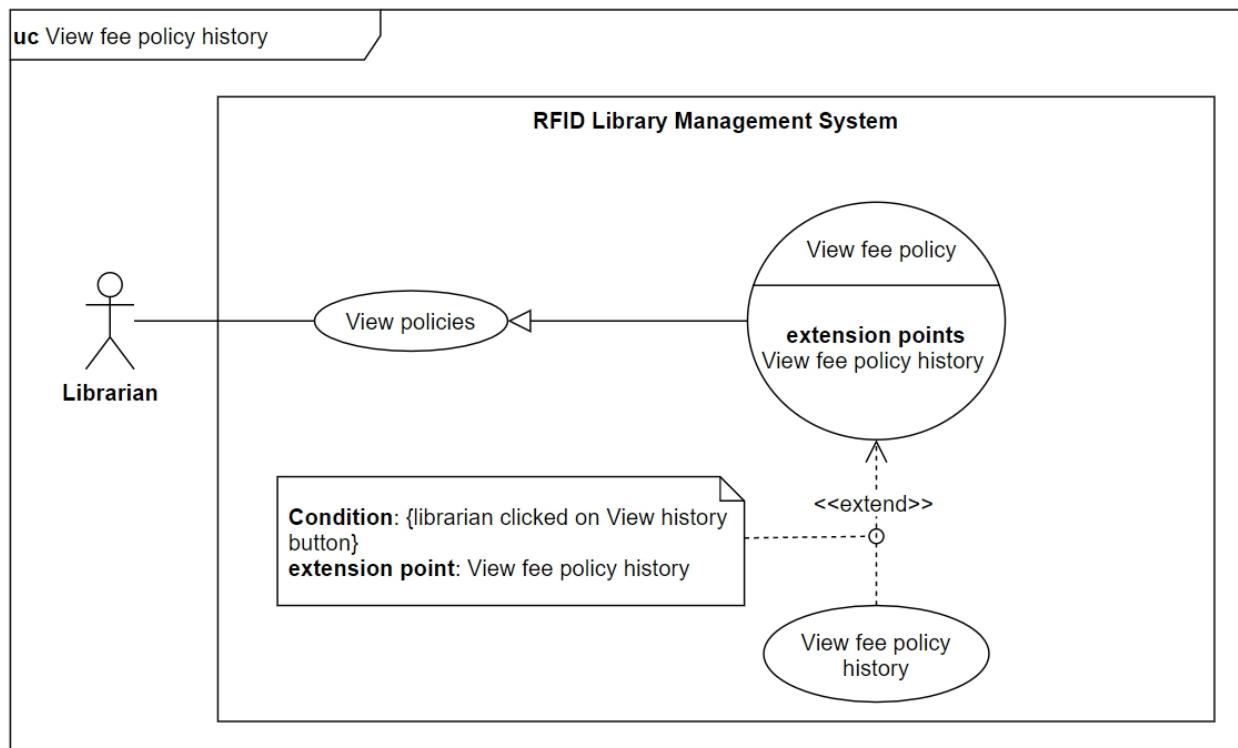


Figure 42 <Use Case> View fee policy history

ID and Name:	UC_33 View fee policy history		
Created By:	HoangPM	Date Created:	24/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to view Fee Policy history		
Trigger:	The Librarian sends request to view the fee policy history		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: RLMS shows a table containing the Fee Policy history		
Normal Flow:	Steps	Actor action	System response
	1	The Librarian sends request to view the Fee Policy	The RLMS processes the request, then shows a table containing the Fee Policy history

	2	Librarian clicks on the View history button	The RLMS processes the request, then shows records of the Fee Policy history
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Each records in the Fee policy history table contains: <ul style="list-style-type: none"> • OVERDUE FINE PER DAY: The fine rate (VND) • MAX OVERDUE FINE: The percentage of the book copy price (as recorded in the system) (max = 100%). This is used to calculate the overdue fine • MISSING DOCUMENT MULTIPLIER: Is used to calculate lost book copy fine in case the lost book is no longer available to re-purchase • DOCUMENT PROCESSING FEE: The fee required to process the lost report, replace the lost book, etc. • CREATED AT: the day the Fee Policy was created and applied 2. The table is sorted descending by CREATED AT (latest date first) 3. The first record is the current Fee Policy 		
Other Information	N/A		
Assumptions:	N/A		

Table 37 <Use Case> View fee policy history

2.2.4.15. View book copies

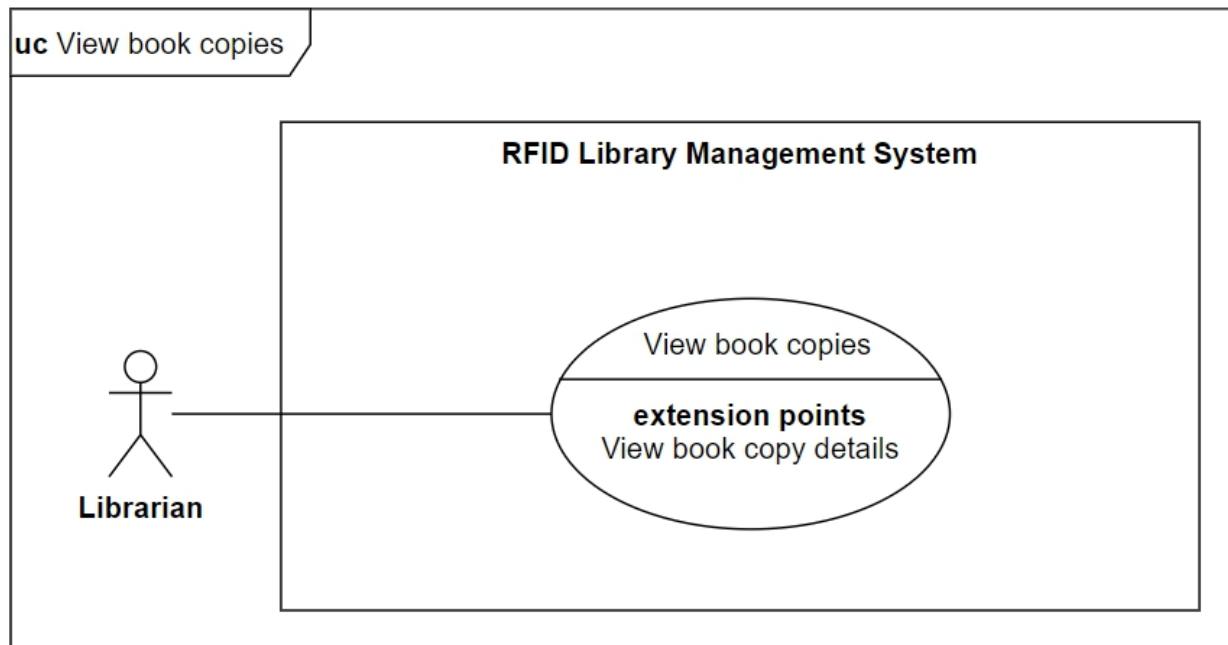


Figure 43 <Use Case> View book copies

ID and Name:	UC_34 View book copies		
Created By:	HoangPM	Date Created:	24/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to view the list of book copies		
Trigger:	The Librarian sends request contain the search query to view book copies		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: RLMS shows a table containing list of book copies POST-2: If there are no record, shows a message saying there is no book copy to show		
Normal Flow:	Step	Actor action	System response
	1	The Librarian sends request to view book copies [Exception 1]	The RLMS processes the request, then returns the search book copy screen

	2	The Librarian enter the search query and clicks on the Search button [Exception 1]	The RLMS processes the request, then shows a table containing list of book copies
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Medium		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. Book copies can be searched by: <ul style="list-style-type: none"> • Like title • Like subtitle • Equals ISBN • Equals Barcode • Equals RFID 2. Book copies can be filtered by book copy status (Available, In process, Borrowed, Out of circulation, Lost, Library use only) 		
Other Information	N/A		
Assumptions:	N/A		

Table 38 <Use Case> View book copies

2.2.4.16. View book copy details

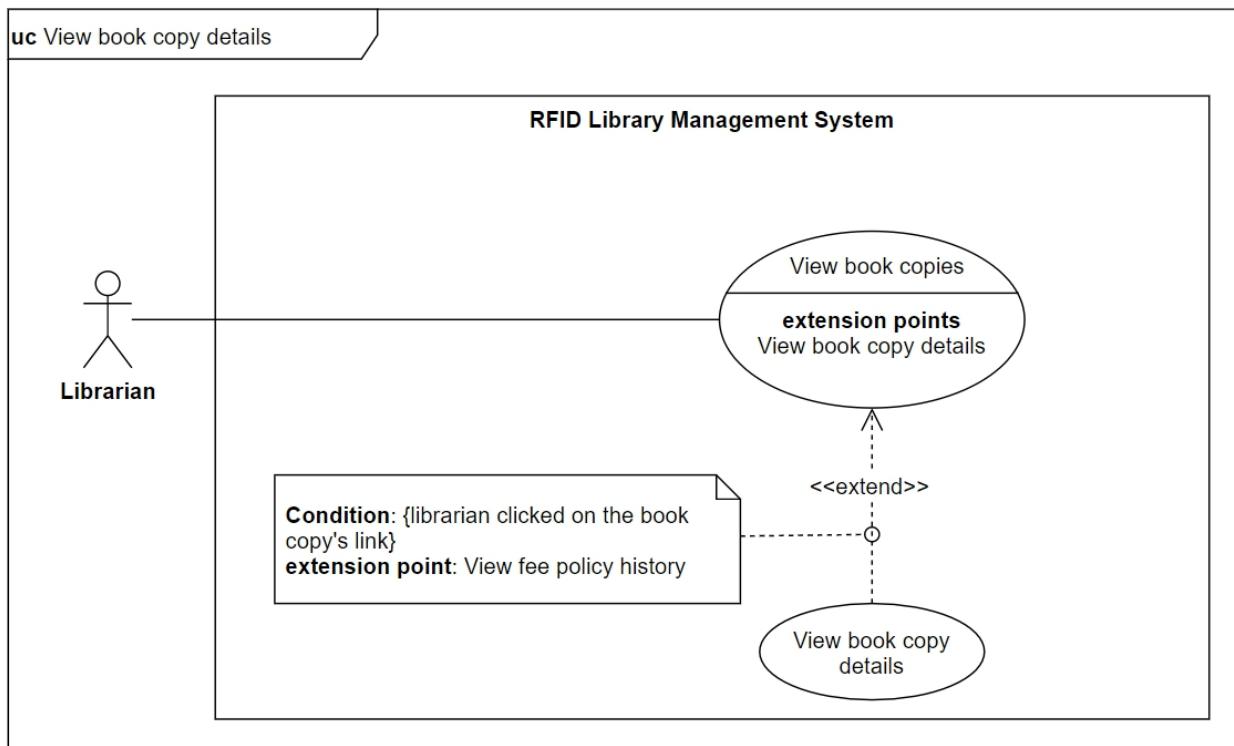


Figure 44 <Use Case> View book copy details

ID and Name:	UC_35 View book copy details		
Created By:	HoangPM	Date Created:	24/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Provide the librarian with the ability to view details of a book copy		
Trigger:	The Librarian sends request to view a book copy details		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: RLMS shows details of a book copy		
Normal Flow:	Step	Actor action	System response
	1	The Librarian sends a request to view a book copy details by clicking on a book copy's url in	The RLMS processes the request, then returns details of a book copy

		the view book copies screen.	
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Usually		
Business Rules:	1. Book copy details contains: <ul style="list-style-type: none"> ● Book copy image ● Title ● Subtitle ● Authors ● ISBN ● Edition ● Publisher ● Number of page ● Language ● Genres ● Barcode ● Call Number ● Book copy location (shelf's name and number of row) ● Book copy type ● Price ● Status 		
Other Information	N/A		
Assumptions:	N/A		

Table 39<Use Case> View book copy details

2.2.4.17. Add books to wishlist for a patron

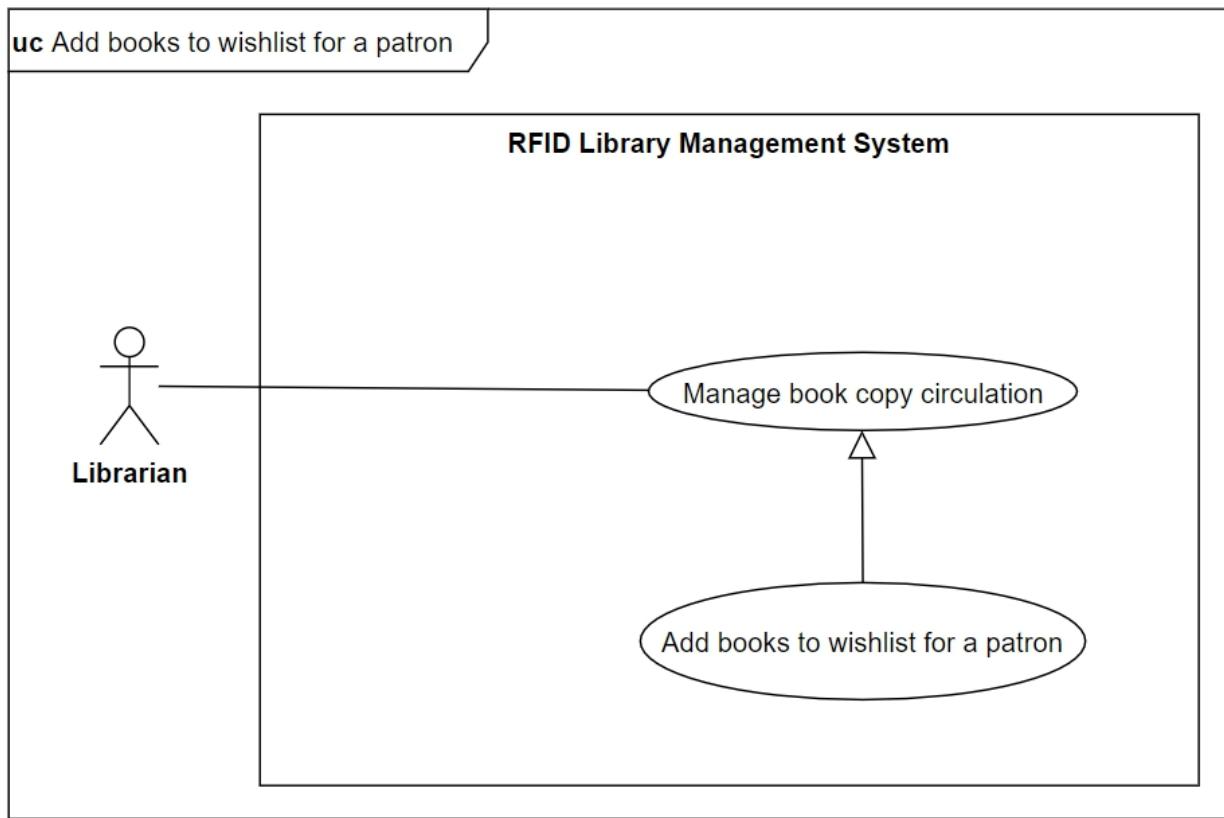


Figure 45 <Use Case> Add books to wishlist for a patron

ID and Name:	UC_36 Add books to wishlist for a patron		
Created By:	TramPH	Date Created:	25/01/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Let the librarian add an unavailable book to a patron's wishlist.		
Trigger:	The librarian sends a request to add a book to a patron's wishlist.		
Preconditions:	PRE-1: The Librarian has logged in successfully as a Librarian role. PRE-2: The book is unavailable in the library.		
Postconditions:	POST-1 The book is added to the patron's wishlist and shows a success message.		
Normal Flow:	Step	Actor action	System response

	1	The librarian clicks on the “Add to wishlist” button next to an available book from the books screen.	RLMS show modal asks the librarian to input the patron's RFID or email.
	2	The librarian inputs the patron's email or RFID and clicks the “Search” button. [Alternative 1]	RLMS returns patron information. [Exception 1] [Exception 2]
	3	The librarian clicks on the “Confirm” button. [Alternative 1] [Alternative 2]	RLMS adds the book to the patron's wishlist. [Exception 3]
Alternative Flows:	Step	Actor action	System response
	1.1	The librarian clicks on the “Cancel” button.	RLMS closes add to wishlist form.
	2.1	The librarian clicks on the “Reset” button.	RLMS deletes found patron information and returns to normal flow at step 2.
Exceptions:	No.	Cause	System response
	1	The system cannot find patron information.	RLMS does not show patron information.
	2	The patron is inactive	RLMS shows the patron information and inactive patron message.
	3	The patron has added that book to his/her wishlist already.	RLMS shows an error message.
Priority:	Medium		
Frequency of Use:	Occasionally		

Business Rules:	<ol style="list-style-type: none"> 1. The librarian can only add the following books into wish list: <ul style="list-style-type: none"> o Books with status “IN_CIRCULATION” have no copy with status “AVAILABLE” 2. The patron cannot have the same book in the wishlist. 3. The system will notify the patron by email when the book is available in the library. 4. The librarian cannot add a book to an inactive patron’s wishlist.
Other Information	N/A
Assumptions:	N/A

Table 40 <Use Case> Add books to wishlist for a patron

2.2.4.18. View lost book reports

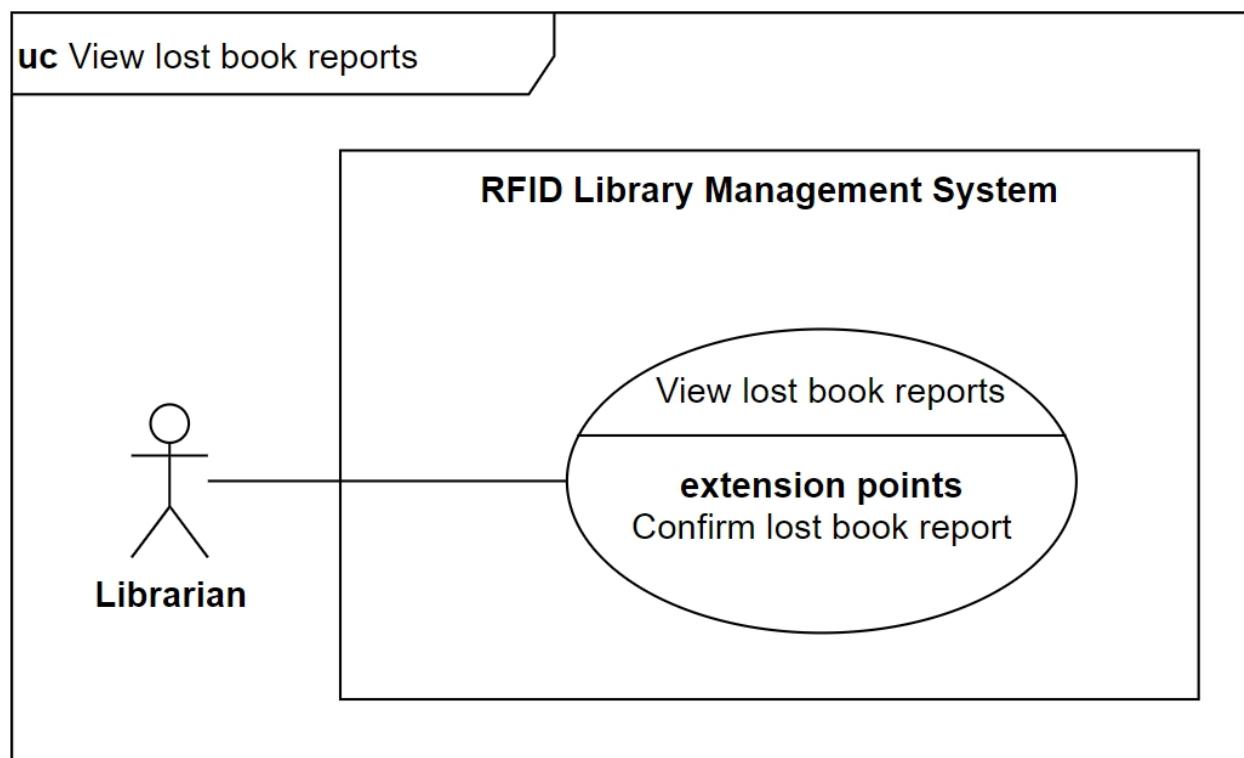


Figure 46 <Use Case> View lost book reports

ID and Name:	UC_37 View lost book reports		
Created By:	KhangNDN	Date Created:	26/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A

Description:	Let the librarian view reports of lost books in a specific period of time.		
Trigger:	The Librarian sends request to view book lost report		
Preconditions:	PRE-1: The User has logged in successfully as a Librarian role.		
Postconditions:	POST-1: RLMS shows a table containing list of lost books POST-2: If there is no result, show message		
Normal Flow:	Step	Actor action	System response
	1	The librarian selects start date and end date to search lost books [Exception 1]	The system searches books based on start date and end date then displays the results to screen.
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Occasionally		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> Lost books can be searched by: Q1, Q2, Q3, Q4, "Today", "Yesterday", "This month", "Last month", "This year" or a custom period. The Librarian can filter the result by reports that are pending and reports that have already been confirmed. Each report will contain: <ul style="list-style-type: none"> Barcode ISBN Title Patron Email Lost At Fine Note 		
Other Information	N/A		

Assumptions:	N/A
--------------	-----

Table 41 <Use Case> View lost book reports

2.2.4.19. Confirm lost book report

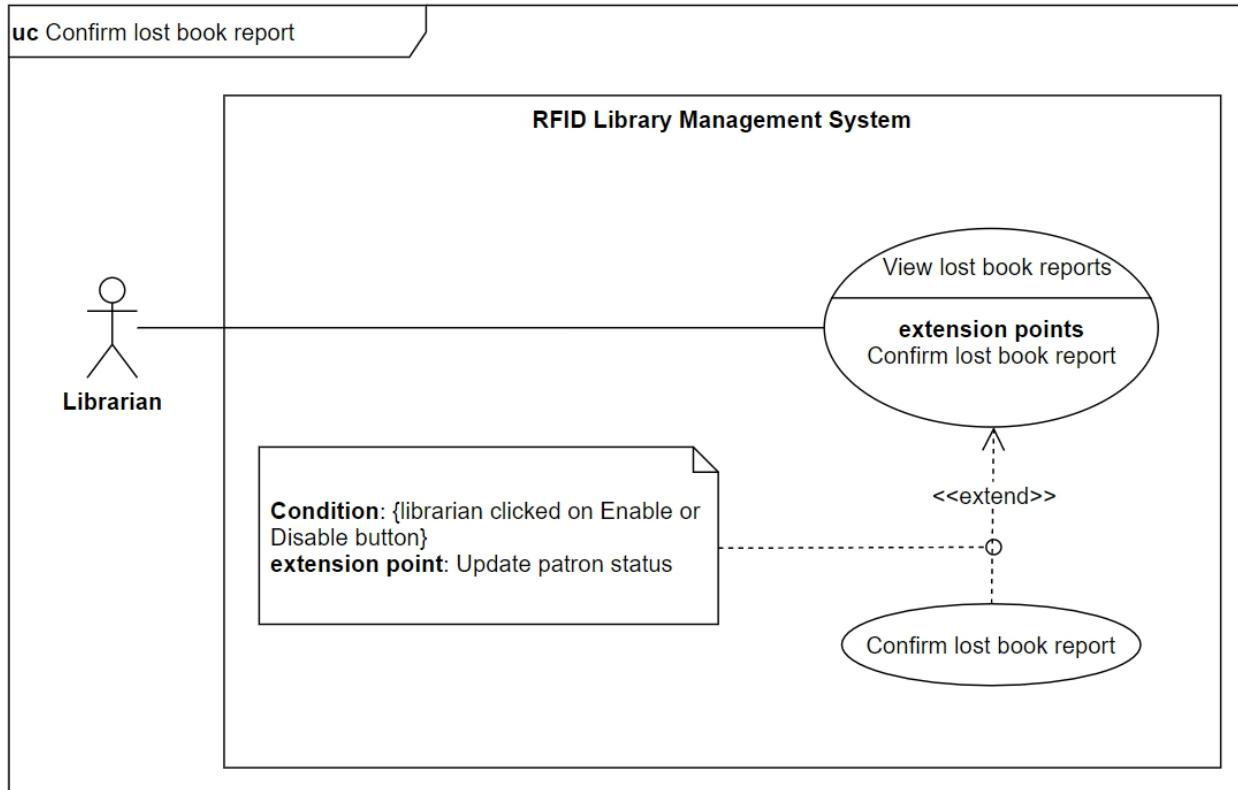


Figure 47 <Use Case> Confirm lost book report

ID and Name:	UC_38 Confirm lost book report		
Created By:	TramPH	Date Created:	26/02/2021
Primary Actor:	Librarian	Secondary Actors:	Patron
Description:	Let the librarian confirm and notify the lost book fine to the patron		
Trigger:	The librarian sends request to confirm the lost book report		
Preconditions:	PRE-1: The Librarian has logged in as a Librarian role. PRE-2: The lost book has been reported by the patron		
Postconditions:	POST-1: The system shows success message, and the patron received a lost		

	fine email POST-2: If fail, the system show error message		
Normal Flow:	Step	Actor action	System response
	1	The librarian click on “Confirm” button with ‘PENDING’ status on the book lost report screen	RLMS shows a confirmation modal
Alternative Flows:	No.	Actor action	System response
	1.1	The librarian click “Cancel” button	RLMS closes the confirmation modal
Exceptions:	No.	Cause	System response
	1	The system cannot send email to the patron	RLMS shows error message
Priority:	Medium		
Frequency of Use:	Occasionally		
Business Rules:	1. The confirmation shows: <ul style="list-style-type: none"> • Book information: ISBN, title, author, edition • Overdue days • Overdue fee • Lost book fine not in the market • Lost book fine in the market • Lost book fine input: required, range 1000 - 1000000000 • Note input: not required, length 0-500 2. Total fine will be lost fine + overdue fine 3. The suggestion lost fine will be calculated as below: <ul style="list-style-type: none"> • When the lost book is not in the market: lost fine = book copy price * missing document multiplier (is taken from <i>Fee Policy</i>) • When the lost book is still in the market: lost fine = book copy price + document processing fee (is taken from <i>Fee Policy</i>) 		

	<p>4. Overdue book copies will require the patron to pay a fine:</p> <ul style="list-style-type: none"> • Fine = overdue day(s) * fine rate (fine rate is taken from <i>Fee Policy</i>). Overdue days does not include Saturdays and Sundays • Max fine = book copy's price * max overdue fine percentage / 100 (max overdue fine percentage is taken from <i>Fee Policy</i>)
Other Information	N/A
Assumptions:	<p>1. The system will not manage finance (lost fine money). It will only show fine suggestions for librarians and they will input the actual lost fine manually.</p>

Table 42 <Use Case> Confirm lost book report

2.2.4.20. Index book copy location

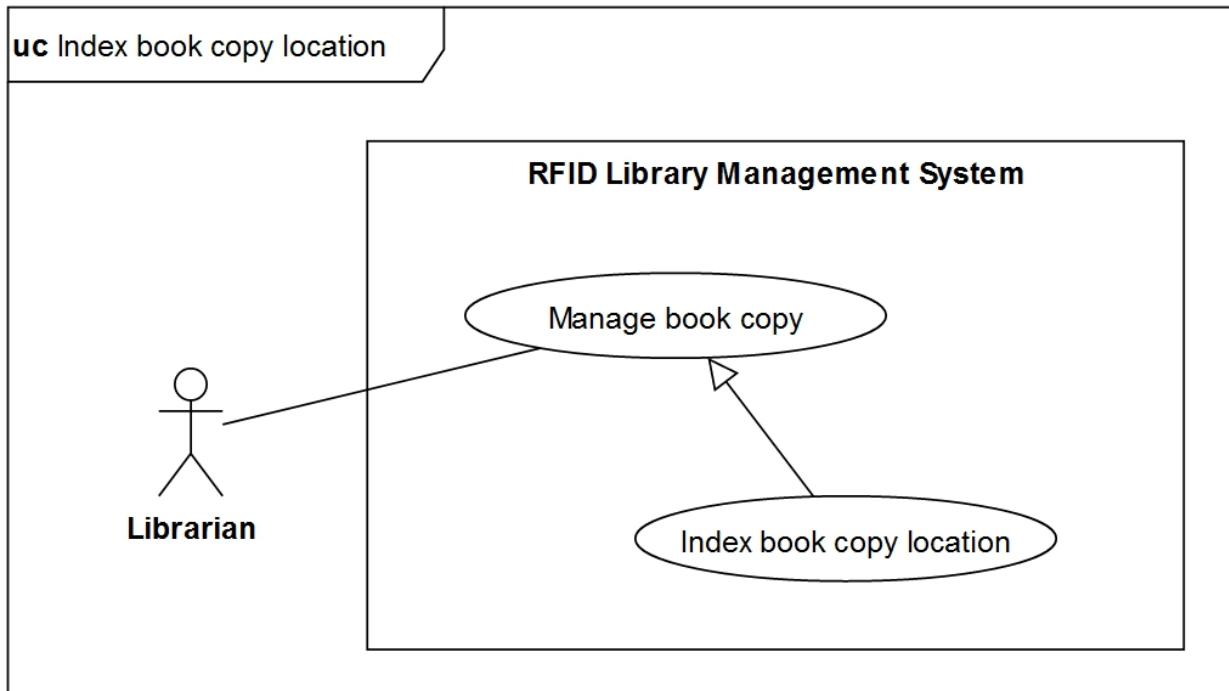


Figure 48 <Use Case> Index book copy location

ID and Name:	UC_39 Index book copy location		
Created By:	KienNT	Date Created:	23/03/2021
Primary Actor:	Librarian	Secondary Actors:	N/A

Description:	Let the librarian index/save book copies to a row of a shelf		
Trigger:	The librarian sends request to save books location		
Preconditions:	PRE-1: The Librarian has logged in successfully as a Librarian role		
Postconditions:	POST-1: The system shows success message POST-2: In case of failure, the system displays an error message		
Normal Flow:	Step	Actor action	System response
	1	The librarian clicks on the Sample button	The RLMS shows a the Sample screen and starts to receive shelf's RFID UID from the UHF reader
	2	The librarian selects shelf and row then clicks on the Scan button [Alternate 1]	The RLMS starts to receive book copy RFID UID from the UHF reader
	3	The librarian uses the UHF Reader to scan books on the shelf's row	The RLMS shows the scanned books table on the screen
	4	The librarian clicks on the Finish Button [Alternate 2] [Exception 1]	The RLMS saves new books onto the shelf's row
Alternative Flows:	Step	Actor action	System response
	1.1	The librarian uses the UHF Reader to scan the shelf's row RFID tag	The RLMS shows the shelf information on screen
	1.2	The librarian clicks on the Scan button	The RLMS starts to receive book copy RFID UID from the UHF reader
	1.3	The librarian uses the UHF Reader to scan books on the shelf's row	The RLMS shows the scanned books table on the screen
	1.4	The librarian clicks on the Finish Button [Alternate 2]	The RLMS saves new books onto the shelf's row

		[Exception 1]	
	2.1	The librarian click “Cancel” button	RLMS clears the scanned book table
Exceptions:	No.	Cause	System response
	1	One or more scanned books is “BORROWED” or “DISCARD” or “LOST”	RLMS shows error message
Priority:	Medium		
Frequency of Use:	Rarely		
Business Rules:		<ol style="list-style-type: none"> 1. Can only save location for books that are “AVAILABLE” or “LIB_USE_ONLY” 2. When saving new books to a shelf’s row, previous books that were on that row would be removed, and their location would become blank 3. Finish button can only be clicked when there are books in the scanned books table 4. Shelf and row must be selected before clicking the Scan button 5. After clicking the Scan button shelf and row cannot be changed unless the librarian clicked on the Cancel button 6. the scanned books table will show the book’s: <ul style="list-style-type: none"> ● Barcode ● RFID ● Call Number ● Title ● Author ● Edition 7. Shelf can be input manually or by scanning shelf RFID tag 8. Books are saved to shelf’s row by shelf’s row 	
Other Information	N/A		
Assumptions:	N/A		

Table 43 <Use Case> Index book copy location

2.2.4.21. Find misplaced book copies

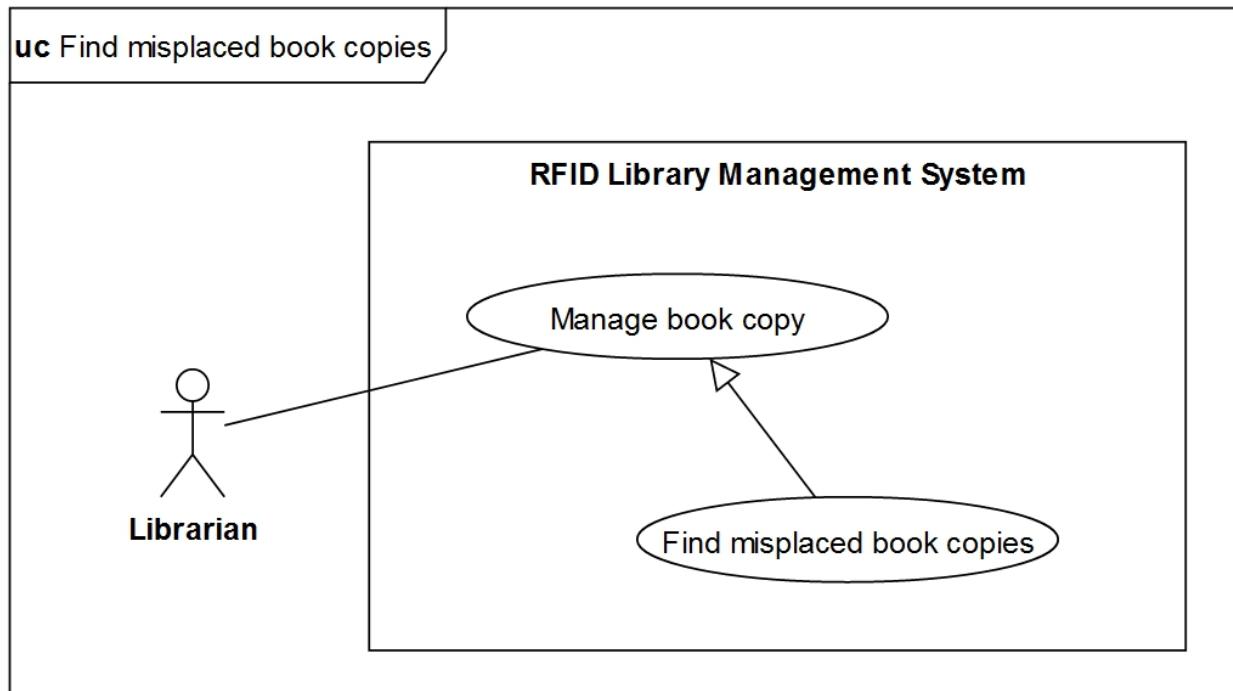


Figure 49 <Use Case> Find misplaced book copies

ID and Name:	UC_40 Find misplaced book copies		
Created By:	KienNT	Date Created:	23/03/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Let the librarian check for misplaced or missing books of each shelf's row		
Trigger:	The librarian sends request to check books location		
Preconditions:	PRE-1: The Librarian has logged in successfully as a Librarian role		
Postconditions:	POST-1: If there are any missing books from the shelf row, show a table of missing books POST-2: If there is any misplaced books on the shelf row, highlight the record in the scanned books table		
Normal Flow:	Step	Actor action	System response
	1	The librarian clicks on the	The RLMS shows a the Check screen

		Check button	and starts to receive shelf's RFID UID from the UHF reader
	2	The librarian selects shelf and row then clicks on the Scan button [Alternate 1]	The RLMS starts to receive book copy RFID UID from the UHF reader
	3	The librarian uses the UHF Reader to scan books on the shelf's row	The RLMS shows the scanned books table on the screen
	4	The librarian clicks on the Finish Button [Alternate 2]	The RLMS clears the scanned books table
Alternative Flows:	Step	Actor action	System response
	1.1	The librarian uses the UHF Reader to scan the shelf's row RFID tag	The RLMS shows the shelf information on screen
	1.2	The librarian clicks on the Scan button	The RLMS starts to receive book copy RFID UID from the UHF reader
	1.3	The librarian uses the UHF Reader to scan books on the shelf's row	The RLMS shows the scanned books table on the screen
	1.4	The librarian clicks on the Finish Button [Alternate 2]	The RLMS clears the scanned books table
	2.1	The librarian click on the Finish button and there are books missing from the shelf's row	RLMS returns a table of missing books
	2.2	The library clicks on the Clear button	RLMS clears the scanned books table
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	Occasionally		

Business Rules:	<ol style="list-style-type: none"> 1. The books table must contain: <ul style="list-style-type: none"> • Initial location (“N/A” if book is not assigned to a location) • Barcode • RFID • Call Number • Title • Author • Edition 2. Only check “AVAILABLE” and “LIB_USE_ONLY” books 3. Book is misplaced if it is put on a different row from its initial row. Misplaced books on the shelf are marked when shown on screen. 4. Shelf can be input manually or by scanning shelf RFID tag 5. Books location are checked by each shelf’s row
Other Information	N/A
Assumptions:	N/A

Table 44 <Use Case> Find misplaced book copies

2.2.4.22. Print book copies barcodes

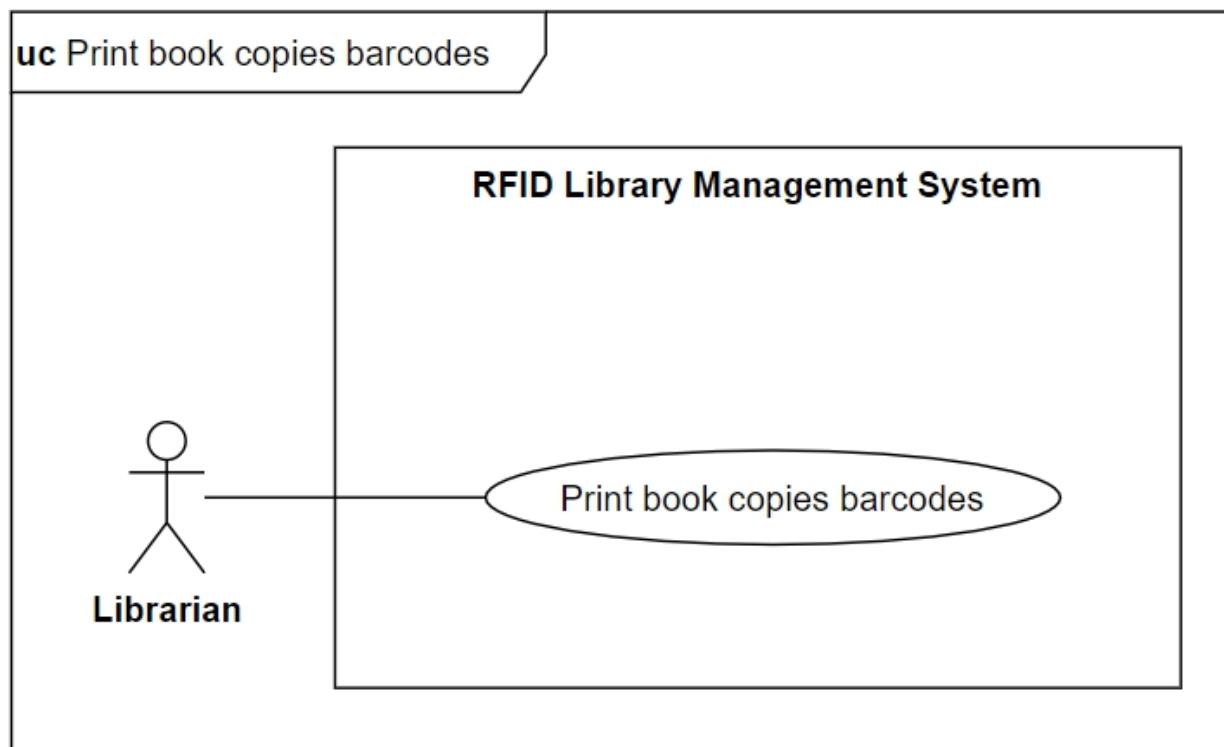


Figure 50 <Use Case> Print book copies barcodes

ID and Name:	UC_41 Print book copies barcodes		
Created By:	TramPH	Date Created:	23/03/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Let the librarian print barcodes of book copies in the system		
Trigger:	The librarian requests to print barcodes		
Preconditions:	PRE-1: The user has logged in successfully as a Librarian role		
Postconditions:	POST-1: The system sends a pdf file and success message POST-2: If print fail, show error message		
Normal Flow:	Steps	Actor action	System response
	1	Librarian ticks on the checkbox of target book copies on the book copy screen	RLMS saves those copies at local
	2	Librarian clicks “Print Barcode” button	RLMS shows confirmation modal
	3	Librarian clicks “OK” button [Alternative 1]	RLMS generates pdf contains those copies’ barcode then send back the librarian, show success message
Alternative Flows:	No.	Actor action	System response
	1.1	Librarian clicks “Close” button	RLMS close the confirmation modal
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	<ol style="list-style-type: none"> 1. The system generates barcode base on Code-39 barcode format 2. The barcodes will be grouped by : ISBN-Book Copy Type-Price 		

Other Information	<ol style="list-style-type: none"> 1. The barcode label has the following dimension: 2.25" x 1" 2. While already having RFID tags, we still keep the barcode system as a fail-safe, in case the RFID tag is broken.
Assumptions:	<ol style="list-style-type: none"> 1. Our team assumes that the library uses 2.25" x 1" barcode labels. 2. The RLMS does not manage the barcode printers, so we assume that the library is equipped with printers according to our barcode label dimension. (We recommend this printer: Zebra GX430t Thermal Transfer Desktop Printer, and this barcode label: BETCKEY - 2.25" x 1" label)

Table 45 <Use Case> Print book copies barcodes

2.2.5. Manager

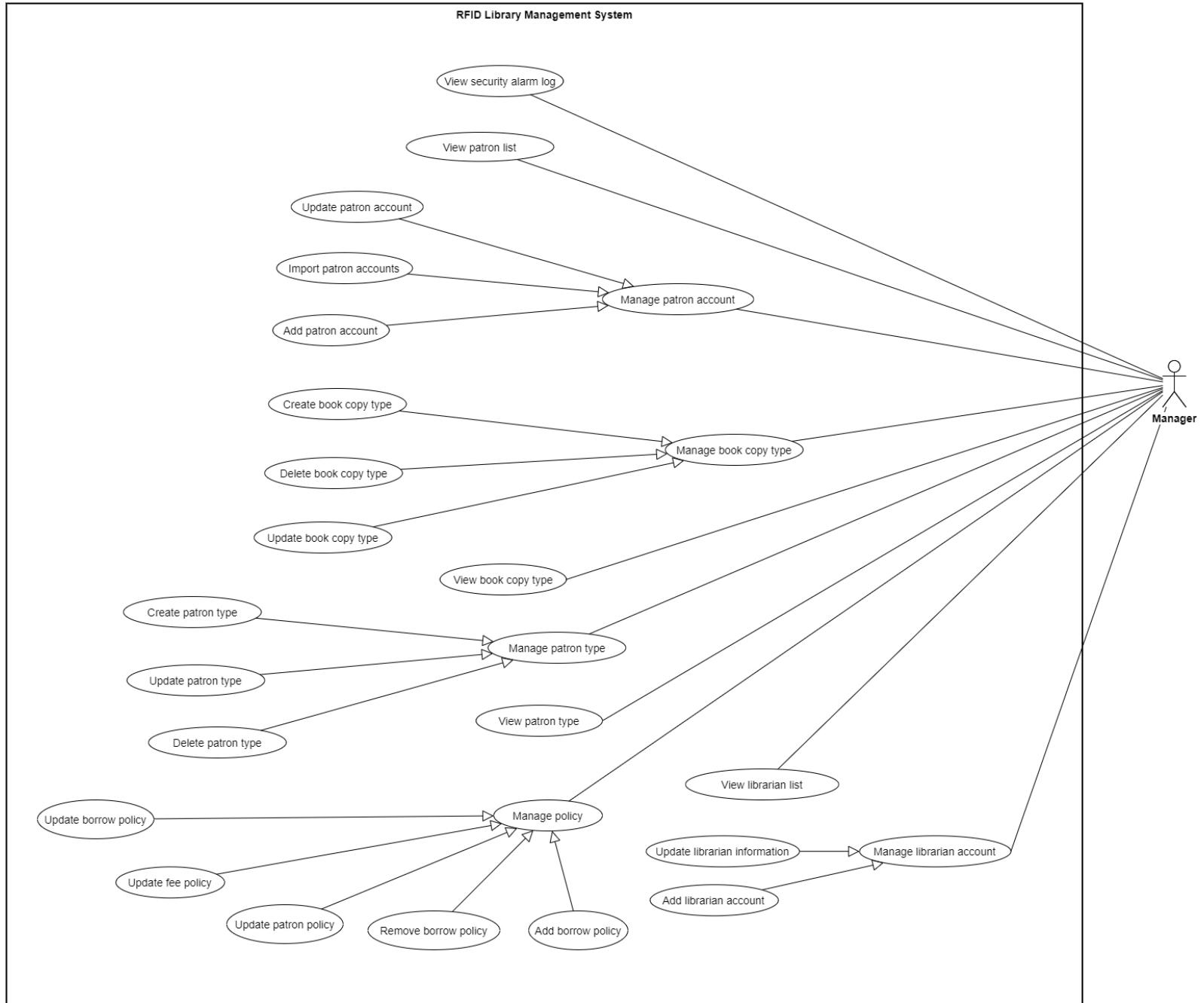


Figure 51 <Use Case Overview> Manager

2.2.5.1. Add borrow policy

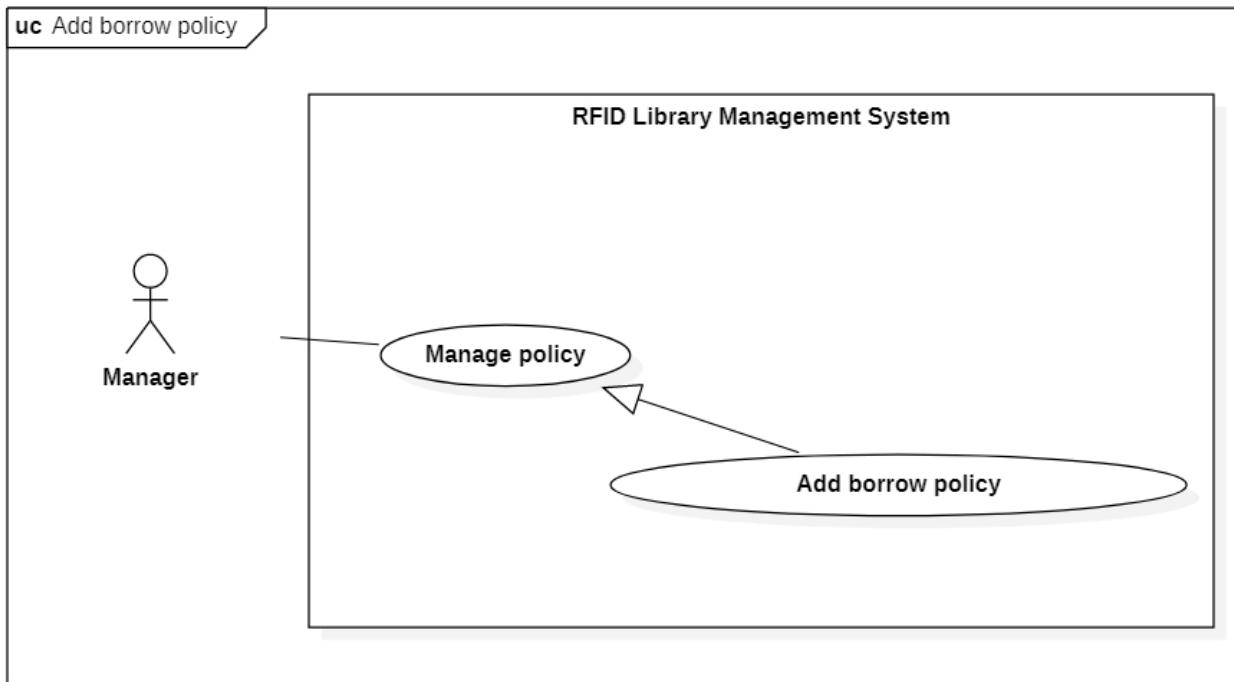


Figure 52 <Use Case> Add borrow policy

ID and Name:	UC_42 Add borrow policy		
Created By:	TramPH	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager set a new borrow policy for patron type and book copy type.		
Trigger:	The manager sends a request to add a borrow policy.		
Preconditions:	PRE-1: The user has logged in as Manager role.		
Postconditions:	POST-2: Borrow policy is added in the system. POST-2: If failed, show error message.		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks on the “Add New” button from the borrow policy screen.	RLMS shows add new borrow policy form.

	2	The manager input required fields and click the “Confirm” button. [Alternative 1]	RLMS checks validation, saves new policy to the database and pops up a successful message. [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Cancel” button	RLMS close add borrow policy form.
Exceptions:	No.	Cause	System response
	1	The manager adds duplicated borrow policies for the same pair of patron type and book copy type.	RLMS shows an error message.
	2	The manager inputs invalid fields.	RLMS shows error messages.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. System allows multiple borrow policy settings based on patron type and book copy type. For example, the policy that patron type Student borrow book copy type A can be different from the policy that patron type Student borrow book copy type B. 2. Adding borrow policy requires the following fields: <ul style="list-style-type: none"> • Patron type: select from patron type list • Book copy type: select from copy type list • Borrow period (days): required in range 1 - 1000 • Checkouts allowed (count): required in range 1 - 100 • Renewals allowed (times): required in range 0 - 100 • Renew period (days): required in range 1 - 1000 3. Manager cannot set duplicated policies for the same pair of patron type and book copy type. 4. The checkout allowed includes the number of copies the patron keeping 		
Other Information	N/A		

Assumptions:	N/A
--------------	-----

Table 46 <Use Case> Add borrow policy

2.2.5.2. Update borrow policy

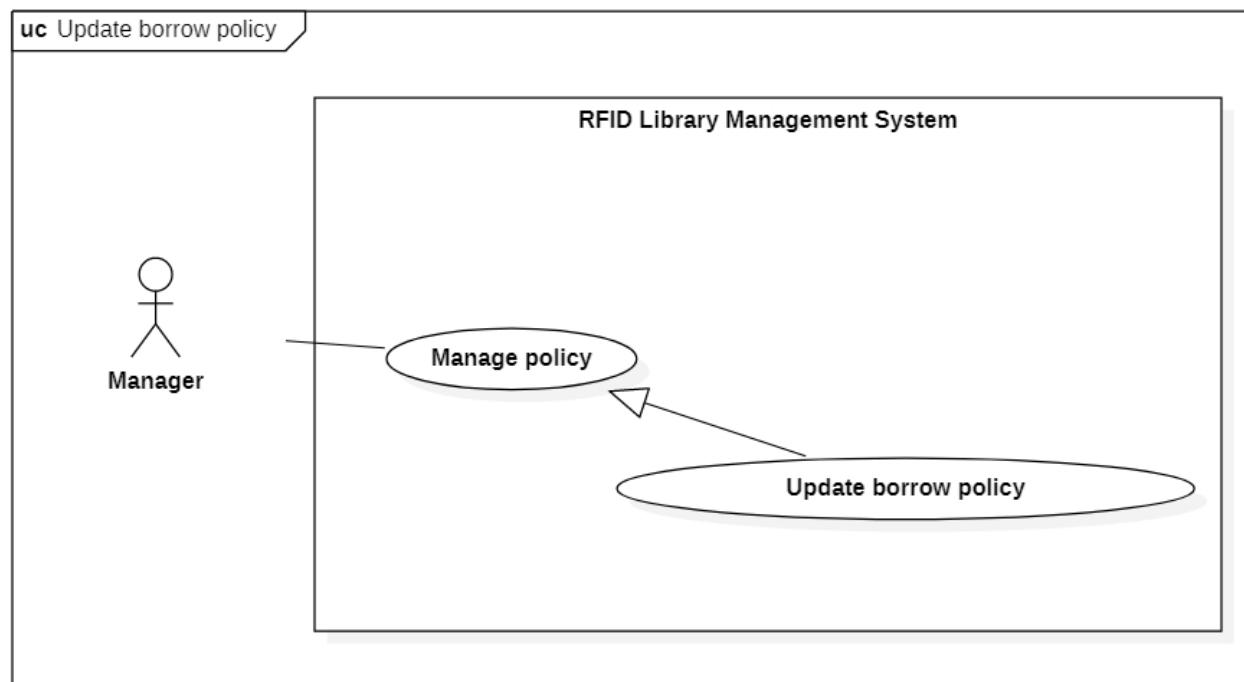


Figure 53 <Use Case> Update borrow policy

ID and Name:	UC_43 Update borrow policy		
Created By:	KienNT	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let manager updates borrow policy for patron type and book copy type.		
Trigger:	The manager sends a request to update a borrow policy.		
Preconditions:	PRE-1: The user has logged in as Manager role. PRE-2: The updating borrow policy is stored in the database.		
Postconditions:	POST-2: Borrow policy is updated in the system.		

	POST-2: If failed, show error message.		
Normal Flow:	Step	Actor action	System response
	1	The manager inputs new values for editable fields and clicks the update icon button from the borrow policy screen.	RLMS shows a confirmation dialog.
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Close” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The system cannot find the updating borrow policy.	RLMS shows an error message.
	2	The manager inputs invalid fields.	RLMS shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. The manager can only update the following fields: <ul style="list-style-type: none"> • Borrow period (days): required in range 1 - 1000 • Checkouts allowed (count): required in range 1 - 100 • Renewals allowed (times): required in range 0 - 100 • Renew period (days): required in range 1 - 1000 2. Manager cannot set duplicated policies for the same pair of patron type and book copy type. 3. The checkout allowed includes the number of copies the patron keeping 		

Other Information	N/A
Assumptions:	N/A

Table 47 <Use Case> Update borrow policy

2.2.5.3. Remove borrow policy

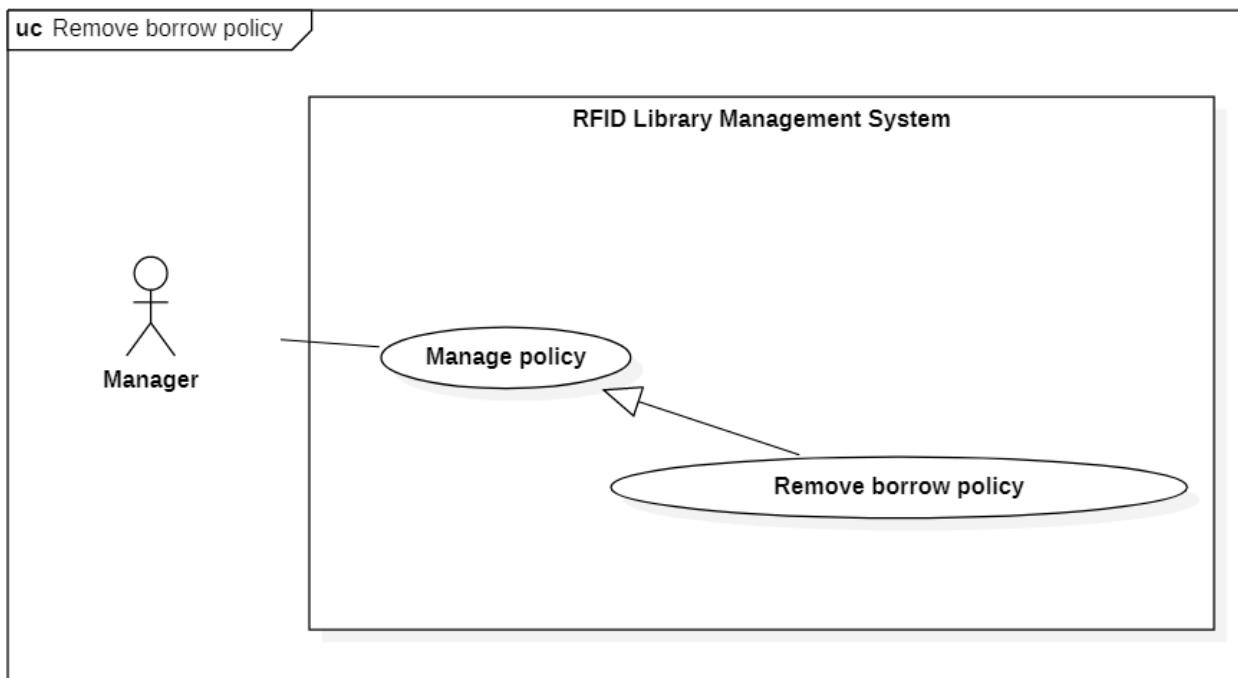


Figure 54 <Use Case> Remove borrow policy

ID and Name:	UC_44 Remove borrow policy		
Created By:	KienNT	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager remove an existing borrow policy.		
Trigger:	The manager requests to remove a borrow policy.		
Preconditions:	PRE-1: The user has logged in as Manager role. PRE-2: The borrow policy is stored in the database.		

Postconditions:	POST-1: The borrow policy is removed in the database. POST-2: If failed, show error message.		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks the remove icon button from the borrow policy screen.	RLMS shows a confirmation dialog.
	2	The manager clicks the “OK” button. [Alternative 1]	RLMS removes that borrow policy from the database and pops up a successful message. [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Close” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The system cannot find the removing borrow policy.	RLMS shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	1. When the manager removes a borrow policy, the patron type from that policy cannot borrow or renew any copy from that book copy type.		
Other Information	N/A		
Assumptions:	N/A		

Table 48 <Use Case> Remove borrow policy

2.2.5.4. Update patron policy

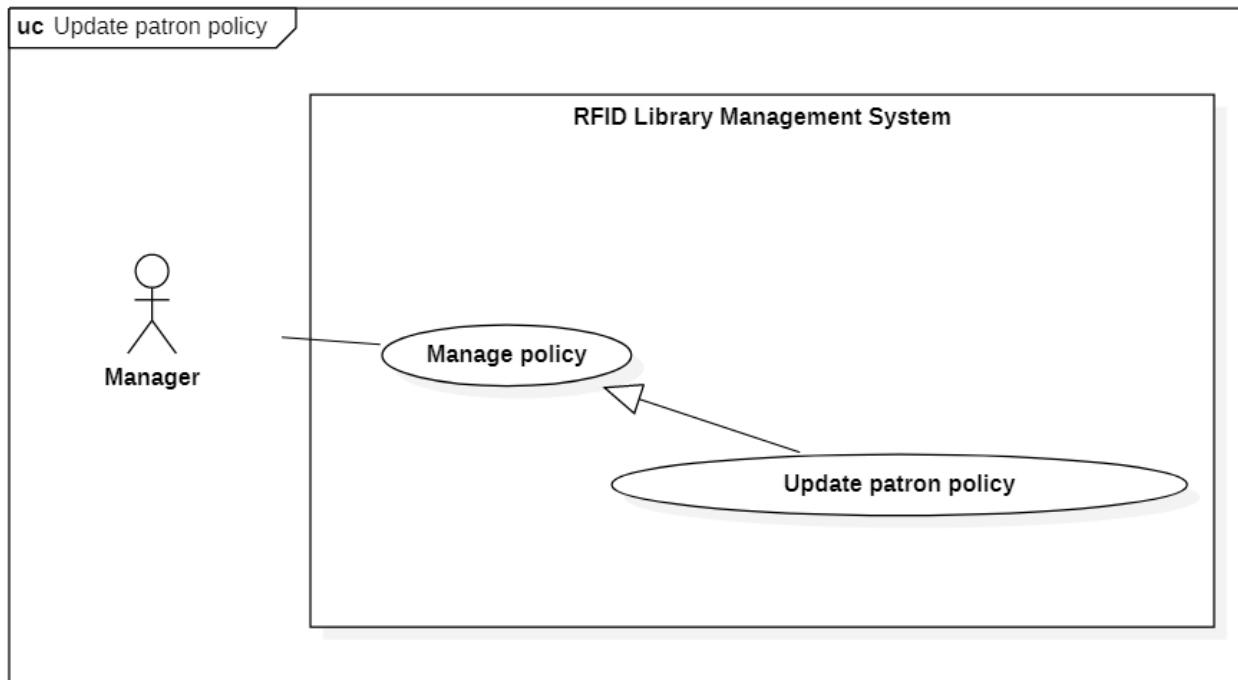


Figure 55 <Use Case> Update patron policy

ID and Name:	UC_45 Update patron policy		
Created By:	KienNT	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager update the patron policy.		
Trigger:	The manager requests to update a patron policy.		
Preconditions:	PRE-1: The user has logged in as Manager role. PRE-2: The patron policy is stored in the database.		
Postconditions:	POST-1: The patron policy is updated in the database. POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager inputs a new “total checkouts allowed” of one patron type and clicks the update icon button from the	RLMS shows a confirmation dialog.

		patron policy screen.	
	2	The manager clicks the “OK” button. [Alternative 1]	RLMS checks validation, updates that patron policy to the database and pops up a successful message. [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Close” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The system cannot find the updating patron policy.	RLMS shows an error message.
	2	The manager inputs invalid max number borrow.	RLMS shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. The manager can only update “total checkouts allowed” (maximum number of copies that patron type can borrow includes the number of copies the patron is keeping). 2. Total checkouts allowed in range 0 - 100 		
Other Information	N/A		
Assumptions:	N/A		

Table 49 <Use Case> Update patron policy

2.2.5.5. Update fee policy

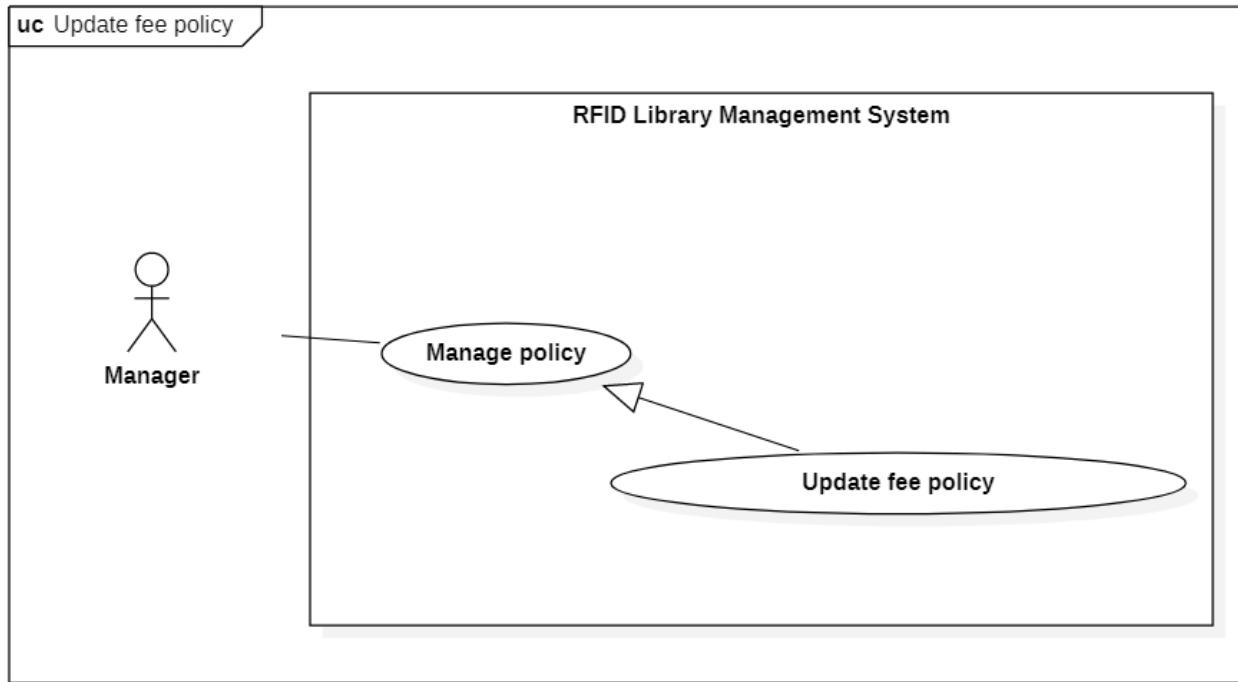


Figure 56 <Use Case> Update fee policy

ID and Name:	UC_46 Update fee policy		
Created By:	KienNT	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager update the fee policy.		
Trigger:	The manager requests to update the fee policy.		
Preconditions:	PRE-1: The user has logged in as Manager role.		
Postconditions:	POST-1: The new fee policy is added in the database. POST-2: The old fee policy is still stored in the database. POST-3: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager inputs editable fields and clicks the update button from the fee policy	RLMS shows a confirmation dialog.

		screen.	
	2	The manager clicks the “OK” button. [Alternative 1]	RLMS checks validation, saves the new fee policy into the database and pops up a successful message. [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Close” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The manager inputs invalid fields.	RLMS shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. The system only adds the updated fee policy with appropriate created time to database instead of updating the fee policy in the database. 2. The system has to save all fee policy history. 3. The managers can only update the following fields: <ul style="list-style-type: none"> • Overdue fine per day: The fine rate (VND) • Max overdue fine (% of book price): The percentage of the book copy price (as recorded in the system) (max = 100%). This is used to calculate the overdue fine • Missing document multiplier: Is used to calculate lost book copy fine • Document processing fee: Is used to calculate lost book copy fine (VND) 4. The Fee Policy is applied right after it is updated 		
Other Information	N/A		
Assumptions:	N/A		

Table 50 <Use Case> Update fee policy

2.2.5.6. View librarian list

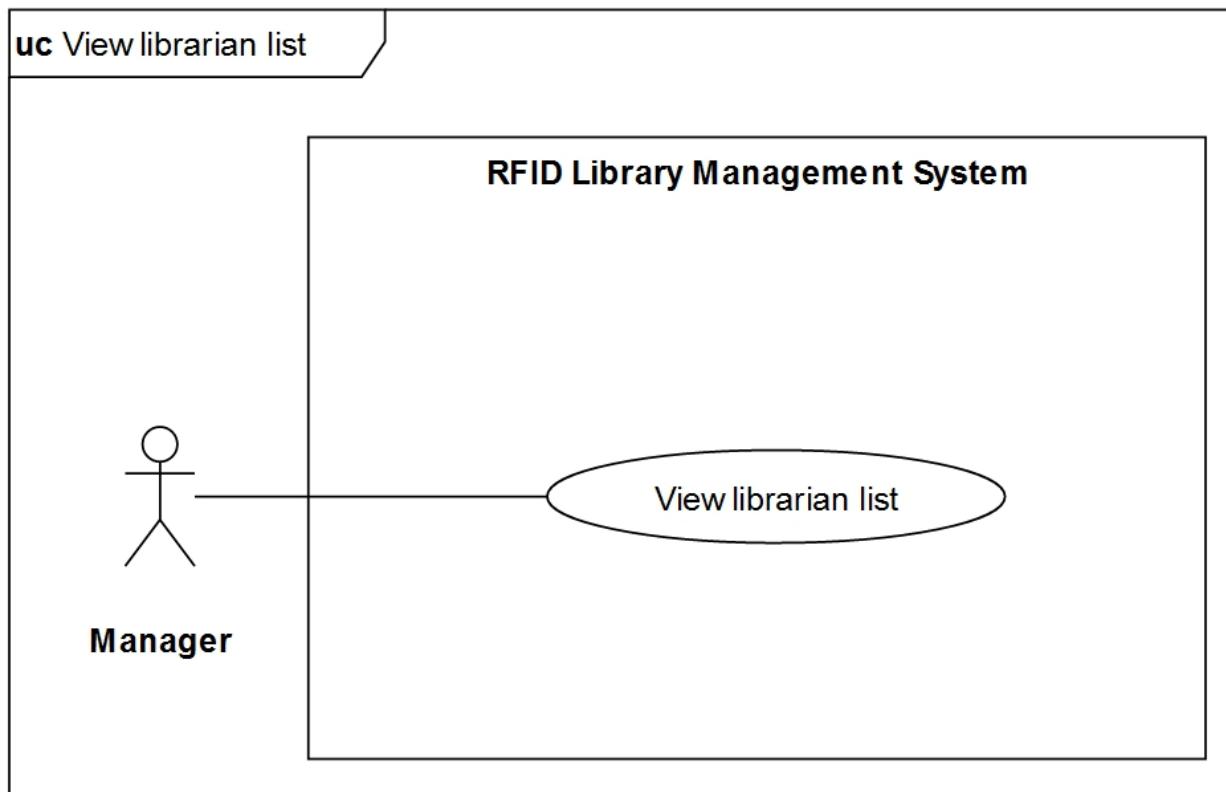


Figure 57 <Use Case> View librarian list

ID and Name:	UC_47 View librarian list		
Created By:	KhangNDN	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Provide the manager with the ability to view a list of librarians' accounts.		
Trigger:	The Manager sends a request to get a list of librarians		
Preconditions:	PRE-1: The user has logged in as Manager role		
Postconditions:	POST-1: RLMS shows a list of librarian's accounts POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response

	1	The manager enters the search input and clicks the Search button. [Exception 1]	The RLMS processes the request and returns a list of librarians.
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Each record in the table includes: <ul style="list-style-type: none"> ● Librarian's Id ● Librarian's email ● Librarian's name ● Librarian's gender ● Librarian's phone ● Librarian's status (Active/Inactive) 2. Search by like librarian's email 		
Other Information	N/A		
Assumptions:	N/A		

Table 51 <Use Case> View librarian list

2.2.5.7. Add librarian account

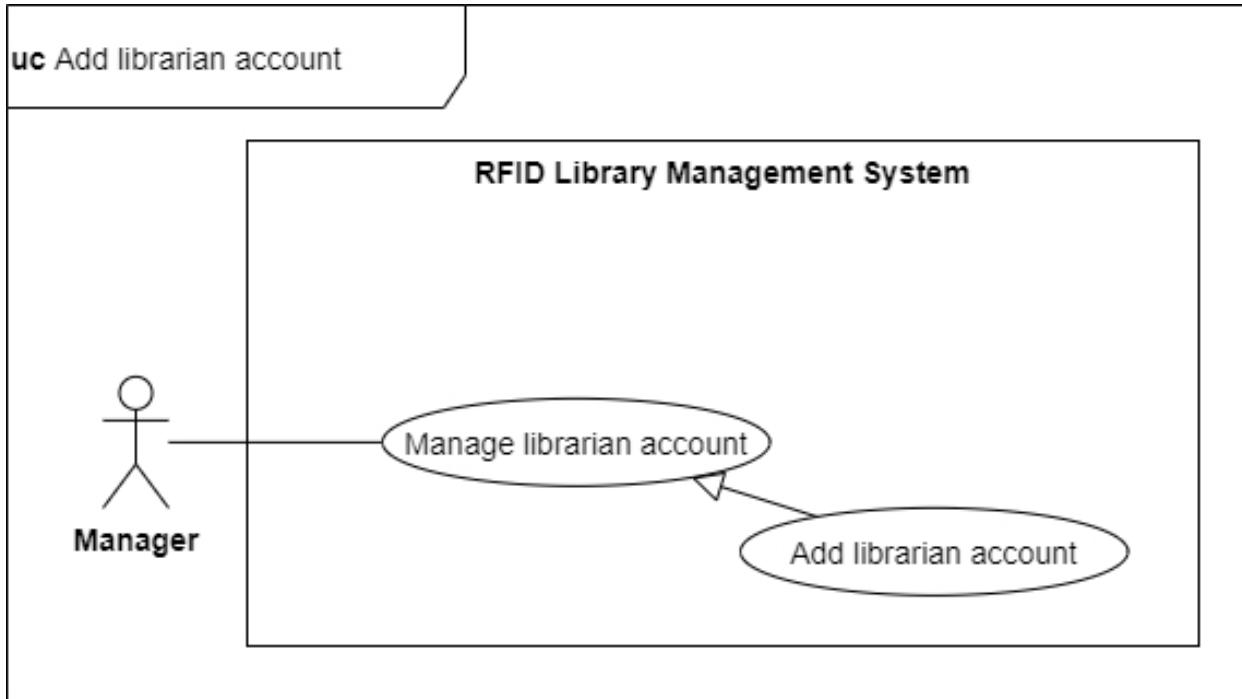


Figure 58 <Use Case> Add librarian account

ID and Name:	UC_48 Add librarian account		
Created By:	KhangNDN	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Manager can add new librarian's account to the system		
Trigger:	The manager sends request to add new librarian's account		
Preconditions:	User has logged in as Manager role		
Postconditions:	POST-1: Show success message POST-2: If add failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks on the "Add librarian" button.	The RLMS requires manager to fill librarian's account information

	2	The manager fills require fields and click Save button [Alternative 1]	The RLMS checks validation and processes the create transaction, then shows the success message. The RLMS also sends email to librarian about account username and password [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	Manager clicks Cancel button	The add librarian's account modal closes
Exceptions:	No.	Cause	System response
	1	Data is invalidated	The system shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Librarian's account information requires following fields: <ul style="list-style-type: none"> • Image: is required, is chosen from the local files • Full name: is required, 1-100 characters • Phone number: is required, 10 digits • Email: is required, unique, match email format • RFID number: is required, using RFID card • Gender: Male or Female 2. Librarian's email will also be the account's username. Librarian's account password will be generated by RLMS 3. After the librarian's account was created successfully, RLMS had to send an email to librarian about the account's email and auto generated password. 4. Generated password (length: 10) should include lower case characters, upper case characters, digits, and special characters. 		
Other Information	N/A		
Assumptions:	N/A		

Table 52 <Use Case> Add librarian account

2.2.5.8. Update librarian information

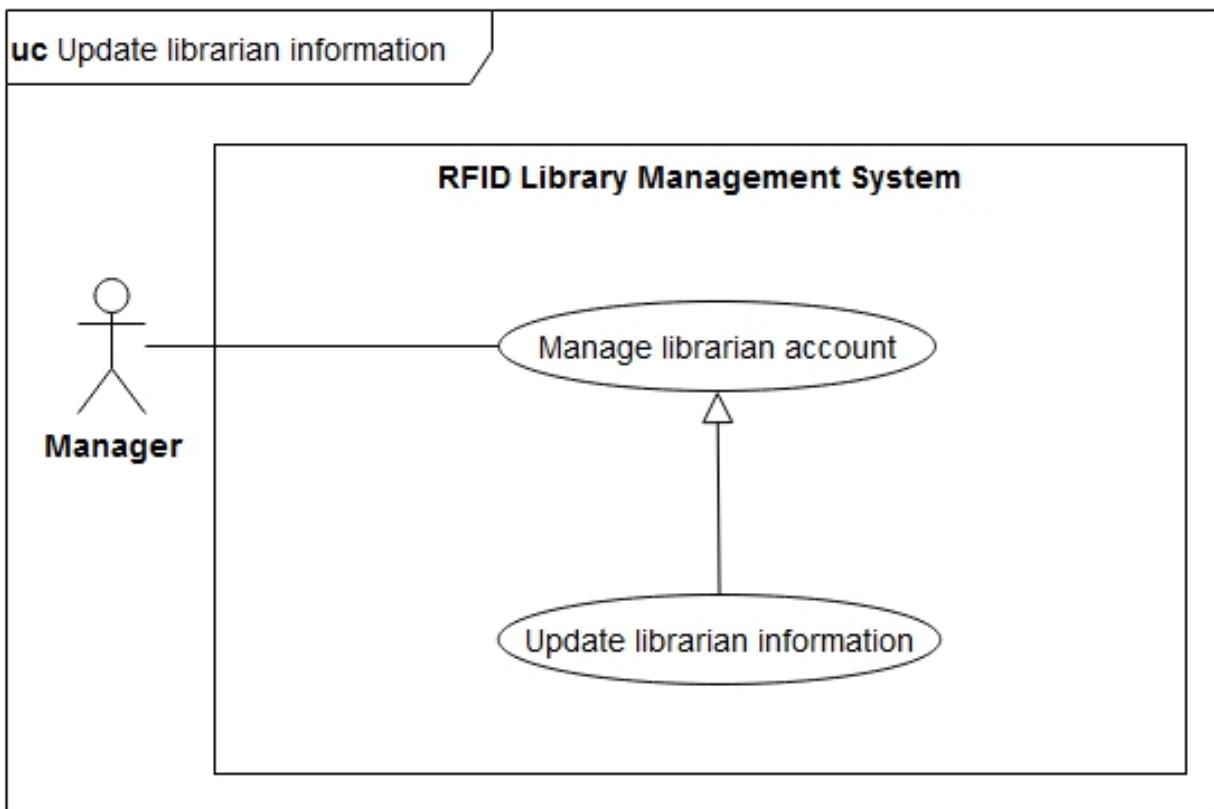


Figure 59 <Use Case> Update librarian information

ID and Name:	UC_49 Update librarian information		
Created By:	KhangNDN	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Manager can update existing librarian's account		
Trigger:	The manager sends request to update existing librarian's account		
Preconditions:	PRE-1: User has logged in as Manager role PRE-2: Updating librarian is stored in the system.		
Postconditions:	POST-1: Show success message POST-2: If update failed, show error message		
Normal Flow:	Step	Actor action	System response

	1	The manager clicks on the update icon button.	The system requires manager to fill librarian's account information
	2	The manager fills require fields and click Save button [Alternative 1]	The RLMS checks validation and processes the update transaction, then shows the success message. [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	Manager clicks Cancel button	The update librarian's account modal closes
Exceptions:	No.	Cause	System response
	1	Data is invalid	The system shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Update librarian's account information requires following fields: <ul style="list-style-type: none"> • Image: is required, is chosen from local files. • Full name: is required, 1-100 characters • Phone number: is required, 10 digits • RFID number: is required, using RFID card • Gender: Male or Female 2. By default, the librarian's current information is shown on the update screen when it is shown to the user. 		
Other Information	N/A		
Assumptions:	N/A		

Table 53 <Use Case> Update librarian information

2.2.5.9. Add patron account

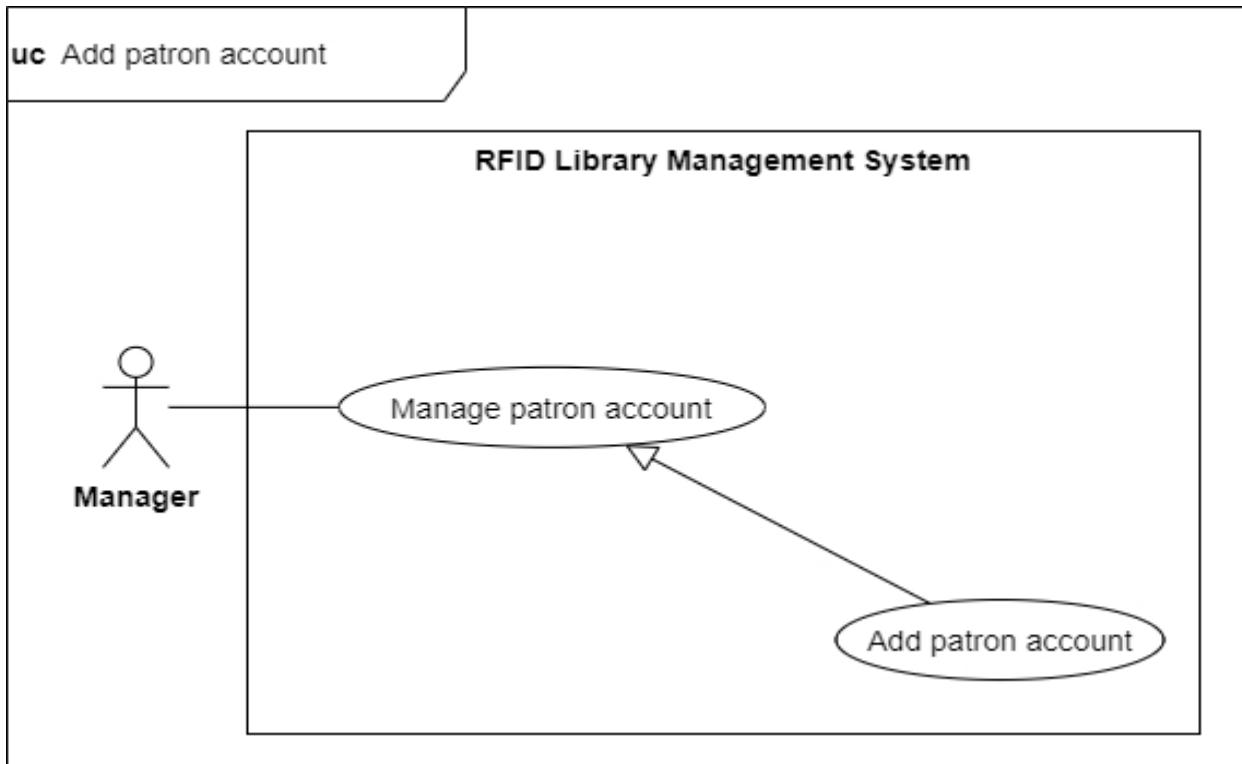


Figure 60 <Use Case> Add patron account

ID and Name:	UC_50 Add patron account		
Created By:	KhangNDN	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Manager can add new patron's account to the system		
Trigger:	The manager sends request to add new patron's account		
Preconditions:	PRE-1: User has logged in as Manager role		
Postconditions:	POST-1: Show success message POST-2: If add failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The managers click on the	The RLMS requires manager to fill

		"Add Patron" button.	patron's account information
	2	The manager fills require fields and click Save button [Alternative 1]	The RLMS checks validation and processes the create transaction, then shows the success message. The RLMS also sends email to patron about account username and password [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	Manager clicks Cancel button	The add patron's account modal closes
Exceptions:	No.	Cause	System response
	1	Data is invalidated	The system shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Patron's account information requires following fields: <ul style="list-style-type: none"> • Image: required, is chosen from local files. • Full name: required, 1-100 characters • Phone number: required, 10 digits • Email: required, unique, match email format • RFID number: required, using RFID card • Patron's type: required, select from patron type list • Gender: Male or Female 2. Patron's email will also be the account's username. 3. Patron's account password will be generated by RLMS. 4. After the patron's account is created successfully, RLMS sends account information to the patron email, including the generated password. 5. Generated password (length: 10) should include lower case characters, upper case characters, digits, and special characters. The user can change the password after log into the system 		

Other Information	N/A
Assumptions:	N/A

Table 54 <Use Case> Add patron account

2.2.5.10. Update patron account

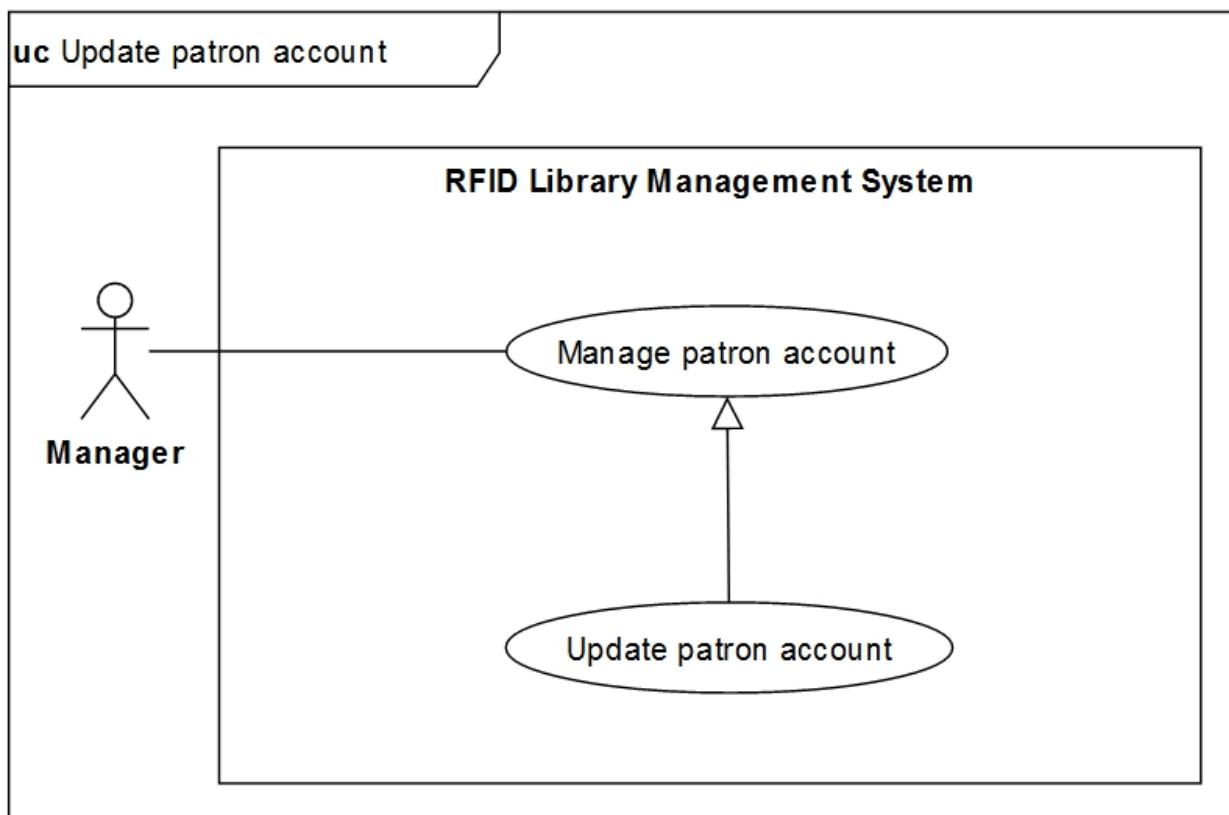


Figure 61 <Use Case> Update patron account

ID and Name:	UC_51 Update patron account		
Created By:	KhangNDN	Date Created:	24/02/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Manager can update existing patron's account		
Trigger:	The manager sends request to update existing patron's account		
Preconditions:	User has logged in as Manager role		

Postconditions:	POST-1: Show success message POST-2: If update failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks on the update icon button.	The system requires manager to fill patron's account information
Alternative Flows:	Step	Actor action	System response
	1.1	Manager clicks Cancel button	The update patron's account modal closes
Exceptions:	No.	Cause	System response
	1	Data is invalidated	The system shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> Update patron's account information requires following fields: <ul style="list-style-type: none"> Avatar: is required, is chosen from local files. Full name: is required, 1-100 characters Phone number: is required, 10 digits RFID number: is required, input by scanning patron card Patron's type: is required, select from patron type list Gender: Male or Female By default, the patron's current information is shown on the update screen when it is shown to the user. 		
Other Information	<ol style="list-style-type: none"> If there is no internet connection, the system cannot update the avatar 		

Assumptions:	N/A
--------------	-----

Table 55 <Use Case> Update patron account

2.2.5.11. View security alarm log

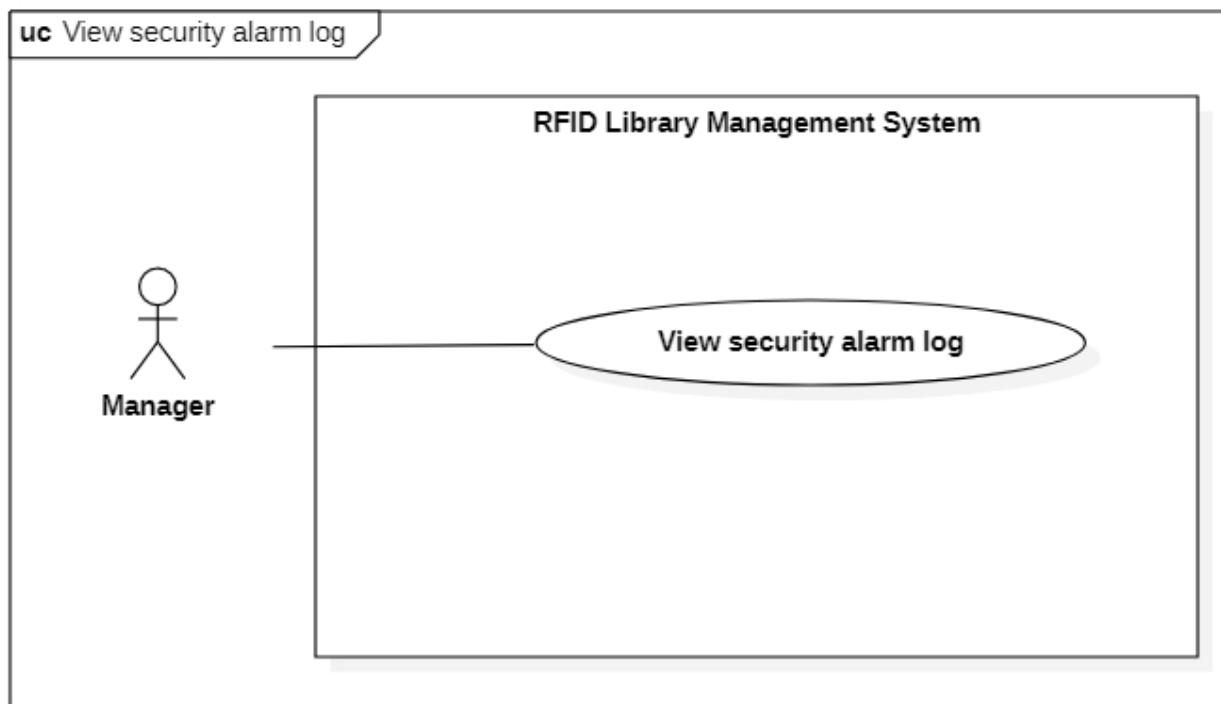


Figure 62 <Use Case> View security alarm log

ID and Name:	UC_52 View security alarm log		
Created By:	HoangPM	Date Created:	03/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Provide the manager with the ability to view security alarm log		
Trigger:	The manager sends request to view security alarm log		
Preconditions:	User has logged in as Manager role		
Postconditions:	POST-1: If there is any log record, shows a table of log records POST-2: If there is no log record, shows no data message		

Normal Flow:	Step	Actor action	System response
	1	The manager sends request to view security alarm log [Exception 1]	The RLMS processes the request and returns a table of log records
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieved an empty list	The system returns a message saying there is no security alarm log
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Security log records are searched by date. Retrieve all log records from 00:00:00 of the query date to 23:59:59 of the query date. 		
Other Information	<ol style="list-style-type: none"> 1. If there is no internet connection, the system cannot load the book image 		
Assumptions:	N/A		

Table 56 <Use Case> View security alarm log

2.2.5.12. Import patron accounts

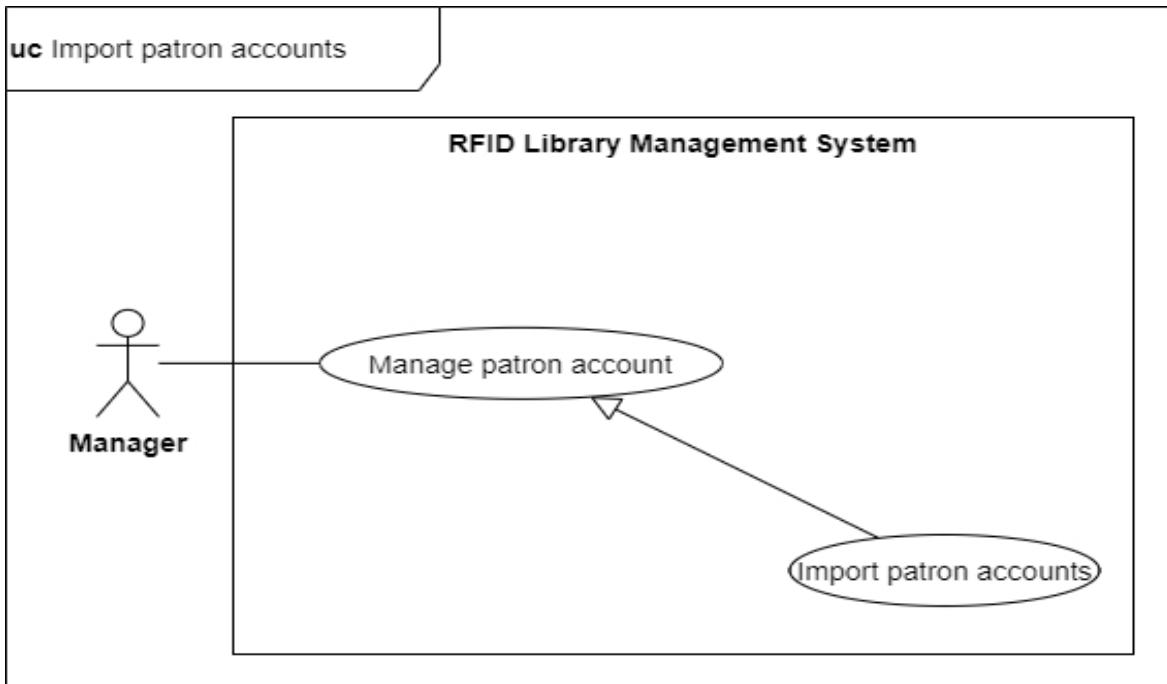


Figure 63 <Use Case> Import patrons

ID and Name:	UC_53 Import patron accounts		
Created By:	TramPH	Date Created:	24/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let manager to import patrons from an excel file		
Trigger:	Manager send request to import patrons		
Preconditions:	User has logged in as Manager role		
Postconditions:	POST-1: The system saves patron accounts and show success message POST-2: If import fail, show error message		
Normal Flow:	Step	Actor action	System response
	1	Manager clicks "Import Patron" button on the patron management screen	RLMS shows import patron form

	2	Manager choose excel file, patron type then clicks “Confirm” button [Alternative 1]	RLMS validates data, saves patron to database and shows success message [Exception 1]
Alternative Flows:	Step	Actor action	System response
	1.1	Manager clicks “Cancel” button	RLMS closes the import patron form
Exceptions:	No.	Cause	System response
	1	Manager chooses non-excel file	The system shows error message
	2	Data in excel file are invalid	The system shows error message
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	1. System allows manager to import multiple new patrons 2. Excel file has 6 columns with the following headers: <ul style="list-style-type: none"> • Email: required, email format, maximum length: 100 • RFID: required, maximum length: 80 • Full Name: required, maximum length: 100 • Phone: required 10 digits • Gender: required, F or M • Avatar: required, maximum length: 500 3. Patron should receive account information email in order to log into the system		
Other Information	1. Before importing, manager should check headers, columns and rows in the excel file for better performance		
Assumptions:	1. The manager can get the excel file that generated from another system of the school with the correct format 2. The school must store the avatar of the patron on the firebase or another storage that can be accessed by the internet with the link		

Table 57 <Use Case> Import patrons

2.2.5.13. View book copy type

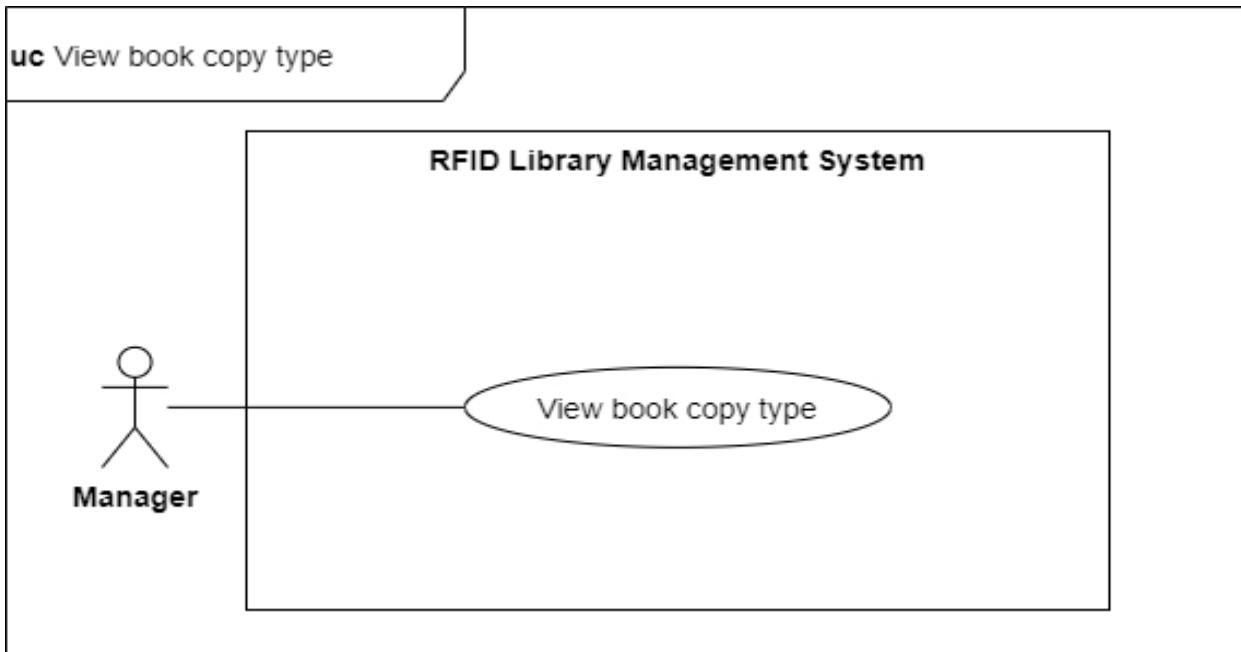


Figure 64 <Use Case> View book copy type

ID and Name:	UC_54 View book copy type		
Created By:	KienNT	Date Created:	25/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Provide the manager with the ability to view <i>Book Copy Type</i> . It is for showing which book copy type is available in the system.		
Trigger:	The Librarian sends request to view the book copy type		
Preconditions:	PRE-1: The user has logged in as Manager role		
Postconditions:	POST-1: RLMS shows a list of book copy types POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks the “Types” tab on the Policy page.	RLMS shows a list of book copy types.

		[Exception 1]	
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. The system allows manage view all the book copy type of the library 2. The manager can classify the book copies into different types. One book copy just belongs to one type. With this classification, the manager can authorize which patron can borrow which copy type 3. The name of book copy type must be unique 		
Other Information	N/A		
Assumptions:	N/A		

Table 58 <Use Case> View book copy type

2.2.5.14. Create Book Copy Type

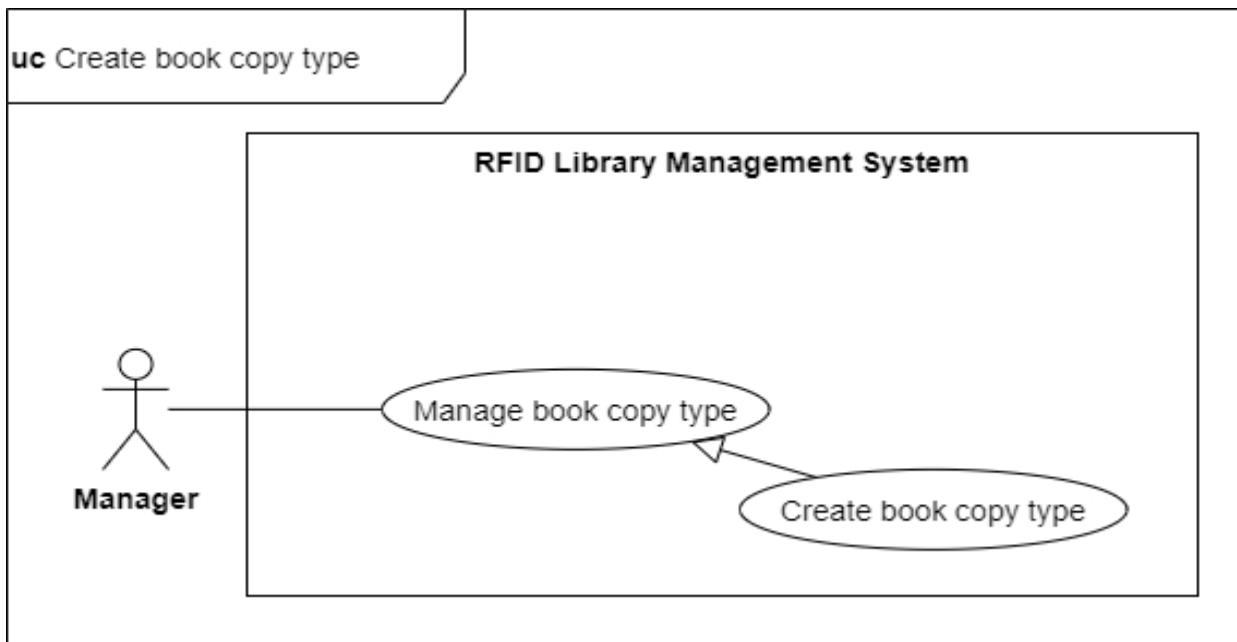


Figure 65 <Use Case> Create book copy type

ID and Name:	UC_55 Create book copy type		
Created By:	KienNT	Date Created:	25/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager add a new book copy type.		
Trigger:	The manager sends a request to add a book copy type.		
Preconditions:	PRE-1: The user has logged in as Manager role.		
Postconditions:	POST-1: Book copy type is added in the system POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks on the "Add New" button from the book copy type screen.	RLMS shows add new book copy type form.

	2	The manager input required fields and click the “Confirm” button. [Alternative 1]	RLMS checks validation, saves new book copy type to the database and pops up a successful message. [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Cancel” button	RLMS close add book copy type form.
Exceptions:	No.	Cause	System response
	1	The manager adds a duplicated name of book copy type.	RLMS shows an error message.
	2	The manager inputs invalid fields.	RLMS shows error messages.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	1. The system allows the manager to add a new book copy type. The name of book copy type must be unique and range 1-100 2. Manager cannot set a duplicated name for book copy type		
Other Information	N/A		
Assumptions:	N/A		

Table 59 <Use Case> Create book copy type

2.2.5.15. Update book copy type

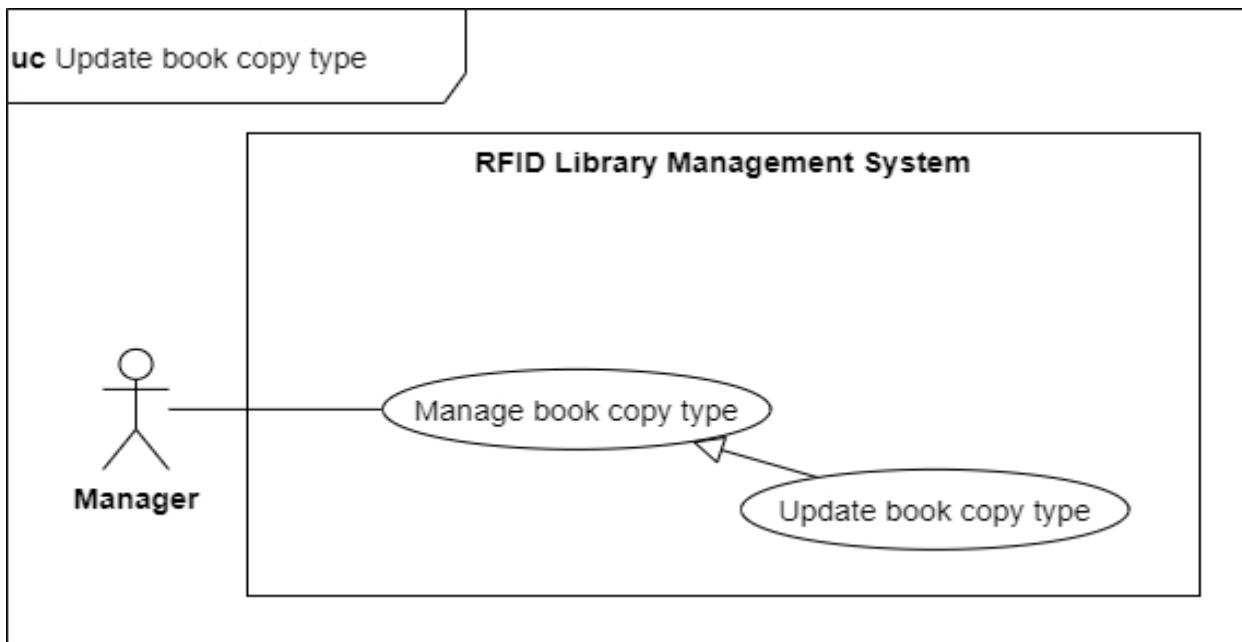


Figure 66 <Use Case> Update book copy type

ID and Name:	UC_56 Update book copy type		
Created By:	KienNT	Date Created:	25/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager update a book copy type.		
Trigger:	The manager sends a request to update a book copy type.		
Preconditions:	PRE-1: The user has logged in as Manager role. PRE-2: The updating book copy type is stored in the database.		
Postconditions:	POST-1: Book copy type is updated in the system POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager inputs new values for editable fields and clicks the update icon button from the book copy type	RLMS shows a confirmation dialog.

		screen.	
	2	The manager clicks the “OK” button. [Alternative 1]	RLMS checks validation, saves updated type to the database and pops up a successful message. [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Close” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The system cannot find the updating book copy type.	RLMS shows an error message.
	2	The manager inputs invalid fields.	RLMS shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	1. The manager can only update the following fields: • Name: required, max 100 2. Manager cannot set a duplicated name for the book copy type.		
Other Information	N/A		
Assumptions:	N/A		

Table 60 <Use Case> Update book copy type

2.2.5.16. Delete book copy type

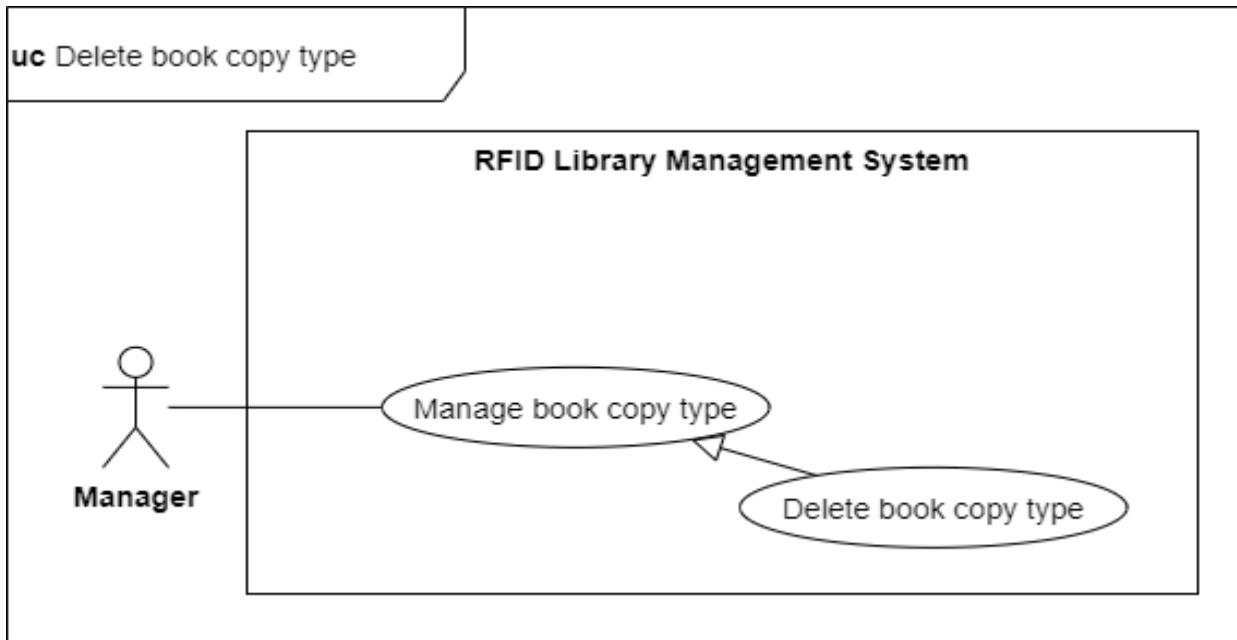


Figure 67 <Use Case> Delete book copy type

ID and Name:	UC_57 Delete book copy type		
Created By:	KienNT	Date Created:	25/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager remove an existing book copy type.		
Trigger:	The manager requests to remove a book copy type.		
Preconditions:	PRE-1: The user has logged in as Manager role. PRE-2: The book copy type is stored in the database.		
Postconditions:	POST-1: The book copy type is removed in the database. POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks the remove icon button from the book copy type screen.	RLMS shows a confirmation dialog.

	2	The manager clicks the “OK” button. [Alternative 1]	RLMS removes that book copy type from the database and pops up a successful message. [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Close” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The system cannot find the removing book copy type.	RLMS shows an error message.
	2	The book copy type is being used.	RLMS shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	1. The manager cannot delete the in-use book copy type.		
Other Information	N/A		
Assumptions:	N/A		

Table 61 <Use Case> Delete book copy type

2.2.5.17. View patron type

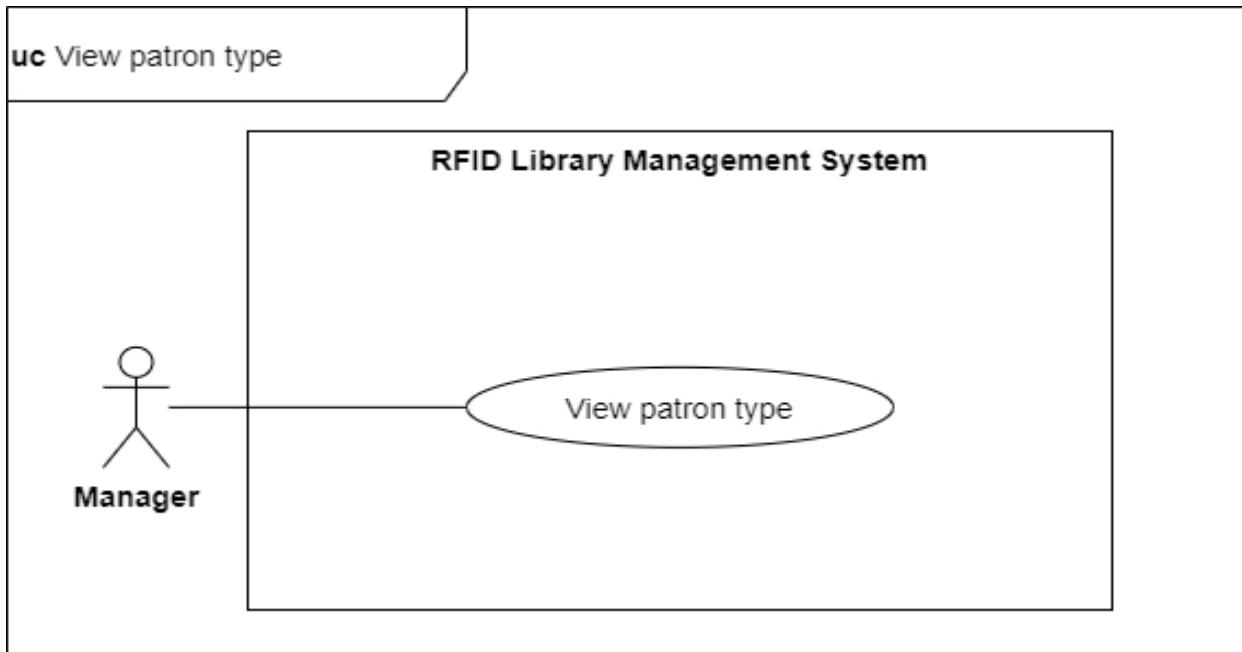


Figure 68 <Use Case> View patron type

ID and Name:	UC_58 View patron type		
Created By:	KienNT	Date Created:	25/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Provide the manager with the ability to view Patron Type. It is for showing which patron type is available in the system.		
Trigger:	The Manager sends request to view the policy		
Preconditions:	PRE-1: The user has logged in as Manager role		
Postconditions:	POST-1: RLMS shows a table containing the Patron Type POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The Manager sends request to view the Patron Type [Exception 1]	The RLMS processes the request, then shows a table containing the Patron Type

Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Each records in the Patron Type table contains: <ul style="list-style-type: none"> • Name: Patron type name • Checkouts Allowed: Maximum number book copies can be borrowed by the corresponding patron type. This number includes the number of copies the patron keeping 2. Patron type name is unique 		
Other Information	N/A		
Assumptions:	N/A		

Table 62 <Use Case> View patron type

2.2.5.18. Create patron type

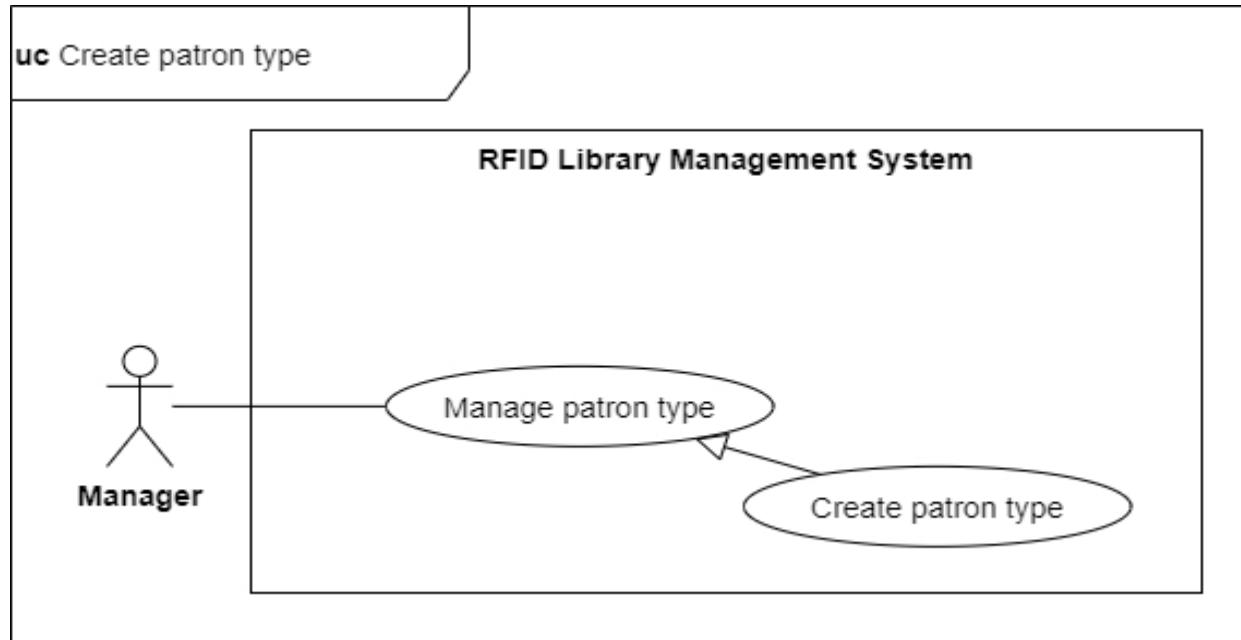


Figure 69 <Use Case> Create patron type

ID and Name:	UC_59 Create patron type		
Created By:	KienNT	Date Created:	25/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager add a new patron type.		
Trigger:	The manager sends a request to add a patron type.		
Preconditions:	PRE-1: The user has logged in as Manager role.		
Postconditions:	POST-1: Patron type is added in the system. POST-2: If failed, show error message		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks on the “Add New” button from the patron type screen.	RLMS shows add a new patron type form.
	2	The manager input required fields and click the “Confirm” button. [Alternative 1]	RLMS checks validation, saves a new patron type to the database and pops up a successful message. [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Cancel” button	RLMS close add patron type form.
Exceptions:	No.	Cause	System response
	1	The manager adds a duplicated name of patron type.	RLMS shows an error message.
	2	The manager inputs invalid fields.	RLMS shows error messages.

Priority:	Low
Frequency of Use:	Rarely
Business Rules:	<ol style="list-style-type: none"> 1. Adding patron type requires the following fields: <ul style="list-style-type: none"> • Name: required, max 100 • Checkouts allowed: required, between 0 to 100 2. Manager cannot set a duplicated name for patron type
Other Information	N/A
Assumptions:	N/A

Table 63 <Use Case> Add patron type

2.2.5.19. Update patron type

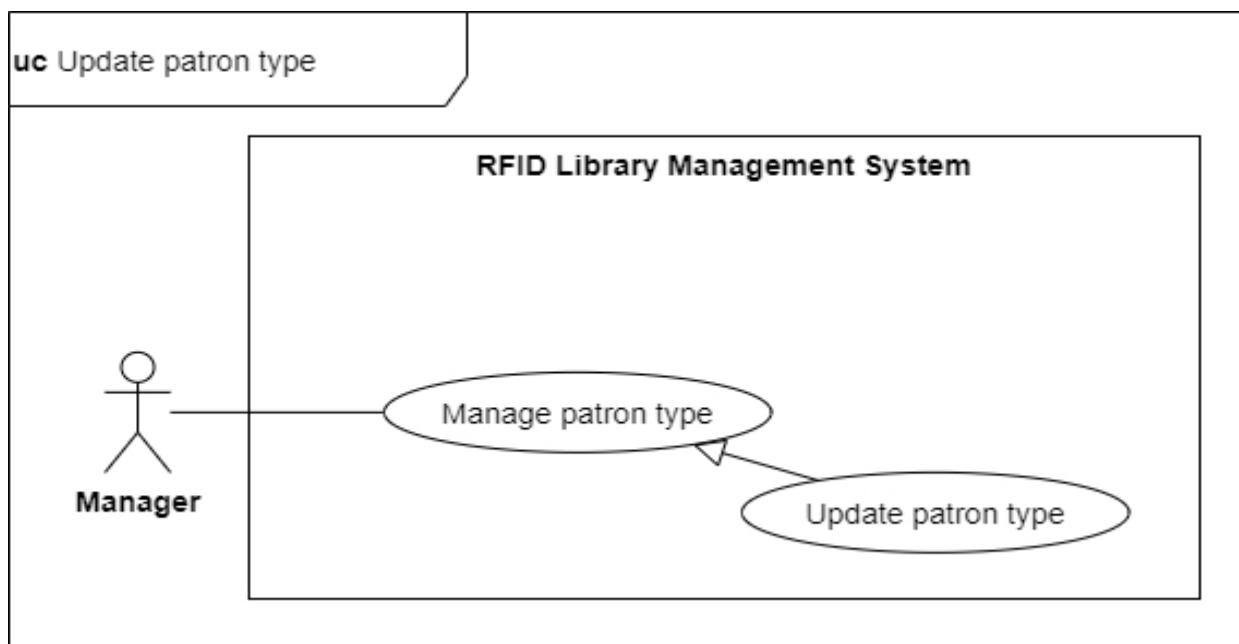


Figure 70 <Use Case> Update patron type

ID and Name:	UC_60 Update patron type		
Created By:	KienNT	Date Created:	25/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A

Description:	Let the manager update a patron type.		
Trigger:	The manager sends a request to update a patron type.		
Preconditions:	PRE-1: The user has logged in as Manager role. PRE-2: The updating patron type is stored in the database.		
Postconditions:	POST-1: Patron type is updated in the system. POST-2: If failed, show error message.		
Normal Flow:	Step	Actor action	System response
	1	The manager inputs new values for editable fields and clicks the update icon button from the patron type screen.	RLMS shows a confirmation dialog.
	2	The manager clicks the “OK” button. [Alternative 1]	RLMS checks validation, saves updated type to the database and pops up a successful message. [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Close” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The system cannot find the updating patron type.	RLMS shows an error message.
	2	The manager inputs invalid fields.	RLMS shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		

Business Rules:	<ol style="list-style-type: none"> 1. The manager can only update the following fields: <ul style="list-style-type: none"> • Name: required, max 100 • Checkouts allowed: required in range 0 - 100 2. Manager cannot set duplicated type for the patron type.
Other Information	N/A
Assumptions:	N/A

Table 64 <Use Case> Update patron type

2.2.5.20 Delete patron type

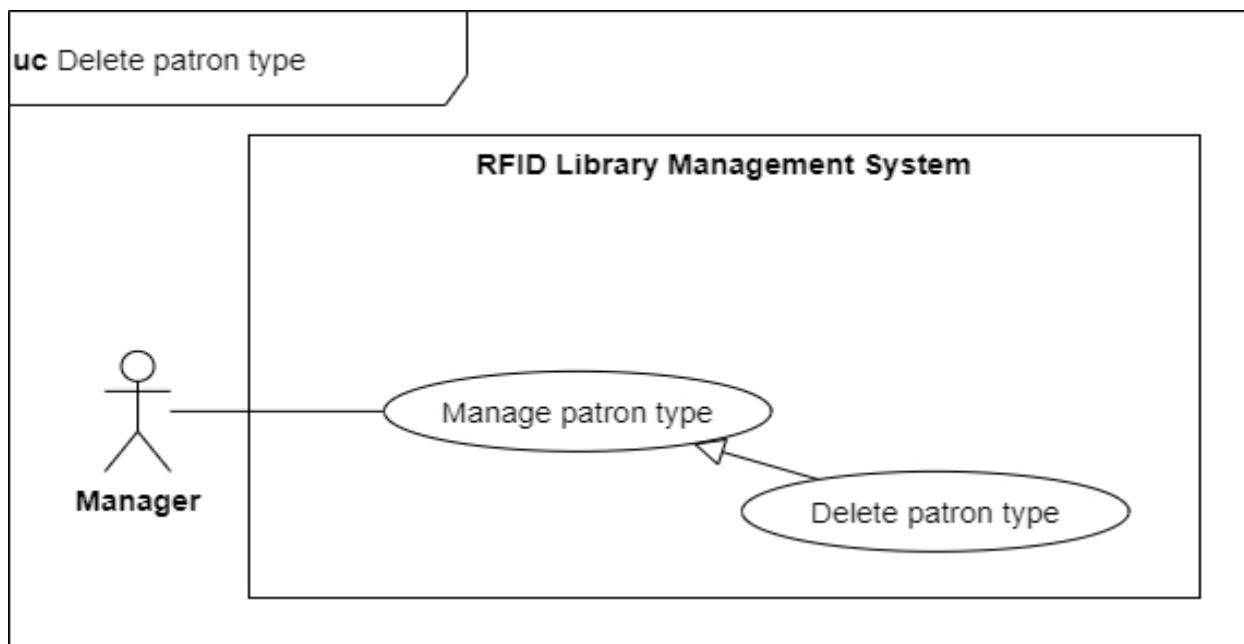


Figure 71 <Use Case> Delete patron type

ID and Name:	UC_61 Delete book copy type		
Created By:	KienNT	Date Created:	25/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Let the manager remove an existing patron type.		
Trigger:	The manager requests to remove a patron type.		

Preconditions:	PRE-1: The user has logged in as Manager role. PRE-2: The patron type is stored in the database.		
Postconditions:	POST-1: The patron type is removed in the database. POST-2: If failed, show error message.		
Normal Flow:	Step	Actor action	System response
	1	The manager clicks the remove icon button from the patron type screen.	RLMS shows a confirmation dialog.
	2	The manager clicks the “OK” button. [Alternative 1]	RLMS removes that patron type from the database and pops up a successful message. [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor action	System response
	1.1	The manager clicks “Close” button	RLMS closes the confirmation dialog.
Exceptions:	No.	Cause	System response
	1	The system cannot find the removing patron type.	RLMS shows an error message.
	2	The patron type is being used.	RLMS shows an error message.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	1. Manager cannot delete the in-use patron type.		
Other Information	N/A		
Assumptions:	N/A		

Table 65 <Use Case> Delete patron type

2.2.4.8. View patron list

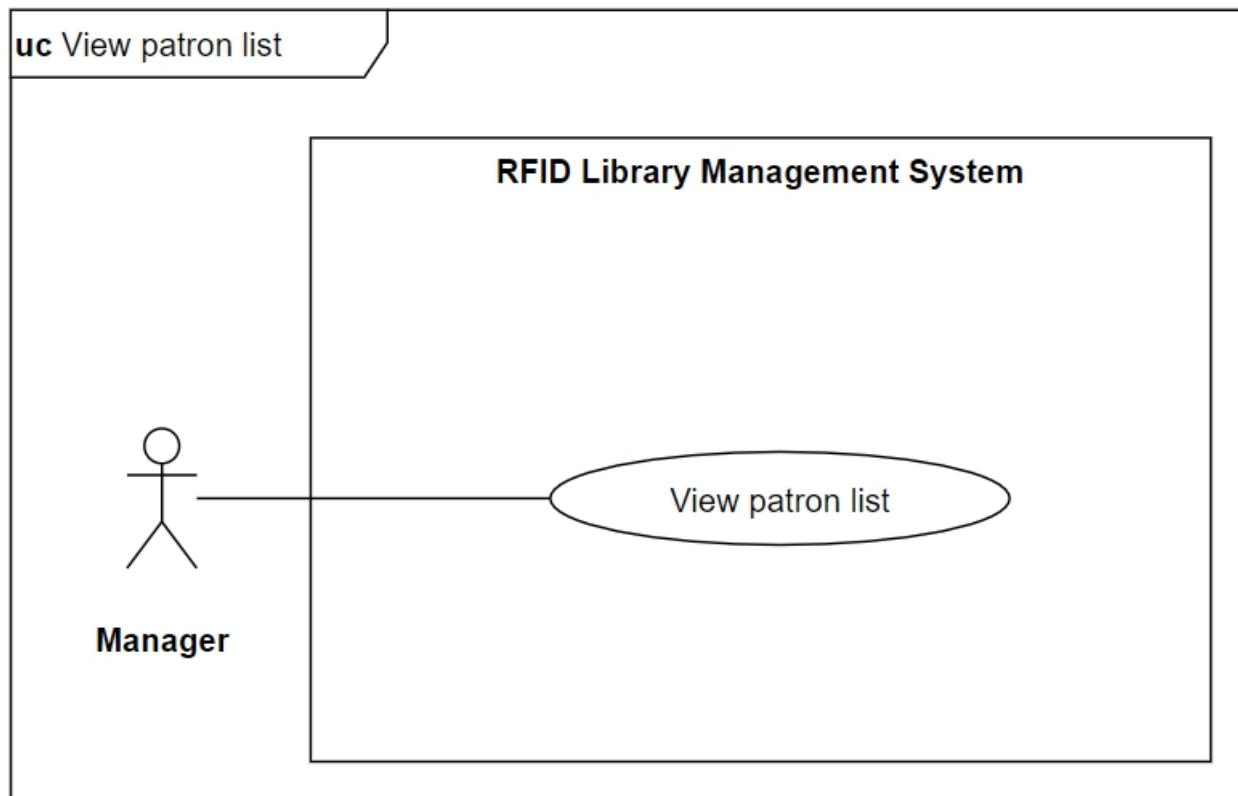


Figure 72 <Use Case> View patron list

ID and Name:	UC_62 View patron list		
Created By:	HoangPM	Date Created:	22/02/2021
Primary Actor:	Librarian	Secondary Actors:	N/A
Description:	Let the librarian search for a patron account		
Trigger:	The Librarian sends a request to get a list of patrons		
Preconditions:	PRE-1: The user has logged in as Librarian role		
Postconditions:	POST-1: If there is any search result, the RLMS shows a table contains a list of patrons POST-2: If there are no search result, the RLMS shows an error message		
Normal Flow:	Step	Actor action	System response

	1	The librarian sends a request to get a list of patrons [Exception 1]	The RLMS processes the request and returns a list of patrons to the UI.
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	The user retrieves an empty list	RLMS shows there is no data
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. Each record in the table includes: <ul style="list-style-type: none"> • Patron's Id • Patron's email • Patron's name • Patron's gender • Patron's phone • Patron's type • Patron's status (Active/Inactive) • A button to Enable or Disable the patron 2. The patron list can be filtered by the patron email 		
Other Information	N/A		
Assumptions:	N/A		

Table 66 <Use Case> View patron list

2.2.6. System Handler

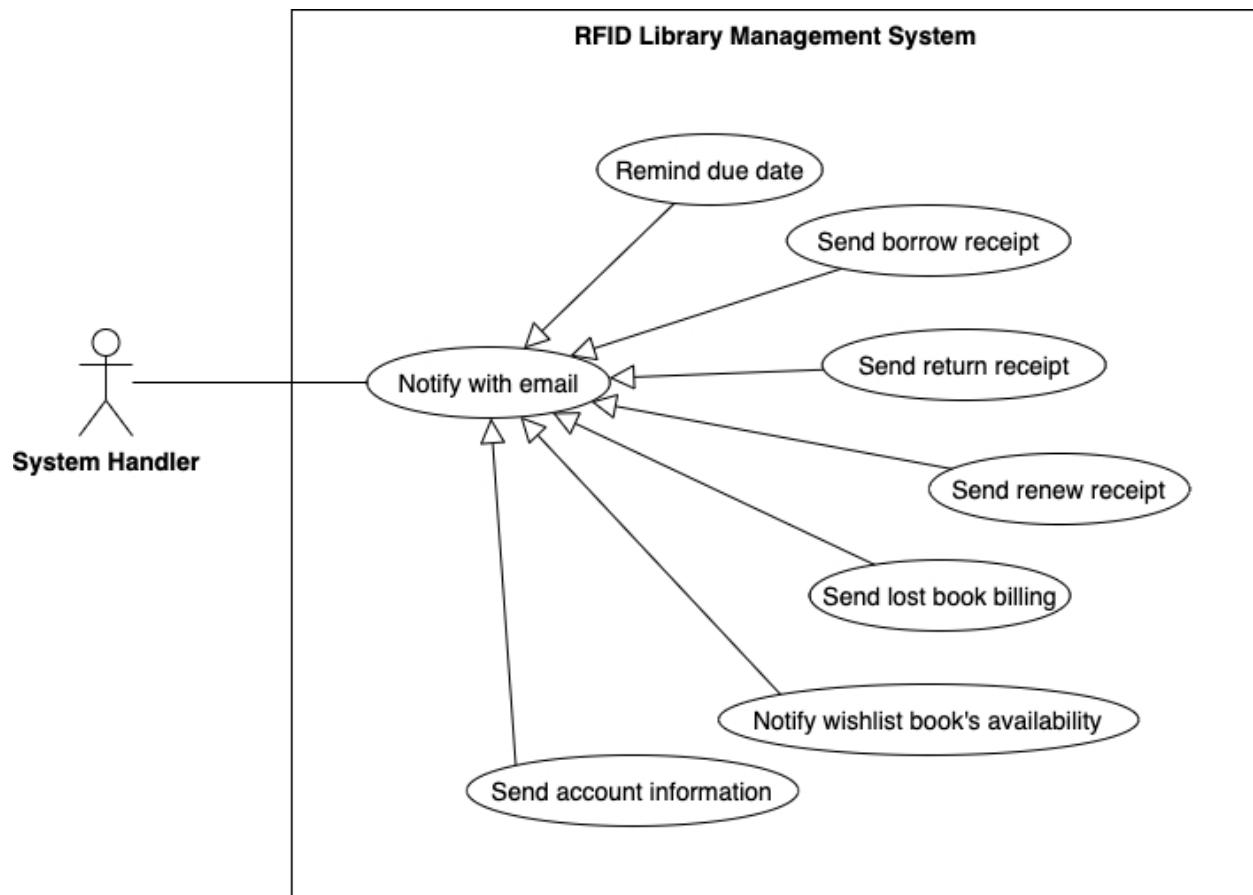


Figure 73 <Use Case Overview> System Handler

2.2.6.1. Send return receipt

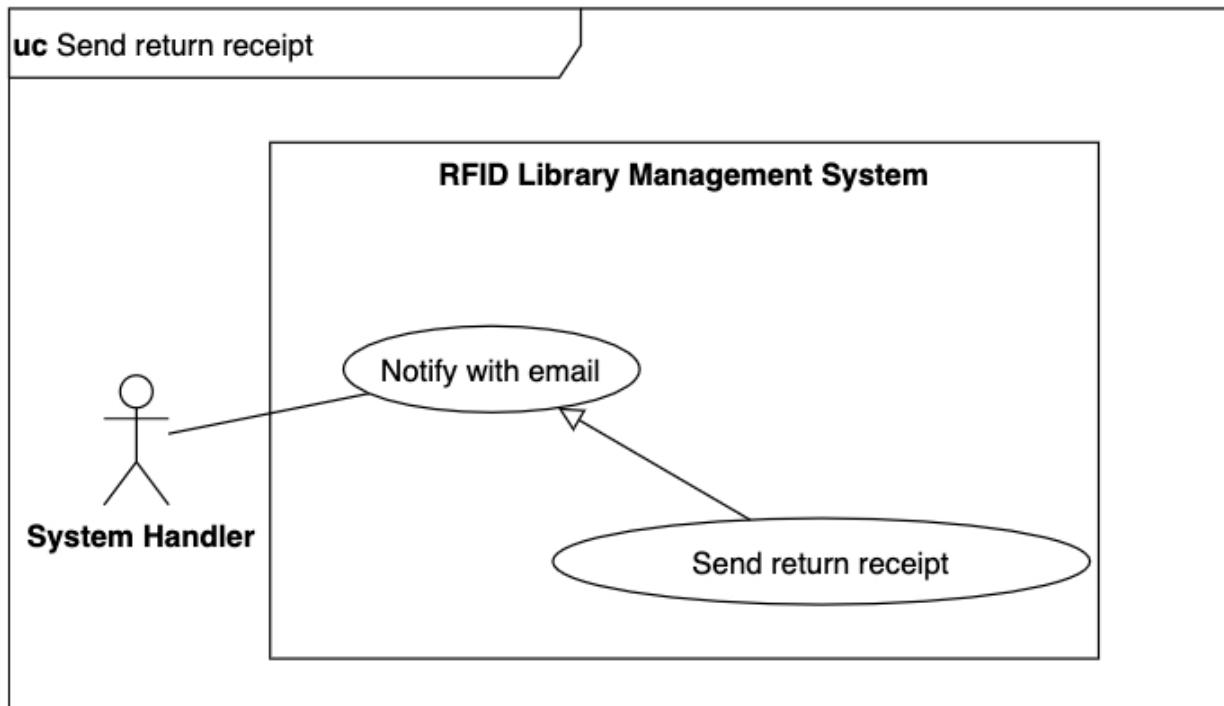


Figure 74 <Use Case> Send return receipt

ID and Name:	UC_63 Send return receipt		
Created By:	KhangNDN	Date Created:	23/02/2021
Primary Actor:	System Handler	Secondary Actors:	Patron, Librarian
Description:	System sends return receipt to patron's email		
Trigger:	After books returned successfully		
Preconditions:	PRE-1: Books are returned successfully.		
Postconditions:	POST-1: Patrons receive return receipt email. POST-2: If failed, log error		
Normal Flow:	Step	Actor action	System response
	1	System handler gets returned book information.	RLMS sends return receipts by email to appropriate patrons.

			[Exception 1]
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	System cannot send email.	System logs error.
Priority:	Low		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. The patrons should receive a return receipt by email after returning any book. 2. System only emails books that returned successfully. 3. The email shows: <ul style="list-style-type: none"> • Patron's full name • A list of returned books: title, subtitle, edition, returned at, overdue days, fine (VND) 		
Other Information	<ol style="list-style-type: none"> 1. The patron need to wait about 4-5s to receive the email 		
Assumptions:	N/A		

Table 67 <Use Case> Send return receipt

2.2.6.2. Send borrow receipt

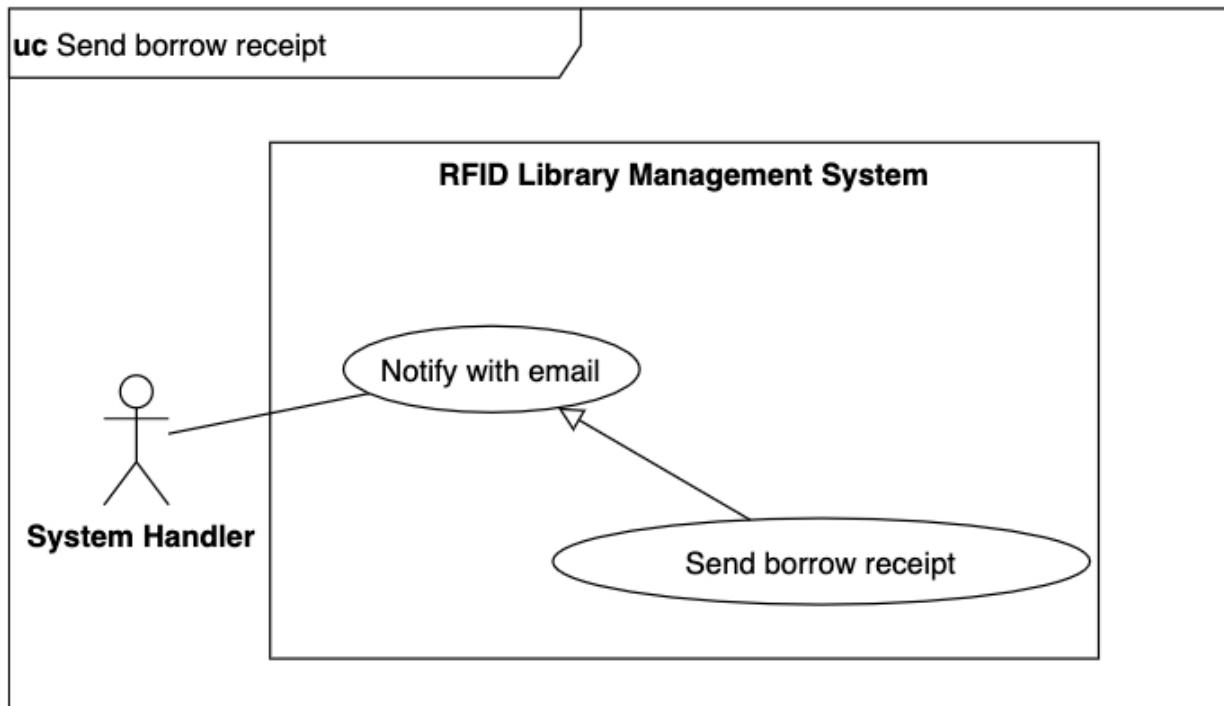


Figure 75 <Use Case> Send borrow receipt

ID and Name:	UC_64 Send borrow receipt		
Created By:	KhangNDN	Date Created:	23/02/2021
Primary Actor:	System Handler	Secondary Actors:	Patron, Librarian
Description:	System sends checkout receipt to patron's email		
Trigger:	After books borrowed successfully		
Preconditions:	PRE-1: Books are borrowed successfully.		
Postconditions:	POST-1: Patrons receive checkout receipt email. POST-2: If failed, log error		
Normal Flow:	Step	Actor action	System response
	1	System handler gets borrowed book information.	RLMS sends checkout receipts by email to appropriate patrons.

			[Exception 1]
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	System cannot send email.	System logs error.
Priority:	Low		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. The patrons should receive a checkout receipt by email after borrowing any book. 2. System only emails books that are borrowed successfully. 3. The email shows: <ul style="list-style-type: none"> • Patron's full name • A list of returned books: title, edition, borrowed at, due date • Applied fine policy 		
Other Information	<ol style="list-style-type: none"> 1. The patron need to wait about 4-5s to receive the email 		
Assumptions:	N/A		

Table 68 <Use Case> Send borrow receipt

2.2.6.3. Remind due date

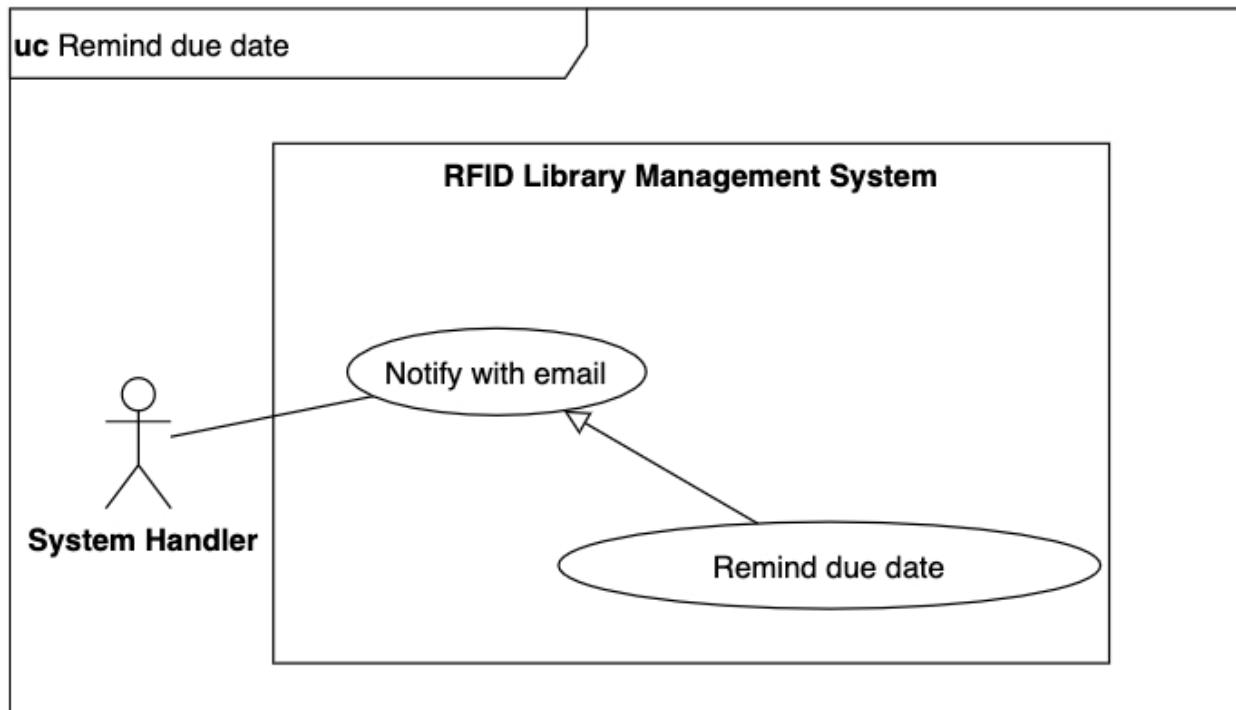


Figure 76 <Use Case> Remind due date

ID and Name:	UC_65 Remind due date		
Created By:	TramPH	Date Created:	23/02/2021
Primary Actor:	System Handler	Secondary Actors:	N/A
Description:	System sends email to remind books about overdue.		
Trigger:	The scheduler runs at 1:00 AM every day.		
Preconditions:	N/A		
Postconditions:	POST-1: Patrons receive overdue reminder email. POST-2: If failed, log error		
Normal Flow:	Step	Actor action	System response
	1	System handler gets all books about overdue.	RLMS sends overdue reminders by email to appropriate patrons.

			[Exception 1]
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	System cannot send email.	System logs error.
Priority:	Low		
Frequency of Use:	Once a day		
Business Rules:	<ol style="list-style-type: none"> 1. The patrons should receive an overdue reminder by email 1 day before the due date. 2. The email shows: <ul style="list-style-type: none"> • Patron's full name • Book information • Fine rate (VND) 3. The system handler checks for overdue soon books and sends emails to remind the patron every day at 1 AM by default 		
Other Information	<ol style="list-style-type: none"> 1. The processing time depend on the number of borrowing record 		
Assumptions:	N/A		

Table 69 <Use Case> Remind due date

2.2.6.4. Notify wishlisted book's availability

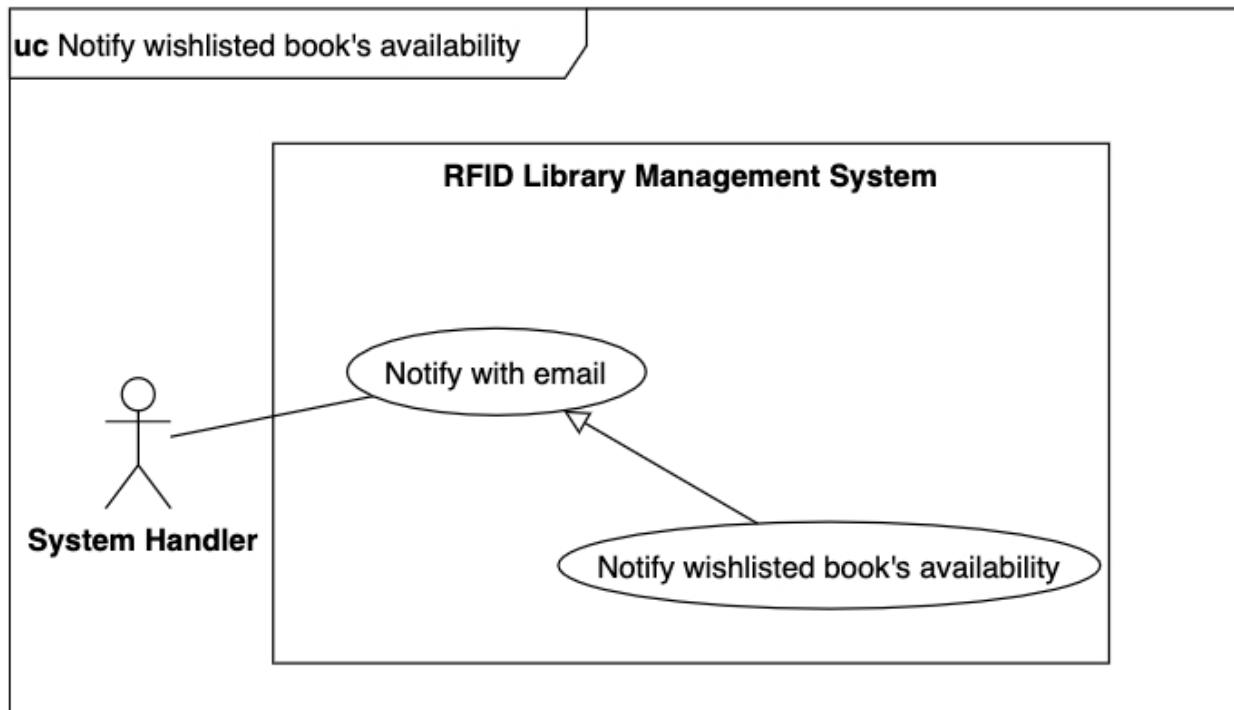


Figure 77 <Use Case> Notify wishlisted book's availability

ID and Name:	UC_66 Notify wishlisted book's availability		
Created By:	TramPH	Date Created:	23/02/2021
Primary Actor:	System Handler	Secondary Actors:	N/A
Description:	System sends email to notify wishlist books are available.		
Trigger:	The scheduler runs at 1:00 AM every day.		
Preconditions:	N/A		
Postconditions:	POST-1: Patrons receive email that notify wishlist books are available. POST-2: If failed, log error		
Normal Flow:	Step	Actor action	System response
	1	System handler gets all books which are available in wishlist	RLMS sends email to notify that wishlist books are available to appropriate patrons. And remove

			the books from patrons wishlist [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	System cannot send email.	System logs error.
Priority:	Low		
Frequency of Use:	Once a day		
Business Rules:	<ol style="list-style-type: none"> 1. The patrons should receive an email to notify that their wishlist books are ready in the library. 2. The email shows: <ul style="list-style-type: none"> • Patron's full name • Book information 3. After the email is sent, remove the book from the patron's wishlist. 4. The system handler checks and sends wishlist emails every day at 1 AM. 		
Other Information	<ol style="list-style-type: none"> 1. The processing time depend on the number of book copy record 		
Assumptions:	N/A		

Table 70 <Use Case> Notify wishlisted book's availability

2.2.6.5. Send account information

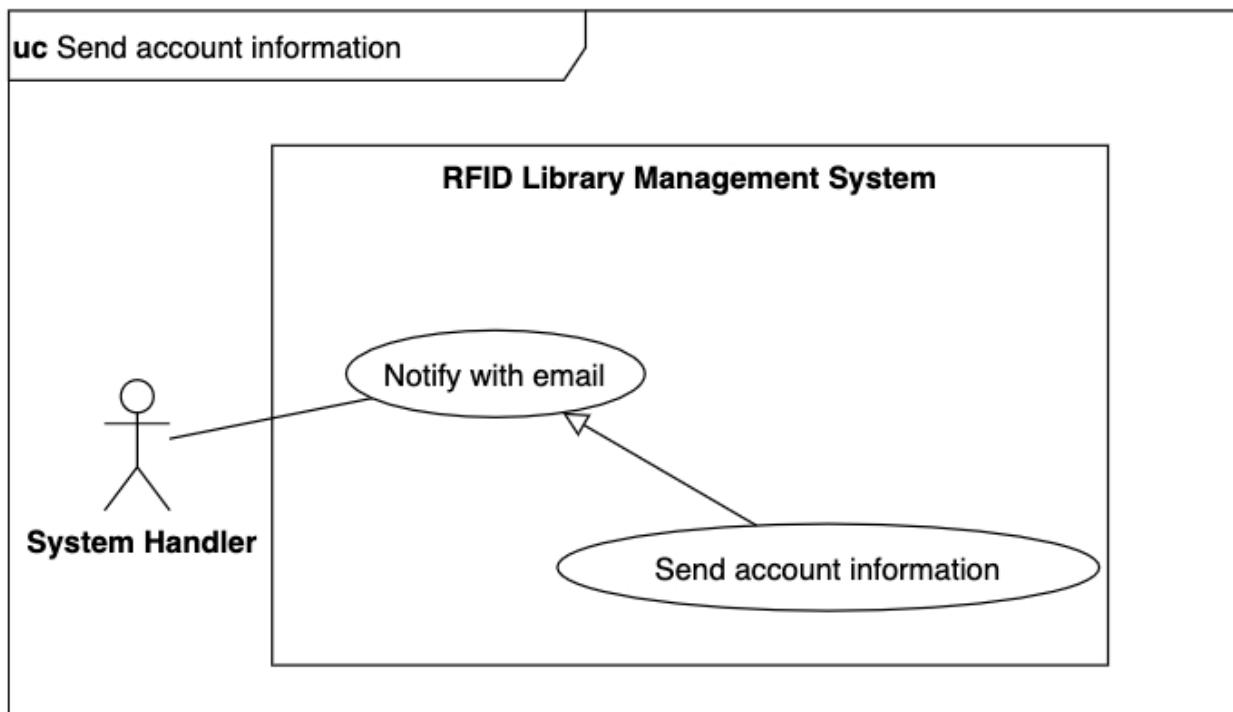


Figure 78 <Use Case> Send account information

ID and Name:	UC_67 Send account information		
Created By:	TramPH	Date Created:	23/02/2021
Primary Actor:	System Handler	Secondary Actors:	Patron, Librarian
Description:	System sends account information for the users to log into the system.		
Trigger:	After user's account is created		
Preconditions:	PRE-1: User's account is created and saved to the database.		
Postconditions:	POST-1: Patrons or librarians receive email that show account information for login. POST-2: If fail, log error		
Normal Flow:	Step	Actor action	System response

	1	System handler gets an account that has been registered.	RLMS sends email to provide registered account information to appropriate users. [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	System cannot send email.	System logs error.
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ol style="list-style-type: none"> 1. The users should receive an email to provide login information after being registered. 2. After receiving the login information, the user can log in to the system and change the password. 3. The email shows: <ul style="list-style-type: none"> • User's full name • Account information: username, password 		
Other Information	<ol style="list-style-type: none"> 1. Each semester the manager will import the patron information to the library system 		
Assumptions:	N/A		

Table 71 <Use Case> Send account information

2.2.6.6. Send lost book billing

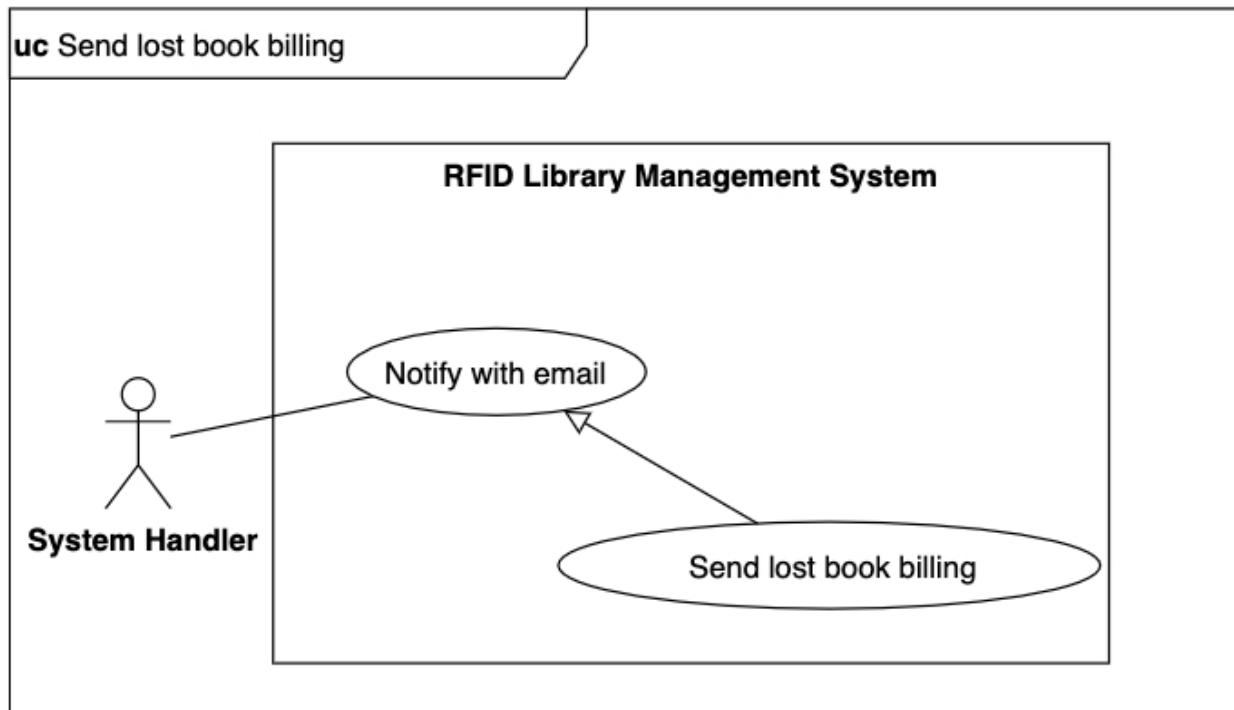


Figure 79 <Use Case> Send lost book billing

ID and Name:	UC_68 Send lost book billing		
Created By:	KhangNDN	Date Created:	24/03/2021
Primary Actor:	System Handler	Secondary Actors:	Patron, Librarian
Description:	System sends lost book billing (notice of fine) to the patron		
Trigger:	After the librarian confirm a lost book		
Preconditions:	PRE-1: The lost book status must be “CONFIRMED”		
Postconditions:	POST-1: The patrons receive a lost book billing via email POST-2: If failed, log error		
Normal Flow:	Step	Actor action	System response
	1	System handler gets the lost book record has been	RLMS send a lost book billing to

		confirmed	appropriate patron
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	System cannot send email.	System logs error.
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	1. The email shows: <ul style="list-style-type: none"> ● Patron's full name ● Time the patron report lost ● Fine (include lost fine and overdue fine) ● Book information 		
Other Information	1. The patron need to wait about 4-5s to receive the email		
Assumptions:	1. The system does not handle finance, only notify total lost fine to the patron		

Table 72 <Use Case> Send lost book billing

2.2.6.7. Send renew receipt

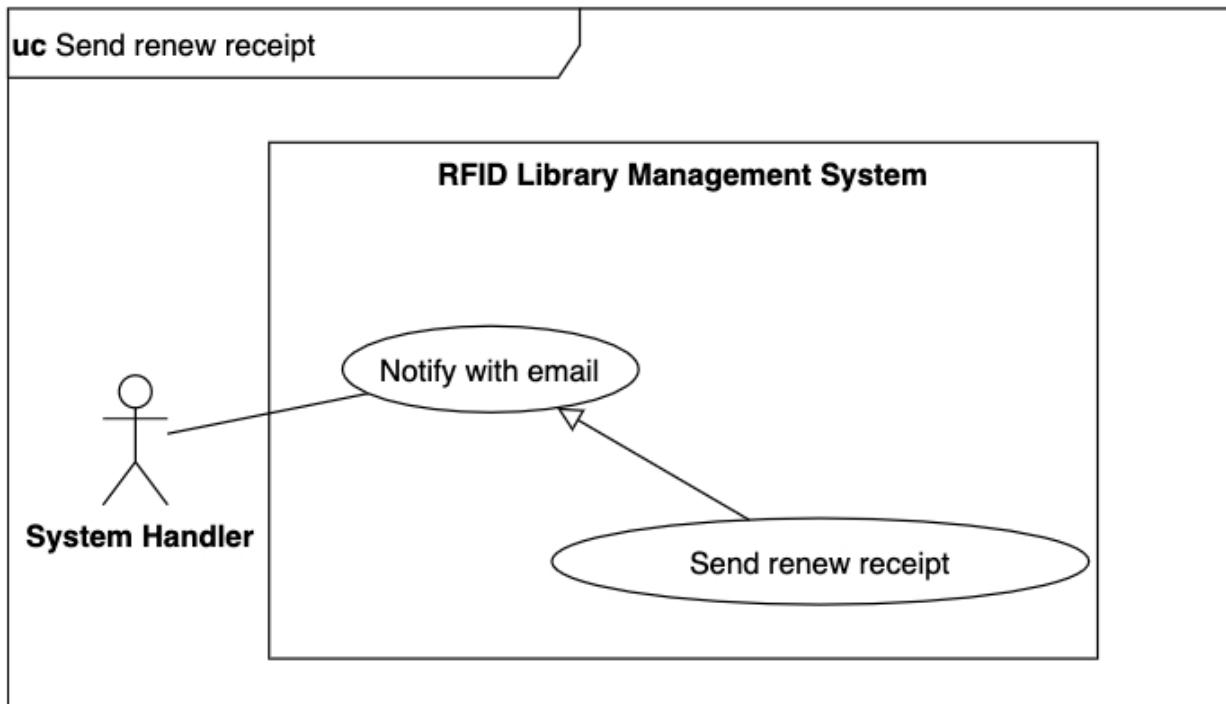


Figure 80 <Use Case> Send renew receipt

ID and Name:	UC_69 Send renew receipt		
Created By:	KhangNDN	Date Created:	24/04/2021
Primary Actor:	System Handler	Secondary Actors:	Patron, Librarian
Description:	System sends the renew receipt to the patron after renewing a book		
Trigger:	After the librarian or the patron renew a book		
Preconditions:	PRE-1: The book has been borrow and not overdue yet		
Postconditions:	POST-1: The patrons receive the renew receipt POST-2: If send fail, log error		
Normal Flow:	Step	Actor action	System response
	1	System handler gets the renew	RLMS send a renew receipt to the

		record	appropriated patron
Alternative Flows:	N/A		
Exceptions:	No.	Cause	System response
	1	System cannot send email.	System logs error.
Priority:	Low		
Frequency of Use:	Occasionally		
Business Rules:	1. The email shows: <ul style="list-style-type: none"> ● Patron's full name ● Renew time ● Old due date ● New due date ● Book information ● Applied fine policy 		
Other Information	1. The patron need to wait about 4-5s to receive the email		
Assumptions:	N/A		

Table 73 <Use Case> Send new receipt

2.2.7. Security Gate

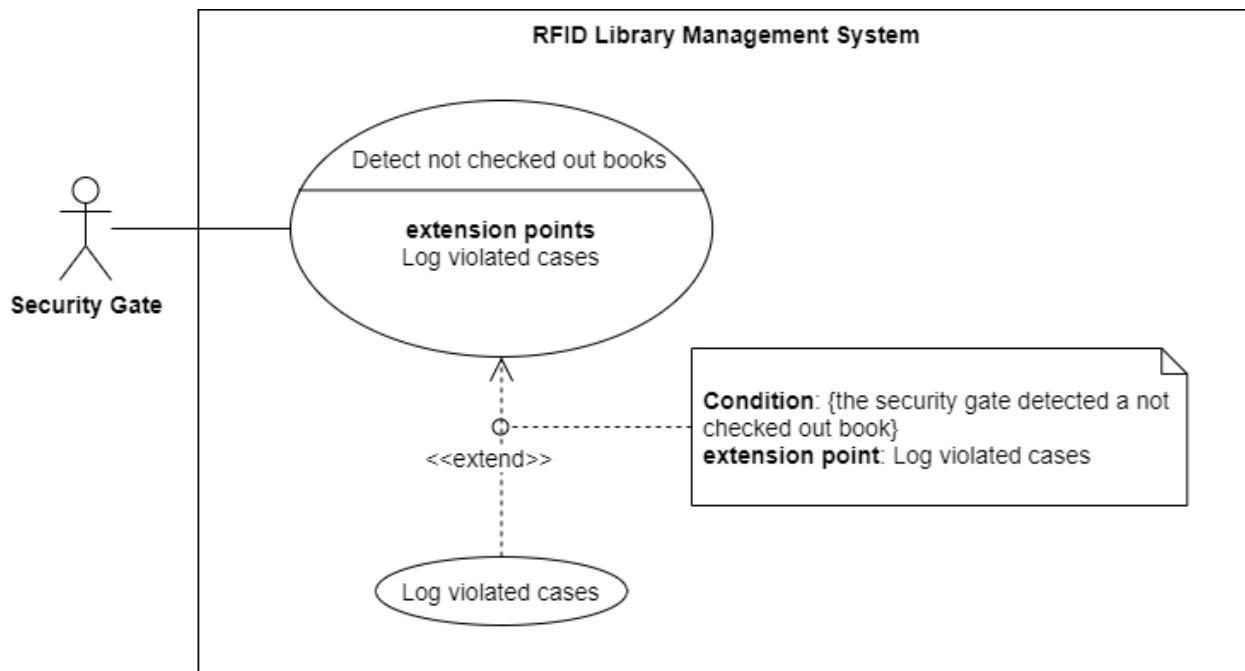


Figure 81 <Use Case Overview> Security Scanner

2.2.7.1. Detect not checked out books

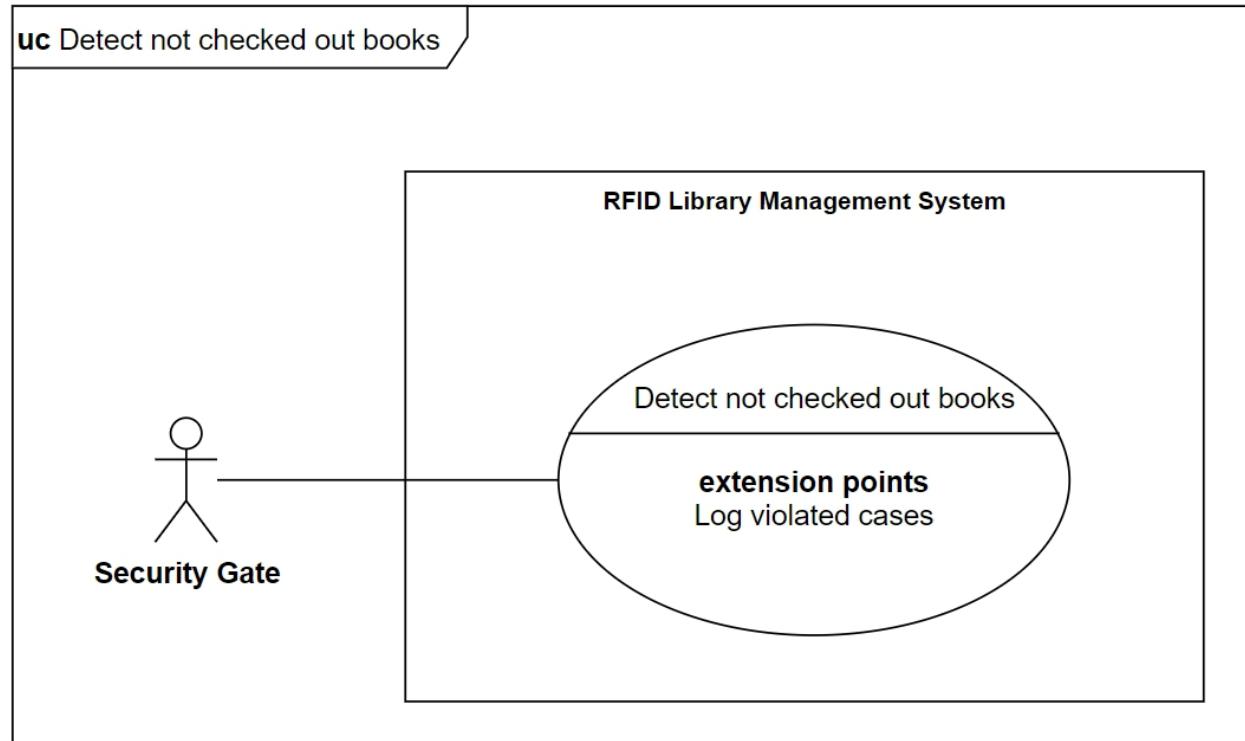


Figure 82 <Use Case> Detect not checked out books

ID and Name:	UC_70 Detect not checked out books		
Created By:	HoangPM	Date Created:	26/02/2021
Primary Actor:	Security Gate	Secondary Actors:	N/A
Description:	The security scanner will be able to detect and raise alarm if it detects a not checked out book passing by		
Trigger:	The security scanner detects a book presence		
Preconditions:	PRE-1: The UHF reader and alarm module is connected to the security gate controller		
Postconditions:	POST-1: Raise the alarm if detects a not checked out book		
Normal Flow:	Step	Actor action	System response
	1	Security Scanner detects a book presence and send request to check that book	RLMS processes the request. If the book is not checked out, the RLMS logs the book information and tells the security scanner to raise the alarm, else do nothing.
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	Usually		
Business Rules:	<ol style="list-style-type: none"> 1. The security gate should be able to check multiple books at a time 2. The security gate logs the book information if the book is not checked out 		
Other Information	N/A		
Assumptions:	1. Patrons are not allowed to bring their bags inside the library		

Table 74 <Use Case> Detect not checked out books

2.2.7.2. Log violated cases

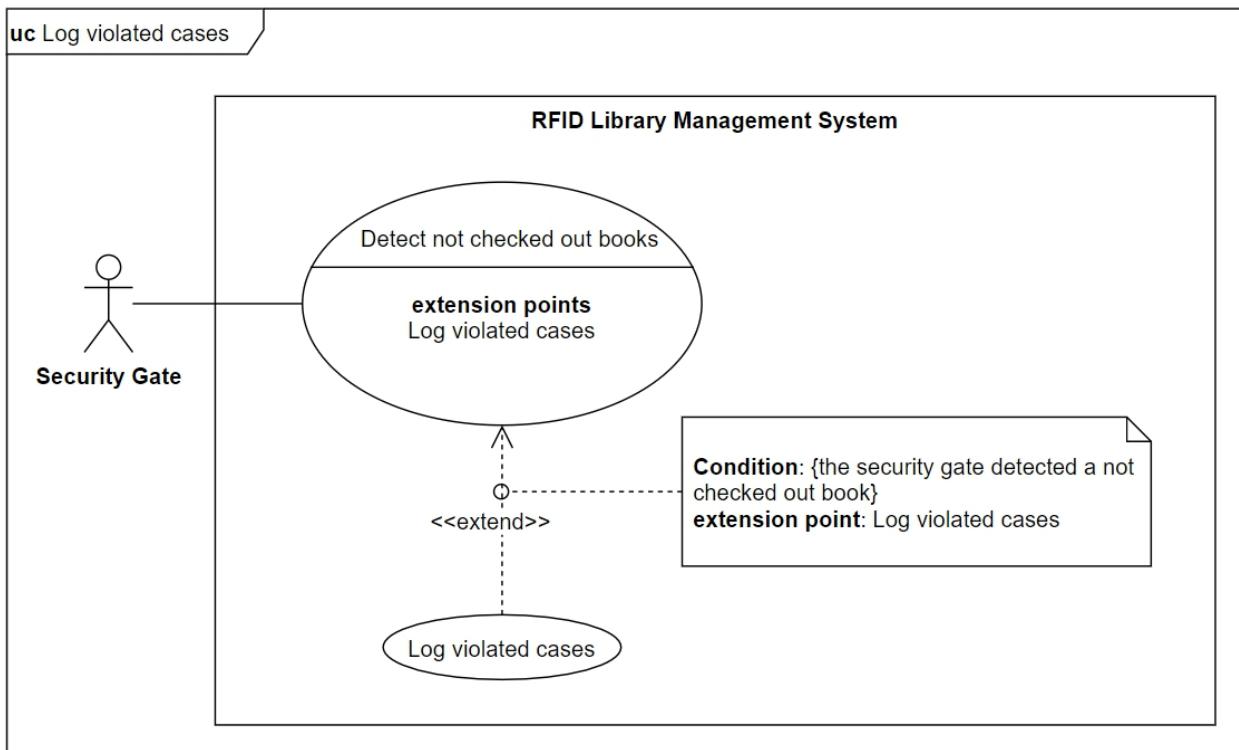


Figure 83 <Use Case> Log violated cases

ID and Name:	UC_71 Log violated cases		
Created By:	HoangPM	Date Created:	03/03/2021
Primary Actor:	Manager	Secondary Actors:	N/A
Description:	Provide the security scanner with the ability to save security alarm log		
Trigger:	After the alarm was raised, the security scanner sends a request to insert a new log record to the database		
Preconditions:	N/A		
Postconditions:	POST-1: A new log record is inserted into the database		
Normal Flow:	Step	Actor action	System response
	1	The security scanner sends a request to log a new record to	The RLMS processes the request and inserts a new log record to the

		the database	database
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	1. The logging time is based on when the alarm was raised		
Other Information	N/A		
Assumptions:	N/A		

Table 75 <Use Case> Log violated cases

3. Functional Requirements

3.1. System Functional Overview

3.1.1. Screen Flow

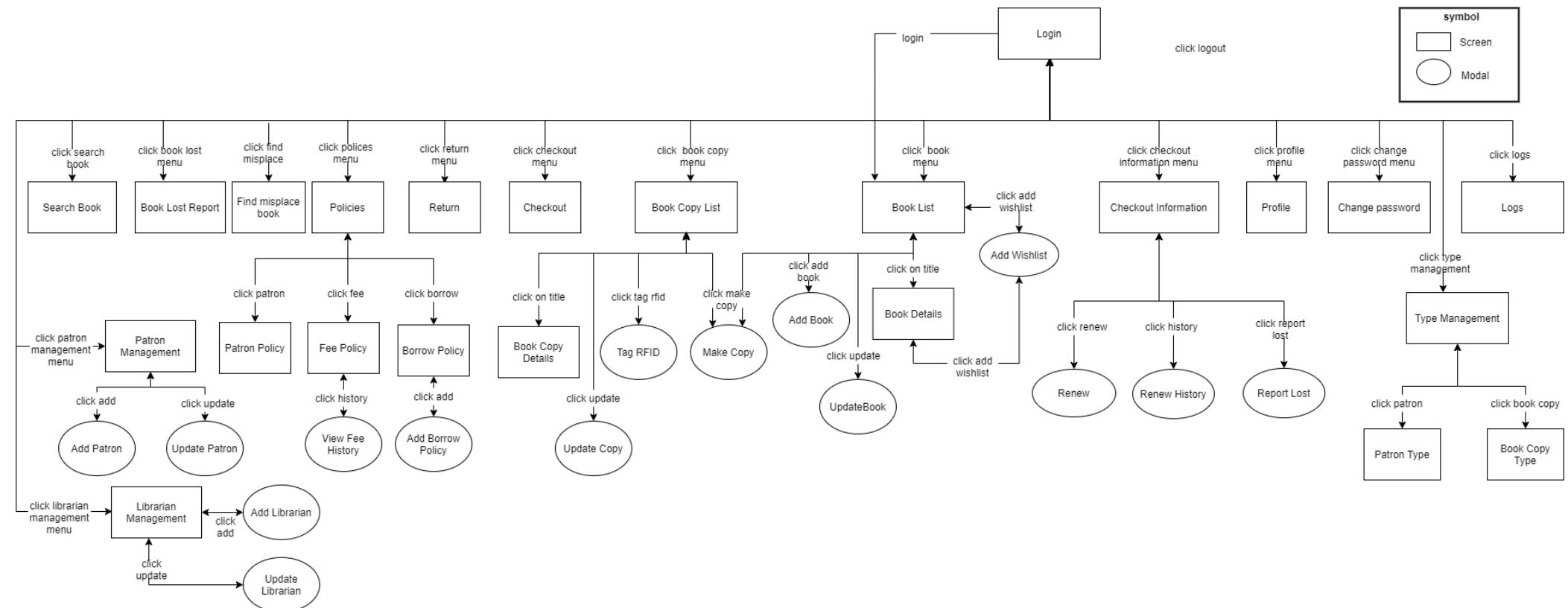
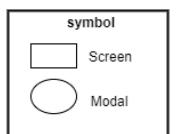


Figure 84 <Screen Flow> Manager Web Application

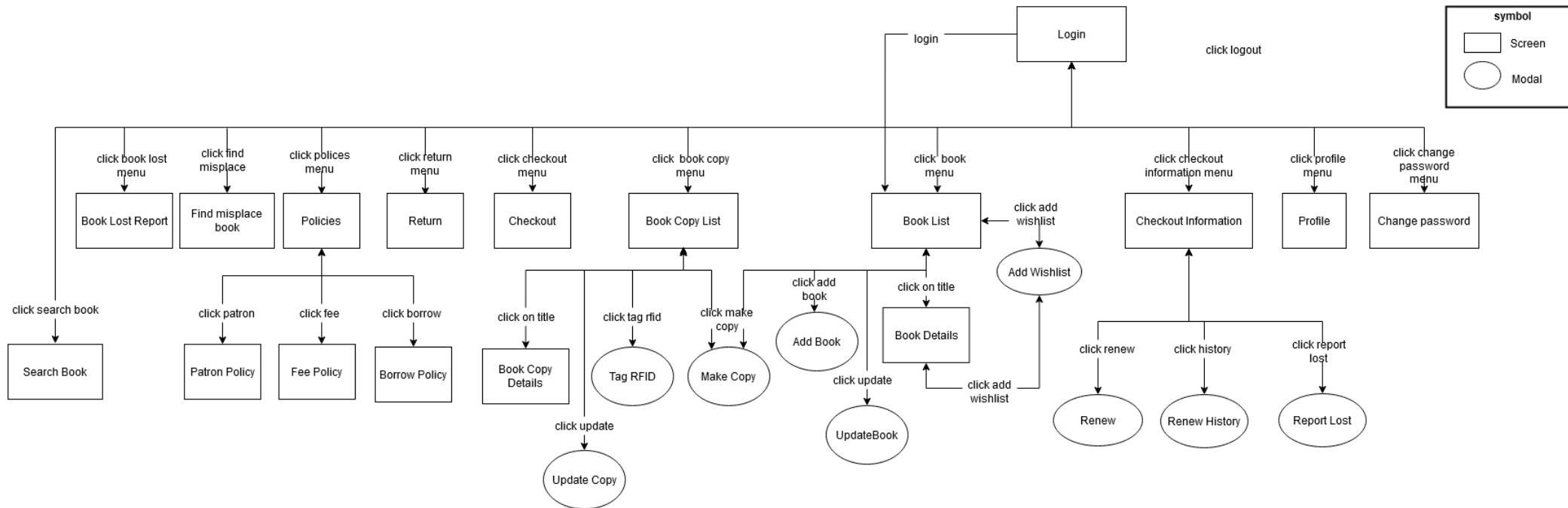


Figure 85 <Screen Flow> Librarian Web Application

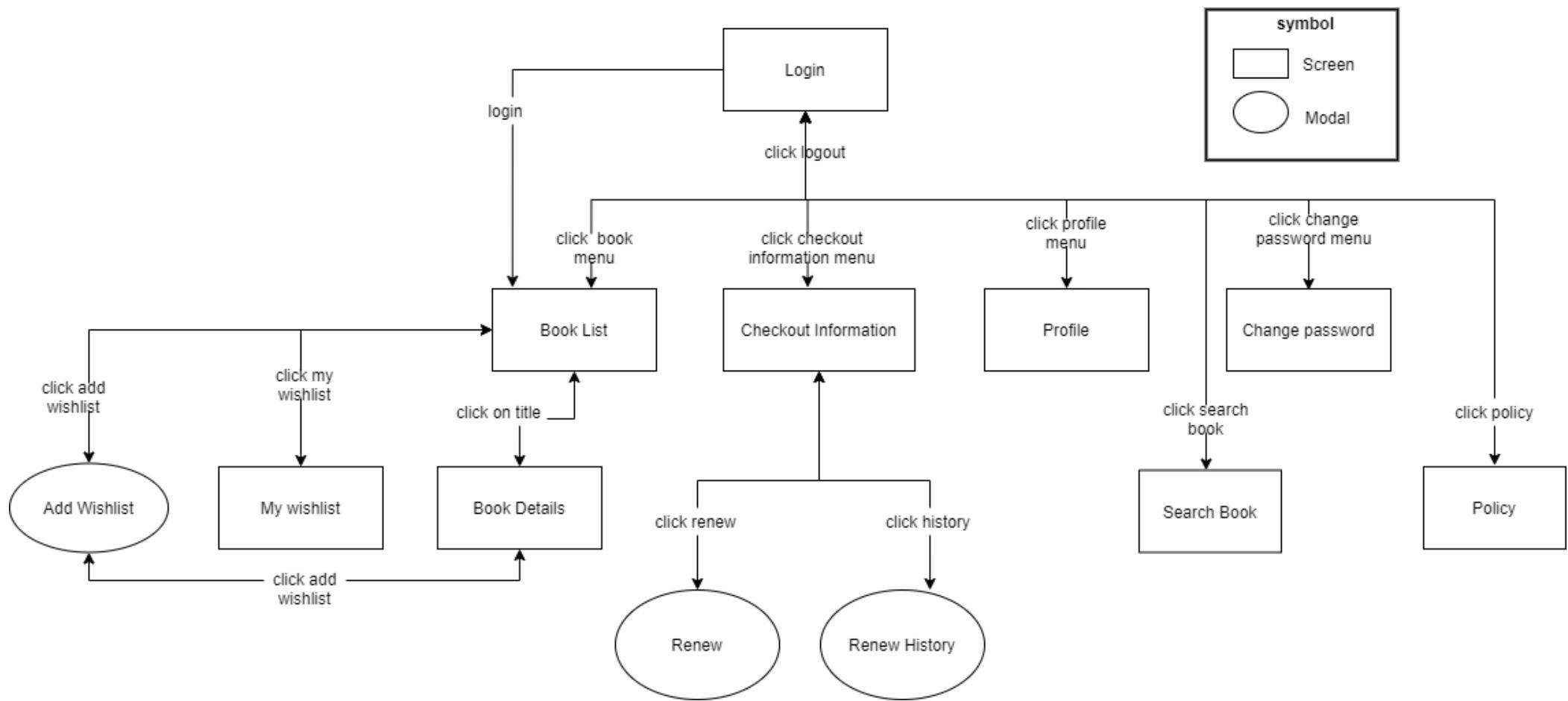


Figure 86 <Screen Flow> Patron Web Application

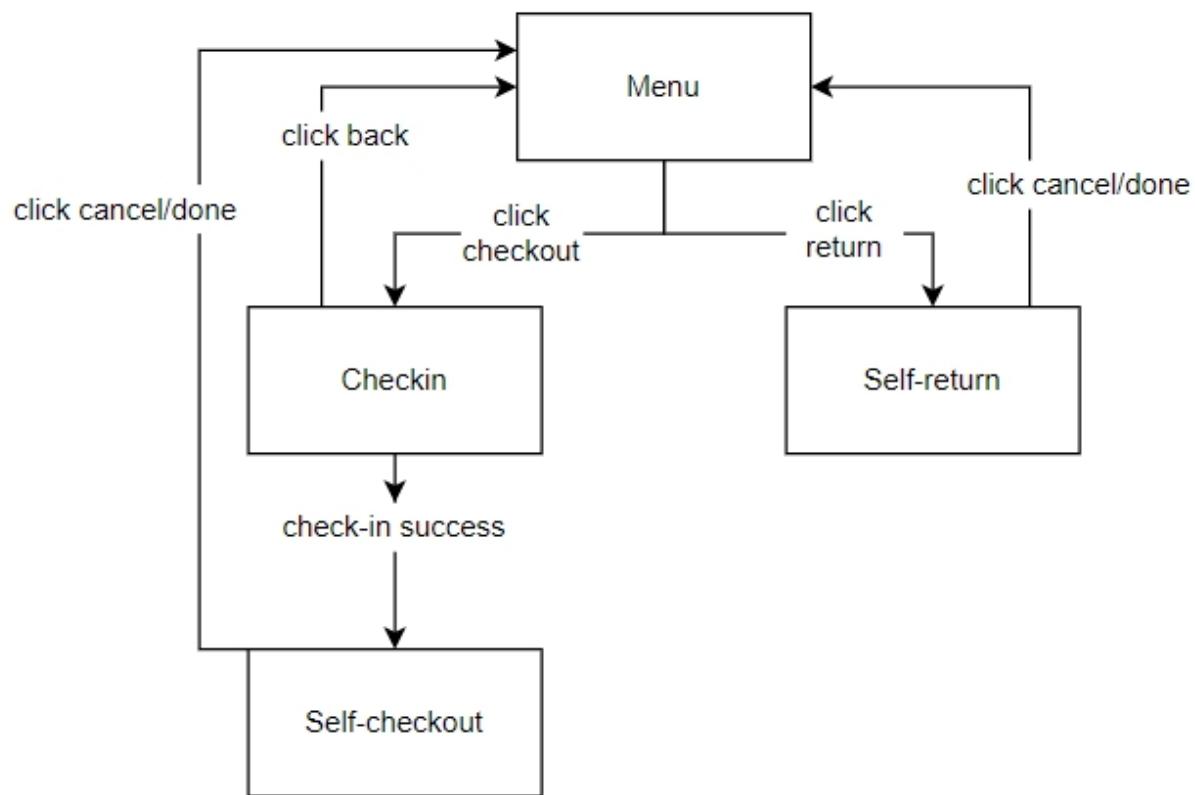


Figure 87 <Screen Flow> Self Return and Checkout Kiosk

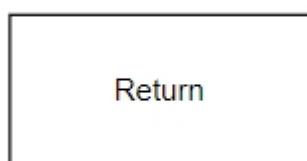


Figure 88 <Screen Flow> Book Drop Application

3.1.2. Screen Details

#	Feature	Screen	Description
01	User Authorization	Login	Determine if the user is a member of the system or not, then check their role to redirect corresponding screens
02	Self-Return Book Drop	Return	Allow users to return at the book drop

			station outside library
03	Self-Return Checkout Kiosk	Menu	Allow users to choose options: return, checkout
04	Self-Return Checkout Kiosk	Check in	Determine if the actor is a patron of the system and have the right to make self-checkout transaction
05	Self-Return Checkout Kiosk	Checkout	Allow patrons to self-checkout inside library
06	Self-Return Checkout Kiosk	Return	Allow users to self-return inside library
07	Book Management	Book	List all books in the system; allow users to add book to wishlist and search list
08	Book Management	Book Details	Show information and locations of a book
09	Book Management	Add Book	Allow users to add a new book
10	Book Management	Update Book	Allow users to update an existing book
11	Book Copy Management	Book Copy	List all copies in the system; allow user (librarian, manager) print multiple barcodes
12	Book Copy Management	Add Book Copy	Allow users to add new copies for a book
13	Book Copy Management	Update Book Copy	Allow users to update an existing copy
14	Book Copy Management	Tag RFID	Allow users to tag RFID for an existing copy
15	Book Copy Management	Book Copy Detail	Show information and location of a copy
16	Wishlist Management	My Wishlist	Show wishlist books that have emailed yet
17	Wishlist Management	Add Wishlist	Allow users to add an unavailable book to the wishlist
18	Checkout Management	Checkout Information	List all borrowed books of a user: returned books, borrowing books, overdue books
19	Checkout Management	Renew	Allow users to renew a borrowing book
20	Checkout Management	Renew History	Allow users to view renew history
21	Checkout For Patron	Checkout	Allow users to check out books for patrons
22	Return for Patron	Return	Allow users to return books for patrons
23	Policy Management	Borrow Policy	Show all borrow policies in the system,

			allow user to update, remove borrow policies
24	Policy Management	Add Borrow Policy	Add new borrow policy
25	Policy Management	Fee Policy	Show the latest fee policy
26	Policy Management	Fee Policy History	Show history of fee policy
27	Policy Management	Patron Policy	Show all the patron policies in the system
28	Lost Management	Book Lost Report	Show all book lost by date
29	Lost Management	Report Lost	Allow user to report lost books
30	Profile Management	Profile	Show profile information, allow user to update profile
31	Profile Management	Change Password	Allow user to change password
32	Librarian Management	Librarian List	Show all librarians in the system, allow user to deactivate, activate librarian
33	Librarian Management	Add Librarian	Allow user to add librarian
34	Librarian Management	Update Librarian	Allow user to update an librarian
35	Patron Management	Patron List	Show all patrons in the system, allow user to deactivate, activate patron
36	Librarian Management	Add Patron	Allow user to add patron
37	Librarian Management	Update Patron	Allow user to update an patron
38	Book Management	Search Book	Allow user to search book by UHF reader
39	Type Management	Patron Type	Allow user to view, edit, delete, add patron type
40	Type Management	Book Copy Type	Allow user to view, edit, delete, add book copy type
41	Security Log	View Log	Allow user to view activated book copies were logged in the system
42	Policy Management	View policies	Allow user to view the library's policies

Table 76 Screen Details

3.1.3. Screen Authorization

Screen	Patron	Librarian	Manager
Login	X	X	X
check in	X		
Self-checkout	X		
Self-return	X		
Book List	X	X	X
Book Details	X	X	X
Add Book		X	X
Update Book		X	X
Book Copy List		X	X
Book Copy Details		X	X
Add Book Copy		X	X
Update Book Copy		X	X
Tag RFID		X	X
My Wishlist	X		
Add Wishlist	X	X	X
Checkout		X	X
Return		X	X
Checkout Information	X	X	X
Renew	X	X	X
Renew History	X	X	X
Borrow Policy		X	X
Add Borrow Policy			X
Fee Policy		X	X
Fee Policy History		X	X

Patron Policy		X	X
Profile	X	X	X
Change Password	X	X	X
Librarian List			X
Add Librarian			X
Update Librarian			X
Patron List		X	X
Add Patron			X
Update Patron			X
View Policies	X	X	X
Search Book	X	X	X
Patron Type Management			X
Book Copy Management			X
Book Lost Report		X	X

Table 77 Screen Authorization

3.1.4. Non-Screen Functions

#	System Functions	Description
1	Automatically send remind overdue email	The system automatically check about overdue books then send email to appropriate patrons
2	Automatically send wishlist available email	The system automatically check which wishlist book is available then send email to appropriate patron and remove the that book from the patron's wishlist
3	Automatically send account information email	The system automatically generate password for new users then send email login information to them
4	Automatically send checkout receipt	The system automatically get checkout information and send the receipt to the patron after checkout immediately
5	Automatically send return receipt	The system automatically get return information and send the receipt to the patron after returning immediately
6	Raise alarm	The system automatically scan and raise security alarm if a

		not checkout book is brought outside of library
7	Save alarm log	When the security alarm was raised, the system automatically logs the current date time and the not checkout copy that was detected
8	Automatically send book lost fine	The system automatically get fine after the librarian confirm the lost book and send the fine information to the patron

Table 78 Non-Screen Functions

3.1.5. Entity Relationship Diagram

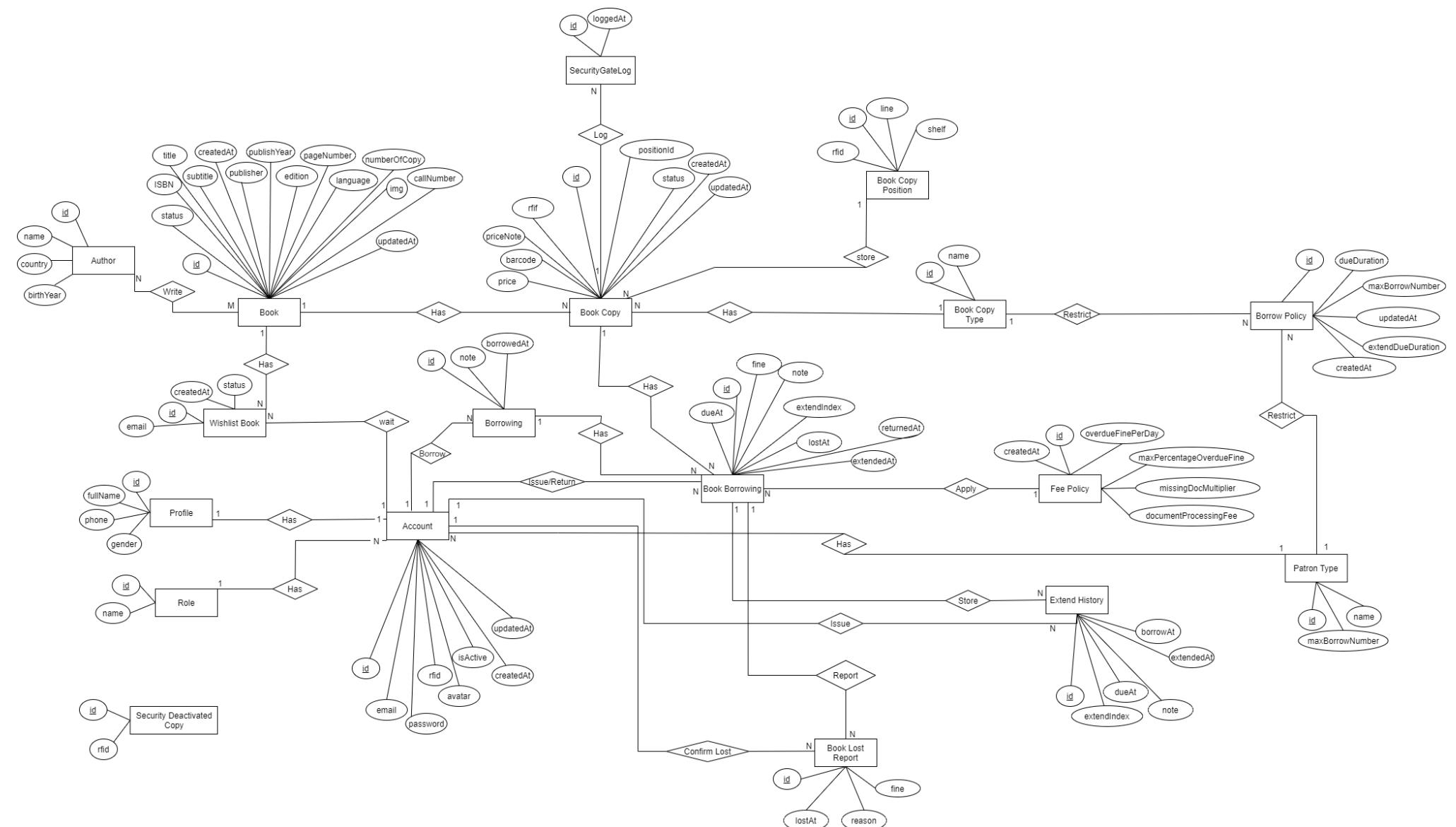


Figure 89 Entity Relationship Diagram

4. Non-Functional Requirements

4.1. External Interfaces

4.1.1. User Interfaces

UI-1: The system shall inform the user with a short and clear message when there is an error.

UI-2: The book drop shall have a user friendly windows form screen for the patron to interact. The patron just need to press 1 button to operate.

UI-3: When a patron brings a book that was not checked out to the library gate, the security alarm shall be raised to warn the user.

UI-4: The system shall allow the users to get book information using UHF RFID reader.

UI-5: The system shall allow the patron to check in with their patron card using HF RFID reader.

4.1.2. Software Interface

SI-1: Firebase

- SI-1.1: The RLMS shall upload image to firebase storage
- SI-1.2: The RLMS shall get image url from the firebase storage

SI-2: UHFReader86Demo V1.11

- SI-2.1: The RLMS shall get RFID tag UID via serial port from UHFReader86Demo V1.11

SI-3: Eltima Virtual Serial Port Driver 6.9

- SI-3.1: The RLMS shall rely on Eltima Virtual Serial Port Driver 6.9 to communicate with other devices like UHF reader, HF reader, drop box, security alarm, etc.

4.1.3. Hardware Interface

HI-1: UHF Reader is required to read books RFID tag.

HI-2: HF Reader is required to read patrons card.

HI-3: Self-service machine self-service machine with GUI is required for patrons to self-checkout and return.

HI-4: Book drop is required for the patron to self-return outside of the library.

HI-5: Barcode reader is required to read the books barcode.

4.2. Quality Attributes

4.2.1. Usability

- UI should be in English
- UI and UX should accommodate the librarian daily tasks
- Text input fields should be validated
- The RLMS does not require training for the patrons, they should be able to use the web application, book drop and self-service machine immediately
- The RLMS requires 2 days of training for the librarian

- The RLMS requires 3 days of training for the manager

4.2.2. Reliability

- In case of application failures or errors, hotfix version can be deployed on the same working day
- The chance of application failure is less than 3%
- Security alarm can detect 95 out of 100 theives every cases

4.2.3. Availability

- The RLMS should be available 24/7
- The RLMS features like borrow and return shall have less than 30 minutes of down time each day for the system scheduler to run

4.2.4 Performance

- Common requests should have response time under 2 seconds
- RFID Reader should return UID of a tag instantly
- Security alarm should be able to scan 5 books at a time in under 1 second

4.2.5. Security

- The RLMS has 3 roles (Patron, Librarian, Manager), each role can only access its own group of functions
- The RLMS checks for authentication and authorization for every requests

4.2.6. Maintainability

- The RLMS applies controller-service-repository pattern based on Spring Framework
- Coding should be clear and follow the team code conventions
- RLMS is divided into components therefore each component can be updated or modified without affecting other the whole system

4.2.7. Portability

- The book drop and self-service machine is windows form application, and can only be used inside library or campus area
- Librarian and Manager's application is web based so it can run on operating system that supports Google Chrome version 90

4.2.8. Design Constraints

- Web application is developed using ReactJS
- Book drop and self-service machine is developed using Windows Form
- Web service is developed using Spring Boot
- Arduino board is coded using Arduino IDE

5.2.9. Purchased Components

- Firebase Storage is limited to 5GB storage in total, 50 000 requests or 1GB bandwidth/day

IV. Software Design Document

1. Overall Description

1.1. Assumptions

This system is designed basing on these following assumptions:

- Software/OS:
 - Window 10
 - Chrome browser version 90
 - WinForm application .Net Framework
 - Java SE 15
 - MySQL 8.0
- Hardware:
 - Computer with internet connection
 - UHF RFID reader
 - Arduino Uno
 - Arduino Pro Micro
 - RFID RC522 NFC
 - Stepper Motor 28BYJ-48 5V
 - Module driver ULN2003
 - NHF card
 - UHF tag
 - Led light RGB
 - Buzzer
 - Board bread
 - Male to female jumper wires
- End users:
 - Patron of school/university library: student, lecturer, school/university staff, etc.
 - Librarian
 - Library manager
- User requirements may change unexpectedly
- Detailed design is flexible to quickly adapt user's requirement

1.2. Design Constraints

This system should be compiled with following items:

- End-user environment: Window OS, Chrome
- Every process in this system must be less than 2 seconds
- Security requirement: prevent all unauthorized requests and prevent not checked out books from exiting the library as much as possible
- Network communication: this system depends on network connection

2. System Architecture Design

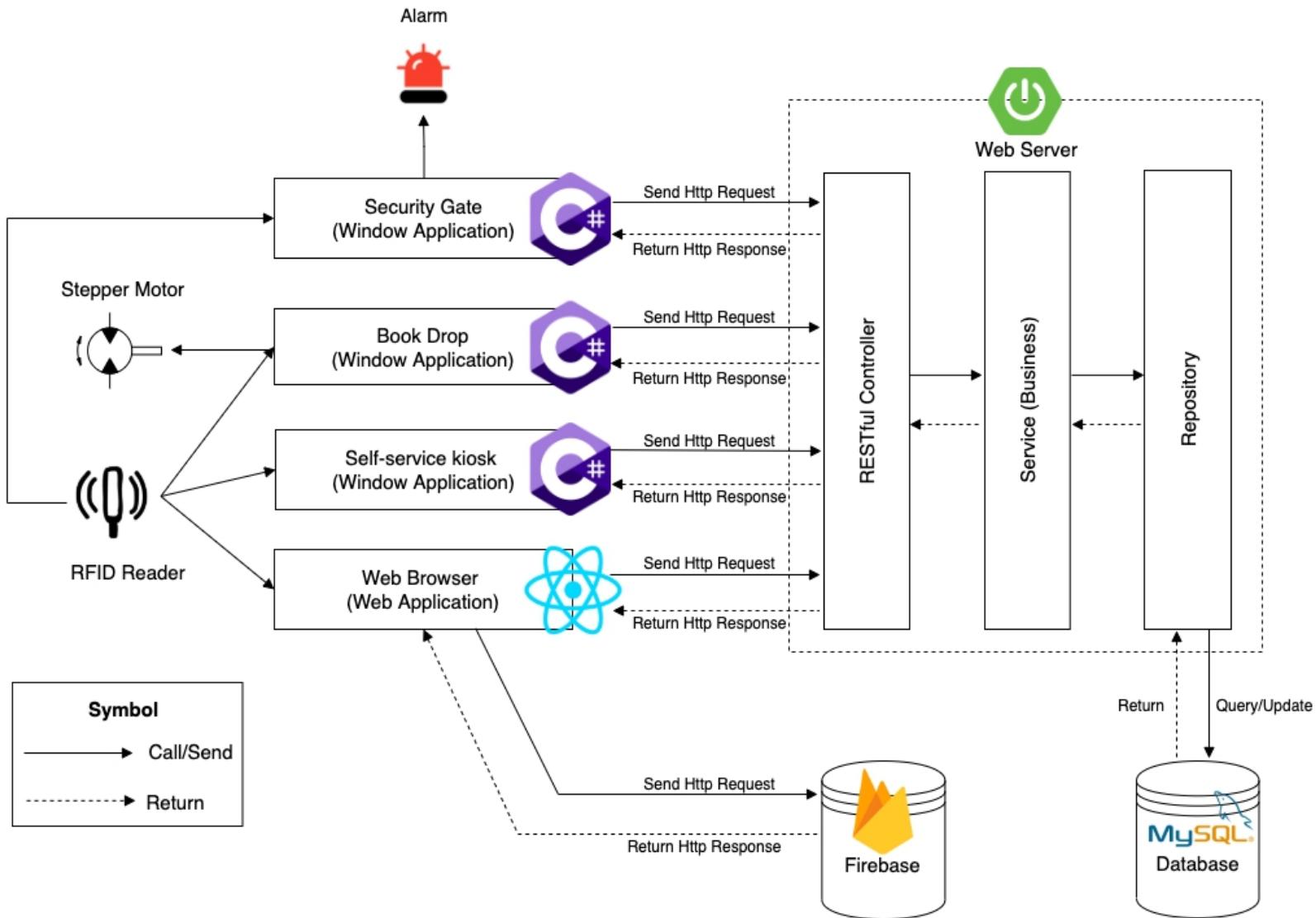


Figure 90 Architecture Diagram

The RLMS consists of following sub-systems:

- Web Server: receives requests from the front end applications (web application, Window application) then returns the corresponding response. Web server is built base on Spring Boot Framework, Restful API, Three Tier Architecture (Controller - Service - Repository), Data Transfer Object pattern
- Security Gate, Book Drop, Self-service kiosk: send http requests to the web server and receive http response from it. They are Window applications using .Net Framework.
- Web Browser: sends http requests to the web server and receives http response from it. Web browser apply: React JS, Redux for managing states

We also use MySQL database to store the operating system information and Firebase storage to store images of the system.

There are supporting devices applied in our system (check Technology Solution for further information)

- RFID readers: scan and send RFID key to sub-systems
- Stepper motor: control the doors of the book drop
- Alarm for security gate

3. Device Description

3.1. UHF RFID Reader (Book Tag Reader)



Figure 91 <Device Description> UHF RFID Reader

UHF Reader Specification	
Purpose	Read books RFID tags and send them to the RLMS. It can send tags UID by emulating keyboard input (USB) or through serial port (RS232).
Model	GEE-UR-2100
Operating Voltage	USB d.c 5V or external d.c +9V
Operating Frequency	860 - 960 MHz

RF output	0 - 26 dBm
Interface	USB, RS232
Reader Distance	1 meter - 0.5 meter
Polarization	Circular polarized
Read speed	50 tags/second
Tag buffer	370 tags@128 bit EPC
Antenna	Built-in circular polarization antenna
Protocol	ISO18000-6C & EPC global Class I Gen 2
Anti-collision	Multiple tags operation allowed
Operating condition	-20 - 60 C, 5- 95% RH, no condensation
Storage condition	-5- 45 C, 5- 95% RH, no condensation

Table 79 <Device Description> UHF RFID Reader Reference:

<https://www.geenfc.com/en/Products/UHFRreader/Fixedreader/2018-05-22/239.html>

3.2. HF RFID Reader Module



Figure 92 <Device Description> HF RFID Reader Module

HF Reader Specification	
Purpose	Read patrons cards and send to Arduino UNO
Model	RC522
Operating Voltage	3.3V
Operating Frequency	13.56MHz
Supported RFID tags	mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire
Interface	SPI
Reader Distance	0~60mm

*Table 80 <Device Description> HF Reader specification. Reference:
<https://nshopvn.com/product/mach-rfid-rc522-nfc-13-56mhz/>*

3.3. Stepper Motor

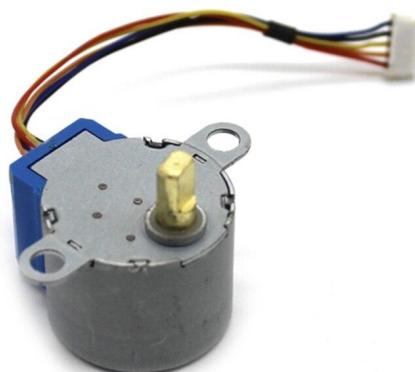


Figure 93 <Device Description> Stepper Motor

Stepper Motor Specification	
Purpose	Open and close book drop door
Model	24BYJ48-5V
Operating Voltage	5V
Reduction Ratio	1/64

Step Angle	5.625 x 1/64
Number of Phases	4
Diameter	28 mm

*Table 81 <Device Description> Stepper motor specification. Reference:
<https://nshopvn.com/product/dong-co-buoc-28byj48-5v/>*

3.4. Arduino UNO Board

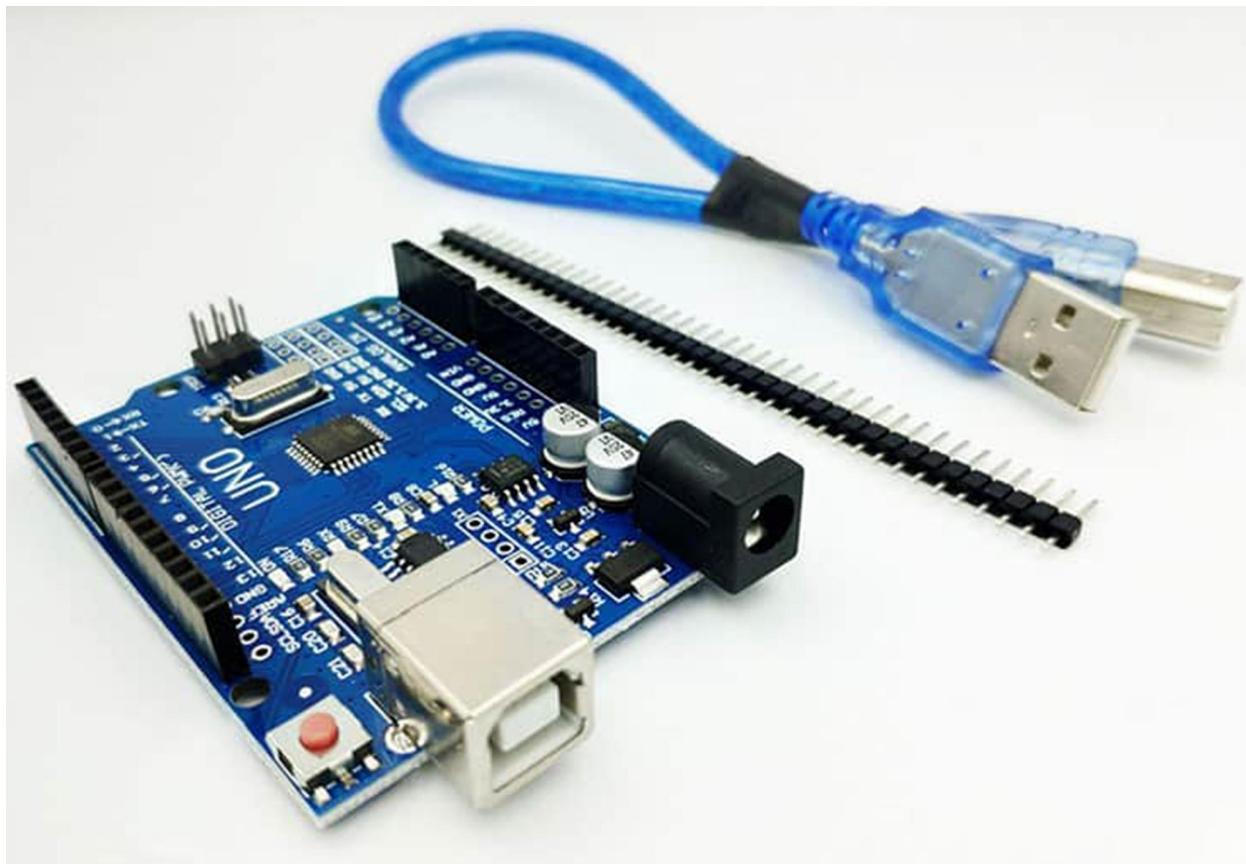


Figure 94 <Device Description> Arduino UNO Board

Arduino UNO Board Specification	
Purpose	Control stepper motor, passive buzzer, LED, etc.
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (limit)	6-20V
ADC pins	6x 10-bit

Digital I/O Pins	14 (of which 6 provide PWM output)
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
DC Current for 5V Pin	500 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
Weight	25 g

Table 82 <Device Description> Arduino UNO Board. Reference: <https://nshopvn.com/product/arduino-uno-r3-dip-kem-cap/> and <https://store.arduino.cc/usar/arduino-uno-rev3>

3.5. Pro Micro Board

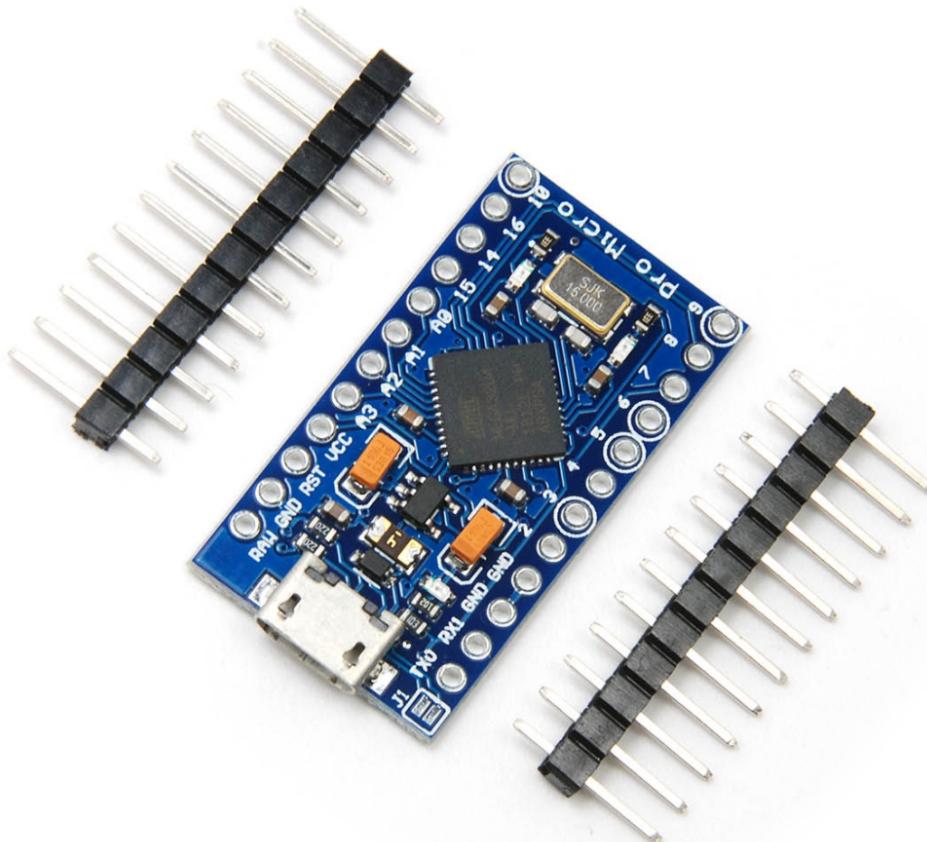


Figure 95 <Device Description> Sparkfun Pro Micro Board

Sparkfun Pro Micro board specification	
Purpose	Gets data from HF Reader and send to RLMS (emulate keyboard)

Microcontroller	ATmega32U4
Operating Voltage	5V
Input Voltage (recommended)	6-20V
ADC pins	9x 10-bit
Digital I/O Pins	12 (of which 5 provide PWM output)
DC Current per I/O Pin	20 mA
Flash Memory	32 Kbytes
SRAM	2 KB (ATmega328P)
USB Connector Style	Micro-B Female

Table 83 <Device Description> Sparkfun Pro Micro Board. Reference:

<https://protosupplies.com/product/pro-micro-5v-16mhz/>

3.6. HF RFID Tags



Figure 96 <Device Description> HF RFID Tags

HF RFID tag specification	
Purpose	Is used for patron cards
Tag type	MIFARE Classic 1K card
Operating Frequency	13.56 MHz
Memory	1 kB, organized in 16 sectors of 4 blocks (one block consists of 16 byte)
UID	7 byte
NUID	4 byte
Anti-Collision	Yes
Range	Up to 100 mm depends on reader

Table 84 <Device Description> HF RFID Tags. Reference: https://www.nxp.com/docs/en/datasheet/MF1S50YYX_V1.pdf

3.7. UHF RFID Tags

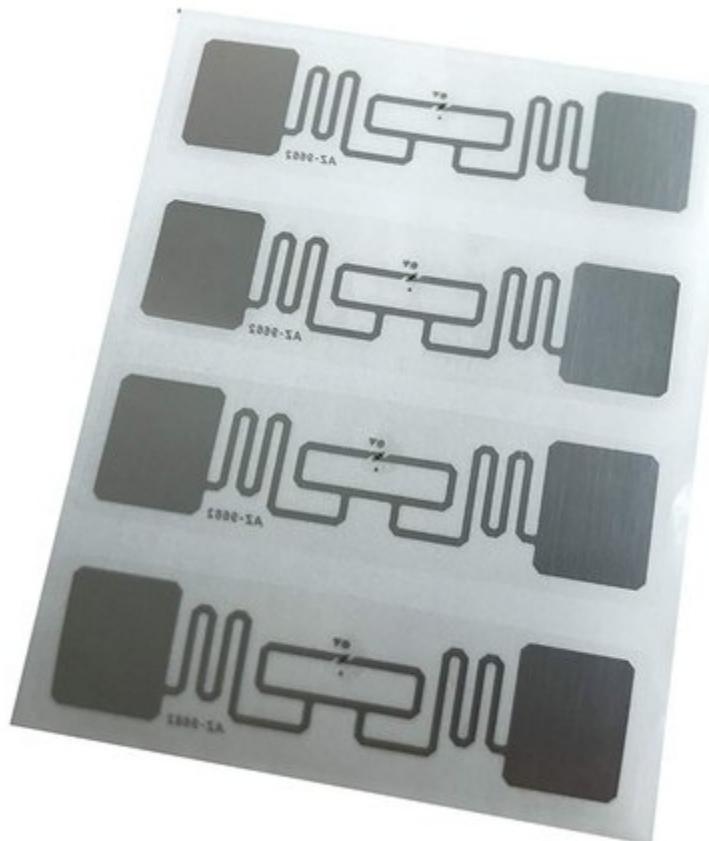


Figure 97 <Device Description> UHF RFID Tags

UHF RFID tag specification	
Purpose	Is used to tag books
Tag type	UHF Class 1 Gen 2
Operating Frequency	860 ~ 960 MHz
Memory	EPC 96-480 Bit, User 512 Bit, TID 32 Bit
Protocol	ISO/IEC 14443A /ISO15693/ EPC global UHF Class 1 Gen 2 (ISO 18000-6C)
Range	Up to 10 m depends on reader

Table 85 <Device Description> UHF RFID Tags. Reference:

<http://www.atdtech.com/index.php/vi/product/rfid/the-dan-rfid-tan-so-cao>

3.8. Passive Buzzer



Figure 98 <Device Description> Passive Buzzer

Passive buzzer specification	
Purpose	Is used for alarm sound
Operating voltage	5V
Operating frequency	2Khz -5Khz
Size	12mm*8.5mm
Weight	1g

Table 86 <Device Description> Passive Buzzer. Reference: <https://nshopvn.com/product/buzz-5v/>

3.9. LED



Figure 99 <Device Description> LED

LED specification	
Purpose	Is used for alarm alerting light
Operating voltage	5V
Length	5mm
Number of legs	2

Table 87 <Device Description> LED. Reference: <https://nshopvn.com/product/led-trong-5mm-xanh-lau-xanh-duong-do-trang-vang/?variant=39855>

3.10. Mini Breadboard



Figure 100 <Device Description> Mini Breadboard

Mini breadboard specification	
Purpose	Is used to connect Arduino boards with modules
Operating voltage	5V
Length	5mm
Number of legs	2

Table 88 <Device Description> Mini Breadboard. Reference: <https://nshopvn.com/product/test-board-cam-mini/>

3.11. RS232 Converter



Figure 101 <Device Description> RS232 Converter

RS232 Converter specification	
Purpose	Is used to connect UHF Scanner to the RLMS through Serial port connection
Main IC	HL-340
Protocol	RS232 (DB9)

Table 89 <Device Description> RS232 Converter. Reference: <https://nshopvn.com/product/cap-chuyen-giao-tiep-usb-rs232/>

3.12. Stepper Motor Controller Board ULN2003

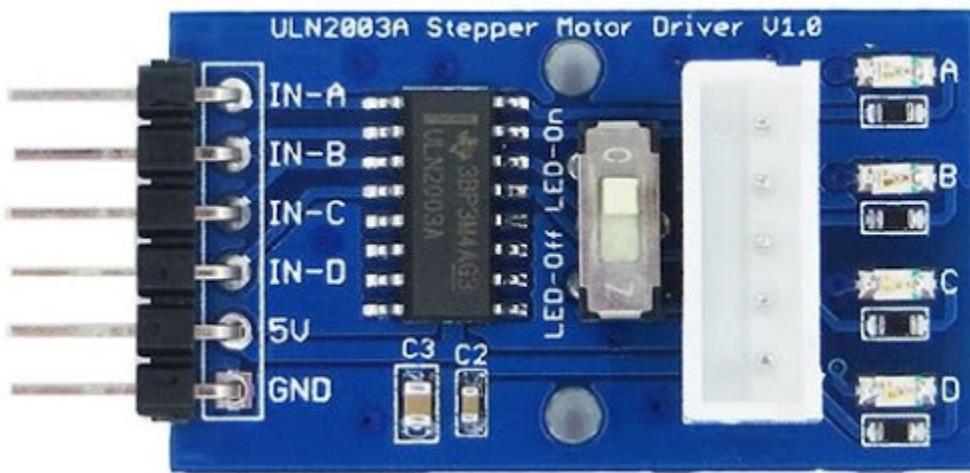


Figure 102 <Device Description> Stepper Motor Controller Board ULN2003

ULN2003 Board Specification	
Purpose	Control the direction and speed of the stepper motor
Model	ULN2003
Operating Voltage	5V - 12V

Table 90 <Device Description> Stepper Motor Controller Board ULN2003. Reference: <https://nshopvn.com/product/mach-dieu-khien-dong-co-buoc-uln2003-dong-co-buoc-5v/>

4. Component Diagram

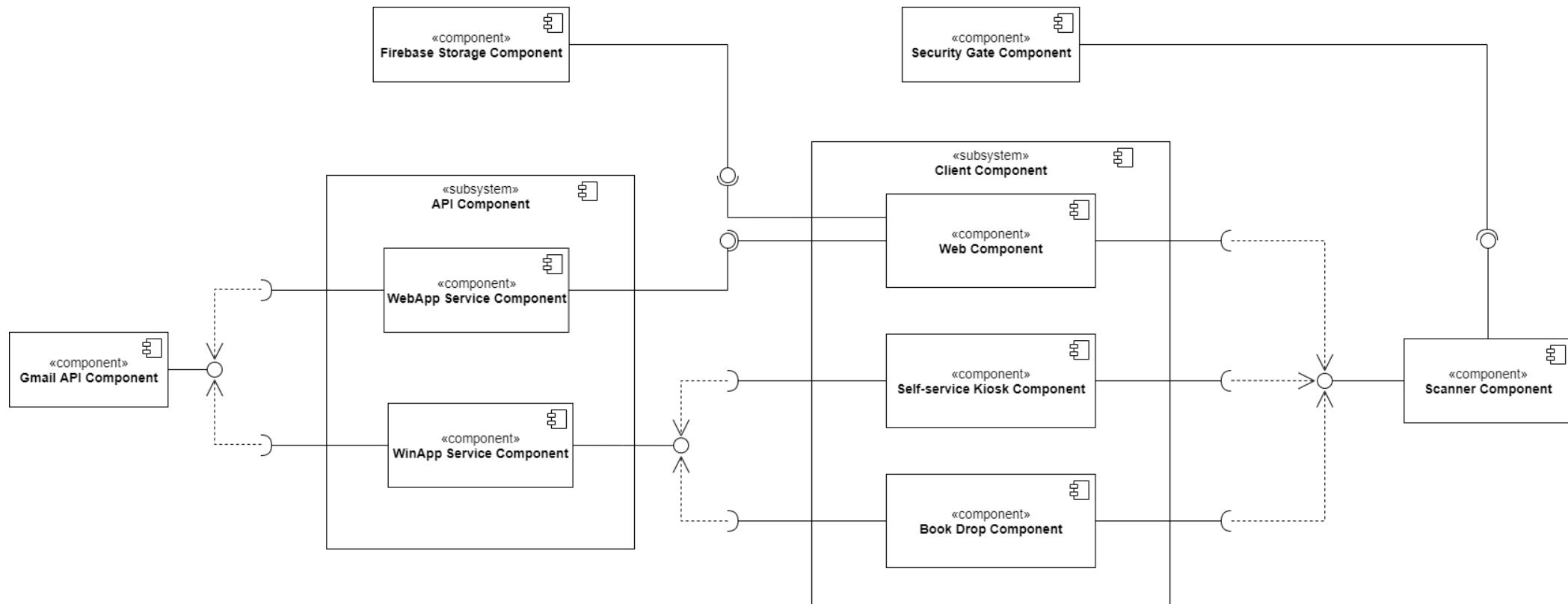


Figure 103 Component Diagram

Component Dictionary	
Component Name	Description
Firebase Storage Component	Is used to upload and store images of the RLMS
Gmail API Component	Is used to send notification emails to the users
WebApp Service Component	Consists of components that provide services like authentication, book management, book circulation management, etc. for the client web application
WinApp Service Component	Consists of components that provide services that allow the user to do self-return and self-checkout at the self-service kiosk and book drop
Web Component	Consists of a web application for patron, librarian and manager
Self-service Kiosk Component	Consists of a Window application for patron to do self-checkout and self-return
Book Drop Component	Consists of a Window application and the Book Drop device for patron to do self-return
Scanner Component	Consists of UHF RFID reader and HF RFID reader that allows functions to read patron cards and book tags
Security Gate Component	Consists of a Window application and an alarm module

5. System Details Design

5.1. Class Diagram

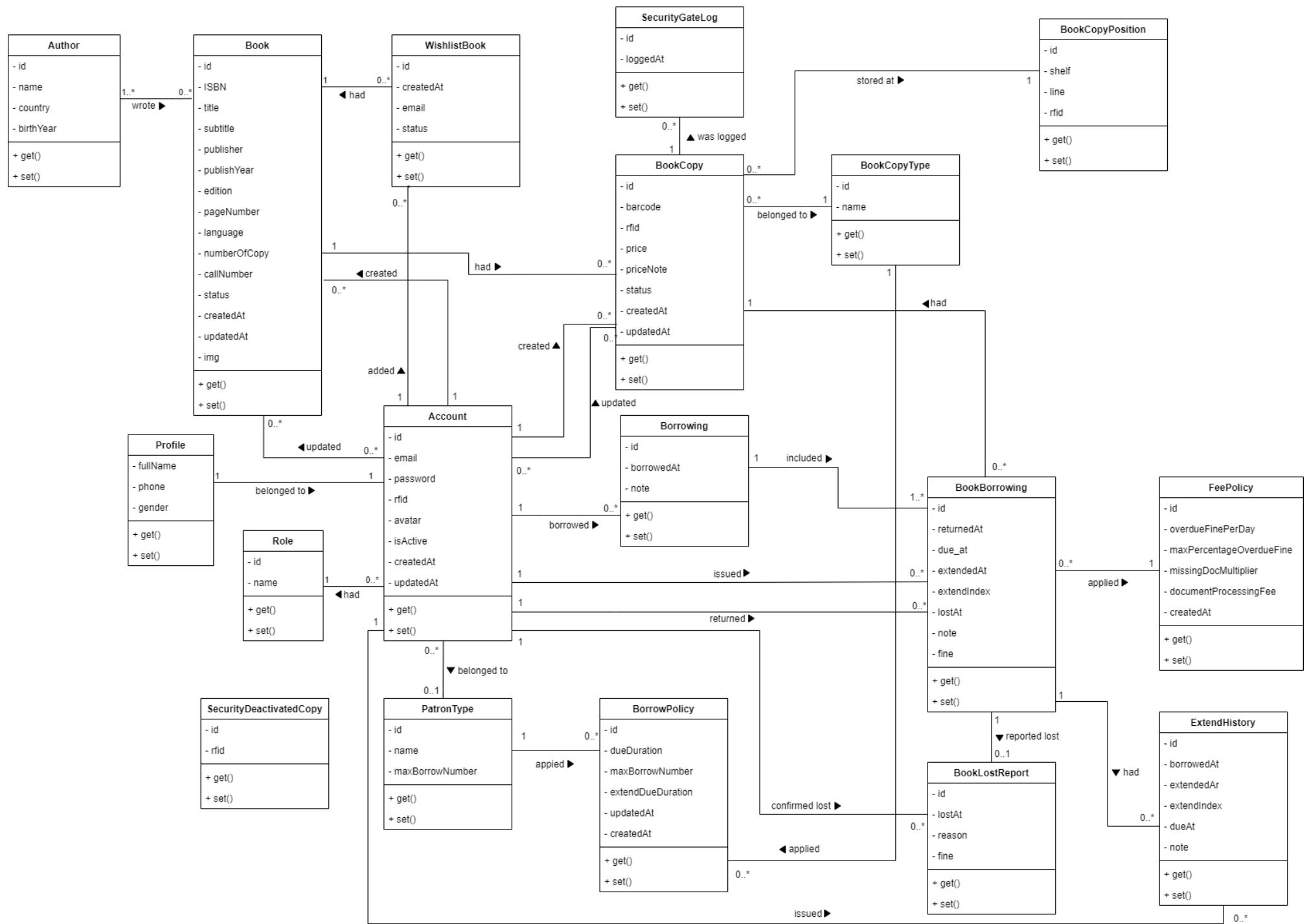


Figure 104 Class Diagram

5.2. Class Specification

5.2.1. Author Class

#	Field Name	Type	Description
1	id	int	Unique identifier of a policy document
2	name	string	Name of the author
3	country	string	Country where the author was born
4	birthYear	int	Year when the author was born

Table 91 <Class Diagram Attributes> Author

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 92 <Class Diagram Methods> Author

5.2.2. Borrowing Class

#	Field Name	Type	Description
1	id	int	Unique identifier of borrowing
2	borrowAt	double	Time when patron borrowed books
3	note	int	Librarian's comment of the borrowed books

Table 93 <Class Diagram Attributes> Borrowing

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 94 <Class Diagram Methods> Borrowing

5.2.3. Security Gate Log Class

#	Field Name	Type	Description
1	id	string	Unique identifier of log
2	loggedAt	LocalDate Time	The time an activated book copy is logged in the system

Table 95 <Class Diagram Attributes> Security Gate Log

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 96 <Class Diagram Methods> Security Gate

5.2.4. Fee Policy Class

#	Field Name	Type	Description
1	id	int	Unique identifier of fee policy
2	overdueFinePerDay	double	Fine charged per overdue day
3	maxPercentageOverdueFee	int	Maximum fine that patron has to pay corresponding to the set percentage of the book price
4	missingDocMultiplier	int	Is used to calculate lost book fine, if the lost book cannot be found in the market.
5	createdAt	DateTime	Time when the fee policy was created
6	documentProcessingFee	double	Fee for processing document

Table 97 <Class Diagram Attributes> Fee Policy

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 98 <Class Diagram Methods> Fee Policy

5.2.5. Book Class

#	Field Name	Type	Description
1	id	int	Unique identifier of book
2	ISBN	string	ISBN of the book
3	title	string	Title of the book
4	subtitle	string	Subtitle of the book
5	publisher	string	Publisher of the book
6	publishYear	int	Year the book was published
7	edition	int	Edition of the book
8	language	string	Language the book was written
9	pageNumber	int	Number of page
10	callNumber	string	Call number of book to identify book's location
11	numberOfCopy	int	Total number of book' copies
12	status	string	Status of the book
13	createdAt	DateTime	Time when the book was created
14	updatedAt	DateTime	Time when the book was updated
15	img	string	Image of the book

Table 99 <Class Diagram Attributes> Book

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 100 <Class Diagram Methods> Book

5.2.6. Profile Class

#	Field Name	Type	Description
1	fullName	string	Full name of the user
2	phone	string	Phone of the user

3	gender	String	Gender of the user: Female, Male
---	--------	--------	----------------------------------

Table 101 <Class Diagram Attributes> Profile

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 102 <Class Diagram Methods> Profile

5.2.7. Role Class

#	Field Name	Type	Description
1	id	string	Unique identifier of role
2	name	string	Name of the role

Table 103 <Class Diagram Attributes> Role

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 104 <Class Diagram Methods> Role

5.2.8. Account Class

#	Field Name	Type	Description
1	id	int	Unique identifier of the account
2	email	string	Email of the account
3	password	string	Password of the account
4	rfid	string	RFID tag ID of the account
5	avatar	string	Avatar of the account
6	isActive	boolean	the status if the account is active or not
7	createdAt	DateTime	Time when the account was created
8	updatedAt	DateTime	Time when the account was updated

Table 105 <Class Diagram Attributes> Account

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 106 <Class Diagram Methods> Account

5.2.9. Book Borrowing Class

#	Field Name	Type	Description
1	id	int	Unique identifier of the book borrowing
2	returnedAt	DateTime	Time when the book borrowing was returned
3	dueAt	Date	Date when the book borrowing needs to be returned
4	extendedAt	DateTime	Time when the book borrowing was renewed
5	extendIndex	int	Current number of renewed times
6	lostAt	DateTime	Time when the book borrowing was reported to be lost
7	note	String	Note of the book borrowing

8	fine	double	Fine of the overdue book borrowing
---	------	--------	------------------------------------

Table 107 <Class Diagram Attributes> Book Borrowing

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 108 <Class Diagram Methods> Book Borrowing

5.2.10. Book Copy Class

#	Field Name	Type	Description
1	id	int	Unique identifier of book copy
2	barcode	string	Barcode of the book copy
3	rfid	string	RFID tag ID of the book copy
4	price	double	Price of the book copy
5	status	string	Status of the book copy
6	createdAt	DateTime	Time when the book copy was created
7	updatedAt	DateTime	Time when the book copy was updated
8	priceNote	string	Note for price of book copy

Table 109 <Class Diagram Attributes> Book Copy

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 110 <Class Diagram Methods> Book Copy

5.2.11. Book Copy Type Class

#	Field Name	Type	Description
1	id	int	Unique identifier of book copy type
2	name	string	Name of the book copy type

Table 111 <Class Diagram Attributes> Book Copy Type

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 112 <Class Diagram Methods> Book Copy Type

5.2.12. Book Lost Report Class

#	Field Name	Type	Description
1	id	int	Unique identifier of book lost report
2	lostAt	DateTime	Time when the book lost report created
3	reason	String	Reason of the book lost report
4	fine	double	Fine of the book lost report

Table 113 <Class Diagram Attributes> Book Lost Report

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 114 <Class Diagram Methods> Book Lost Report

5.2.13. Borrow Policy Class

#	Field Name	Type	Description
1	id	int	Unique identifier of borrow policy
2	dueDuration	int	Due duration when borrow book
3	maxBorrowNumber	int	Max borrow number of borrow policy

4	maxExtendTime	int	Max renewable times
5	extendDueDuration	int	Due duration when renew book
6	createdAt	DateTime	Time when the borrow policy was created
7	updatedAt	DateTime	Time when the borrow policy was updated

Table 115 <Class Diagram Attributes> Borrow Policy

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 116 <Class Diagram Methods> Borrow Policy

5.2.14. Extend History Class

#	Field Name	Type	Description
1	id	int	Unique identifier of extend history
2	borrowedAt	DateTime	Time when the book was borrowed
3	extendedAt	DateTime	Time when the extend history was created
4	extendIndex	int	Index of the extend history
5	dueAt	Date	Date when the book needs to be returned
6	note	string	Note of the extend history

Table 117 <Class Diagram Attributes> Extend History

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 118 <Class Diagram Methods> Extend History

5.2.15. Book Copy Position Class

#	Field Name	Type	Description
1	id	int	Unique identifier of the book copy position

2	shelf	string	Shelf on which the book copy was put
3	line	int	Line on which the book copy was put
4	rfid	string	RFID of shelf-line

Table 119 <Class Diagram Attributes> Book copy position

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 120 <Class Diagram Methods> Book copy position

5.2.16. Patron Type Class

#	Field Name	Type	Description
1	id	int	Unique identifier of patron type
2	name	string	Name of patron type
3	maxBorrowNumber	int	Total number of book copies that patron with this patron type can borrow at a time

Table 121 <Class Diagram Attributes> Patron Type

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 122 <Class Diagram Methods> Patron Type

5.2.17. Wishlist Book Class

#	Field Name	Type	Description
1	id	int	Unique identifier of patron type
2	createdAt	DateTime	Time when the wishlist book was created
3	email	string	Patron email who wishes to borrow the book
4	status	string	Status of wishing book

Table 123 <Class Diagram Attributes> Wishlist Book

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 124 <Class Diagram Methods> Wishlist Book

5.2.18. Security Deactivated Copy Class

#	Field Name	Type	Description
1	id	string	Unique identifier of deactivated book copy
2	rfid	string	RFID of book copy

Table 125 <Class Diagram Attributes>Security Deactivated Copy

#	Method	Return Type	Description
1	get	datatype	Get attribute value
2	set	void	Set attribute value

Table 126 <Class Diagram Methods> Security Deactivated Copy

5.3. State Machine Diagrams

5.3.1. Book Status

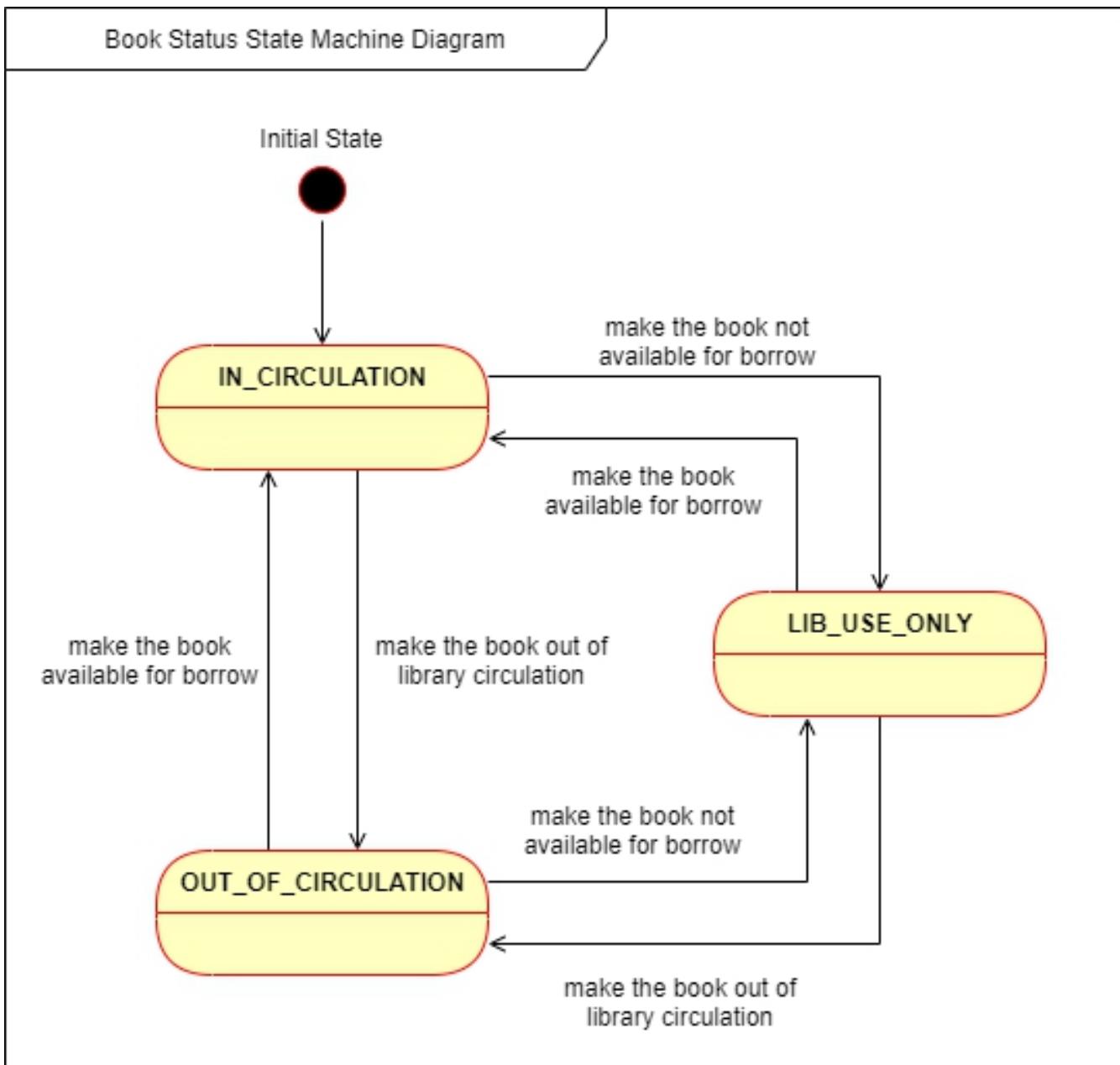


Figure 105 <State Machine Diagram> Book Status

5.3.2. Book Copy Status

Book Copy Status State Machine Diagram

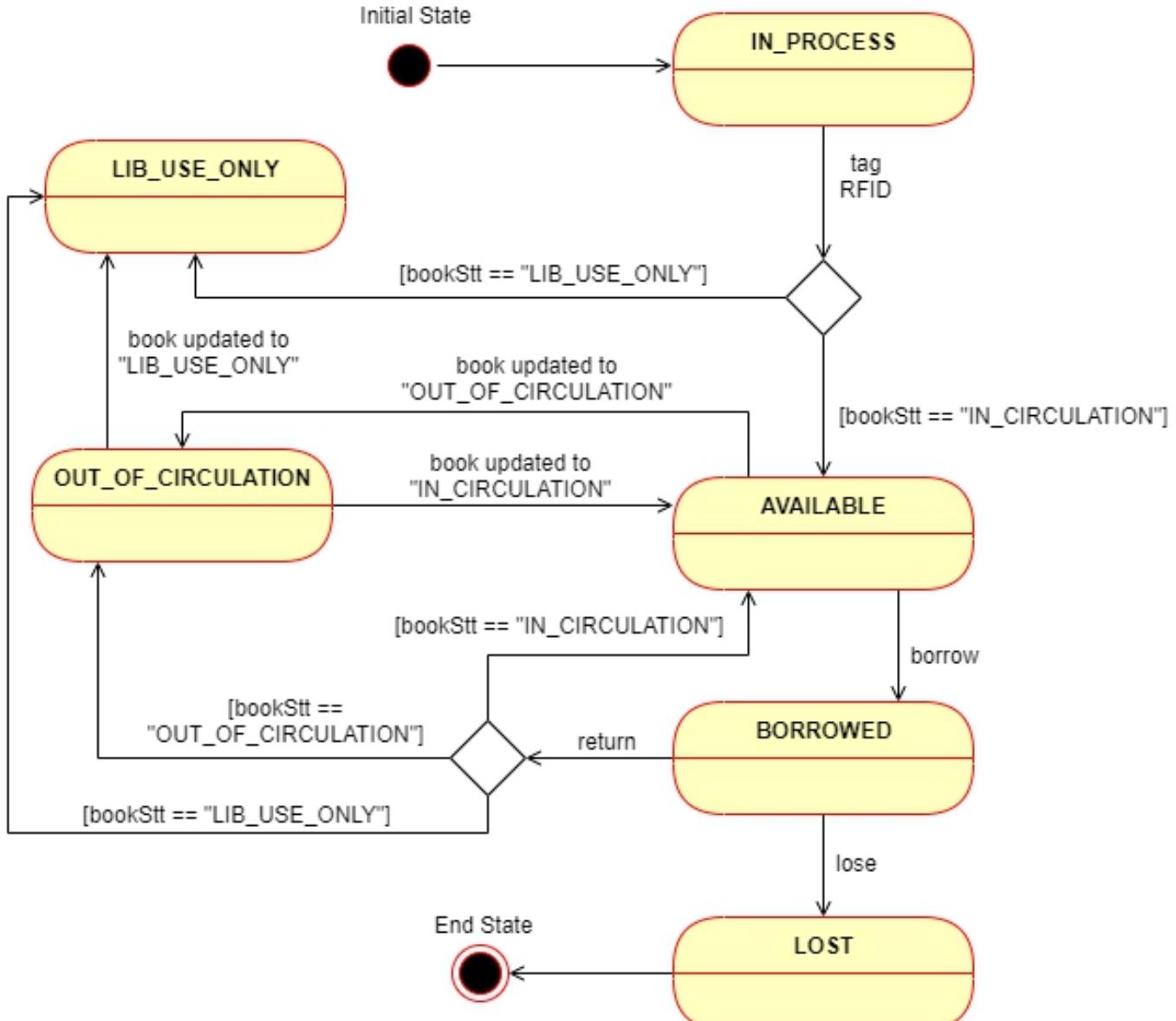


Figure 106 <State Machine Diagram> Book Copy Status

5.4. Sequence Diagrams

5.4.1. Return Book at Book Drop

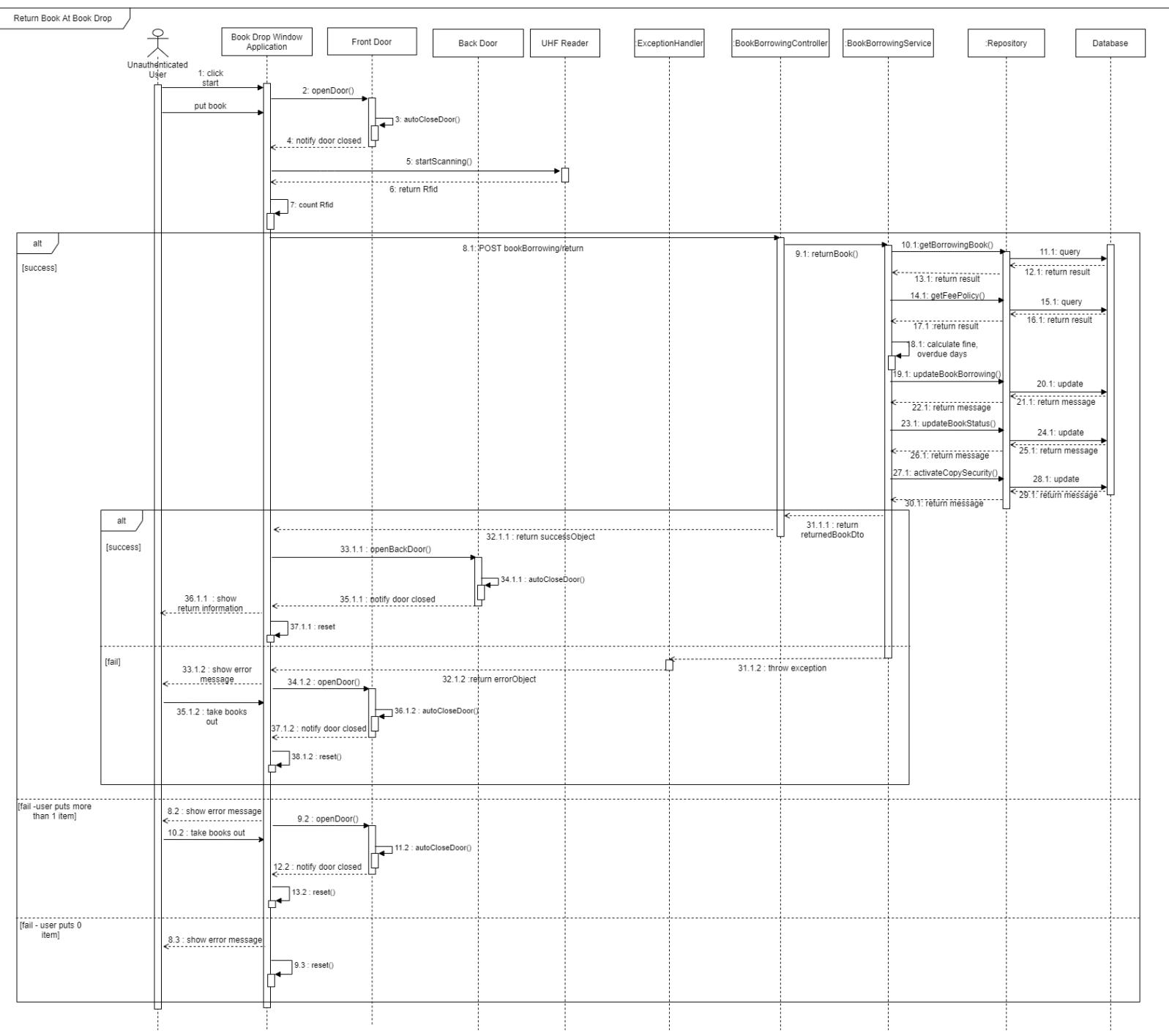


Figure 107 <Sequence Diagram> Return Book at Book Drop

5.4.2. Librarian return books

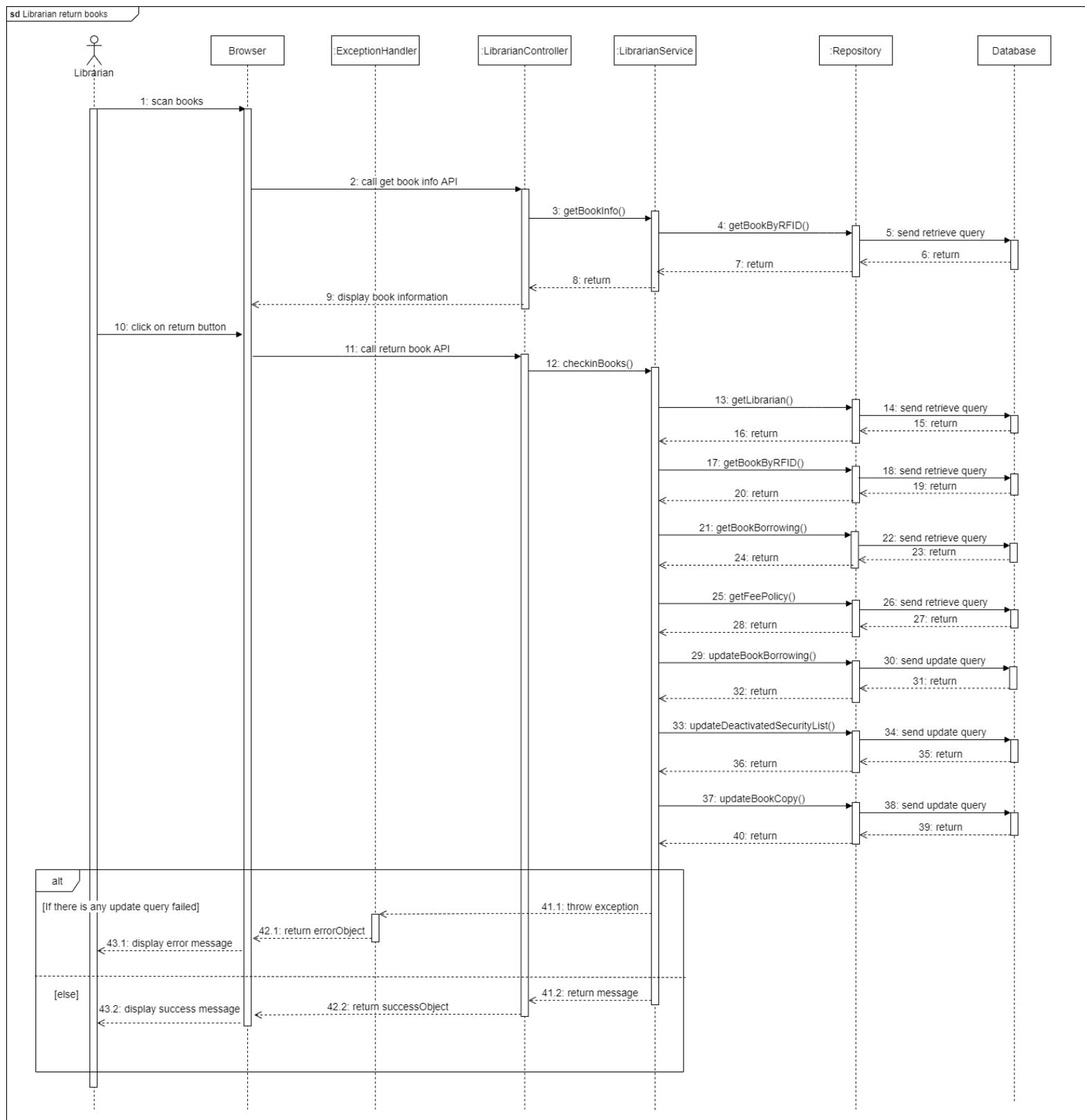


Figure 108 <Sequence Diagram> Librarian return books

5.4.3 Librarian checkout book

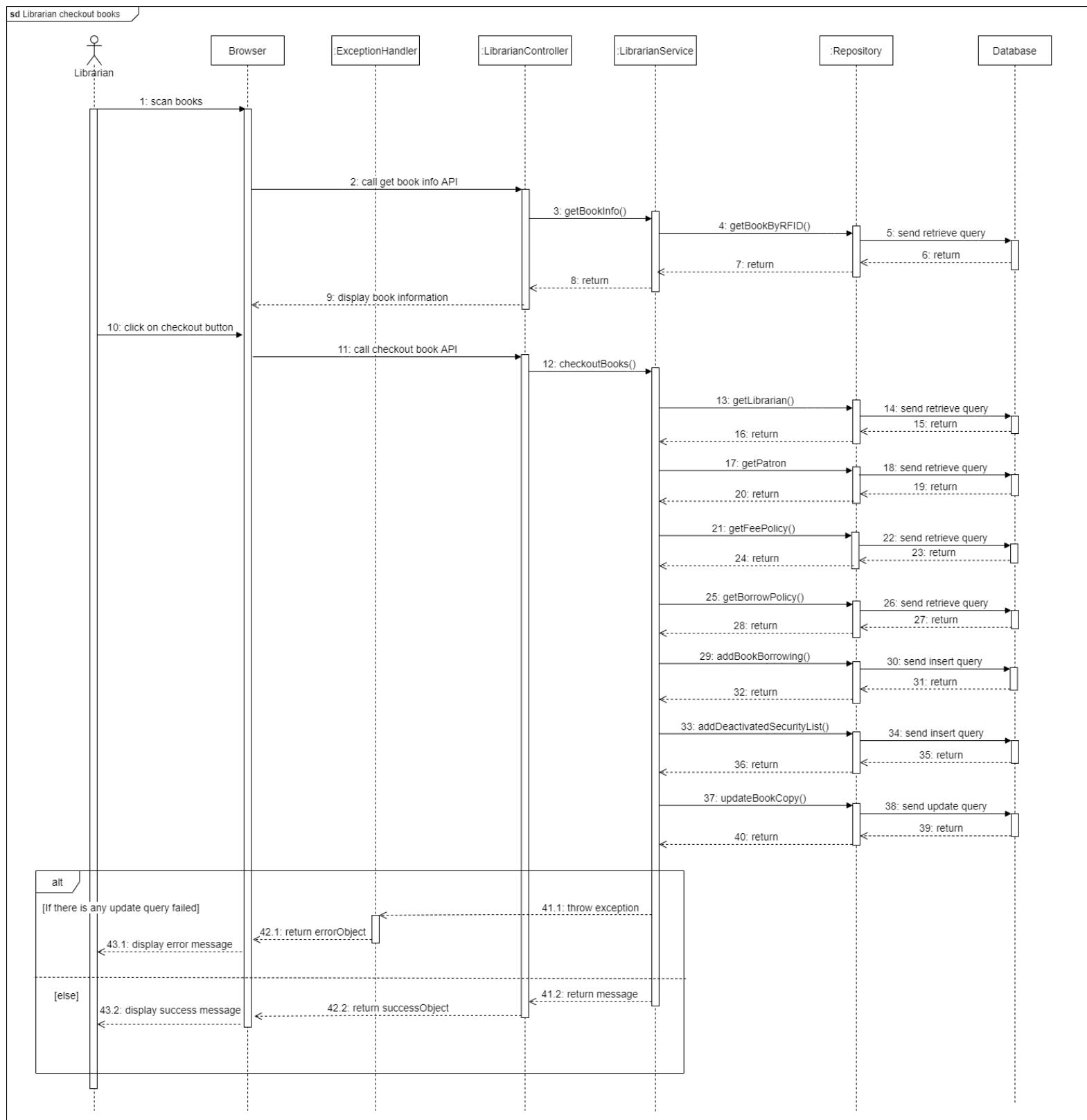


Figure 109 <Sequence Diagram> Librarian checkout book

5.4.4. Librarian renew for a patron

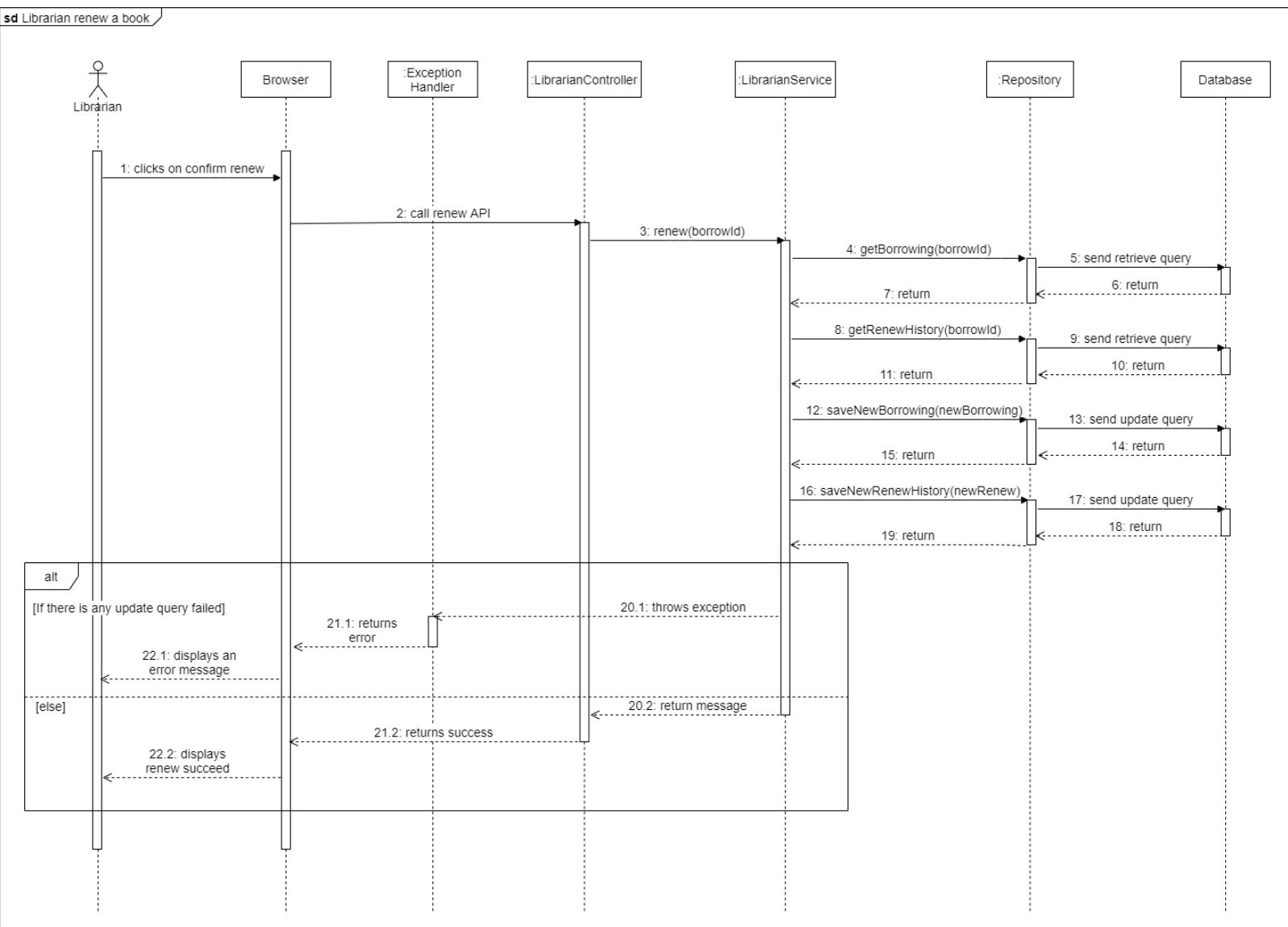


Figure 110 <Sequence Diagram> Librarian renew for a patron

6. Data & Database Design

6.1. Physical Diagram

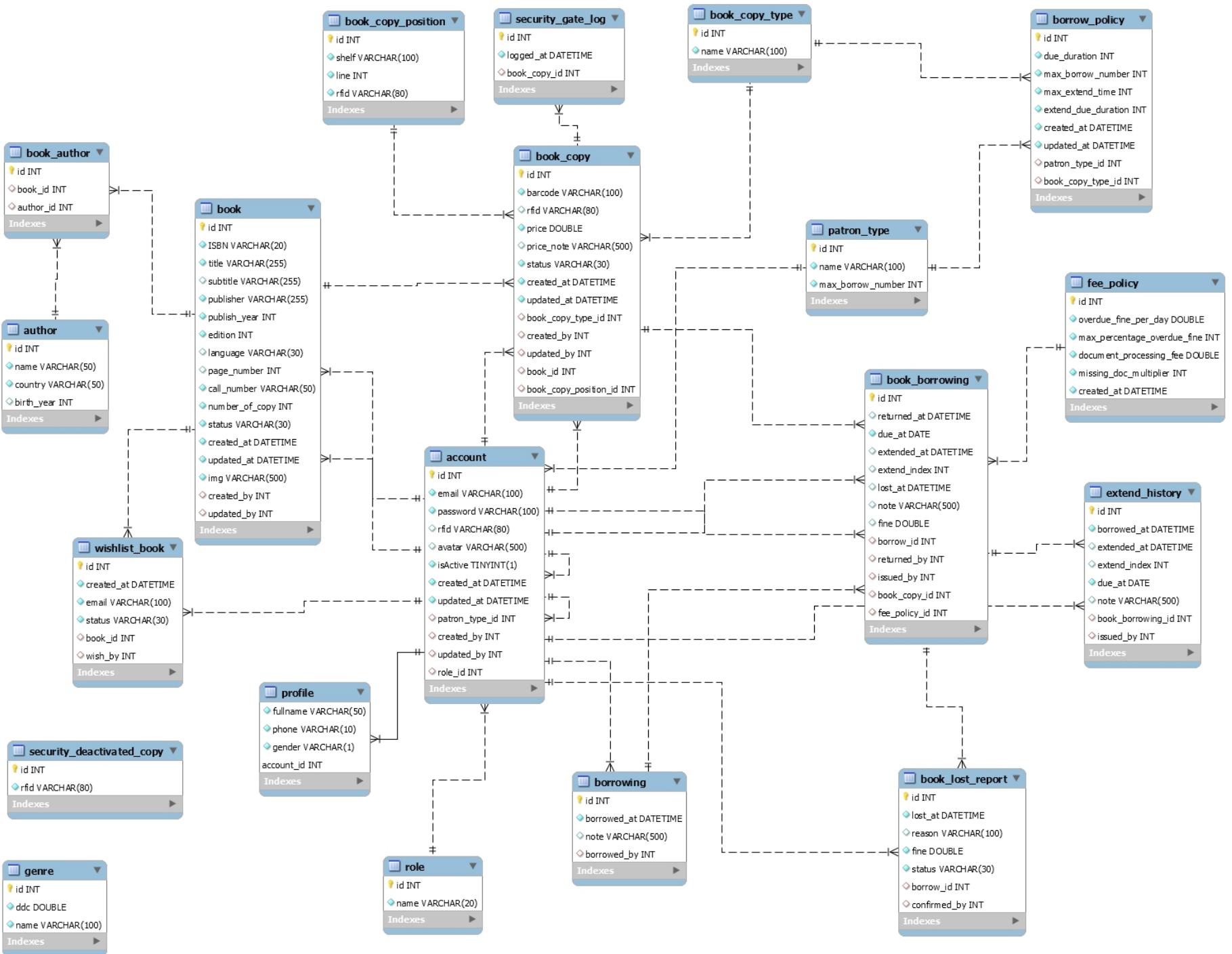


Figure 111 Physical Diagram

6.2. Database Description

6.2.1. author

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of a policy document
2	name	varchar	50	No	Yes		Name of the author
3	country	varchar	50	No	Yes		Country where the author was born
4	birth_year	int		No	No		Year when the author was born

Table 127 <Physical Diagram> author

6.2.2. book_author

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of book author
2	book_id	int		No	No	FK	Reference to book table
3	author_id	int		No	No	FK	Reference to author table

Table 128 <Physical Diagram> book_author

6.2.3. genre

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of genre
2	genre	varchar	100	No	Yes		Name of the genre base on the DDC
3	ddc	double		Yes	Yes		The Dewey Decimal Classification (DDC), colloquially the Dewey Decimal System, is a proprietary library

							classification system
--	--	--	--	--	--	--	-----------------------

Table 129 <Physical Diagram> genre

6.2.4. security_gate_log

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of book genre
2	book_id	int		No	No	FK	Reference to book table
3	logged_at	DateTime		No	No		The time when an activated book copy is logged

Table 130 <Physical Diagram> book_genre

6.2.5. book

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of book
2	ISBN	varchar	20	Yes	Yes		ISBN of the book
3	title	varchar	255	No	Yes		Title of the book
4	subtitle	varchar	255	No	No		Subtitle of the book
5	publisher	varchar	255	No	Yes		Publisher of the book
6	publish_year	int		No	Yes		Year the book was published
7	edition	int		No	Yes		Edition of the book
8	language	varchar	30	No	No		Language the book was written
9	page_number	int		No	No		Number of page
10	call_number	varchar	50	No	Yes		Call number of book to identify book's location
11	number_of_copy	int		No	Yes		Total number of book' copies
12	status	varchar	30	No	Yes		Status of the book

13	created_at	DateTime		No	Yes		Time when the book was created
14	updated_at	DateTime		No	Yes		Time when the book was updated
15	img	varchar	500	No	Yes		Image of the book
16	created_by	int		No	No	FK	Reference to account table
17	updated_by	int		No	No	FK	Reference to account table

Table 131 <Physical Diagram> book

6.2.6. profile

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	account_id	int		Yes	Yes	PK,FK	Use the same PK with account table
2	fullName	varchar	50	No	Yes		Full name of the user
3	phone	varchar	10	No	Yes		Phone of the user
3	gender	varchar	1	No	Yes		Gender of the user

Table 132 <Physical Diagram> profile

6.2.7. role

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of role
2	name	varchar	20	Yes	Yes		Name of the role

Table 133 <Physical Diagram> role

6.2.8. account

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of the account
2	email	varchar	100	Yes	Yes		Email of the account
3	password	varchar	100	No	Yes		Password of the

							account
4	rfid	varchar	80	Yes	No		RFID tag ID of the account
5	avatar	varchar	500	No	No		Avatar of the account
6	isActive	tinyint	1	No	Yes		the status if the account is active or not
7	created_at	DateTime		No	No		Time when the account was created
8	updated_at	DateTime		No	No		Time when the account was updated
9	patron_type_id	int		No	No	FK	Reference to patron_type table
10	created_by	int		No	No	FK	Reference to account table
11	updated_by	int		No	No	FK	Reference to account table
12	role_id	int		No	No	FK	Reference to role table

Table 134 <Physical Diagram> account

6.2.9. book_borrowing

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of the book borrowing
2	returned_at	DateTime		No	No		Time when the book borrowing was returned
3	due_at	Date		No	Yes		Date when the book borrowing needs to be returned
4	extended_at	DateTime		No	No		Time when the book borrowing was renewed
5	extend_index	int		No	No		Current number of renewed times

6	lost_at	DateTime		No	No		Time when the book borrowing was reported to be lost
7	note	varchar	500	No	No		Note of the book borrowing
8	fine	double		No	No		Fine of the overdue book borrowing
9	borrow_id	int		No	No	FK	Reference to borrowing table
10	returned_by	int		No	No	FK	Reference to account table
11	issued_by	int		No	No	FK	Reference to account table
12	book_copy_id	int		No	No	FK	Reference to book_copy table
13	fee_policy_id	int		No	No	FK	Reference to fee_policy table

Table 135 <Physical Diagram> book_borrowing

6.2.10. book_copy

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of book copy
2	barcode	varchar	100	Yes	Yes		Barcode of the book copy
3	rfid	varchar	80	Yes	No		RFID tag ID of the book copy
4	price	double		No	Yes		Price of the book copy
5	status	varchar	30	Yes	No		Status of the book copy
6	created_at	DateTime		No	Yes		Time when the book copy was created
7	updated_at	DateTime		No	Yes		Time when the book copy was updated

8	book_copy_type_id	int		No	No	FK	Reference to book_type table
9	created_by	int		No	No	FK	Reference to account table
10	updated_by	int		No	No	FK	Reference to account table
11	book_id	int		No	No	FK	Reference to book table
12	book_copy_position_id	int		No	No	FK	Reference to book copy position table
13	price_note	varchar	500	No	Yes		Note for price of book copy

Table 136 <Physical Diagram> book_copy

6.2.11. book_copy_type

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of book copy type
2	name	varchar	100	Yes	Yes		Name of the book copy type

Table 137 <Physical Diagram> book_copy_type

6.2.12. book_lost_report

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of book lost report
2	lost_at	DateTime		No	Yes		Time when the book lost report created
3	reason	varchar	100	No	Yes		Reason of the book lost report
4	fine	double		No	Yes		Fine of the book lost report
5	borrow_id	int		No	No	FK	Reference to borrowing table
6	reported_by	int		No	No	FK	Reference to account

							table
--	--	--	--	--	--	--	-------

Table 138 <Physical Diagram> book_lost_report

6.2.13. borrow_policy

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of borrow policy
2	due_duration	int		No	Yes		Due duration when borrow book
3	max_borrow_number	int		No	Yes		Max borrow number of borrow policy
4	max_extend_time	int		No	Yes		Max renewable times
5	extend_due_duration	int		No	Yes		Due duration when renew book
6	created_at	DateTime		No	Yes		Time when the borrow policy was created
7	updated_at	DateTime		No	Yes		Time when the borrow policy was updated
8	patron_type_id	int		No	No	FK	Reference to account table
9	book_copy_type_id	int		No	No	FK	Reference to account table

Table 139 <Physical Diagram> borrow_policy

6.2.14. extend_history

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of extend history
2	borrow_at	DateTime		No	Yes		Time when the book was borrowed
3	extended_at	DateTime		No	No		Time when the extend history was created
4	extend_index	int		No	No		Index of the extend history

5	due_at	Date		No	Yes		Date when the book needs to be returned
6	note	varchar	500	No	No		Note of the extend history
7	book_borrowing_id	int		No	No	FK	Reference to book_borrowing table
8	issued_by	int		No	No	FK	Reference to account table

Table 140 <Physical Diagram> extend_history

6.2.15. book_copy_position

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of the book copy position
2	line	int		No	Yes		Line of shelf on which the book copy was put
3	shelf	varchar	50	No	Yes		Shelf on which the book copy was put
4	rfid	varchar	80	Yes	Yes		Unique RFID tag ID for each shelf-line

Table 141 <Physical Diagram> book_copy_position

6.2.16. patron_type

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of patron type
2	name	varchar	100	No	Yes		Name of patron type
3	max_borrow_number	int		No	Yes		Total number of book copies that patron with this patron type can borrow at a time

Table 142 <Physical Diagram> patron_type

6.2.17. wishlist_book

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes

1	id	int		Yes	Yes	PK	Unique identifier of patron type
2	created_at	DateTi me		No	Yes		Time when the wishlist book was created
3	email	varchar	100	No	Yes		Patron email who wishes to borrow the book
4	status	varchar	30	No	Yes		Status of wishing book
5	book_id	int		No	No		Reference to book table
6	wish_by	int		No	No		Reference to account table

Table 143 <Physical Diagram> patron_type

6.2.18. security_deactivated_copy

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes		PK	Unique identifier of deactivated book copy
2	rfid	varchar	80	Yes	Yes		RFID tag ID of book copy

Table 144 <Physical Diagram> security_deactivated_copy

6.2.19. fee_policy

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of fee policy
2	overdue_fine_per_d ay	double		No	Yes		Fine charged per overdue day
3	max_percentage_overdue	int		No	Yes		Maximum fine that patron has to pay corresponding to the set percentage of the book price
4	missing_doc_multiplier	int		no	Yes		Is used to calculate lost book fine, if the lost book cannot be found

							in the market.
5	created_at	DateTime		No	Yes		Time when the fee policy was created
6	documentProcessingFee	double		No	Yes		Fee for processing document

Table 145 <Physical Diagram> fee_policy

6.2.20. borrowing

#	Field Name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		Yes	Yes	PK	Unique identifier of borrowing
2	borrowed_at	DateTime		No	Yes		Time when patron borrowed books
3	note	varchar	500	No	No		Librarian's comment of the borrowed books
4	borrow_by	int		no	No	FK	Reference to account table

Table 146 <Physical Diagram> borrowing

7. Technology Solution

7.1. RFID Technology

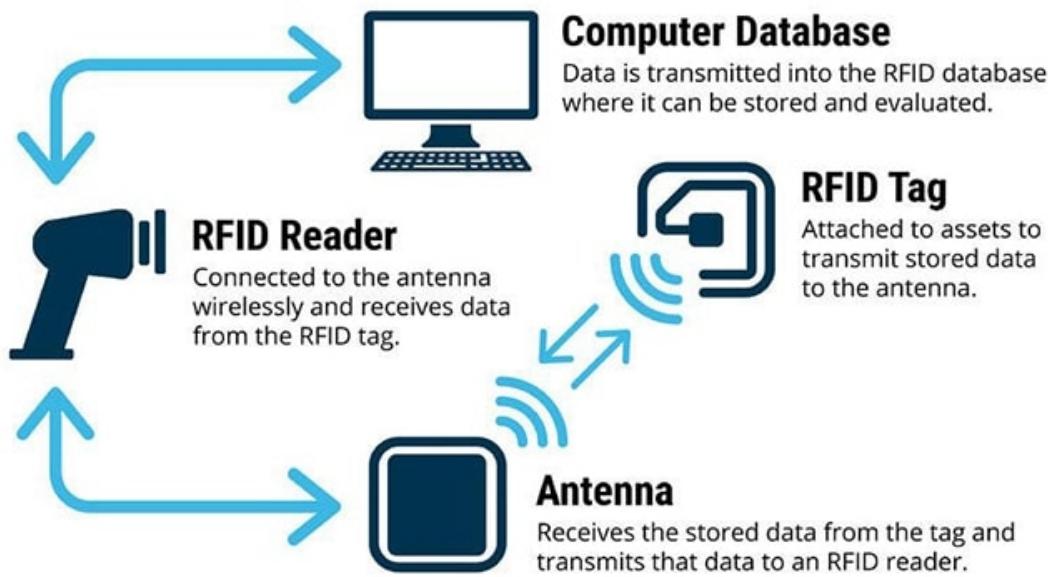
7.1.1. Definition

RFID or Radio Frequency Identification is a technology that allows digital data encoded in RFID tags (or labels) to be read by an RFID reader via radio waves (the tags and readers must have the same operating frequency).

A RFID reader sends out radio waves in a specific frequency through its antenna. If an object with RFID tag is within the reading range of the reader, it will send data back to the reader. By processing the data returned by the tag, the reader can identify the object. The read distance is determined by the capacity of the reader to emit power and by the operating frequency of the RFID system.

There are 2 main types of RFID tag:

- Passive tag: does not have an internal power supply, is powered by the reader's radio wave
- Active tag: has an internal power supply, does not relies to be powered by the reader



How RFID system works. Reference: <https://comparsoft.com/assets-tracking-software/rfid-asset-tracking/>

RFID systems can be categorized by their operating frequency band. Each category has their characteristics:

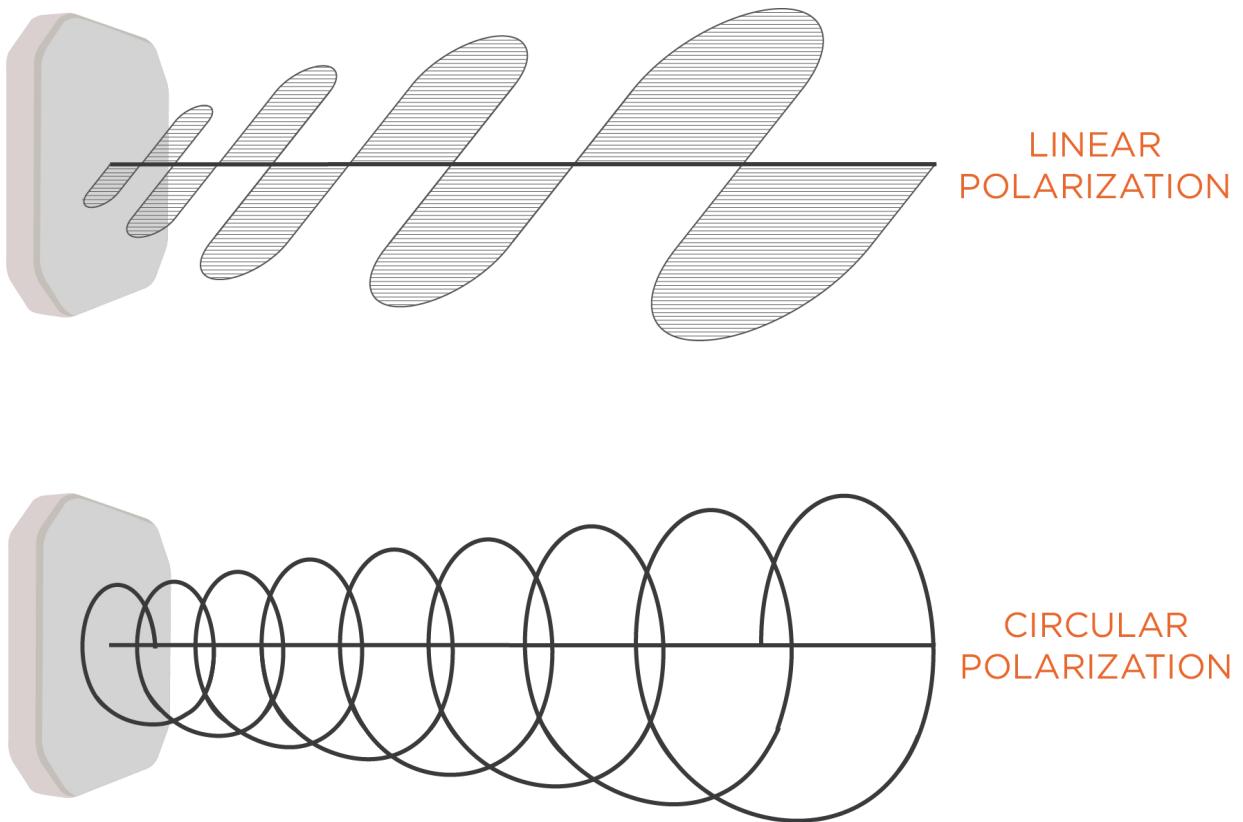
- Low frequency band (LF)
 - 30KHz - 300KHz
 - Reading range: <10cm (depends on reader)
 - Reading speed is lower than the other two
- High frequency band (HF)
 - 3MHz - 30MHz
 - Reading range: <1m (depends on reader)
- Ultra-high frequency band (UHF)
 - 300MHz - 3GHz
 - Reading range: <12m (depends on reader)
 - Reading speed is highest of the 3 categories
 - More sensitivity to radio wave interference

An RFID tag's internal memory is divided into 4 sections (EPC, TID, User Memory and the reversed). TID or Tag Identifier is the section where the tag's UID (Unique ID) is stored. This section cannot be edited. EPC or Electronic Product Code is the section where the tag's NUID is stored (Non-unique ID). This section is editable; users can use it to store the ID of the object they want to track with the RFID tag.

There are different types of RFID reader antennas (Linearly Polarized antennas and Circularly Polarized antennas). Polarization can be understood as the direction of the radio wave's oscillation.

Linearly Polarized antennas radiate waves on a horizontal plane. The receiving tag must be horizontal to the reader as well as being in a consistent distance for the reader to be able to read.

Circular polarization antennas are more flexible because the receiving tag can be read from different angles. But the downside is that we cannot control the precise reading direction of the reader.



Antenna Polarization. Reference: <https://www.atlasrfidstore.com/9-tactics-for-choosing-an-rfid-antenna/#circularlinear>

7.1.2. Application to our project

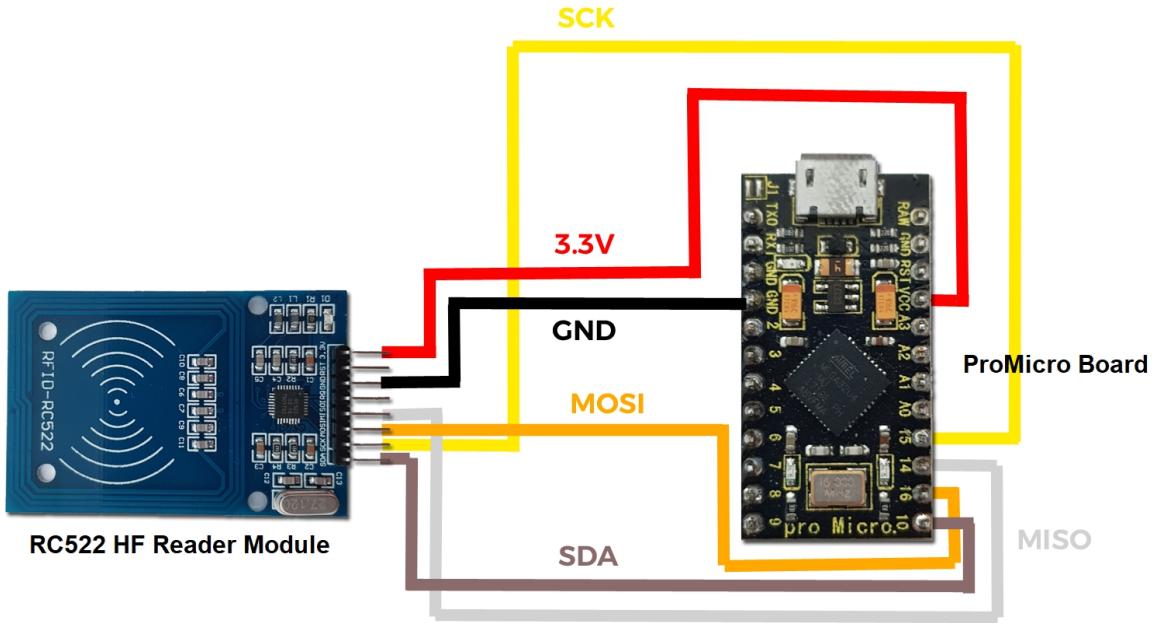
Our team has decided to choose passive tags for our project, they are more reliable than active tags since we don't have to worry about the internal power supply going bad. And they are also cheaper, making maintenance easier since we only need to replace the damaged RFID tag with a new tag.

For RFID operating frequency, we have chosen HF for patron cards and UHF for book tags. We chose HF for the patron card because it has a low reading range, so the card can only be read when the patron intendedly put their card on the scanner. When HF is compared with LF, HF tag has higher memory capabilities while being relatively at the same price. We use UHF for book tags because it has the highest reading range as well as reading speed, allowing us to quickly identify multiple books from distanced.

7.1.3. RFID Readers and Tags

7.1.3.1. Patron Card Reader - HF Reader

For HF readers our team decided to use a RC522 HF reader module because it is capable of both reading and writing while being relatively cheap and easy to find in the market. Instead of connecting RC522 with the popular Arduino UNO board, our team chose the ProMicro Board because it has a ATmega32U4 chip, capable of simulating keystrokes of a computer keyboard.



HF Reader Wiring Schematics. Reference: <https://www.hackster.io/xxlukas84/unlock-your-pc-by-arduino-using-phone-or-rfid-card-970d27>

7.1.3.2. Patron Card - HF Tags



MIFARE Classic 1K card

7.1.3.3. Book Tags Reader - UHF Reader

Because UHF reader modules are hard to find in the market, our team decided to build our own patron card reader but buy a commercial book tag reader from aliexpress.

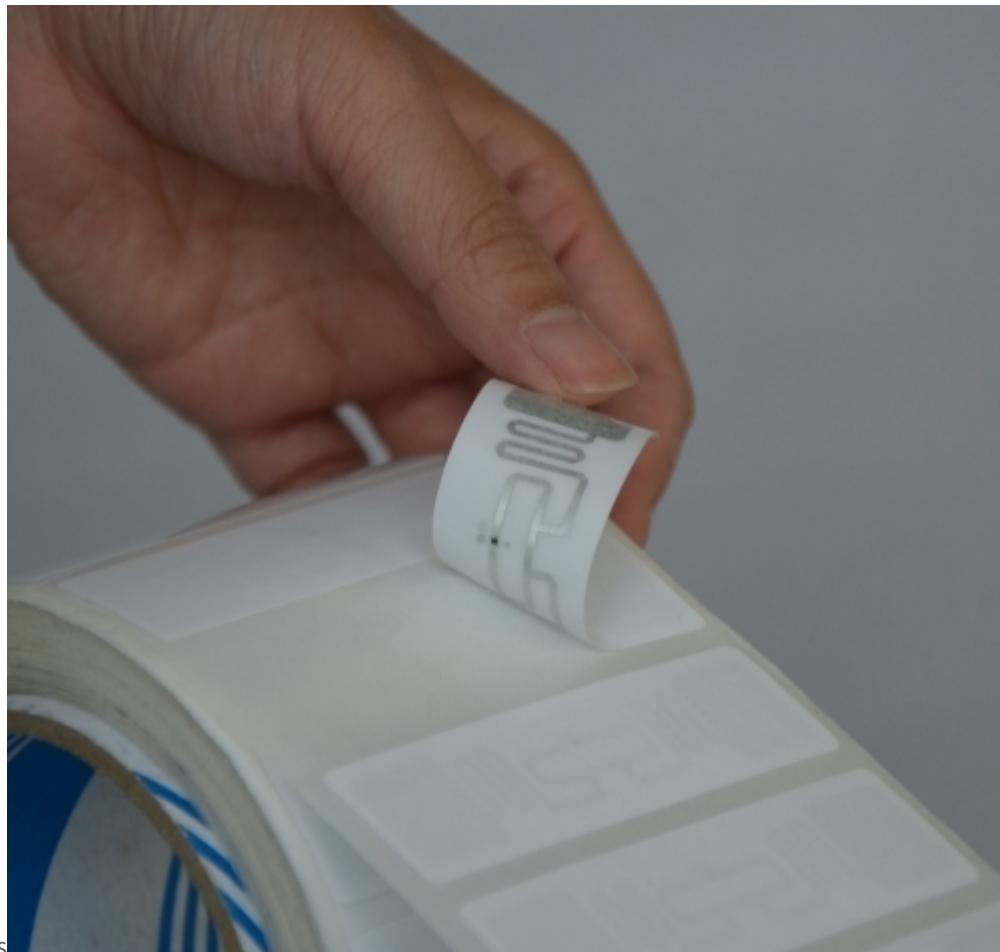


Geenfc GEE-UR-2100 Desktop UHF RFID Reader

Physical	
Material	ABS enclosure
Size	122.4mm(L) x84 mm (W) x 20mm (H) Default: USB, RS232/ RS485, GPIO
Interface	Option A : USB, RS232/ RS485, WG Option B: USB HID virtual keyboard, RS232/ RS485, WG (once configured as virtual keyboard, write function will be deactivated permanently)
Antenna	Built-in circular polarization antenna
Weight	0.6 kg
Protocol	ISO18000-6C & EPCglobal Class I Gen 2
RF & Electrical	
Frequency	US 902~ 928 Mhz, EU 865~ 868 Mhz
RF ouput	0- 26 dBm (software adjustable)
Read/ write Distance	1 meter / 0.5 meter
Polarization	circular polarized
Inventory speed	> 50 tags/ second
Tag buffer	370 tags@128 bit EPC
RSSI	Support
Power Supply	USB d.c 5V or external d.c +9V
Other features	
Software	Demo software, SDK, DLL
Anti-collision	Tag processing algorithm, multiple tags operation allowed
Extension	Allow connect to access controller
Environment	
Operating condition	-20 - 60 C, 5- 95% RH, no condensation
Storage condition	-5- 45 C, 5- 95% RH, no condensation
Certification	CE, ROHS
Accessory	
USB cable	1x

GEE-UR-2100 Specification. Reference: <https://www.aliexpress.com/item/32965153182.html>

7.1.3.4. Book Tags - UHF



Tags

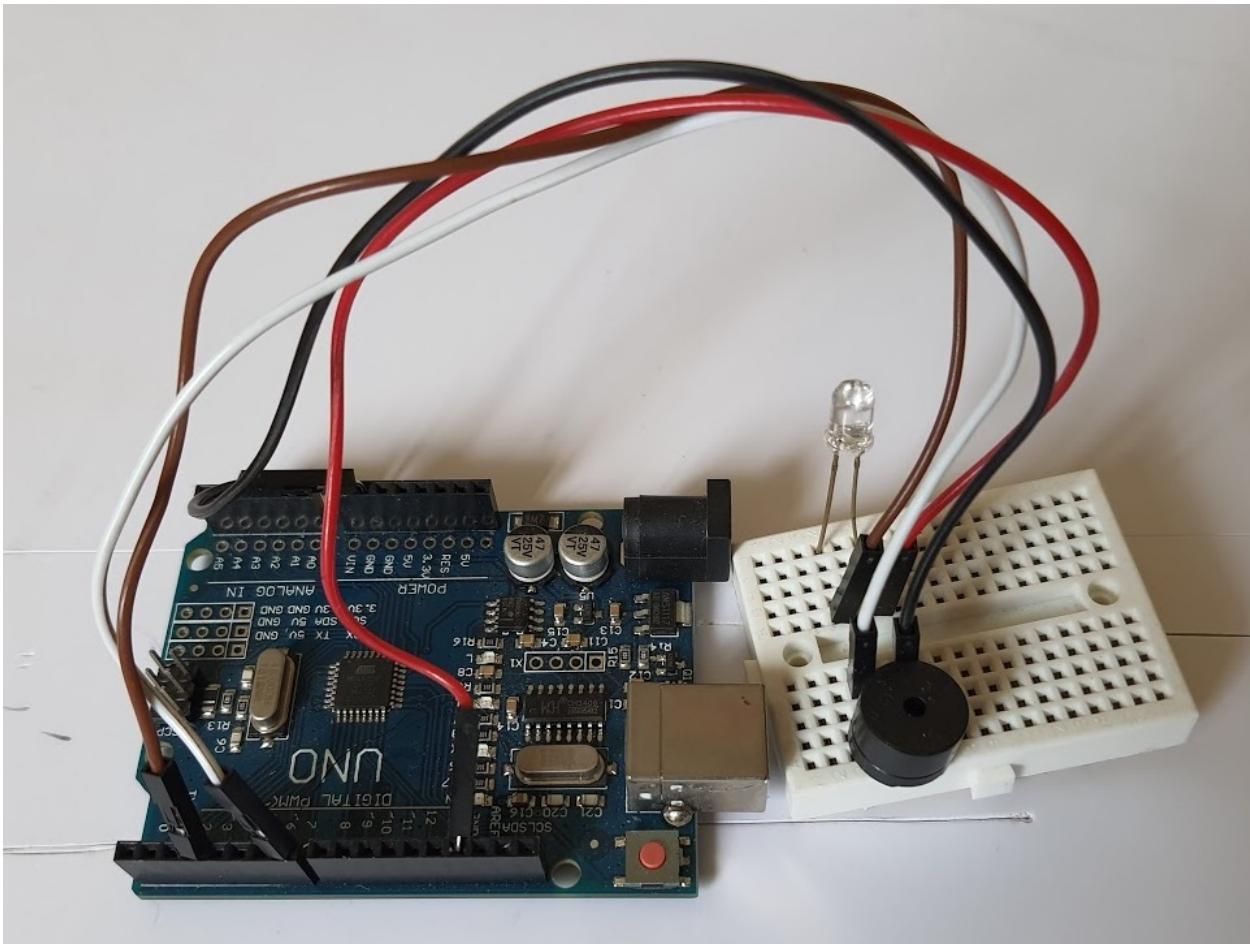
860MHz tag. Reference: <http://www.atdtech.com/index.php/vi/product/rfid/the-dan-rfid-tan-so-cao>

7.2. Security Gate

7.2.1 Description

The purpose of the security gate is to identify and raise the alarm if necessary. The security gate consists of:

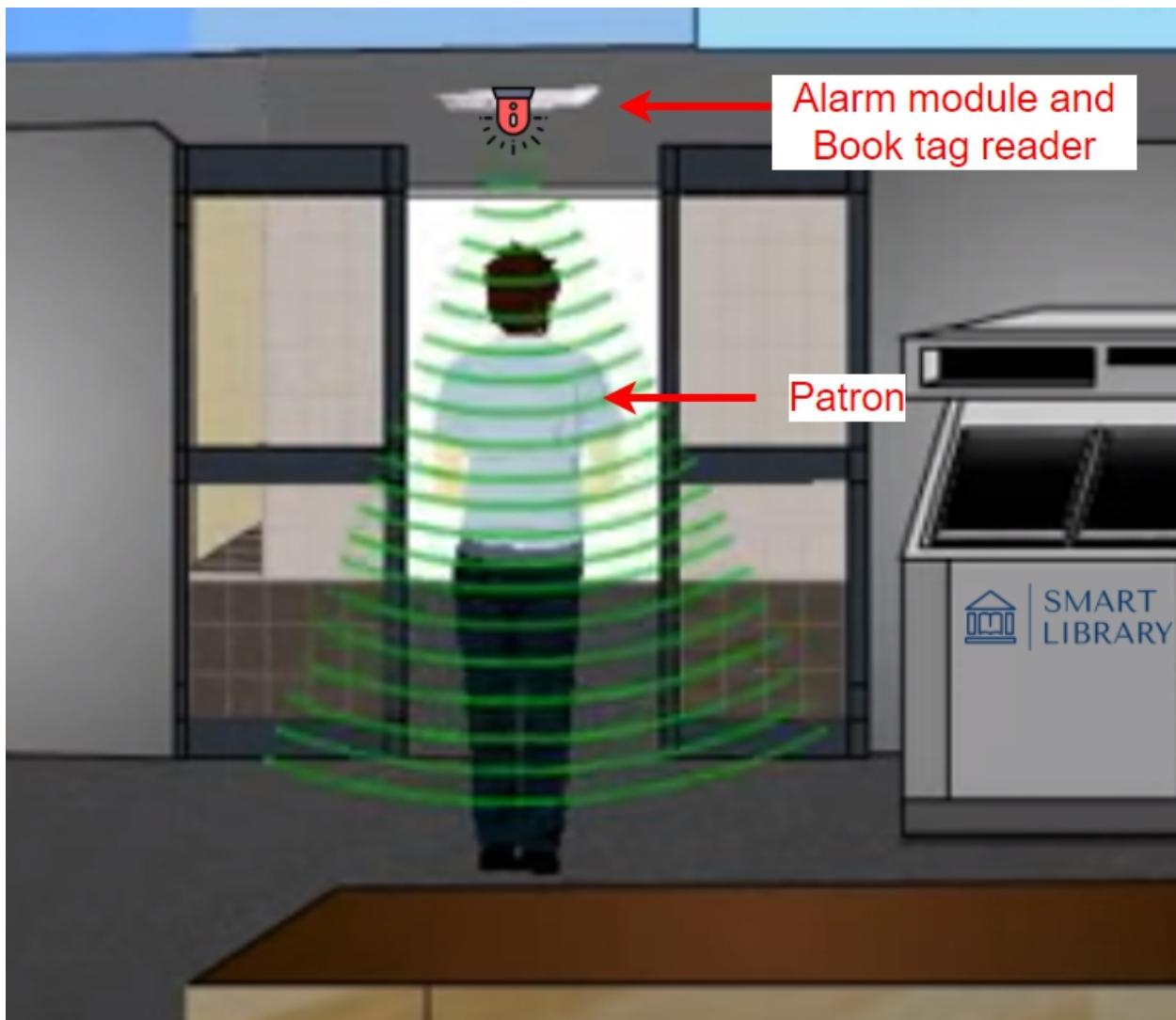
- Book tag reader (refers to 7.1.3.3)
- Alarm module:
 - [Passive buzzer](#) (refers to 3.8)
 - [LED](#) (refers to 3.9)
 - [Arduino UNO](#) (refers to 3.4)
 - [Breadboard](#) (refers to 3.10)



Alarm Module

7.2.2 Installation

The security gate will be installed above the entrance of the library with the direction of the book tag reader antenna faces downward.



Security Gate Installation

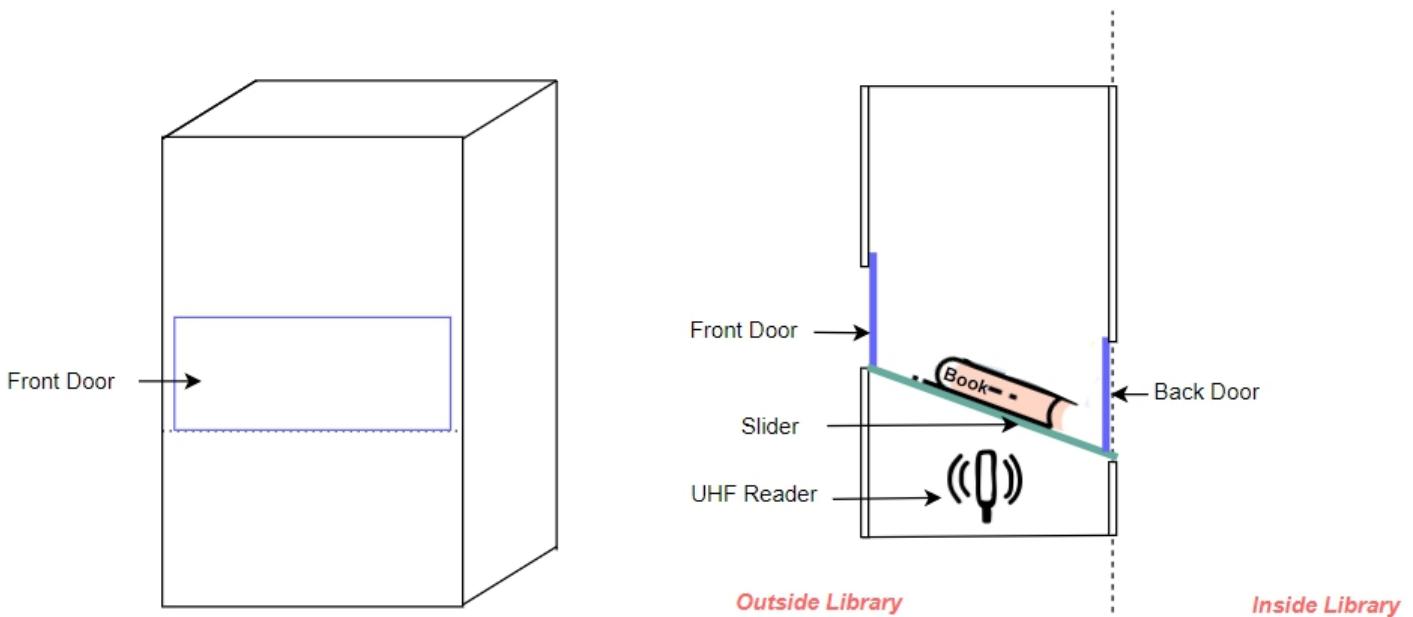
7.2.3 Limitation

The security gate heavily depends on the quality of the book tag reader (reading range and accuracy of the reader).

7.3. Book Drop

7.3.1. Description

We built a window application that is embedded in the book drop station that allows the patron to return a book outside the library. The book drop is designed as figure below:



Book Drop Design

The book drop is made from plastic foams which has the following main components:

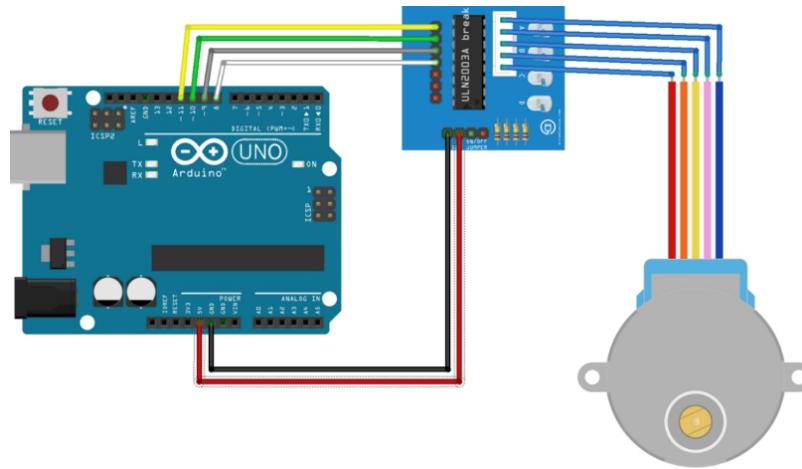
- Front door: allow the patron to open and put the book on the slider inside the box
- Back door: only open to let the book go inside the library when that book is returned successfully
- Slider inside the box from front door to back door: that allows the book to slide down smoothly to the library's storage
- An upper drawer that store stepper modules that monitor the motion of two doors

There is a window application that simulates a whole book drop from front door to back door and the RFID reader. The application runs on a computer that connects with two stepper motors and the UHF reader through USB. Each device is assigned to a particular port and communicates with each other through serial communication. The window application plays as a main controller that orders those devices to do particular functions. For example, the application sends a message asking the stepper motor to rotate in order to open the door.

7.3.2. Devices

To implement to book drop station, we use the following devices:

- Book tag reader (refers to 7.1.3.3) to scan book
- Stepper motor modules to monitor front door and back door:
 - Stepper motor (refers to 3.3)
 - [Stepper motor controller](#) (refers to 3.12)
 - [Arduino UNO](#) (refers to 3.4)



Stepper Motor Wiring Schematics. Reference: <https://www.aranacorp.com/en/control-a-stepper-motor-with-arduino/>

7.3.3. Limitation

The book drop heavily depends on the quality of the UHF reader (reading range and accuracy of the reader). The scanning range must not reach out the box to make high security. However, we have limits to control the UHF's scanning range.

7.4 Obstacles and Solutions

7.4.1 Security Gate

Because the devices are not compatible with each other out of the box, our team has implemented some middleware applications to control and connect devices together via serial ports.

For the security gate, we have implemented an application to connect the Book tag reader with the alarm module via serial port. Each time the book tag reader detects a book it will send the book's ID to our application, the application will decide whether to tell the Alarm module to raise the alarm. (Refers to illustration 7.4.1)

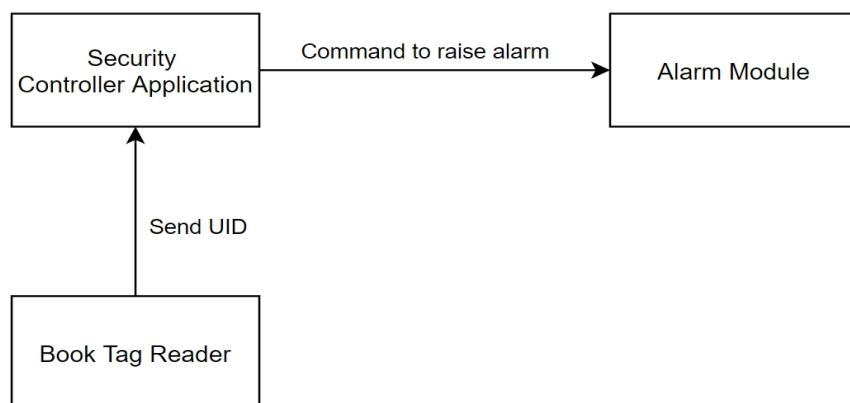


Illustration 7.4.1

7.4.2 Book Tag Reader (UHF RFID Reader)

Because the UHF RFID reader will be read from a far - not in close contact like a patron card reader (HF RFID reader), polarization is an important factor of book tag reader. Although having a less precise reading direction, because of the ability to read tags from different angles is required for the security gate our team has decided to pick a circularly polarized antenna for our book tag reader. This affects the accuracy of locating books location functionality of our system. To overcome the problem, our team has decided to reduce the reading range of the book tag reader when it is used to locate book location and increase the reading range when it is used as the security gate.

The book tag reader has 2 data-output modes: simulating a keyboard or output through serial port. We cannot use the keyboard mode for the book drop and security gate. Because in that mode, the reader only reads each book's RFID tag once when it detects a tag presence (in order to read the same tag again, we would have to take the card out of the reading range first before trying to read it again). And because this mode is not fast enough for the security gate. So we chose the output via serial port mode which can quickly read every tag within the reading range of the reader at the same time. But this mode can only be used via "UHFReader86Demo V1.11" - an application provided by the manufacturer. However, our team has limited control over the third party application and it does not work well with our system.

So we have customized the 3rd party application source code and implemented a console application called Middleware application for the book drop. Our Middleware application will act as a buffer between UHFReader86Demo V1.11 and our system. When the patron is returning a book, after the book drop front door has closed completely, the book drop will request the Middleware application to return the book's RFID tag information that it received from the 3rd party application (Refers to illustration 7.4.2).

For the security gate, our Security Controller Application is able to work directly with the customized UHFReader86Demo V1.11 without having to go through the Middleware application. Every time the UHF RFID reader detects a book tag, it will send the tag information to UHFReader86Demo V1.11 and then UHFReader86Demo V1.11 will send that information to our system.

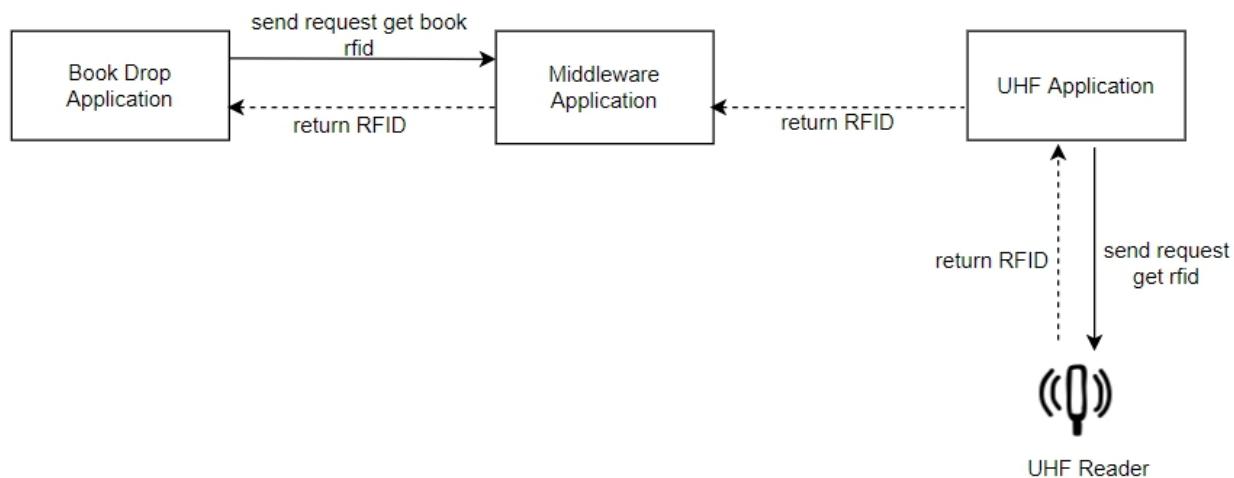


Illustration 7.4.2

7.4.3 Serial Ports

Finding a way to connect different devices and 3rd party applications to our system was a big challenge for our team. Since there isn't any interface for them to communicate with each other. After doing research, our team has decided to create an interface for the components to communicate by letting them send and receive messages to each other via serial ports. In order to do that, we created virtual COM ports and linked them together using Eltima Software's Virtual Serial Port Driver (<https://www.eltima.com/products/vspdxp/>). And then we implemented each component in our system so that they can send and receive messages from other components through serial ports.

V. Software Testing Documentation

1. Overall Description

1.1. Test Model

We choose Agile Testing as the following reasons:

- Agile testing provides an ongoing feedback loop into the development process
- We can quickly and flexibly adapt with requirements changes
- Every member of the team is responsible for testing after implementation

1.2. Testing Levels

We choose the following testing levels for our project:

- Unit testing: only use for self-checkout, self-return, checkout, return, security, tagging, search book, check misplace or missing book
- Acceptance testing: let users try the application, we can get feedback from them.

2. Test Plan

2.1. Test Stages

Type of Test	Stage of Test			
	Unit	Integration	System	Acceptance
Functional Test	X	X		
User Interface Test				X

2.2. Resources

2.2.1. Human Resources

Worker/Doer	Role	Specific Responsibilities/Comments
Phan Hoàng Trâm	Leader	Planning, verifying test deliverables
Phạm Minh Hoàng	Member	Do unit testing as planned
Nguyễn Trung Kiên	Member	Do unit testing as planned
Nguyễn Đỗ Nhật Khang	Member	Do unit testing as planned

2.3. Test Milestones

Milestone Task	Effort (md)	Start Date	End Date
Self-services (self-return, self-checkout) Security Gate	2	28/01/2021	29/01/2021
Manage book Checkout Return	3	10/02/2021	11/02/2021
Manage wishlist Manage account	2	25/02/2021	26/02/2021
Manage lost book Manage policies	2	11/03/2021	12/03/2021
Search book Find misplaced, missing item	2	25/03/2021	26/03/2021
Optimization	1	08/04/2021	08/04/2021

3. Test Cases

[Test Case Document](#)

4. Test Reports

[Test Report Document](#)

VI. Release Package & User Guides

1. Deliverable Package

1.1. Source codes & documents

No.	Items	Sub-Items	Type	Version
Code Package				
1	API for Web	Library-BE	New	1.0
2	API for Win	SelfCheckOutReturnBE	New	1.0
3	Front End	LibraryWebApp	New	1.0
4	Self Service Wn App	LibrarySelfCheckOut	New	1.0
5	Book Drop Win App	BookDrop	New	1.0
6	Security Win App	SecurityGate	New	1.0
Database				
1	Tables	V7_full_script.sql	New	1.0
Documents				
1	Requirement	RFID_Library_Management_System_Full_Final.pdf	Modify	Final

1.2. Known Issues, Limitations & Restriction

Our software depends highly on the hardware and devices: UHF reader, NHF reader, stepper motor. The network connection needs to be fast to load the image.

2. Installation Guides

2.1. System Requirements

2.1.1. Hardware Requirements

Hardware	Specification
Processor	Intel i5, 4 cores
Computer Memory	8GB RAM or more

Storage Space	500GB
Internet Connection	Cable, Wi-Fi

2.1.1. Software Requirements

Software	Name/Version	Description
Operation System	Window 10	Operating system and platform for development
DBMS	MySQL 8.0	Manage database
Web Browser	Chrome	Browser
Web server	Apache Tomcat	Development environment
IDE	IntelliJ, Visual Studio Code, Visual Studio, Arduino	Programming tools
UML	Star UML, Draw.io	Design diagram

2.2. Setup Files

Script command/File name	Description
git clone <repository-link>	Clone the project
npm install	Install library for front end
npm start	Run the web front end on the browser
mvn clean install	Remove all generated file and install the package in to the local repository
mvn spring-boot:run	Run the installed spring boot project in local server

2.3. Installation Instruction

2.3.1. Database

- Clone the project from git repository
- Open RLMS_full_script on MySQL workbench
- Run the script

2.3.1. Web Server

- Clone the project from git repository
- Install package: run in command prompt “mvn clean install”
- Run the server: run in command prompt “mvn spring-boot:run”

2.3.2. Front End

- Clone the project from git repository
- Install packages: npm install
- Run on browser: npm start

2.3.3. Window Application

- Clone the project from git repository: get clone
- Run on browser: double click on file .exe

3. User Manual

3.1. Terms and definitions

Term	Definition
Book	Is the information of a book title in the system
Book copy	Is information of a physical copy of a book inside the library
Borrow Policy	A policy that determines: <ul style="list-style-type: none">● Which patron type can borrow which book copy type. For example, Students can only borrow Textbooks, but Teachers can borrow Textbooks and Reference books.● How many copies of a copy type can be borrowed by a patron type. For example, a student can borrow 3 reference books but can borrow 10 textbooks.● How many times a book copy can be renewed● How long a book copy can be borrowed by the patron.● How long a book can be borrowed after it is renewed.
Patron Policy	A policy that determines the maximum books that a patron type can borrow at a time. For example, a student can borrow 20 book copies in total across all book copy types.
Fee Policy	A policy that determines the parameters in the fine calculating formula.

Call number	Is an address for a book. A call number also represents some of the book information like what is the content of the book, etc. Books in the library are arranged in call number order (numerical and alphabetical). The two common methods for classifying call numbers are DDC and LC. For this project, the call number format will be: DDC + Author information + publish year. For example "100 JOH 2007"
Borrow Period (days)	The time period that a patron is allowed to keep a borrowed book copy
Renew Period (days)	The time period that will be added to the current due date of a renewing book copy.
Checkouts Allowed (count)	The number of book copies from a specific a patron is allowed to borrow
Renewals Allowed (times)	The number of times a patron is allowed to renew a book copy
Patron Type	Different types of patrons, for example: students, lecturers, etc. Each patron type will follow different policies.
Book Copy Type	Different types of book copies, for example: reference books, textbooks, etc. Each book copy type will have different borrow policies.

Table 3.1. Terms and definitions

3.2. System Requirements

3.2.1. Hardware Requirements

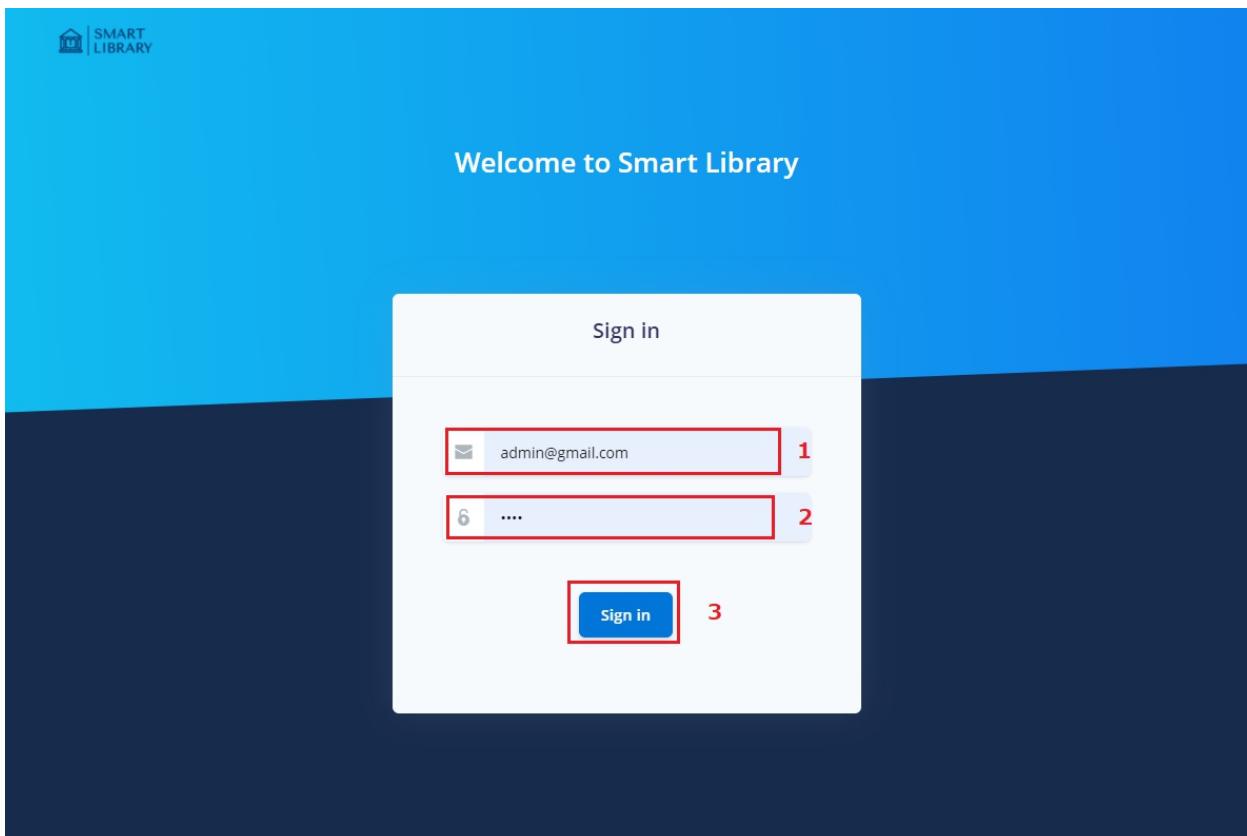
Hardware	Specification
Processor	Intel i5, 4 cores
Computer Memory	8GB RAM or more
Storage Space	500GB
Internet Connection	Cable, Wi-Fi

3.2.1. Software Requirements

Software	Name/Version	Description
Operation System	Window 10	Operating system and platform for development
DBMS	MySQL 8.0	Manage database
Web Browser	Chrome	Browser
Web server	Apache Tomcat	Development environment
IDE	IntelliJ, Visual Studio Code, Visual Studio, Arduino	Programming tools
UML	Star UML, Draw.io	Design diagram

3.3. Application Usage

3.3.1. <Unauthenticated User> Login

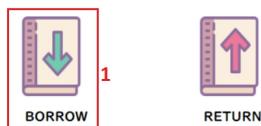


Step	Description
1	Fill in account email
2	Fill in account password
3	Click “Sign in” to login

3.3.2. <Unauthenticated User> Check in



WELCOME TO SMART LIBRARY





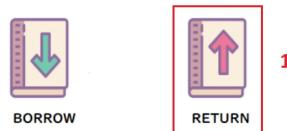
PLEASE SCAN YOUR ID CARD

Step	Description
1	Click on “BORROW” on the main screen
2	Scan the patron card by card reader

3.3.3. <Unauthenticated User> Self-return at the kiosk inside library



WELCOME TO SMART LIBRARY



RETURN SERVICE

NUMBER OF SCANNED BOOKS: 1

Monday, 29 March 2021

The Sun Is Also A Star
Edition: 1
Author(s): G. Willow
Group: REFERENCE

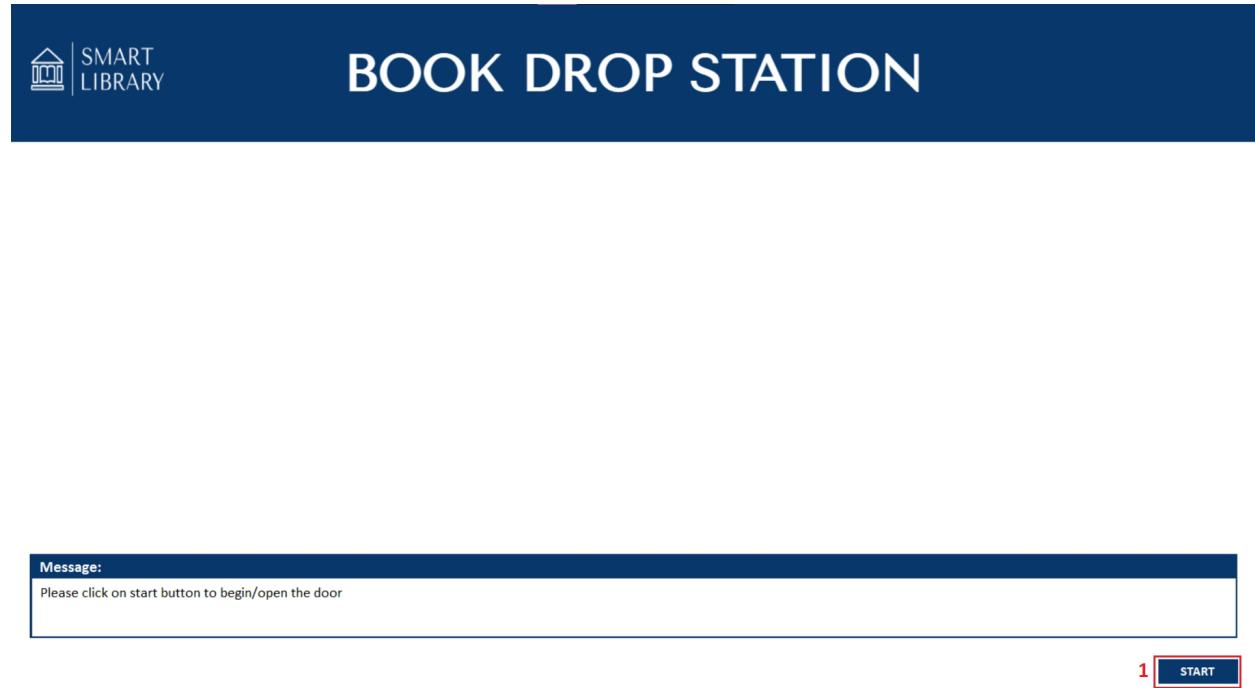
SESSION TIMEOUT: 235

CANCEL RETURN 3

Step	Description
1	Click on “RETURN” on the main screen

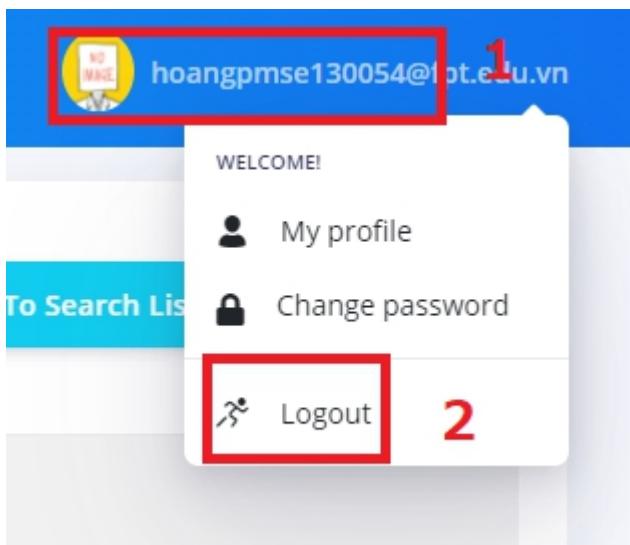
2	Put books on the scanner pad
3	Click on “RETURN” button

3.3.4. <Unauthenticated User> Self-return at the book drop outside library



Step	Description
1	Click on “START” on the main screen
2	Put a book into the box

3.3.5. <Authenticated User> Log out



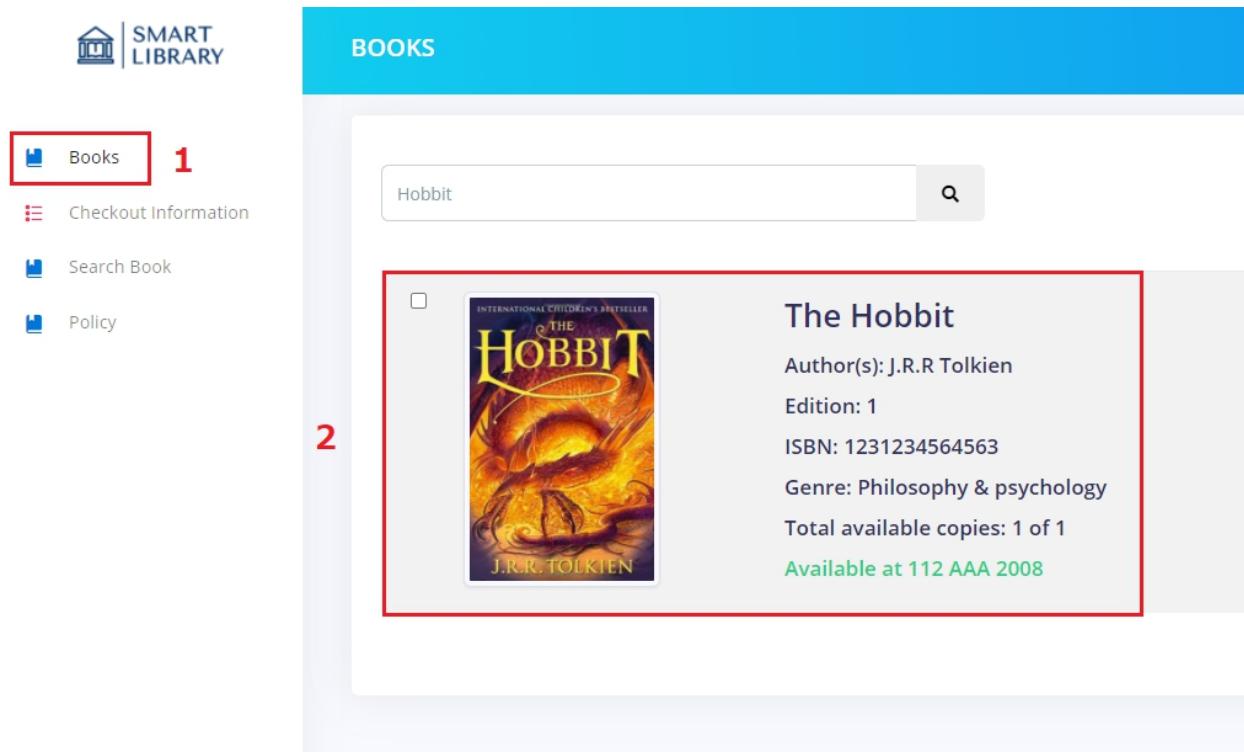
Step	Description
1	Click on the email on the header
2	Click "Logout" to log out

3.3.6. <Authenticated User> Search books

The screenshot shows a library search interface. On the left, a sidebar menu includes 'Books' (highlighted with a red box and the number '1'), 'Checkout Information', 'Search Book', and 'Policy'. The main area has a blue header bar with the word 'BOOKS'. Below the header, a search bar contains the text 'Hobbit' (highlighted with a red box and the number '2')). To the right of the search bar is a magnifying glass icon (highlighted with a red box and the number '3')). On the right side, a book cover for 'The Hobbit' by J.R.R. Tolkien is shown, along with its details: Author(s): J.R.R. Tolkien, Edition: 1, ISBN: 1231234564563, Genre: Philosophy & psychology, Total available copies: 1 of 1, and Available at 112 AAA 2008.

Step	Description
1	Click on “Books” on the sidebar
2	Enter book’s title or ISBN
3	Click on the search button to search

3.3.7. <Authenticated User> View book’s details



The image shows the Smart Library application interface. On the left, there is a sidebar with a logo and four options: Books, Checkout Information, Search Book, and Policy. The 'Books' option is highlighted with a red box and the number '1' above it. The main area has a blue header bar with the word 'BOOKS'. Below the header is a search bar containing the text 'Hobbit' with a magnifying glass icon to its right. A red box labeled '2' highlights a book card for 'The Hobbit' by J.R.R. Tolkien. The book cover features a dragon and the title 'THE HOBBIT'. To the right of the cover, the book's details are listed: Author(s): J.R.R. Tolkien, Edition: 1, ISBN: 1231234564563, Genre: Philosophy & psychology, Total available copies: 1 of 1, and Available at 112 AAA 2008.

Step	Description
1	Click on “Books” on the sidebar
2	Click on a book in the book list to view the book details

3.3.8. <Authenticated User> Change password

The screenshot shows a 'CHANGE PASSWORD' page. At the top right, there is a user profile icon with the email 'tuongnt1@fpt.edu.vn' and a red box labeled '1'. Below the profile, a 'WELCOME!' message is displayed with links: 'My profile' (with a person icon), 'Change password' (with a lock icon, red box labeled '2'), and 'Logout' (with a person icon). The main form area contains three input fields: 'Current password*' (red box labeled '3'), 'New password*' (red box labeled '3'), and 'Confirm password*' (red box labeled '3'). Below these fields is a note '* Required field'. At the bottom right of the form is a green 'Save' button (red box labeled '4').

Step	Description
1	Click on username in the navigation bar
2	Click on “Change password” button
3	Fill in required fields
4	Click on “Save” button

3.3.9. <Authenticated User> Add books to a searching list

The screenshot shows the SMART LIBRARY application interface. On the left, a sidebar has 'Books' selected (Step 1). The main area is titled 'BOOKS' and contains a search bar. Two book entries are listed:

- Bố Già : Mario Puzo**
Author(s): J.R.R Tolkien
Edition: 1
ISBN: 1231234564567
Genre: Philosophy & psychology
Total available copies: 2 of 10
Available at 111 AAA 2008
- The Hobbit**
Author(s): J.R.R Tolkien
Edition: 1
ISBN: 1231234564563
Genre: Philosophy & psychology
Total available copies: 1 of 1
Available at 112 AAA 2008

A modal dialog box titled 'Add books to search list' (Step 3) is open, asking 'Are you sure?' (Step 4). It includes 'Close' and 'OK' buttons.

Step	Description
1	Click on “Books” on the sidebar
2	Check books to add to search list
3	Click on “Add to Search List”
4	Click on “OK” to add to search list

3.3.10. <Authenticated User> Search books by UFH reader

SEARCH BOOK

CLICK "START" TO SCAN

SEARCH BOOK

SCANNING BOOKS

Step	Description
1	Click on “Search Book” on the sidebar
2	Click “Start” button to start searching and move the scanner
3	Tick on book that was found
4	Click on “Finish” to finish searching

3.3.11. <Patron> Self-checkout at the kiosk inside library

CHECK-OUT SERVICE

hoangpmse130054@fpt.edu.vn

NUMBER OF SCANNED BOOKS: 1

Monday, 29 March 2021

Nicholas

Nicolas: Part 1

Edition: 1

Author(s): Cline

Group: REFERENCE



SESSION TIMEOUT: 235

CANCEL

CHECK OUT

2

Step	Description
1	Put books on the scanner pad
2	Click on “CHECK OUT” button

3.3.12. <Patron> Add book the wishlist

The screenshot shows a library interface. On the left, there's a sidebar with links: 'Books' (highlighted with a red box and the number '1'), 'Checkout Information', 'Search Book', and 'Policy'. The main content area has a blue header bar with the word 'BOOKS'. Below it, a book cover for 'Understanding Dreams' by Ernest is displayed, featuring two sheep. To the right of the cover, the book's title and details are listed: 'Understanding Dreams : (Collins Gem) Paperback – March 7, 2005', 'Author(s): Ernest', 'Edition: 1', 'ISBN: 0007183984', 'Genre: Philosophy & psychology', and 'Not available'. A red box highlights the 'Add to Wishlist' button at the top right, with the number '2' indicating it's been interacted with.

Step	Description
1	Click on “Books” on the sidebar
2	Click on “Add to Wishlist” on unavailable books to add the book to wishlist

3.3.13. <Patron> View wishlist

The screenshot shows the SMART LIBRARY website interface. On the left sidebar, there are links for Books (highlighted with a red box and labeled '1'), Checkout Information, Search Book, and Policy. The main content area is titled 'BOOKS' and features a search bar with placeholder text 'Search by ISBN or title' and a magnifying glass icon. To the right of the search bar are two buttons: '+ Add To Search List' and 'My Wishlist' (also highlighted with a red box and labeled '2'). Below the search bar, there are two book entries:

- Bố Già : Mario Puzo**
Author(s): J.R.R Tolkien
Edition: 1
ISBN: 1231234564567
Genre: Philosophy & psychology
Total available copies: 2 of 10
Available at 111 AAA 2008
- The Hobbit**
Author(s): J.R.R Tolkien
Edition: 1
ISBN: 1231234564563
Genre: Philosophy & psychology
Total available copies: 1 of 1
Available at 112 AAA 2008

Step	Description
1	Click on “Books” on the sidebar
2	Click on “My Wishlist” to view wishlist

3.3.14. <Patron> View profile

The screenshot shows the SMART LIBRARY website with a user profile dropdown menu. The menu items are:

- 1 **hoangpmse130054@fpt.edu.vn** (Email address, highlighted with a red box)
- 2 **My profile** (Profile link, highlighted with a red box)
- Change password
- Logout

Step	Description
1	Click on the email on the header
2	Click on “My Profile” to view your profile

3.3.15. <Patron> Update profile



2 Save

Fullname: Pham Minh Hoang

Email: hoangpmse130054@fpt.edu.vn

Phone: 0938072402 1

Step	Description
1	Edit phone number
2	Click on “Save” to update profile

3.3.16. <Patron> View borrow history

The screenshot shows the Smart Library interface. At the top left is the library logo with the text "SMART LIBRARY". To the right is a large blue header bar with the text "CHECKOUT INFORMATION". Below this is a sidebar with several options: "Books", "Checkout Information" (which is highlighted with a red border and has a red number "1" to its left), "Search Book", and "Policy". To the right of the sidebar, three sections are listed under the "CHECKOUT INFORMATION" header: "Overdue Books", "Borrowing Books", and "Returned Books".

Step	Description
1	Click on “Checkout Information” on the sidebar to view checkout information

3.3.17. <Patron> Renew book

The image consists of two vertically stacked screenshots of a library management system interface.

Screenshot 1 (Top): Checkout Information - Renewal Step

- Step 1:** The sidebar shows "Checkout information" highlighted with a red box.
- Step 2:** A row in the borrowing list has a "Renew" button highlighted with a red box.
- Step 3:** A modal dialog titled "Renew Due Date" shows the new due date as "2021-05-07". The "Confirm" button is highlighted with a red box.

BOOK	BORROWED AT	DUUE DATE	Actions
Harmony - Edition[1]	2021-04-09, 02:46:07	2021-04-30	Renew History Lost
Harmony - Edition[1]	2021-04-09, 01:05:00	2021-04-30	Renew History Lost

Screenshot 2 (Bottom): Checkout Information - Confirmation Step

- Step 4:** A success message dialog is displayed, stating "Renewed successfully" with a green checkmark icon. The "Close" button is highlighted with a red box.

BOOK	BORROWED AT	DUUE DATE	Actions
Harmony - Edition[1]	2021-04-09, 02:46:07	2021-04-30	Renew History Lost
Harmony - Edition[1]	2021-04-09, 01:05:00	2021-04-30	Renew History Lost

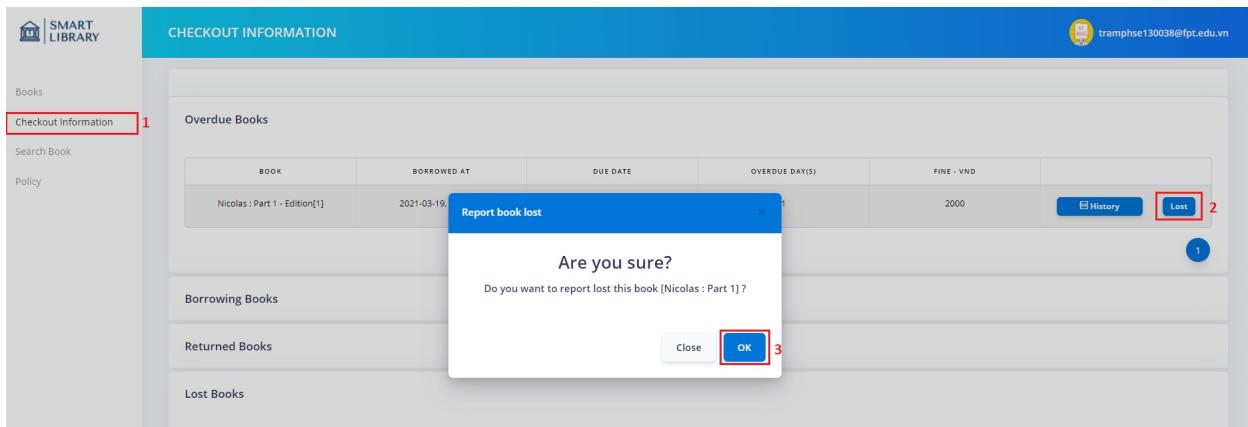
Step	Description
1	Click on “Checkout information” on the sidebar
2	Click on the “Renew” button on a borrowing from the borrowing list
3	Click on “Confirm”
4	Click “Close” to finish

3.3.18. <Patron> View renew history

The screenshots illustrate the process of viewing a patron's renew history. In the first screenshot, the user has selected a borrowing record from the list. Step 1 highlights the 'Checkout information' link in the sidebar. Step 2 highlights the 'History' button in the row of actions for the selected borrowing record. In the second screenshot, a modal window is open, displaying the renewal history for the selected book. It shows two entries: one initial borrow and one renewal. The 'Close' button is visible at the bottom right of the modal.

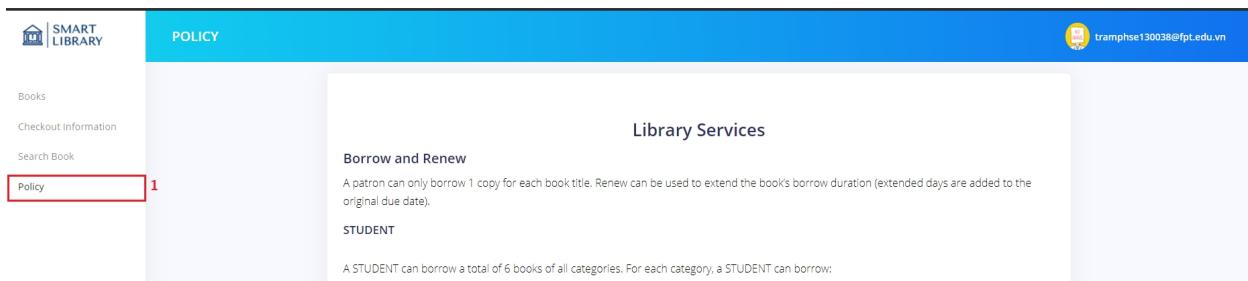
Step	Description
1	Click on “Checkout information” on the sidebar
2	Click on the history button on a borrowing from the borrowing list

3.3.19. <Patron> Report lost book



Step	Description
1	Click on “Checkout Information” on the sidebar to view checkout information
2	Click on “Lost” to report that book was lost
3	Click on “OK” to confirm

3.3.20. <Patron> View library's policy



Step	Description
1	Click on “Policy” on the sidebar to view policies of library

3.3.21. <Librarian> Checkout books

CHECK OUT BOOKS

SCAN PATRON'S CARD

SCAN CHECK OUT BOOKS

Patron information

Checking out book(s)

Check out books 1

Search by patron's email 2

2.1

Q

Clear

Check out 4

3.1

3

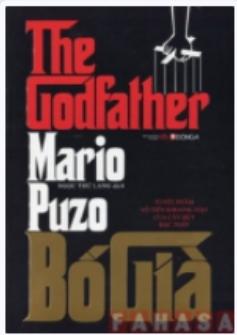
0

01123400000012

Bố Già: Mario Puzo
by J.R.R Tolkien
Edition: 1
Barcode: 01123400000012
Book Type: REGULAR
ISBN: 1231234564567
Overdue date: 2021-01-05

Confirm checkout

X



Bố Già: Mario Puzo

by J.R.R Tolkien

Edition: 1

Barcode: 01123400000012

Book Type: REGULAR

ISBN: 1231234564567

Overdue at: 2021-04-05

1

5

* Required field

Close

Confirm

Step	Description
1	Click on “Check out books” on the sidebar to enter the checkout screen
2	Scan the patron card or enter the email with keyboard
2.1	Click search to get the checkout patron information (Skip this step if scanning patron card in step 2)
3	Scan the book RFID tag or enter the book barcode with keyboard
3.1	Click search to get the checkout book information (Skip this step if scanning RFID tag in step 3)
4	Click on “Check out” button
5	Click on “Confirm” button to finish checkout

3.3.22. <Librarian> Return books

1

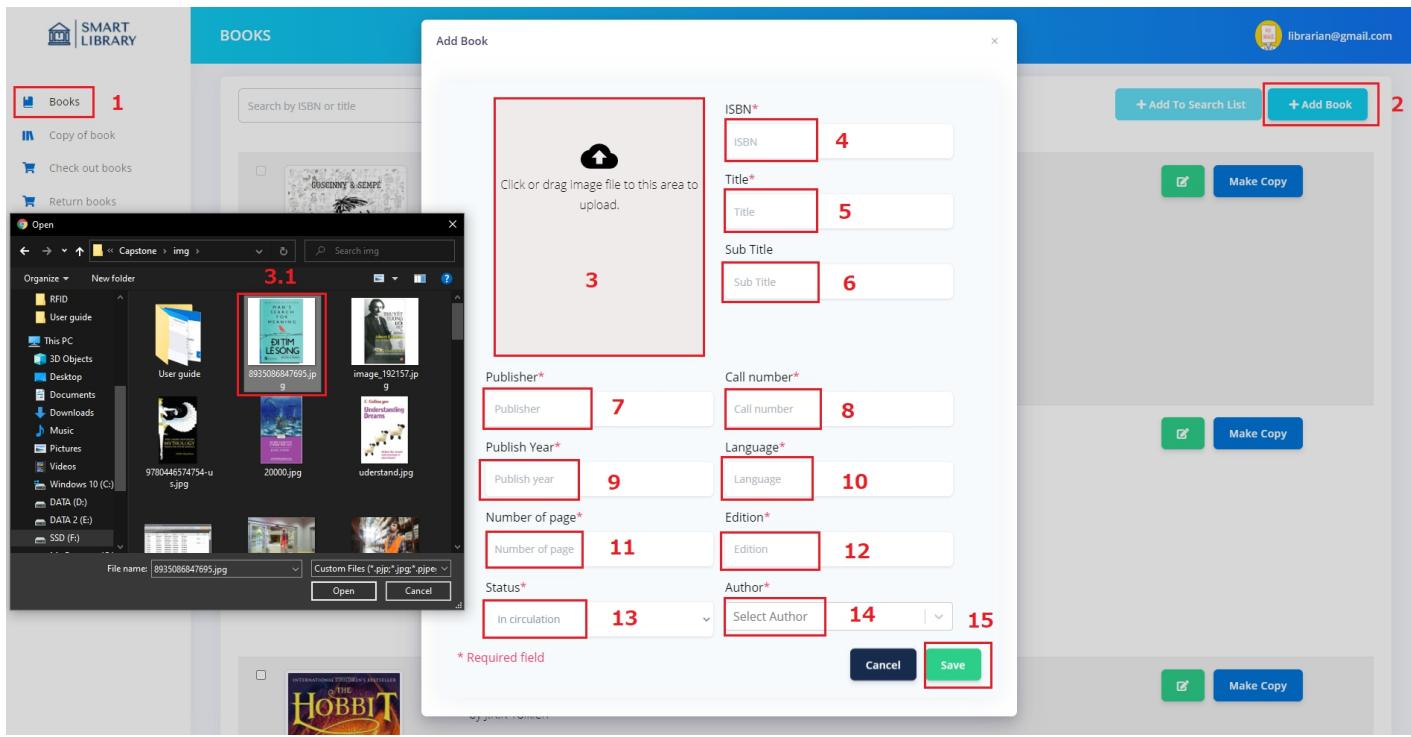
2

3

4

Step	Description
1	Click on “Return books” on the sidebar to enter the checkout screen
2	Scan the book RFID tag or enter the book barcode with keyboard
2.1	Click search to get the return book information (Skip this step if scanning RFID tag in step 2)
3	After all the return books is shown on screen, click on the “Return” button
4	Click on “Confirm” button to finish return

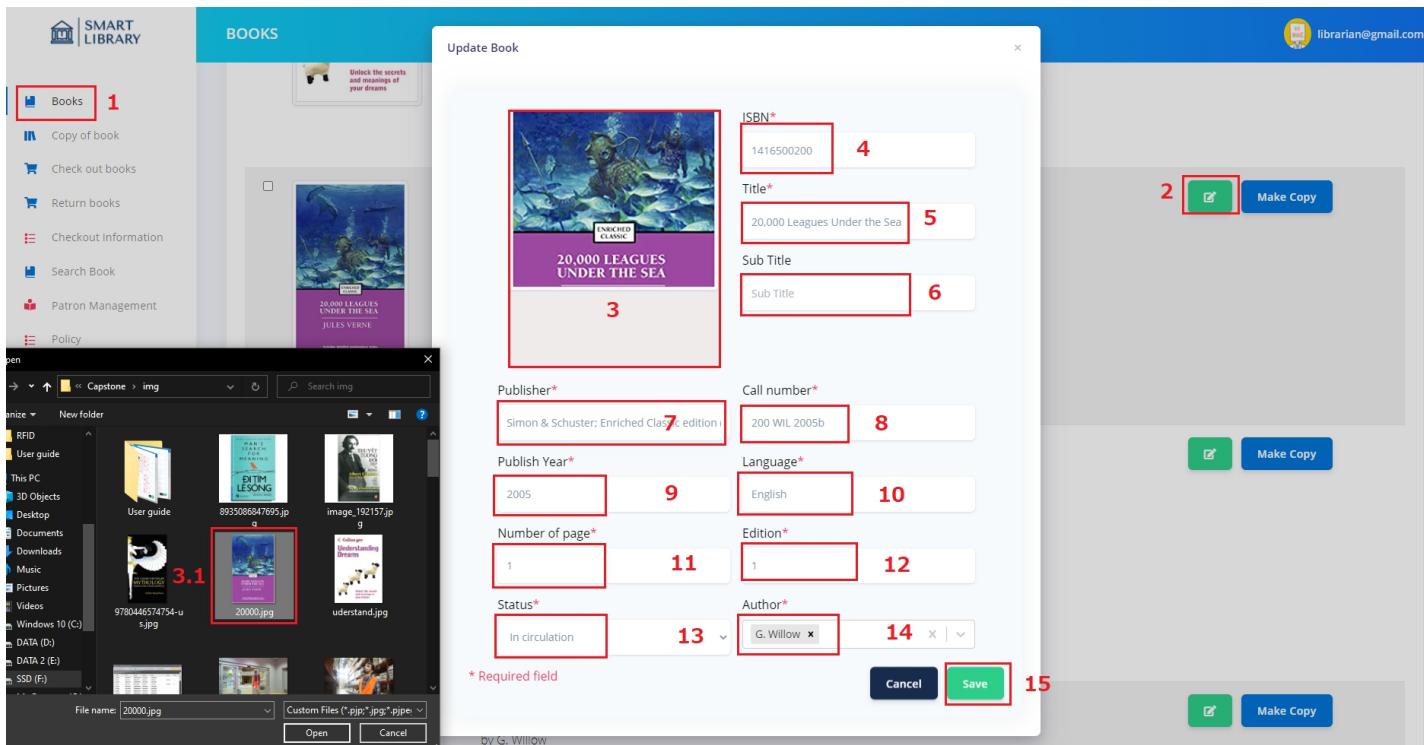
3.3.23. <Librarian> Add book



Step	Description
1	Click on "Books" on the sidebar
2	Click on "Add Book"
3	Click to open my computer
3.1	Pick a book image from the computer
4	Enter the book's ISBN
5	Enter the book's title
6	Enter the book's subtitle
7	Enter the book's publisher
8	Enter the book's call number
9	Enter the book's publish year
10	Enter the book's language
11	Enter the book's number of page

12	Enter the book's edition
13	Pick a status for the book
14	Pick the book's author(s)
15	Click on "Save" to finish adding book

3.3.24. <Librarian> Update book



Step	Description
1	Click on "Books" on the sidebar
2	Click on the edit button on a book from the book list
3	Click to open my computer (Optional)
3.1	Pick a new book image from the computer
4	Enter the book's new ISBN (Optional)
5	Enter the book's new title (Optional)
6	Enter the book's new subtitle (Optional)

7	Enter the book's new publisher (Optional)
8	Enter the book's new call number (Optional)
9	Enter the book's new publish year (Optional)
10	Enter the book's new language (Optional)
11	Enter the book's new number of page (Optional)
12	Enter the book's new edition (Optional)
13	Pick a new status for the book (Optional)
14	Pick the book's new author(s) (Optional)
15	Click on "Save" to finish update book

3.3.25. <Librarian> Add books to wishlist for patron

The screenshot illustrates the process of adding a book to a patron's wishlist. The interface is divided into two main sections: a sidebar on the left and a main content area on the right.

Left Sidebar:

- Books** (highlighted with a red box and labeled 1): This is the active menu item.
- Copy of book
- Check out books
- Return books
- Checkout Information
- Search Book
- Patron Management
- Policy
- Book Lost Report
- Check misplaced books

Main Content Area:

The main content area shows a book titled "The Hobbit" by J.R.R. Tolkien. The book cover features a dragon breathing fire. Below the book details, there is a modal window titled "Add to wishlist".

Modal Window Fields:

- Full Name: (Input field)
- RFID/Email*: (Input field, highlighted with a red box and labeled 3)
- Email: (Input field)
- Phone: (Input field)
- Gender: (Input field)
- Status: (Input field)
- * Required field (Text)
- Cancel (Button)
- Search (Button, highlighted with a red box and labeled 4)
- Make Copy (Button)
- Add to wishlist (Button, highlighted with a red box and labeled 2)

Patron Details (Visible in Main Area):

- Full Name: Pham Minh Hoang
- Email: hoangpmse130054@fpt.edu.vn
- Phone: 0657111000
- Gender: M
- Status: Active

Bottom Footer:

- * Required field (Text)
- Cancel (Button)
- Reset (Button)
- Confirm (Button, highlighted with a red box and labeled 5)

Step	Description
1	Click on “Books” on the sidebar
2	Click on “Add to wishlist” on an unavailable book from the book list
3	Scan the patron card or enter the email with keyboard
4	Click “Search” to get the patron information
5	Click “Confirm” to finish add book to a patron’s wishlist

3.3.26. <Librarian> Search book copies

The screenshot shows the SMART LIBRARY software interface. On the left, a sidebar lists various options: Books (1), Copy of book (highlighted with a red box), Check out books, Return books, Checkout Information, Search Book, Patron Management, Policy, and Book Lost Report. The main area is titled "COPY OF BOOK". It features a search bar with an ISBN/EAN field containing "E28068940000400BB95758AE" (2), a search button (3), and a dropdown menu labeled "Select..." (4). Below the search bar, there's a checkbox followed by a thumbnail image of a book cover for "THE CLASSIC BESTSELLER MYTHOLOGY Timeless Tales of Gods and Heroes" by Edith Hamilton. To the right of the image, the book's title, author, edition, price, ISBN, barcode, and call number are displayed.

Step	Description
1	Click on “Copy of Book” on the sidebar
2	Enter book’s ISBN, barcode, title or scan book’s RFID tag
3	Click on the search button to search for copies of the book
4	Choose a book copy status to filter the result (Optional)

3.3.27. <Librarian> View book copy's details

Step	Description
1	Click on “Copy of Book” on the sidebar
2	Click on a book copy in the book copy list

3.3.28. <Librarian> Tag a book copy

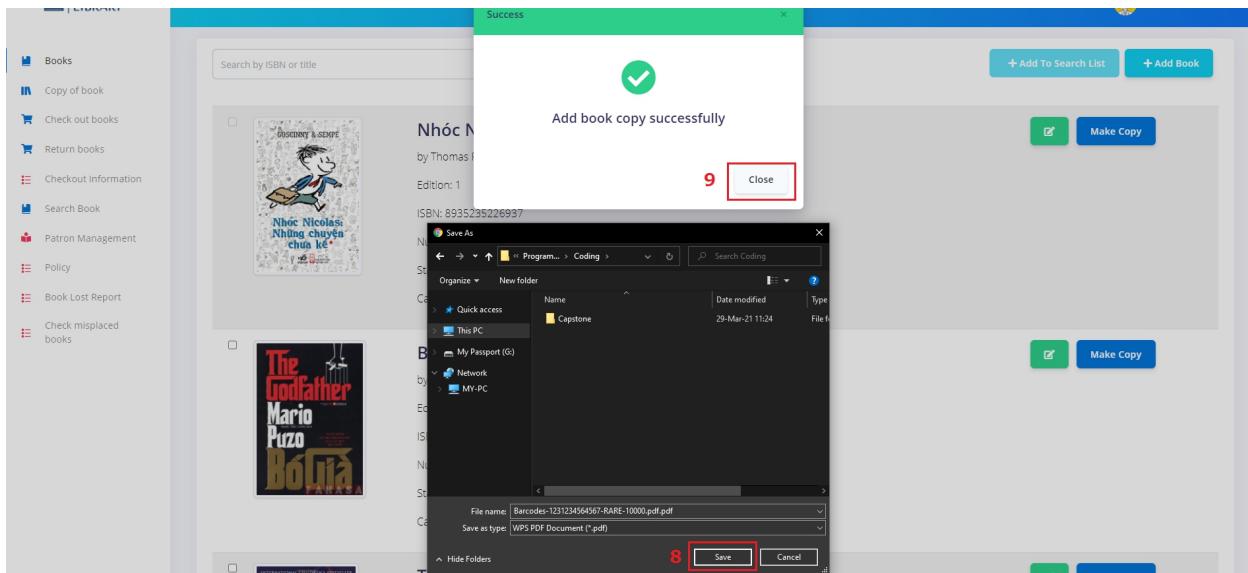
Step	Description
1	Click on “Copy of Book” on the sidebar
2	Click on “Tag RFID”

3	Enter book copy's barcode
4	Scan new RFID tag
5	Click "Confirm" to finish tagging a book copy

3.3.29. <Librarian> Add book copies

The screenshots illustrate the steps for adding book copies:

- Screenshot 1:** Shows the main menu with the "Books" option highlighted (1).
- Screenshot 2:** Shows the "Make Book Copy" dialog for "Bố Già". Fields filled: ISBN (1231234564567), Title (Bố Già), Number of copy (1), Status: In circulation, Call number (RARE) (3). Buttons: "Save" (6).
- Screenshot 3:** Shows the "Confirm Book Copy" dialog for "Bố Già". It displays the book details and two barcodes. The bottom barcode has the number 03123400000029. Buttons: "Cancel" (1), "Confirm" (7).



Step	Description
1	Click on “Books” on the sidebar
2	Click on “Make Copy”
3	Choose a book copy type
4	Enter book copy price (VND)
5	Enter total number of copies adding
6	Click “Save”
7	Click “Confirm” to save new copies to system and generate barcode file
8	Choose a location for the barcode file and click “Save”
9	Click “Close” to finish add book copies

3.3.30. <Librarian> Update book copy

The screenshot shows two consecutive steps in the SMART LIBRARY software for updating a book copy.

Step 1: The user is on the "COPY MANAGEMENT" page. On the left sidebar under "Books", "Book copies" is selected. A red box labeled "1" highlights the "Copy management" option. The main area shows a list of books, with one titled "Harmony" by Project Itoh selected. A red box labeled "2" highlights the edit button (pencil icon) next to the book's thumbnail.

Step 2: A modal window titled "Edit Copy" is open for the book "Harmony". The modal contains fields for ISBN, Barcode, Price, Copy Type, Subtitle, Author, and Edition. Red boxes numbered 3 through 6 highlight specific fields: "Price" (1200), "Copy Type" (REGULAR), "RFID Code" (987456), and the "Save" button. A red box labeled "7" highlights the "Close" button at the bottom right of the modal.

Step	Description
1	Click on “Copy management” on the sidebar
2	Click on the edit button on a copy from the copy list
3	Enter the new Price (Optional)
4	Select the new Copy Type (Optional)
5	Enter the new RFID Code (Optional)

6	Click on “Save”
7	Click “Close” to finish

3.3.31. <Librarian> View patron policy

PATRON TYPE	TOTAL CHECKOUTS ALLOWED	ACTION
STUDENT	6	
LECTURER	10	
TEST	1	

Step	Description
1	Click on “Policy setting” on the sidebar
2	Click on the patron tab
3	Select patron type filter (Optional)

3.3.32. <Librarian> View borrow policy

The screenshot shows the SMART LIBRARY POLICY page. On the left sidebar, 'Policy' is highlighted with a red box and the number 1. The 'Borrow' tab is also highlighted with a red box and the number 2. Below the tabs, there are two dropdown menus: 'Select patron type...' and 'Select copy type...', both highlighted with red boxes and numbered 3 and 4 respectively. A blue 'Add New' button is visible. The main area displays a table with six rows of borrowing policies. A blue circle with the number 1 is located in the bottom right corner of the table.

PATRON TYPE	BOOK COPY TYPE	BORROW PERIOD (DAYS)	CHECKOUTS ALLOWED (COUNT)	RENEWALS ALLOWED (TIMES)	RENEWAL PERIOD (DAYS)
STUDENT	REGULAR	7	4	2	7
STUDENT	REFERENCE	7	4	2	7
LECTURER	REGULAR	7	4	2	7
LECTURER	REFERENCE	7	4	2	7
LECTURER	RARE	7	4	2	7

Step	Description
1	Click on “Policy setting” on the sidebar
2	Click on the borrow tab
3	Select patron type filter (Optional)
4	Select copy type filter (Optional)

3.3.33. <Librarian> View fee policy

localhost:3000/librarian/policy*

Step	Description
1	Click on “Policy setting” on the sidebar
2	Click on the fee tab

3.3.34. <Librarian> View history of fee policy

localhost:3000/librarian/policy*

Step	Description
------	-------------

1	Click on “Policy setting” on the sidebar
2	Click on the fee tab
3	Click on the “History” button

3.3.35. <Librarian> View patron checkout information

Step	Description
1	Click on “Checkout information” on the sidebar
2	Enter RFID or Email
3	Click on the “Search” button to search for result

3.3.36. <Librarian> Renew borrowing book for patron

The screenshots show the 'CHECKOUT INFORMATION' page for a user named 'Pham Minh Hoang'. The sidebar on the left includes options like Books, Book copies, Check out books, Return books, and Checkout information (which is highlighted with a red box). The main area shows a list of borrowed books:

BOOK	BORROWED AT	DUE DATE	
Harmony - Edition[1]	2021-04-09, 02:46:07	2021-04-30	Renew Email
Harmony - Edition[1]	2021-04-09, 01:05:00	2021-04-30	Renew Email

In the first screenshot, a modal dialog is open for renewing the second book. It shows the 'Renew Due Date' set to '2021-05-07', a 'Back' button, and a 'Confirm' button (highlighted with a red box) which has the number '3' next to it.

In the second screenshot, a success message box is displayed: 'Renewed successfully' with a green checkmark icon. A 'Close' button (highlighted with a red box) has the number '4' next to it.

Step	Description
1	Click on “Checkout information” on the sidebar
2	Click on the “Renew” button on a borrowing from the borrowing list
3	Click on “Confirm”
4	Click “Close” to finish

3.3.37. <Librarian> View renew history of a borrowed book

The screenshot shows the SMART LIBRARY system interface. The top navigation bar includes links for Books, Book copies, Check out books, Return books, and Checkout information (which is highlighted with a red box and labeled '1'). Below the navigation is a search bar with placeholder text 'Get checkout Informations by a student's RFID or Email, e.g. 130111, example@smart.edu.vn' and a 'Search' button. The main content area displays a user profile for 'Pham Minh Hoang' with email 'hoangpmse130054@fpt.edu.vn'. Below the profile is a table titled 'Borrowing Books' showing two entries:

BOOK	BORROWED AT	DU DATE	
Harmony - Edition[1]	2021-04-09, 02:46:07	2021-04-30	<button>Renew</button> <button>History</button> <button>Details</button>
Harmony - Edition[1]	2021-04-09, 01:05:00	2021-04-30	<button>Renew</button> <button>History</button> <button>Details</button>

A blue circle with '1' is positioned at the bottom right of the table.

The modal window shows the following details for the borrowed book:

Book: Harmony - Edition [1]
Barcode: 02123400000004
Borrowed At: 2021-04-09, 01:05:00

The borrowing history table has columns: #, RENEWED AT, and DUE DATE. It contains three rows:

#	RENEWED AT	DUE DATE
0		2021-04-16
1	2021-04-09, 01:19:33	2021-04-23
2	2021-04-09, 02:29:32	2021-04-30

A blue circle with '1' is at the bottom right of the table, and a blue circle with '2' is at the bottom right of the modal window.

Step	Description
1	Click on "Checkout information" on the sidebar
2	Click on the history button on a borrowing from the borrowing list

3.3.38. <Librarian> View lost books

The screenshot shows the "BOOK LOST REPORT" section of the library management system. On the left sidebar, under the "Books" category, the "Book Lost Report" link is highlighted with a red box and labeled "1". The main area displays a calendar for March and April 2021, with specific dates highlighted in blue. A dropdown menu for "Status" is open, showing "Pending" as the selected option, which is also highlighted with a red box and labeled "2". Below the calendar, a table lists a single lost book entry:

Barcode	Title	Patron Email	Lost At	Fine-VND	Note	Action
02123400000001	Nicolas: Part 1	trampn130038@fpt.edu.vn	2021-03-29, 15:12:23	-	-	<button>Confirm</button>

A small blue circular icon with the number "1" is located in the bottom right corner of the table.

Step	Description
1	Click on “Book Lost Report” on the sidebar to show all the lost books
2	Choose the time when lost books were reported
3	Choose status

3.3.39. <Librarian> Confirm lost book

This screenshot is identical to the one above, showing the "BOOK LOST REPORT" page. The "Book Lost Report" link in the sidebar is highlighted with a red box and labeled "1". The calendar and status dropdown are also highlighted with red boxes and labeled "2" and "3" respectively. The table below shows the same lost book entry. A blue rectangular button labeled "Confirm" is overlaid on the "Action" column of the table, and it is also highlighted with a red box and labeled "4".

Confirm lost book X

ISBN: 9780552574242 Title: Nicolas Author: Cline Edition: 1 Overdue Day(s): 1 Overdue Fee: 2000 VND Lost book fine not in market: 250000 VND Lost book fine in market: 80000 VND	Lost book fine* <div style="border: 2px solid red; padding: 5px; display: inline-block;"> Lost book fine VND 5 </div> Note <div style="border: 2px solid red; padding: 5px; display: inline-block;"> Note 5 </div> <p>* Required field</p> <div style="text-align: right; margin-top: 10px;"> Cancel Confirm 6 </div>
---	---

Step	Description
1	Click on “Book Lost Report” on the sidebar to show all the lost books
2	Choose the time when lost books were reported
3	Choose Pending status
4	Click on “Confirm” to show confirm form
5	Fill in required fields
6	Click on “Confirm” to confirm the lost book and automatically send lost billing to the patron

3.3.40. <Librarian> Save books' location

CHECK MISPLACED BOOKS

tuongnt1@fpt.edu.vn

Books
Book copies ►
Check out books
Return books
Checkout information
Search book
Check misplaced books 1
Security gate logs
Lost book management ►
Group management
Librarian management
Patron management
Policy setting

Sampling - Please SELECT or SCAN position before start...

3.1 A1 | 1 | 3.2

2 Sample Check

4 Start Finish

Scanned Books

Barcode	RFID	Call Number	Title	Author	Edition
There is no data to display					

CHECK MISPLACED BOOKS

tuongnt1@fpt.edu.vn

Books
Book copies ►
Check out books
Return books
Checkout information
Search book
Check misplaced books 1
Security gate logs
Lost book management ►
Group management
Librarian management
Patron management
Policy setting

Sampling

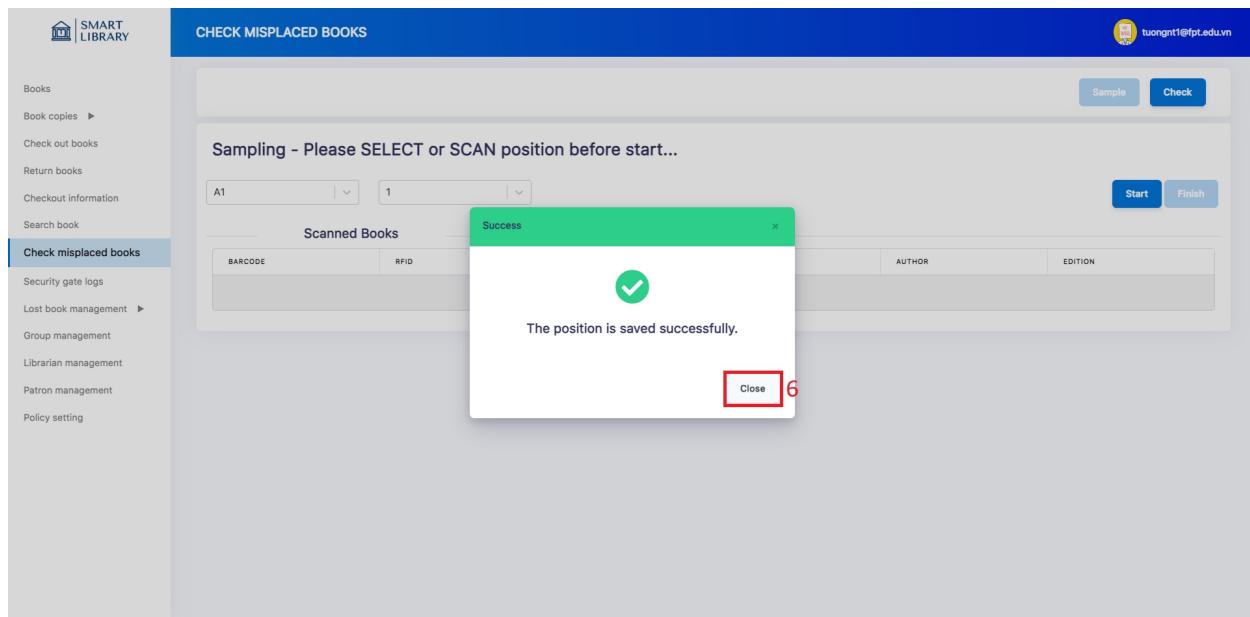
A1 | 1

5 Cancel Finish

Scanned Books

Barcode	RFID	Call Number	Title	Author	Edition
02123400000002	E28068940000400BB95768AE	900.1 VER 2005	20,000 Leagues Under the Sea:(Enrich)	Jules Verne	1
03123400000001	E28068940000400BB95758AE	200. TOL 2012	The Hobbit	J.R.R Tolkien	2

1



Step	Description
1	Click on “Check misplaced books” on the sidebar
2	Click on “Sample” option
3	Scan row rfid on bookshelf
3.1	Select a shelf (skip this step if scanning rfid tag in step 3)
3.2	Select a row(skip this step if scanning rfid tag in step 3)
4	Click “Start” button
5	Scan books then click “Finish” button
6	Click “Close” button

3.3.41. <Librarian> Find misplaced and missing books at a location

SIMPLY LIBRARY

CHECK MISPLACED BOOKS

tuongnt1@fpt.edu.vn

Books
Book copies ►
Check out books
Return books
Checkout information
Search book
Check misplaced books 1
Security gate logs
Lost book management ►
Group management
Librarian management
Patron management
Policy setting

Sample Check 2

Checking - Please SELECT or SCAN position before start...

3.1 A1 3.2 2 3.3 3.2

4 Start Finish

Scanned Books

INITIAL POSITION	BARCODE	RFID	CALL NUMBER	TITLE	AUTHOR	EDITION
There is no data to display						

CHECK MISPLACED BOOKS

tuongnt1@fpt.edu.vn

Books
Book copies ►
Check out books
Return books
Checkout information
Search book
Check misplaced books 5
Security gate logs
Lost book management ►
Group management
Librarian management
Patron management
Policy setting

Sample Check

Checking

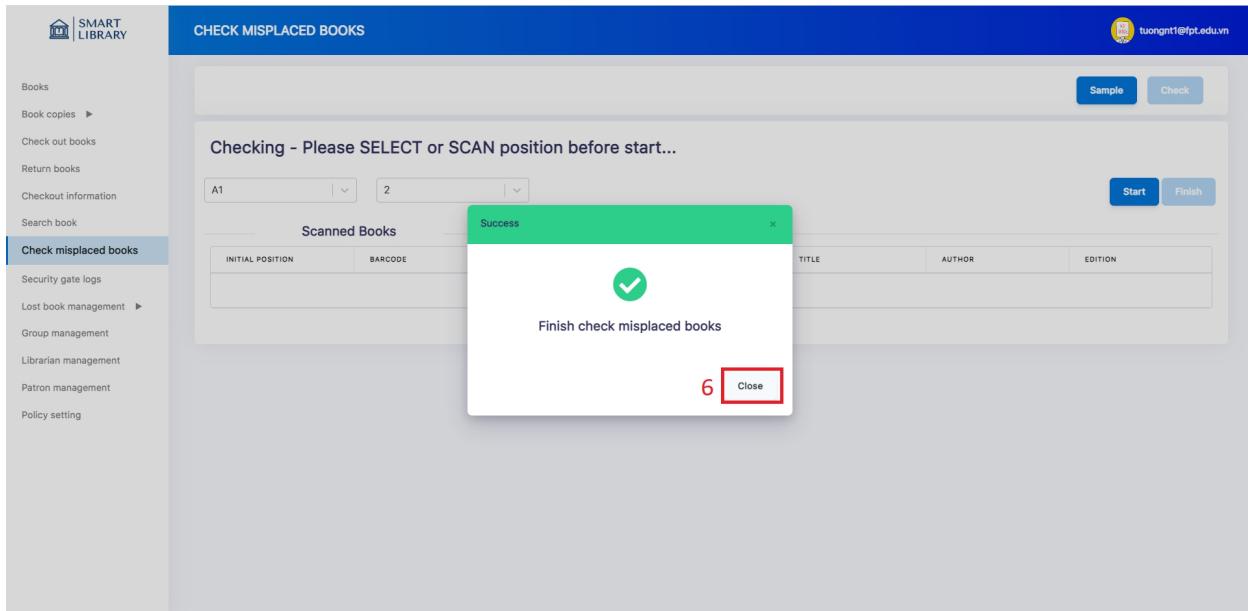
A1 2 3

Cancel Finish

Scanned Books

INITIAL POSITION	BARCODE	RFID	CALL NUMBER	TITLE	AUTHOR	EDITION
Shelf: A1 - Row: 2	01123400000005	E28068940000500BB95...	200. LIN 2012	Thần Đồng Đất Việt 36:Số 1	Lê Linh	1
Shelf: A1 - Row: 2	01123400000004	E28068940000500BB95...	200. LIN 2012	Thần Đồng Đất Việt 36:Số 1	Lê Linh	1
N/A	01123400000003	E28068940000400BB95...	200. LIN 2012	Thần Đồng Đất Việt 36:Số 1	Lê Linh	1

1



Step	Description
1	Click on “Check misplaced books” on the sidebar
2	Click on “Check” option
3	Scan row rfid on bookshelf
3.1	Select a shelf (skip this step if scanning rfid tag in step 3)
3.2	Select a row(skip this step if scanning rfid tag in step 3)
4	Click “Start” button
5	Scan books then click “Finish” button
6	Click “Close” button

3.3.42. <Librarian> Print multiple barcodes

1 Book copies ▾

2 Copy management

3 THE SUBTLE ART OF NOT GIVING A F*CK

4 Print Barcodes

5 Selected

6 OK

The screenshot shows the SMART LIBRARY system's Copy Management interface. On the left sidebar, under the "Book copies" section, the "Copy management" option is selected. The main area displays a list of book copies found, with one item selected. A modal window is overlaid, showing the book's cover ("THE SUBTLE ART OF NOT GIVING A F*CK" by Ernest J.R.R. Tolkien) and its details: Edition: 1, Price: 350, ISBN: 0439708184, Barcode: 0112340000003, Call Number: 123.ABC, and Status: AVAIL. At the bottom of the modal, a green success message says "Print barcode successfully" with a checkmark icon. A red box highlights the "Close" button at the bottom right of the modal. The background shows the list of books, with the selected book's details partially visible.

Step	Description
1	Click on “Book copies” on the sidebar
2	Click on “Copy management” option
3	Select barcodes that need to be printed on book copy list
4	Click on “Print barcodes” button
5	Choose “Selected” option
6	Click “OK” button
7	Click “Close” button

3.3.43. <Librarian> Search patron

The screenshot shows the 'PATRON MANAGEMENT' page. On the left sidebar, 'Patron management' is highlighted with a red box and labeled '1'. In the center, there is a search bar with 'Search by email' and a magnifying glass icon, both outlined in red and labeled '2'. Below the search bar is a table with two rows of patron data. The first row for 'Phan Hoang Tram' has a 'Disable' button outlined in red and labeled '3'. The second row for 'Pham Minh Hoang' has a 'Disable' button.

ID	EMAIL	NAME	GENDER	PHONE	TYPE	STATUS	ACTION
6	trampmse130038@fpt.edu.vn	Phan Hoang Tram	Female	0965457000	STUDENT	ACTIVE	<input type="button" value="Disable"/>
7	hoangpmse130054@fpt.edu.vn	Pham Minh Hoang	Male	0657111000	STUDENT	ACTIVE	<input type="button" value="Disable"/>

Step	Description
1	Click on “Patron management” on the sidebar
2	Type patron’s email to search
3	Click “Search” button

3.3.44. <Librarian> Activate patron

The screenshot shows the 'PATRON MANAGEMENT' page. On the left sidebar, 'Patron management' is highlighted with a red box and labeled '1'. In the center, there is a search bar with 'Search by email' and a magnifying glass icon. Below the search bar is a table with two rows of patron data. The first row for 'Phan Hoang Tram' has an 'Enable' button outlined in red and labeled '2'. The second row for 'Pham Minh Hoang' has a 'Disable' button.

ID	EMAIL	NAME	GENDER	PHONE	TYPE	STATUS	ACTION
6	trampmse130038@fpt.edu.vn	Phan Hoang Tram	Female	0965457000	STUDENT	DISABLE	<input checked="" type="button" value="Enable"/>
7	hoangpmse130054@fpt.edu.vn	Pham Minh Hoang	Male	0657111000	STUDENT	ACTIVE	<input type="button" value="Disable"/>

PATRON MANAGEMENT

tuongnt@fpt.edu.vn

Search by email

ID	EMAIL	NAME	GENDER	PHONE	TYPE	STATUS	ACTION
6	trampmse130038@fpt.edu.vn	Phan Hoang Tram	Female	0965457000	STUDENT	DISABLE	<input checked="" type="button"/> Enable
7	hoangpmse130054@fpt.edu.vn				STUDENT	ACTIVE	<input checked="" type="button"/> Disable

Activate Patron

Are you sure?
Do you want to activate this patron?

Close 3

PATRON MANAGEMENT

tuongnt@fpt.edu.vn

Search by email

ID	EMAIL	NAME	GENDER	PHONE	TYPE	STATUS	ACTION
6	trampmse130038@fpt.edu.vn	Phan Hoang Tram	Female	0965457000	STUDENT	ACTIVE	<input checked="" type="button"/> Disable
7	hoangpmse130054@fpt.edu.vn				STUDENT	ACTIVE	<input checked="" type="button"/> Disable

Success

Change patron status successfully

Close 4

Step	Description
1	Click on “Patron management” on the sidebar
2	Click “Enable” button on patron in patron list
3	Click “OK” button
4	Click “Close” button

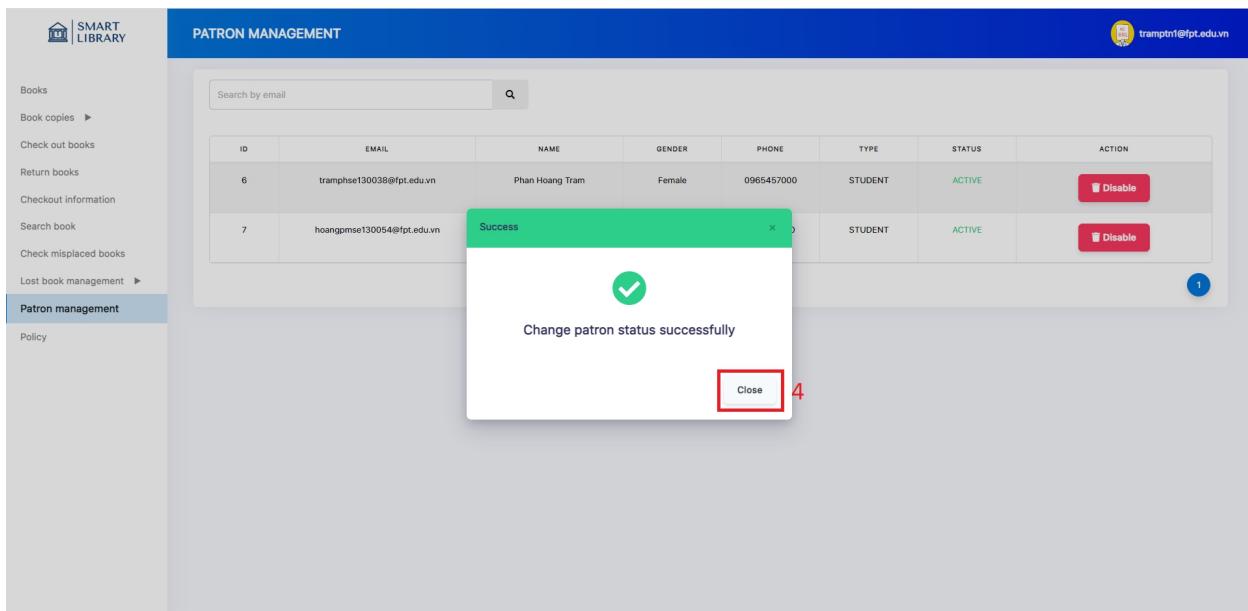
3.3.45. <Librarian> Deactivate

The screenshot shows the 'PATRON MANAGEMENT' page. On the left sidebar, under 'Patron management', the 'Disable' button for patron ID 6 is highlighted with a red box and labeled '2'. The 'OK' button in a confirmation dialog is also highlighted with a red box and labeled '3'.

ID	EMAIL	NAME	GENDER	PHONE	TYPE	STATUS	ACTION
6	trampnse130058@fpt.edu.vn	Phan Hoang Tram	Female	0965457000	STUDENT	ACTIVE	<input checked="" type="checkbox"/> Disable
7	hoangpmse130054@fpt.edu.vn	Pham Minh Hoang	Male	0657111000	STUDENT	ACTIVE	<input checked="" type="checkbox"/> Disable

The screenshot shows the 'PATRON MANAGEMENT' page with a confirmation dialog titled 'Are you sure?'. The 'OK' button is highlighted with a red box and labeled '3'.

ID	EMAIL	NAME	TYPE	STATUS	ACTION
6	trampnse130058@fpt.edu.vn	Phan Hoang Tram	STUDENT	ACTIVE	<input checked="" type="checkbox"/> Disable
7	hoangpmse130054@fpt.edu.vn	Pham Minh Hoang	STUDENT	ACTIVE	<input checked="" type="checkbox"/> Disable



Step	Description
1	Click on “Patron management” on the sidebar
2	Click “Disable” button on patron in patron list
3	Click “OK” button
4	Click “Close” button

3.3.46. <Manager> Add borrow

This screenshot shows the 'POLICY SETTING' page under the 'Borrow' tab. The left sidebar includes links for Books, Book copies, Check out books, Return books, Checkout information, Search book, Check misplaced books, Security gate logs, Lost book management, Group management, Librarian management, Patron management, and Policy setting (which is highlighted with a red box). The main area displays a table of borrowing policies with columns for Patron Type, Book Copy Type, Borrow Period (Days), Checkouts Allowed (Count), Renewals Allowed (Times), Renewal Period (Days), and Action. A red box highlights the '+ Add New' button in the top right corner of the table header.

This screenshot shows the 'POLICY SETTING' page with the 'Add new' dialog box open. The dialog has fields for Patron Type*, Copy Type*, Borrow Period (Days)*, Checkouts Allowed (Count)*, Renewals Allowed (Times)*, and Renewal Period (Days)*. Each field is highlighted with a red box. At the bottom of the dialog, there are 'Cancel' and 'Confirm' buttons, with 'Confirm' highlighted with a red box. The background shows the same table of policies as the previous screenshot, with a red box around the '1' icon in the bottom right corner of the table.

PATRON TYPE	BOOK COPY TYPE	DUE DURATION (DAYS)	MAX BORROW NUMBER	MAX RENEW TIME	RENEW DUE DURATION (DAYS)	ACTION
STUDENT	REGULAR	7	4	2	7	
STUDENT	REFERENCE					
LECTURER	REGULAR			2	7	
LECTURER	REFERENCE			2	7	
LECTURER	RARE			2	7	

Step	Description
1	Click on “Policy setting” on the sidebar
2	Click on “Borrow” tab
3	Click “Add New” button
4	Select Patron type
5	Select Book copy type
6	Fill in Due Duration
7	Fill in Max borrow number
8	Fill in Max renew time
9	Fill in Renew Due Duration
10	Click “Confirm” button
11	Click “Close” button

3.3.47. <Manager> Update borrow

POLICY SETTING

Borrow Patron Fee

Select patron type... Select copy type... + Add New

PATRON TYPE	BOOK COPY TYPE	BORROW PERIOD (DAYS)	CHECKOUTS ALLOWED (COU...)	RENEWALS ALLOWED (TIMES)	RENEWAL PERIOD (DAYS)	ACTION
STUDENT	REGULAR	3	4	5	6	7
STUDENT	REFERENCE	7	4	2	7	
LECTURER	REGULAR	7	4	2	7	
LECTURER	REFERENCE	7	4	2	7	
LECTURER	RARE	7	4	2	7	

1

policy

POLICY SETTING

Borrow Patron Fee

Select patron type... Select copy type... + Add New

PATRON TYPE	BOOK COPY TYPE	BORROW PERIOD (DAYS)	CHECKOUTS ALLOWED (COU...)	RENEWALS ALLOWED (TIMES)	RENEWAL PERIOD (DAYS)	ACTION
STUDENT	REGULAR	7	4	2	7	
STUDENT	REFERENCE	7	4	2	7	
LECTURER	REGULAR	7	4	2	7	
LECTURER	REFERENCE	7	4	2	7	
LECTURER	RARE	7	4	2	7	

8

Close OK

Step	Description
1	Click on “Policy setting” on the sidebar
2	Click on “Borrow” tab
3	Update “Due Duration” field
4	Update “Max borrow number” field
5	Update “Max renew time” field
6	Update “Renew Due Duration” field
7	Click “Update” button on borrow policy in borrow policy list
8	Click “OK” button
9	Click “Close” button

3.3.48. <Manager> Remove borrow policy

POLICY SETTING

Borrow Patron Fee

PATRON TYPE	BOOK COPY TYPE	BORROW PERIOD (DAYS)	CHECKOUTS ALLOWED (CO...)	RENEWALS ALLOWED (TIMES)	RENEWAL PERIOD (DAYS)	ACTION
STUDENT	REGULAR	7	4	2	7	3
STUDENT	REFERENCE	7	4	2	7	
LECTURER	REGULAR	7	4	2	7	
LECTURER	REFERENCE	7	4	2	7	
LECTURER	RARE	7	4	2	7	

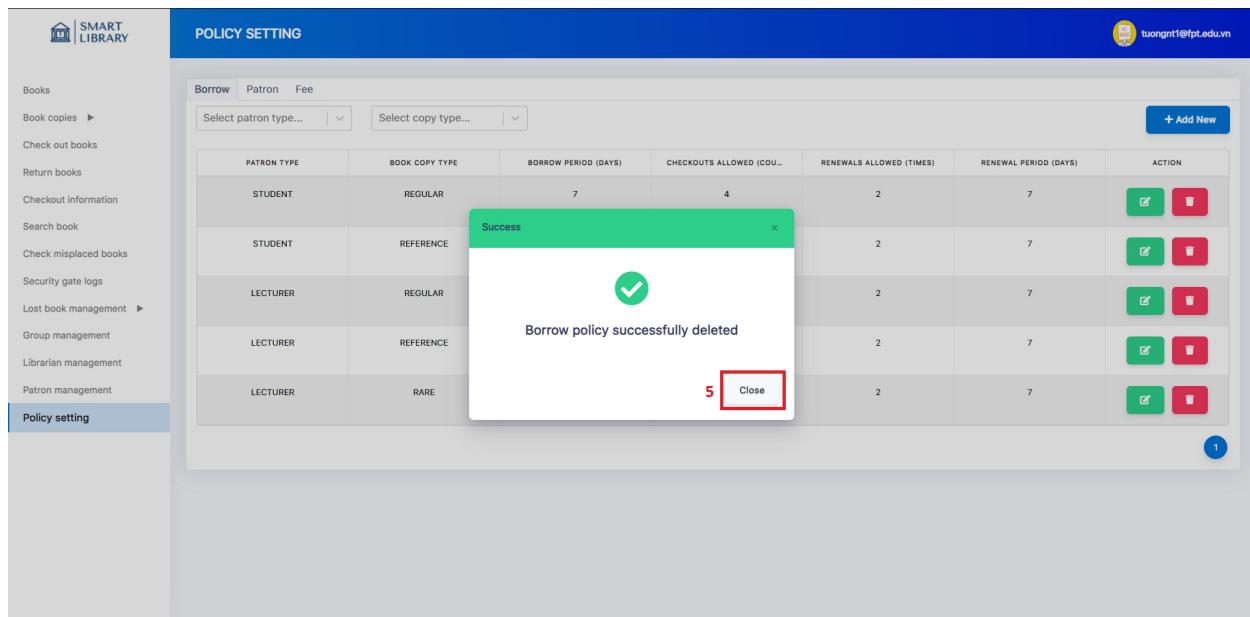
1 Policy setting

POLICY SETTING

Borrow Patron Fee

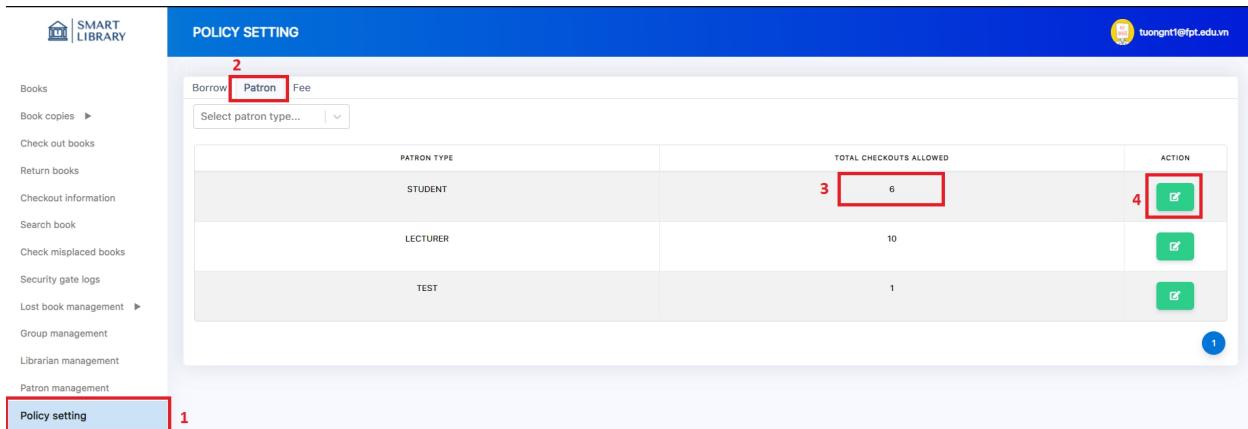
PATRON TYPE	BOOK COPY TYPE	BORROW PERIOD (DAYS)	CHECKOUTS ALLOWED (CO...)	RENEWALS ALLOWED (TIMES)	RENEWAL PERIOD (DAYS)	ACTION
STUDENT	REGULAR	7	4	2	7	
STUDENT	REFERENCE	7	4	2	7	
LECTURER	REGULAR	7	4	2	7	
LECTURER	REFERENCE	7	4	2	7	
LECTURER	RARE	7	4	2	7	

1 Policy setting



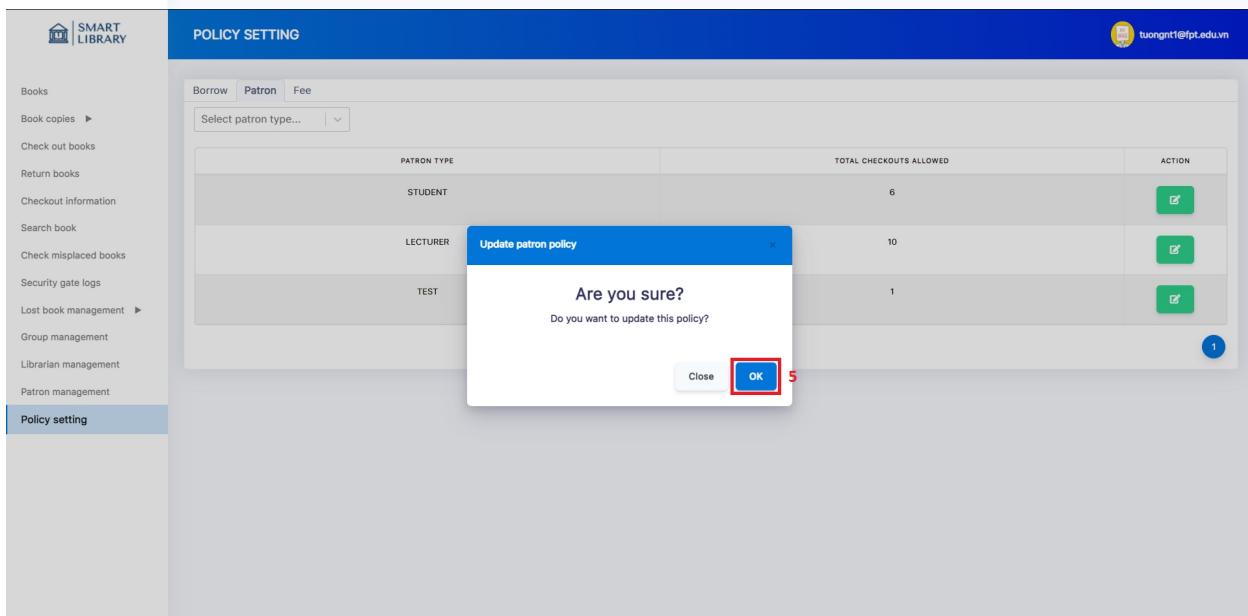
Step	Description
1	Click on “Policy setting” on the sidebar
2	Click on “Borrow” tab
3	Click “Delete” button on borrow policy in borrow policy list
4	Click “OK” button
5	Click “Close” button

3.3.49. <Manager> Update patron policy

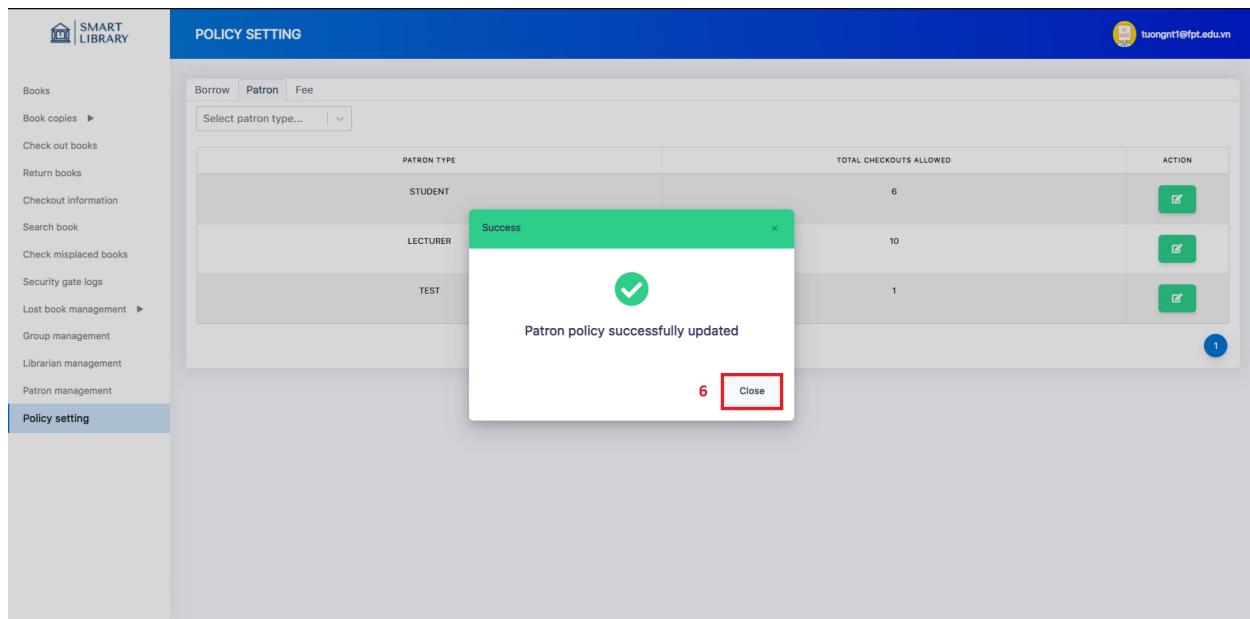


The screenshot shows the 'POLICY SETTING' page with the 'Patron' tab selected (highlighted with a red box). The main content area displays a table of patron types and their allowed checkouts. A modal window at the bottom right contains a confirmation message: 'Are you sure? Do you want to update this policy?' with 'OK' and 'Close' buttons.

PATRON TYPE	TOTAL CHECKOUTS ALLOWED	ACTION
STUDENT	6	
LECTURER	10	
TEST	1	

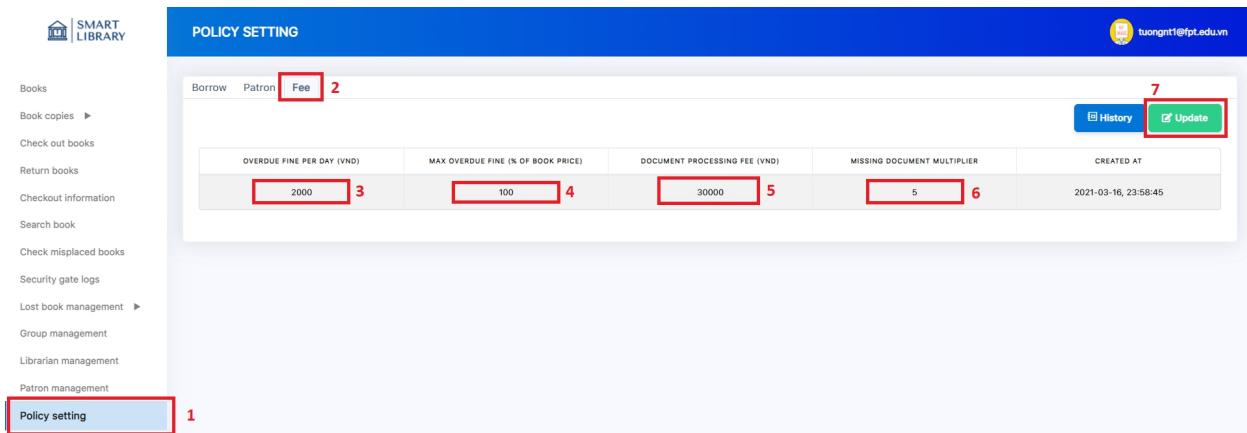


The screenshot shows the 'POLICY SETTING' page with the 'Patron' tab selected. A confirmation dialog box is open in the center, asking 'Are you sure? Do you want to update this policy?' with 'OK' and 'Close' buttons. The 'OK' button is highlighted with a red box.



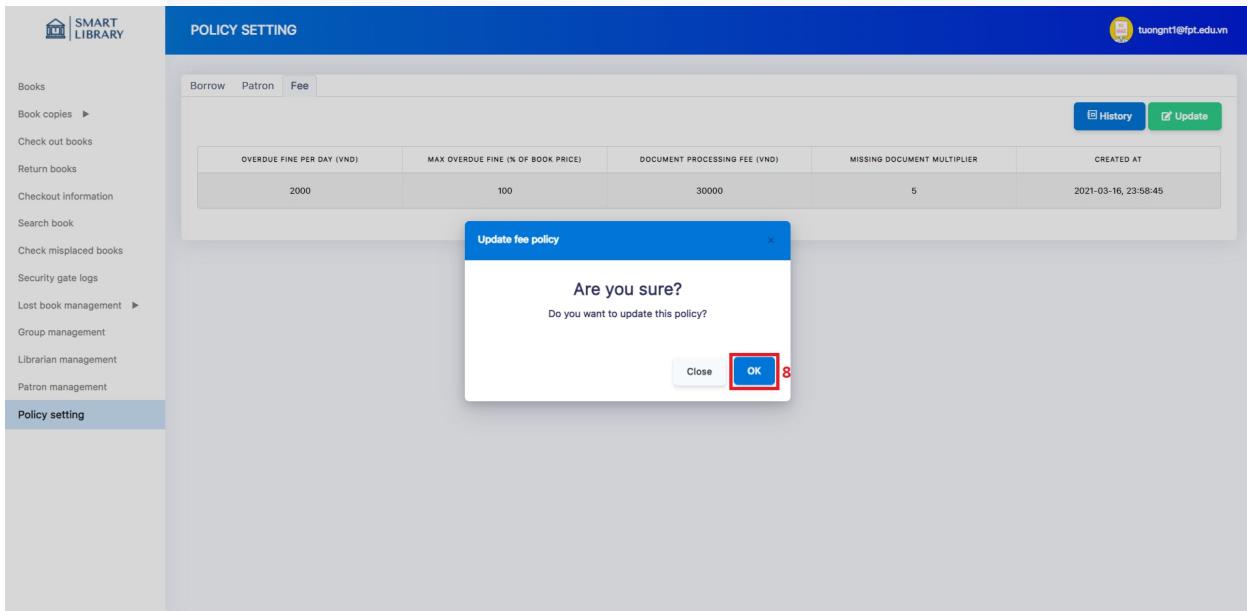
Step	Description
1	Click on “Policy setting” on the sidebar
2	Click on “Patron” tab
3	Update “Max Borrow Number” field
4	Click “Update” button on patron policy in borrow policy list
5	Click “OK” button
6	Click “Close” button

3.3.50. <Manager> Update fee policy

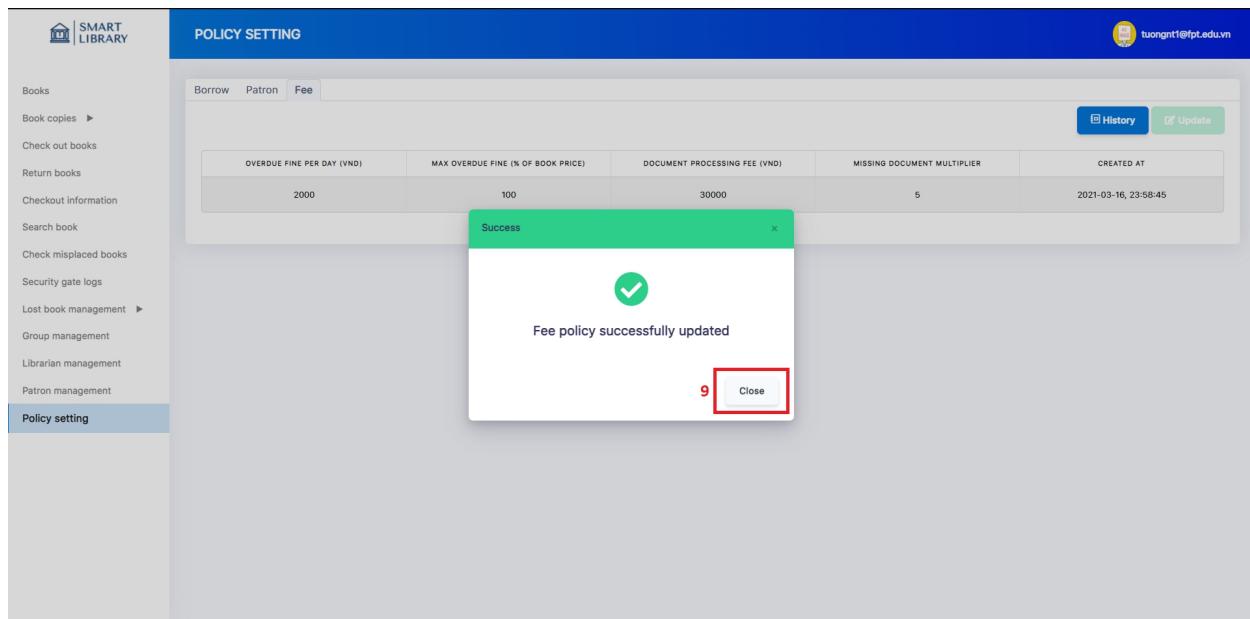


The screenshot shows the 'POLICY SETTING' page with the 'Fee' tab selected. On the left sidebar, 'Policy setting' is highlighted with a red box and the number 1. The main content area displays a table with five columns: OVERDUE FINE PER DAY (VND), MAX OVERDUE FINE (% OF BOOK PRICE), DOCUMENT PROCESSING FEE (VND), MISSING DOCUMENT MULTIPLIER, and CREATED AT. The values are 2000, 100, 30000, 5, and 2021-03-16, 23:58:45 respectively. A red box surrounds the 'Fee' tab in the navigation bar, and another red box surrounds the 'Update' button in the top right corner. The numbers 2, 3, 4, 5, 6, 7, and 8 are placed near their respective elements.

OVERDUE FINE PER DAY (VND)	MAX OVERDUE FINE (% OF BOOK PRICE)	DOCUMENT PROCESSING FEE (VND)	MISSING DOCUMENT MULTIPLIER	CREATED AT
2000	100	30000	5	2021-03-16, 23:58:45



The screenshot shows the 'POLICY SETTING' page with the 'Fee' tab selected. The 'Policy setting' item in the sidebar is highlighted with a red box and the number 1. A modal dialog titled 'Update fee policy' is open in the center, asking 'Are you sure?' and 'Do you want to update this policy?'. The 'OK' button is highlighted with a red box and the number 8. The 'Close' button is also visible. The numbers 2 through 7 are present in the top right corner of the main content area.



Step	Description
1	Click on “Policy setting” on the sidebar
2	Click on “Fee” tab
3	Update “Overdue Fine Per Day” field
4	Update “Max Percentage Overdue Fine” field
5	Update “Document Processing Fee” field
6	Update “Missing Document Multiplier” field
7	Click “Update” button
8	Click “OK” button
9	Click “Close” button

3.3.51. <Manager> Search librarian

The screenshot shows the 'LIBRARIAN MANAGEMENT' page. On the left sidebar, 'Librarian management' is highlighted with a red box and labeled '1'. A search bar at the top has 'Search by email' and a magnifying glass icon, with '2' next to it. Below the search bar is a table with columns: ID, EMAIL, NAME, GENDER, PHONE, STATUS, and ACTION. The table contains five rows of data. The fifth row, corresponding to the search result, has a status of 'DISABLED' and an 'Enable' button highlighted with a red box and labeled '3'.

ID	EMAIL	NAME	GENDER	PHONE	STATUS	ACTION
2	trampton1@fpt.edu.vn	Pham Thi Tram	Female	0965457361	ACTIVE	
3	thuypt2@fpt.edu.vn	Pham Thi Thuy	Female	0657111491	ACTIVE	
4	huynm3@fpt.edu.vn	Nguyen Minh Huy	Male	0948634582	ACTIVE	
5	oanhph4@fpt.edu.vn	Phan Hoang Oanh	Female	0916741852	DISABLE	3

Step	Description
1	Click on “Librarian management” on the sidebar
2	Type librarian’s email to search
3	Click “Search” button

3.3.52. <Manager> Activate librarian’s account

The screenshot shows the 'LIBRARIAN MANAGEMENT' page. The sidebar 'Librarian management' is highlighted with a red box and labeled '1'. The table below shows a row for user ID 5 with a status of 'DISABLED'. The 'Enable' button in the 'ACTION' column is highlighted with a red box and labeled '2'.

ID	EMAIL	NAME	GENDER	PHONE	STATUS	ACTION
2	trampton1@fpt.edu.vn	Pham Thi Tram	Female	0965457361	ACTIVE	
3	thuypt2@fpt.edu.vn	Pham Thi Thuy	Female	0657111491	ACTIVE	
4	huynm3@fpt.edu.vn	Nguyen Minh Huy	Male	0948634582	ACTIVE	
5	oanhph4@fpt.edu.vn	Phan Hoang Oanh	Female	0916741852	DISABLE	2

The screenshot shows the Librarian Management page with a modal dialog titled "Activate Patron". The dialog asks "Are you sure?" and "Do you want to activate this librarian?". It contains two buttons: "Close" and "OK". The "OK" button is highlighted with a red box.

ID	EMAIL	NAME	GENDER	PHONE	STATUS	ACTION
2	trampmt1@fpt.edu.vn	Pham Thi Tram	Female	0965457361	ACTIVE	
3	thuypt2@fpt.edu.vn			1491	ACTIVE	
4	huynm3@fpt.edu.vn			4582	ACTIVE	
5	oanhph4@fpt.edu.vn			1852	DISABLE	

The screenshot shows the Librarian Management page with a modal dialog titled "Success" containing a green checkmark icon and the message "Change librarian status successfully". It has a "Close" button. The "Close" button is highlighted with a red box.

ID	EMAIL	NAME	GENDER	PHONE	STATUS	ACTION
2	trampmt1@fpt.edu.vn	Pham Thi Tram	Female	0965457361	ACTIVE	
3	thuypt2@fpt.edu.vn			1491	ACTIVE	
4	huynm3@fpt.edu.vn			4582	ACTIVE	
5	oanhph4@fpt.edu.vn			1852	DISABLE	

Step	Description
1	Click on “Librarian management” on the sidebar
2	Click “Enable” button on librarian in librarian list
3	Click “OK” button
4	Click “Close” button

3.3.53. <Manager> Deactivate librarian's account

LIBRARIAN MANAGEMENT

ID	EMAIL	NAME	GENDER	PHONE	STATUS	ACTION
2	tramptn1@fpt.edu.vn	Pham Thi Tram	Female	0965457361	ACTIVE	
3	thuypt2@fpt.edu.vn	Pham Thi Thuy	Female	0657111491	ACTIVE	
4	huynm3@fpt.edu.vn	Nguyen Minh Huy	Male	0948634582	ACTIVE	
5	canhph4@fpt.edu.vn	Phan Hoang Oanh	Female	0916741852	ACTIVE	2

account

LIBRARIAN MANAGEMENT

ID	EMAIL	NAME	GENDER	PHONE	STATUS	ACTION
2	tramptn1@fpt.edu.vn	Pham Thi Tram	Female	0965457361	ACTIVE	
3	thuypt2@fpt.edu.vn	Pham Thi Thuy	Female	0657111491	ACTIVE	
4	huynm3@fpt.edu.vn	Nguyen Minh Huy	Male	0948634582	ACTIVE	
5	canhph4@fpt.edu.vn	Phan Hoang Oanh	Female	0916741852	ACTIVE	

The screenshot shows the 'LIBRARIAN MANAGEMENT' page. On the left sidebar, 'Librarian management' is selected. The main area displays a table of librarians with columns: ID, EMAIL, NAME, GENDER, PHONE, STATUS, and ACTION. A modal window titled 'Success' is open in the center, stating 'Change librarian status successfully'. At the bottom right of the modal is a red-bordered 'Close' button. The status of the fifth librarian in the list has been changed from ACTIVE to DISABLED.

ID	EMAIL	NAME	GENDER	PHONE	STATUS	ACTION
2	tramptm1@fpt.edu.vn	Pham Thi Tram	Female	0965457361	ACTIVE	
3	thuypt2@fpt.edu.vn	Pham Thi Thuy	Female	0657111491	ACTIVE	
4	huynm3@fpt.edu.vn	Nguyen Minh Huy	Male	0948634582	ACTIVE	
5	oanhph4@fpt.edu.vn	Phan Hoang Oanh	Female	0916741852	DISABLE	

Step	Description
1	Click on “Librarian management” on the sidebar
2	Click “Disable” button on librarian in librarian list
3	Click “OK” button
4	Click “Close” button

3.3.54. <Manager> Add librarian's account

The screenshot shows the 'LIBRARIAN MANAGEMENT' page. On the left sidebar, 'Librarian management' is selected. The main area displays a table of librarians with columns: ID, EMAIL, NAME, GENDER, PHONE, STATUS, and ACTION. A red box labeled '1' highlights the 'Librarian management' link in the sidebar. A red box labeled '2' highlights the '+ Add Librarian' button in the top right corner of the main area. The status of the fifth librarian in the list has been changed from DISABLE to ACTIVE.

ID	EMAIL	NAME	GENDER	PHONE	STATUS	ACTION
2	tramptm1@fpt.edu.vn	Pham Thi Tram	Female	0965457361	ACTIVE	
3	thuypt2@fpt.edu.vn	Pham Thi Thuy	Female	0657111491	ACTIVE	
4	huynm3@fpt.edu.vn	Nguyen Minh Huy	Male	0948634582	ACTIVE	
5	oanhph4@fpt.edu.vn	Phan Hoang Oanh	Female	0916741852	ACTIVE	

LIBRARIAN MANAGEMENT

Add Librarian

RFID Number* 741258

Name* Nguyen Van An

Email* annv1@fpt.edu.vn

Phone* 0912685563

Gender* Male

ACTION

Disable

Disable

Disable

Enable

Disable

LIBRARIAN MANAGEMENT

+ Add Librarian

ID 2

NAME Pham Thi Tram

GENDER Female

PHONE 0965457361

STATUS ACTIVE

ACTION

Disable

Disable

Disable

Enable

Success

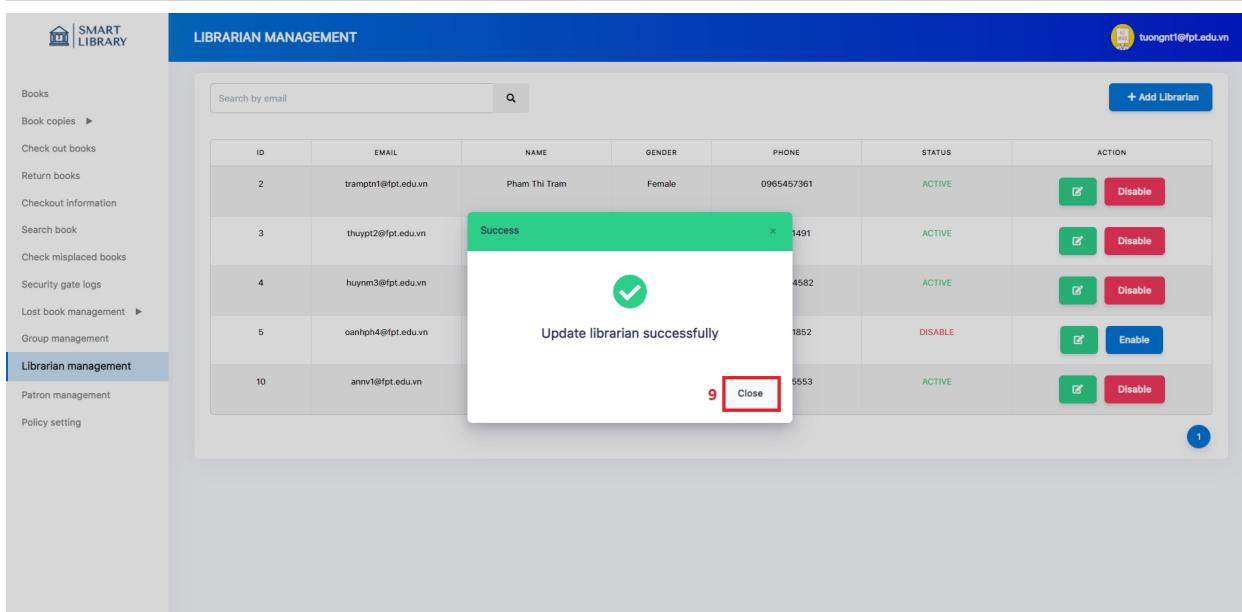
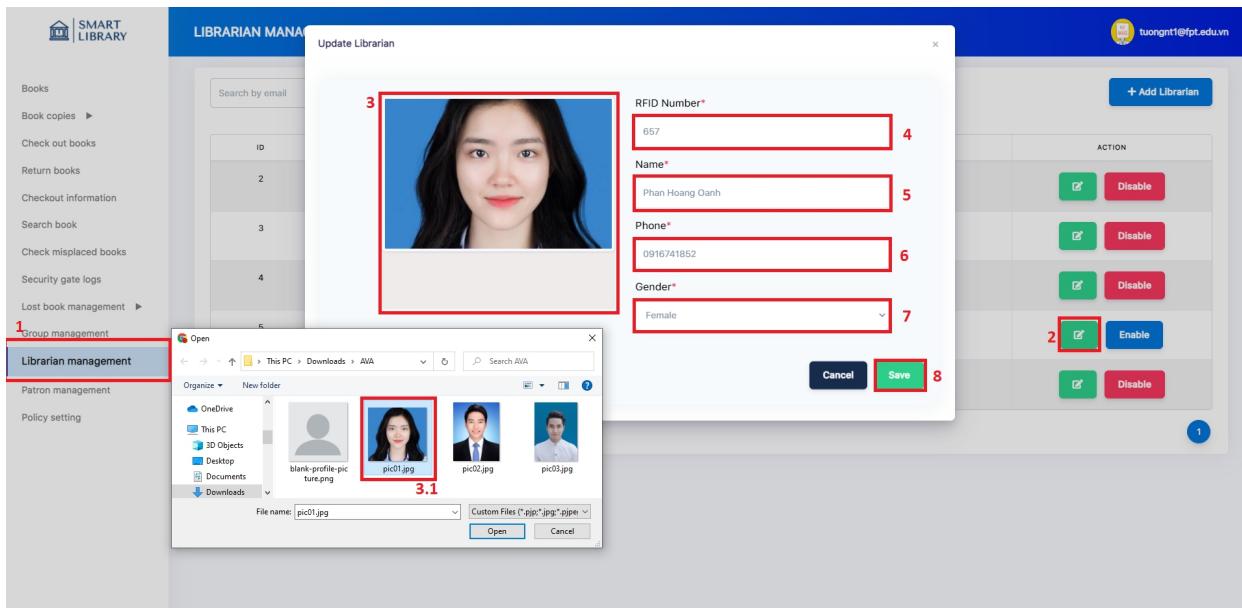
Add librarian successfully

CLOSE

Step	Description
1	Click on “Librarian management” on the sidebar
2	Click on “Add Librarian” button
3	Click to open file chooser
3.1	Pick an image from the computer
4	Fill in “RFID Number” field
5	Fill in “Name” field

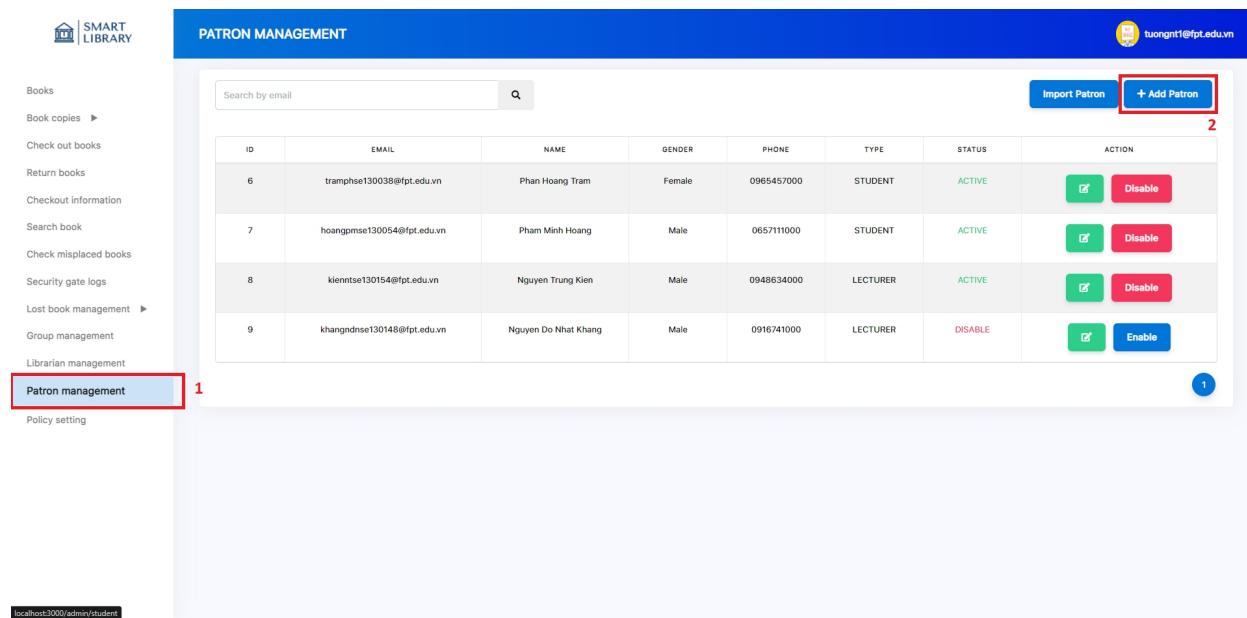
6	Fill in “Email” field
7	Fill in “Phone” field
8	Select the “Gender” field
9	Click on “Confirm” button
10	Click “Close” to finish add new librarian

3.3.55. <Manager> Update librarian's account



Step	Description
1	Click on “Librarian management” on the sidebar
2	Click on the edit button on a librarian from the librarian list
3	Click to open file chooser (Optional)
3.1	Pick a new image from the computer
4	Enter the new RFID Number (Optional)
5	Enter the new Name (Optional)
6	Enter the new Phone (Optional)
7	Enter the new Gender (Optional)
8	Click on “Save”
9	Click “Close” to finish

3.3.56. <Manager> Add patron's account



The screenshot shows the 'PATRON MANAGEMENT' page. On the left sidebar, under the 'Books' section, the 'Patron management' link is highlighted with a red box and labeled '1'. At the top right, there is a user icon and the email 'tuongnt1@fpt.edu.vn'. Below the header, there is a search bar with placeholder 'Search by email' and a magnifying glass icon. To the right of the search bar are two buttons: 'Import Patron' and '+ Add Patron', with the latter also having a red box around it and labeled '2'. The main area displays a table of patron data:

ID	EMAIL	NAME	GENDER	PHONE	TYPE	STATUS	ACTION
6	trampmse130038@fpt.edu.vn	Phan Hoang Tram	Female	0965457000	STUDENT	ACTIVE	
7	hoangpmse130054@fpt.edu.vn	Pham Minh Hoang	Male	0657111000	STUDENT	ACTIVE	
8	kienntse130154@fpt.edu.vn	Nguyen Trung Kien	Male	0948634000	LECTURER	ACTIVE	
9	khangndse130148@fpt.edu.vn	Nguyen Do Nhat Khang	Male	0916741000	LECTURER	DISABLE	

At the bottom left of the page, the URL 'localhost:3000/admin/student' is visible.

PATRON MANAGE

Add Patron

RFID Number*

852147

Name*

Nguyen Van Binh

Email*

binhnvse111111@fpt.edu.vn

Phone*

0965369361

Gender*

Male

Patron Type*

STUDENT

Cancel Save

Open

Organize New folder

OneDrive This PC 3D Objects Desktop Documents Downloads

blank-profile-picture.png pic01.jpg pic02.jpg pic03.jpg

File name: pic03.jpg Custom Files (*.jpg;*.png;*.jpeg)

Open Cancel

3 3.1 4 5 6 7 8 9 10

PATRON MANAGEMENT

Search by email

Import Patron + Add Patron

ID	EMAIL	NAME	GENDER	PHONE	TYPE	STATUS	ACTION
6	trampmse130038@fpt.edu.vn	Phan Hoang Tram	Female	0965457000	STUDENT	ACTIVE	
7	hoangpmse130054@fpt.edu.vn				STUDENT	ACTIVE	
8	kienntmse130154@fpt.edu.vn				LECTURER	ACTIVE	
9	khangndmse130148@fpt.edu.vn				LECTURER	DISABLE	

Success

Add patron successfully

Close 11

tuongnt1@fpt.edu.vn

Step	Description
1	Click on “Patron management” on the sidebar
2	Click on “Add Patron” button
3	Click to open file chooser
3.1	Pick an image from the computer
4	Fill in “RFID Number” field
5	Fill in “Name” field

6	Fill in “Email” field
7	Fill in “Phone” field
8	Select the “Gender” field
9	Select the “Patron Type” field
10	Click on “Confirm” button
11	Click “Close” to finish add new patron

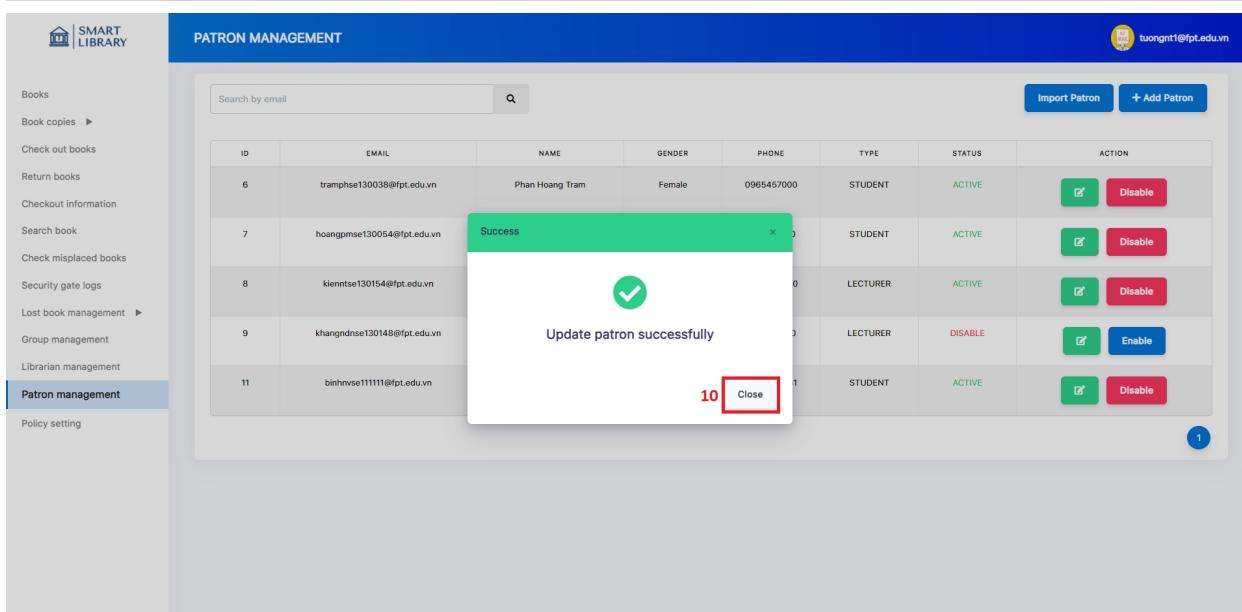
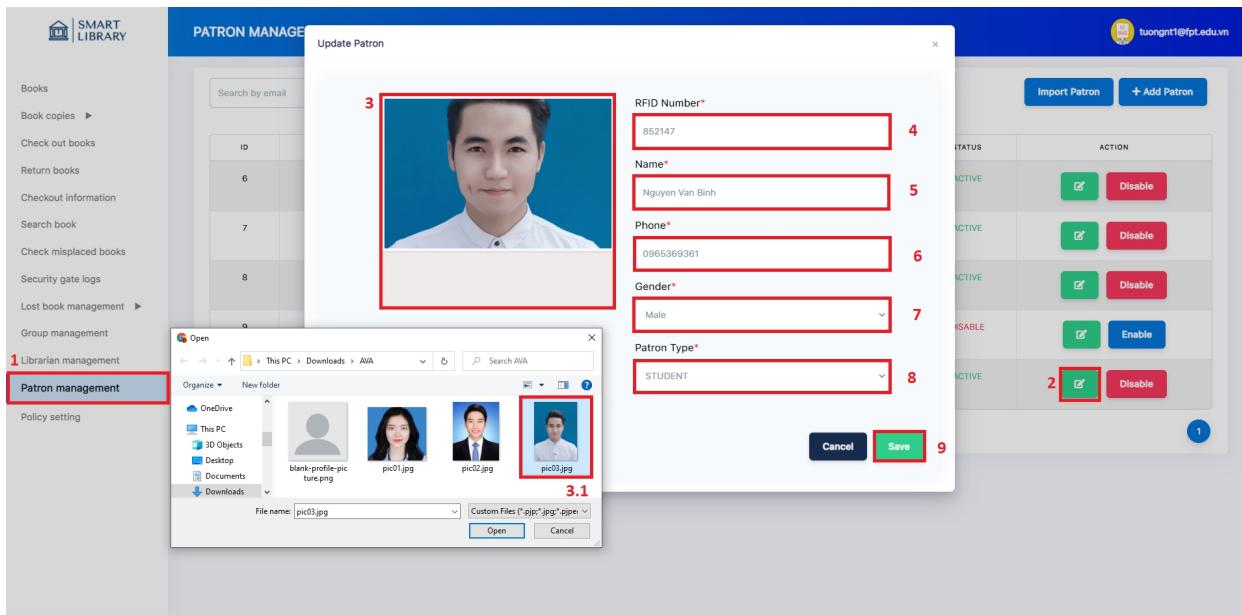
3.3.57. <Manager> Import patrons' account

The screenshot shows the SMART LIBRARY application's Patron Management interface. On the left sidebar, under 'Librarian management', 'Patron management' is selected (Step 1). At the top right, the email 'tuongnt1@fpt.edu.vn' is displayed. In the center, there is a modal window titled 'Import Patron'. Inside the modal, 'Patron Type' is set to 'STUDENT' (Step 3), and the 'Import file' section has a 'Choose File' button with a red box around it (Step 4). Below the file input is a 'Save' button with a red box around it (Step 5). In the background, the main table lists patrons with columns for ID, EMAIL, NAME, GENDER, PHONE, TYPE, STATUS, and ACTION. One row is highlighted with a green success message: 'Import patron successfully' with a checkmark icon, and a 'Close' button with a red box around it (Step 6).

Step	Description
1	Click on “Patron management” on the sidebar
2	Click on “Import Patron” button
3	Select “Patron type”
4	Click to open file chooser

4.1	Pick a file from the computer
5	Click on “Save”
6	Click “Close” to finish

3.3.58. <Manager> Update patron's account



Step	Description
------	-------------

1	Click on “Patron management” on the sidebar
2	Click on the edit button on a patron from the patron list
3	Click to open file chooser (Optional)
3.1	Pick a new image from the computer
4	Enter the new RFID Number (Optional)
5	Enter the new Name (Optional)
6	Enter the new Phone (Optional)
7	Enter the new Gender (Optional)
8	Enter the new Patron Type (Optional)
9	Click on “Save”
10	Click “Close” to finish

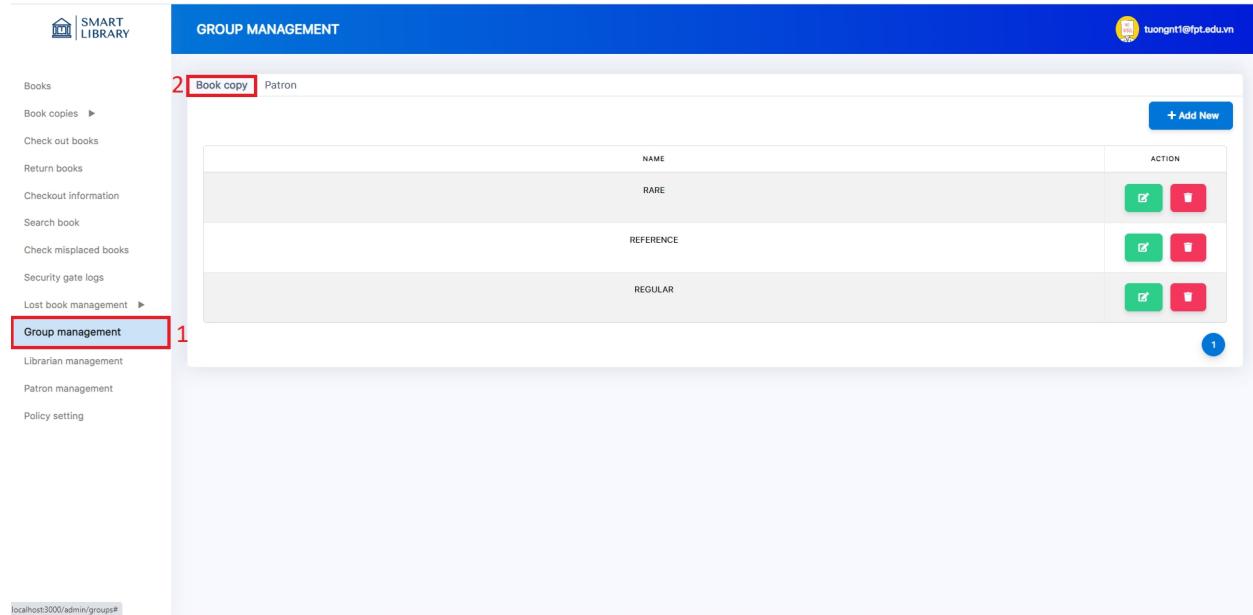
3.3.59. <Manager> View security gate log

The screenshot shows the Smart Library application interface. The left sidebar contains links for Books, Book copies, Check out books, Return books, Checkout information, Search book, Check misplaced books, and Security gate logs (which is highlighted with a red box and labeled 1). The main content area is titled "SECURITY GATE LOGS". It features a date picker with the date "2021-03-24" highlighted (labeled 2). Below the date picker is a table listing security logs. The first log entry shows a thumbnail of a book titled "HARMONY PROJECT", the name "Harmony", the barcode "02123400000002", and the timestamp "2021-03-24, 15:07:25". The second log entry shows the same details with the timestamp "2021-03-24, 15:08:01". The third log entry shows the same details with the timestamp "2021-03-24, 15:10:06". A blue "Apply" button is visible next to the date picker.

Step	Description
1	Click on “Security gate logs” on the sidebar
2	Click to show date picker

2.1	Choose a date
3	Click “Apply” to finish

3.3.60. <Manager> View book copy type



The screenshot shows the SMART LIBRARY Group Management interface. On the left sidebar, under 'Lost book management' (highlighted with a red box labeled '1'), the 'Group management' tab is selected (highlighted with a blue box labeled '2'). The main area is titled 'GROUP MANAGEMENT' and shows a table for 'Book copy'. The table has columns for 'NAME' and 'ACTION'. It lists three categories: 'RARE' (with edit and delete icons), 'REFERENCE' (with edit and delete icons), and 'REGULAR' (with edit and delete icons). A blue button '+ Add New' is located at the top right of the table area. The URL 'localhost:3000/admin/groups#1' is visible at the bottom of the browser window.

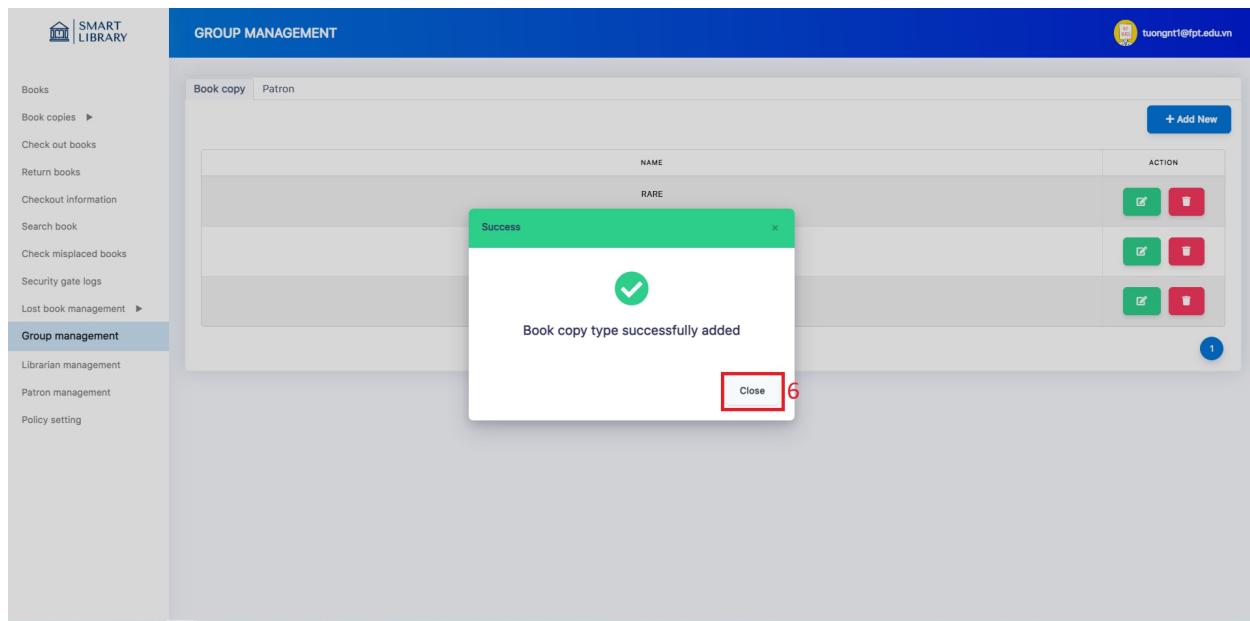
Step	Description
1	Click on “Group management” on the sidebar
2	Click on “Book copy” tab

3.3.61. <Manager> Add book copy type

The screenshots show the 'GROUP MANAGEMENT' section of the SMART LIBRARY application.

Screenshot 1: The 'Group management' menu is selected (1). The 'Book copy' tab is active (2). A modal window titled 'Add new book copy' is open, showing a 'Name*' field which is empty and highlighted with a red border (4). Below it is a note: '* Required field'. At the bottom are 'Cancel' and 'Confirm' buttons, with 'Confirm' highlighted with a green border (5).

Screenshot 2: The 'Group management' menu is selected (1). The 'Book copy' tab is active (2). A table lists three book copy types: RARE, REFERENCE, and REGULAR. Each row has edit and delete icons. A blue circular badge with the number '1' is visible in the bottom right corner of the table area. At the top right of the main content area is a '+ Add New' button (3), also highlighted with a red border.

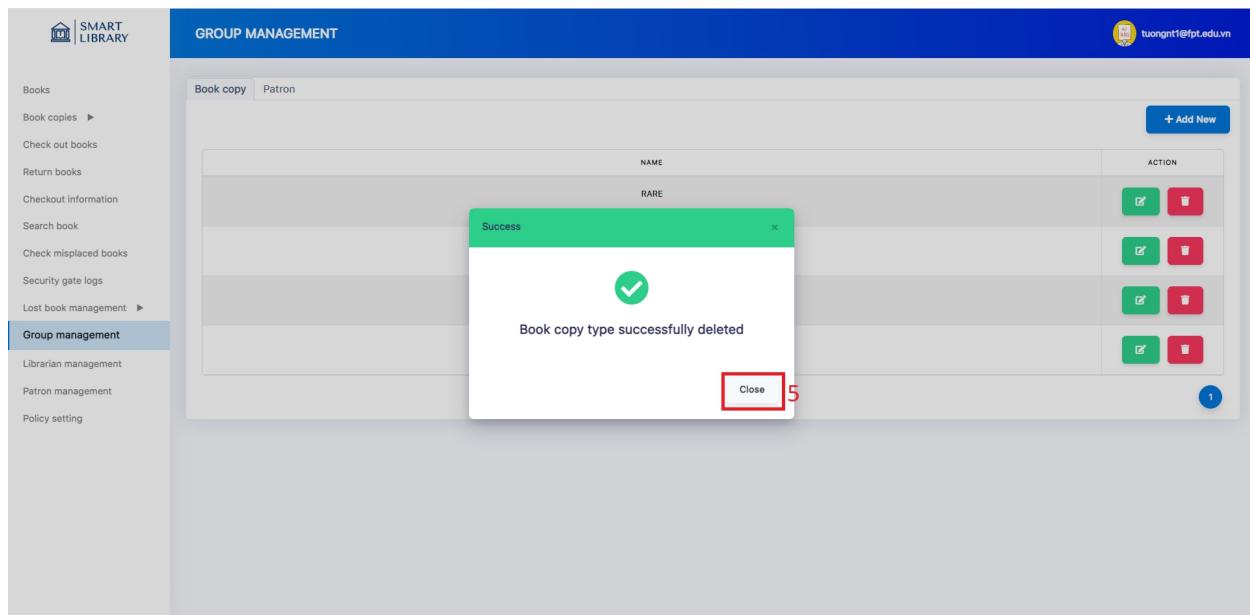


Step	Description
1	Click on “Group management” on the sidebar
2	Click on “Book copy” tab
3	Click on “Add New” button
4	Fill in “Name” field
5	Click on “Confirm” button
6	Click “Close” to finish add new book copy type

3.3.62. <Manager> Remove book copy

This screenshot shows the 'GROUP MANAGEMENT' section of the library software. On the left, a sidebar lists various management options. The 'Group management' option is highlighted with a red box and the number '1'. In the main area, a table titled 'Book copy' displays three rows: 'RARE', 'REFERENCE', and 'REGULAR'. Each row has two columns: 'NAME' and 'ACTION'. The 'ACTION' column contains icons for edit and delete. The 'Delete' icon for the 'RARE' row is highlighted with a red box and the number '3'. The URL 'localhost:3000/admin/groups#type' is visible at the bottom.

This screenshot shows the same 'GROUP MANAGEMENT' interface after a selection has been made. A confirmation dialog box titled 'Delete book copy' is centered over the table. The dialog asks 'Are you sure?' and 'Do you want to delete this book copy type?'. It has 'Close' and 'OK' buttons. The 'OK' button is highlighted with a red box and the number '4'. The URL 'localhost:3000/admin/groups#type' is visible at the bottom.



Step	Description
1	Click on “Group management” on the sidebar
2	Click on “Book copy” tab
3	Click on “Delete” button on book copy type in book copy type list
4	Click on “Confirm” button
5	Click “Close” to finish delete book copy type

3.3.63. <Manager> Updated book copy

GROUP MANAGEMENT

tuongnt1@pt.edu.vn

Books

Book copies ►

Check out books

Return books

Checkout information

Search book

Check misplaced books

Security gate logs

Lost book management ►

Group management 1

Librarian management

Patron management

Policy setting

localhost3000/admin/groups#

NAME	ACTION
RARE	4
REGULAR	

type

GROUP MANAGEMENT

tuongnt1@pt.edu.vn

Books

Book copies ►

Check out books

Return books

Checkout information

Search book

Check misplaced books

Security gate logs

Lost book management ►

Group management

Librarian management

Patron management

Policy setting

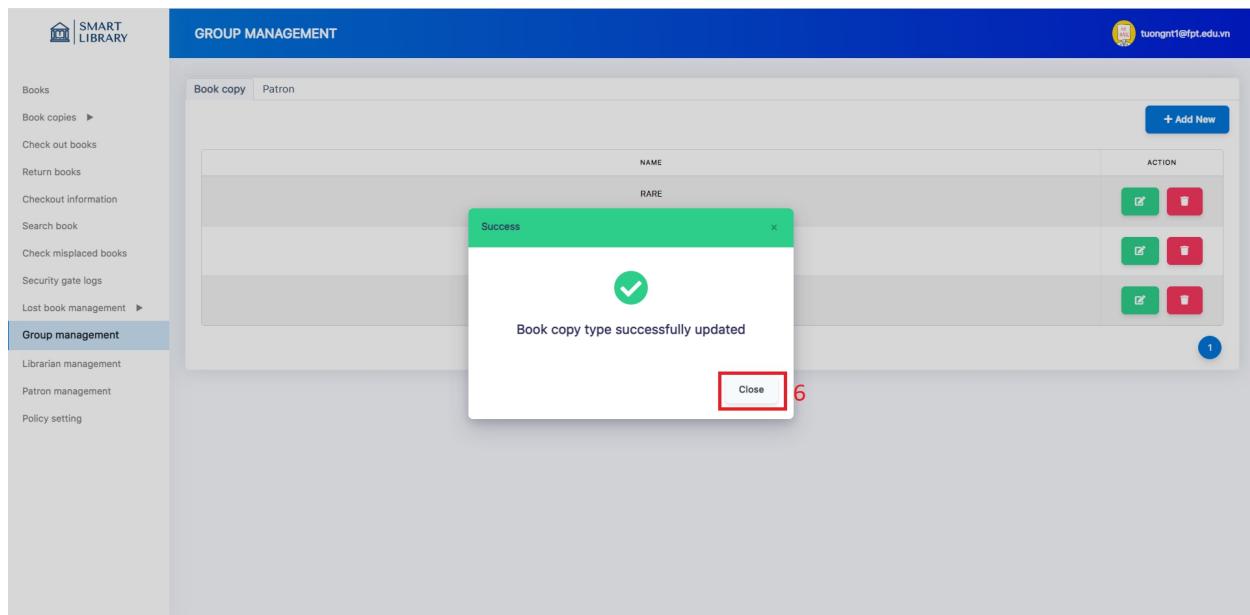
localhost3000/admin/groups#

Update book copy

Are you sure?

Do you want to update this book copy type?

Close
OK
5



Step	Description
1	Click on “Group management” on the sidebar
2	Click on “Book copy” tab
3	Enter Book copy type’s name
4	Click on “Update” button on book copy type in book copy type list
5	Click on “Confirm” button
6	Click “Close” to finish update book copy type

3.3.64. <Manager> View patron type

The screenshot shows the 'GROUP MANAGEMENT' page. On the left sidebar, 'Group management' is highlighted with a red box and labeled '1'. The 'Patron' tab is selected, indicated by a red box and labeled '2'. The main content area shows a table with three rows:

NAME	TOTAL CHECKOUTS ALLOWED	ACTION
STUDENT	6	
LECTURER	10	
TEST	1	

At the bottom right of the table, there is a blue button labeled '1'.

URL: localhost:3000/admin/groups#

Step	Description
1	Click on “Group management” on the sidebar
2	Click on “Patron” tab

3.3.65. <Manager> Add patron type

The screenshot shows the 'GROUP MANAGEMENT' page. On the left sidebar, 'Group management' is highlighted with a red box and labeled '1'. The 'Patron' tab is selected. At the top right, there is a blue button labeled '+ Add New' with a red box and labeled '3'.

The main content area shows the same table as the previous screenshot, listing STUDENT, LECTURER, and TEST patron types.

URL: localhost:3000/admin/groups#

The screenshot shows the SMART LIBRARY application's Group Management screen. On the left sidebar, under the 'Group management' section, the 'Patron management' option is selected. The main area displays a table of existing patron types: STUDENT (Name: TEST, Total Checkouts Allowed: 6), LECTURER (Total Checkouts Allowed: 10), and TEST (Total Checkouts Allowed: 1). A modal window titled 'Add new patron' is open, prompting for 'Name*' (containing 'Name' with red box 4) and 'Total Checkouts Allowed*' (containing 'Total Checkouts Allowed' with red box 5). A note at the bottom indicates '* Required field'. At the bottom of the modal are 'Cancel' and 'Confirm' buttons, with 'Confirm' highlighted with red box 6. In the top right corner of the main screen, there is an email address: tuongnt1@fpt.edu.vn.

The screenshot shows the same Group Management screen after the new patron type has been added. The table now includes a new row for 'LECTURER' with 'SUCCESS' in the Name column. A success message box is displayed in the center, containing a green checkmark icon and the text 'Patron type successfully added'. At the bottom of this message box is a 'Close' button highlighted with red box 7. The rest of the interface remains the same, including the sidebar and the top right email address.

Step	Description
1	Click on “Group management” on the sidebar
2	Click on “Patron” tab
3	Click on “Add New” button
4	Fill in “Name” field
5	Fill in “Max borrow number” field
6	Click on “Confirm” button

7

Click “Close” to finish add new patron type

3.3.66. <Manager> Remove patron type

GROUP MANAGEMENT

Book copy Patron 2

Select patron type... + Add New

NAME	TOTAL CHECKOUTS ALLOWED	ACTION
STUDENT	6	
LECTURER	10	
GUEST	1	Delete 3

Books Book copies ► Check out books Return books Checkout information Search book Check misplaced books Security gate logs Lost book management ► Group management 1 Librarian management Patron management Policy setting

GROUP MANAGEMENT

Book copy Patron

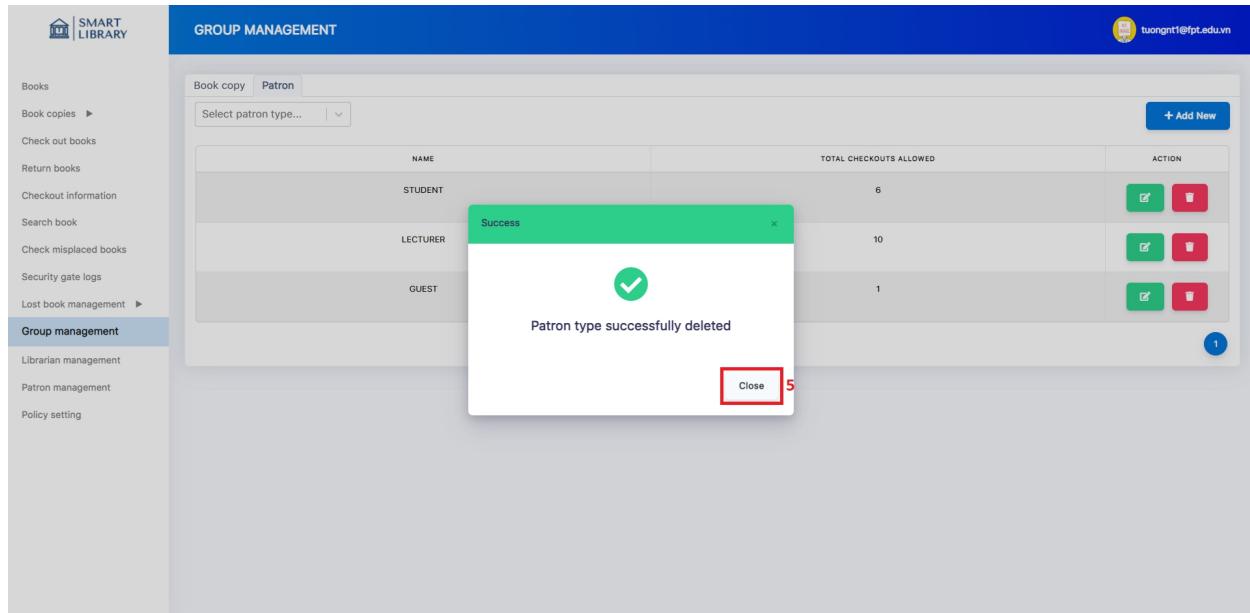
Select patron type... + Add New

NAME	TOTAL CHECKOUTS ALLOWED	ACTION
STUDENT	6	
LECTURER	10	
GUEST	1	

Are you sure?
Do you want to delete this patron type?

Close OK 4

Books Book copies ► Check out books Return books Checkout information Search book Check misplaced books Security gate logs Lost book management ► Group management 1 Librarian management Patron management Policy setting



Step	Description
1	Click on “Group management” on the sidebar
2	Click on “Patron” tab
3	Click on “Delete” button on patron type in patron type list
4	Click on “Confirm” button
5	Click “Close” to finish delete patron type

3.3.67. <Manager> Update patron type

GROUP MANAGEMENT

tuongnt1@fpt.edu.vn

Books
Book copies ►
Check out books
Return books
Checkout information
Search book
Check misplaced books
Security gate logs
Lost book management ►
Group management 1
Librarian management
Patron management
Policy setting

Book copy Patron 2

Select patron type... ▾

+ Add New

NAME	TOTAL CHECKOUTS ALLOWED	ACTION
STUDENT 3	6 4	5
LECTURER	10	

1

GROUP MANAGEMENT

tuongnt1@fpt.edu.vn

Books
Book copies ►
Check out books
Return books
Checkout information
Search book
Check misplaced books
Security gate logs
Lost book management ►
Group management
Librarian management
Patron management
Policy setting

Book copy Patron

Select patron type... ▾

+ Add New

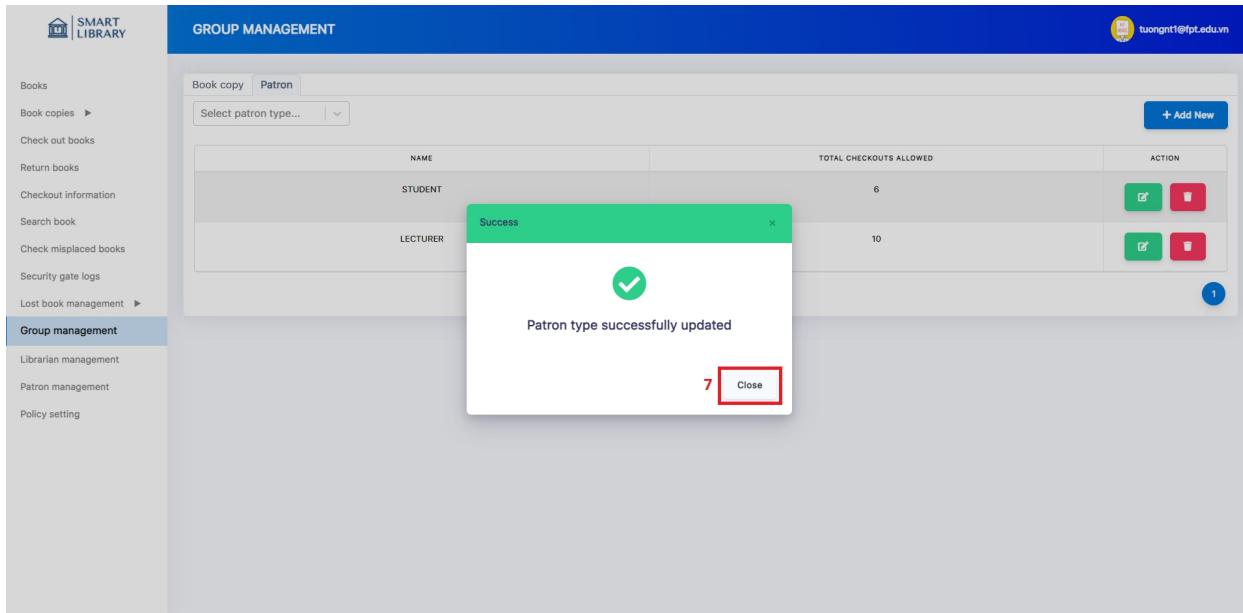
NAME	TOTAL CHECKOUTS ALLOWED	ACTION
STUDENT	6	
LECTURER	10	

Are you sure?

Do you want to update this patron type?

Close OK 6

1



Step	Description
1	Click on “Group management” on the sidebar
2	Click on “Patron” tab
3	Enter patron type’s name
4	Enter patron type’s max borrow number
5	Click on “Update” button on patron type in patron type list
6	Click on “Confirm” button
7	Click “Close” to finish update patron type

VII. Appendix

1. References

1. Fakhroutdinov, K. (n.d.). The unified modeling language. Retrieved January 10, 2021, from <https://www.uml-diagrams.org/>
2. React – a JavaScript library for building user interfaces. (n.d.). Retrieved January 10, 2021, from <https://reactjs.org/>
3. Spring boot. (n.d.). Retrieved January 10, 2021, from <https://spring.io/projects/spring-boot>
4. .Net: Free. cross-platform. open source. (n.d.). Retrieved January 10, 2021, from <https://dotnet.microsoft.com/>

5. Arduino. (n.d.). Retrieved January 10, 2021, from <https://www.arduino.cc/>
6. Fundamentals | Firebase. (n.d.). Retrieved January 10, 2021, from <https://firebase.google.com/docs/guides>
7. Gmail API | Google developers. (n.d.). Retrieved January 10, 2021, from <https://developers.google.com/gmail/api>
8. What is Rfid? HF VS. UHF explained. (2020, May 05). Retrieved January 10, 2021, from <https://www.tersosolutions.com/what-is-rfid/>
9. Different types of rfid systems. (n.d.). Retrieved January 10, 2021, from <https://www.impinj.com/products/technology/how-can-rfid-systems-be-categorized>
10. How librarians use isbns and barcodes. (2011, December 15). Retrieved January 10, 2021, from <http://authormaps.com/how-librarians-use-isbns-and-barcodes/>
11. Smiley, S. (2017, March 24). Encoding RFID tags: 3 things to know. Retrieved January 10, 2021, from <https://www.atlasrfidstore.com/rfid-insider/encoding-rfid-tags-3-things-to-kno>
12. Admin. (2020, July 06). Rfid in libraries: Privacy and confidentiality guidelines. Retrieved January 10, 2021, from <http://www.ala.org/advocacy/intfreedom/statementspolis/otherpolicies/rfidguidelines>
13. RFID and the 7 questions you must know. (n.d.). Retrieved January 10, 2021, from https://www.ableid.com/rfid_the_7_questions_you_must_know.html
14. Comparing LC and Dewey. (n.d.). Retrieved January 10, 2021, from <http://home.olemiss.edu/~tharry/LC/LCvsDDC.html>
15. List of Dewey Decimal classes. (2021, April 09). Retrieved January 10, 2021, from https://en.wikipedia.org/wiki/List_of_Dewey_Decimal_classes
16. Ứng dụng công nghệ Rfid trong quản lý và tự động hóa thư viện. (n.d.). Retrieved January 10, 2021, from <https://idtvietnam.vn/en/ung-dung-cong-nghe-rfid-trong-quan-ly-va-tu-dong-hoa-thu-vien-571>
17. Finding books: How to read dewey decimal call numbers. (n.d.). Retrieved January 10, 2021, from <https://libguides.pratt.edu/finding-books/reading-dewey-call-numbers>
18. Library RFID System. (n.d.). Retrieved January 10, 2021, from https://www.rfid-library.com/eng_shelf.html
19. Lưu hành tài liệu. (n.d.). Retrieved February 20, 2021, from <https://lib.tdtu.edu.vn/vi/dich-vu/luu-hanh>
20. Nội quy thư viện Đại Học Quốc Gia. (n.d.). Retrieved January 10, 2021, from <http://lib.hcmussh.edu.vn/Default.aspx?ArticleId=187d7085-1224-4f47-aa0b-5b1f72d6a1bb>
21. Find a book @ ellis library: Understand circulation status codes. (n.d.). Retrieved March 10, 2021, from <https://libraryguides.missouri.edu/c.php?g=583142&p=4026632>
22. Kuklovskaia, E. (n.d.). Barcode Label 1"-2.25". Retrieved April 19, 2021, from https://www.amazon.com/dp/B072NZF5CC/ref=emc_b_5_i
23. Dmarter@middlebury.edu. (2020, November 17). Library borrowing policy. Retrieved April 19, 2021, from <https://www.middlebury.edu/library/about/policies/library-borrowing>