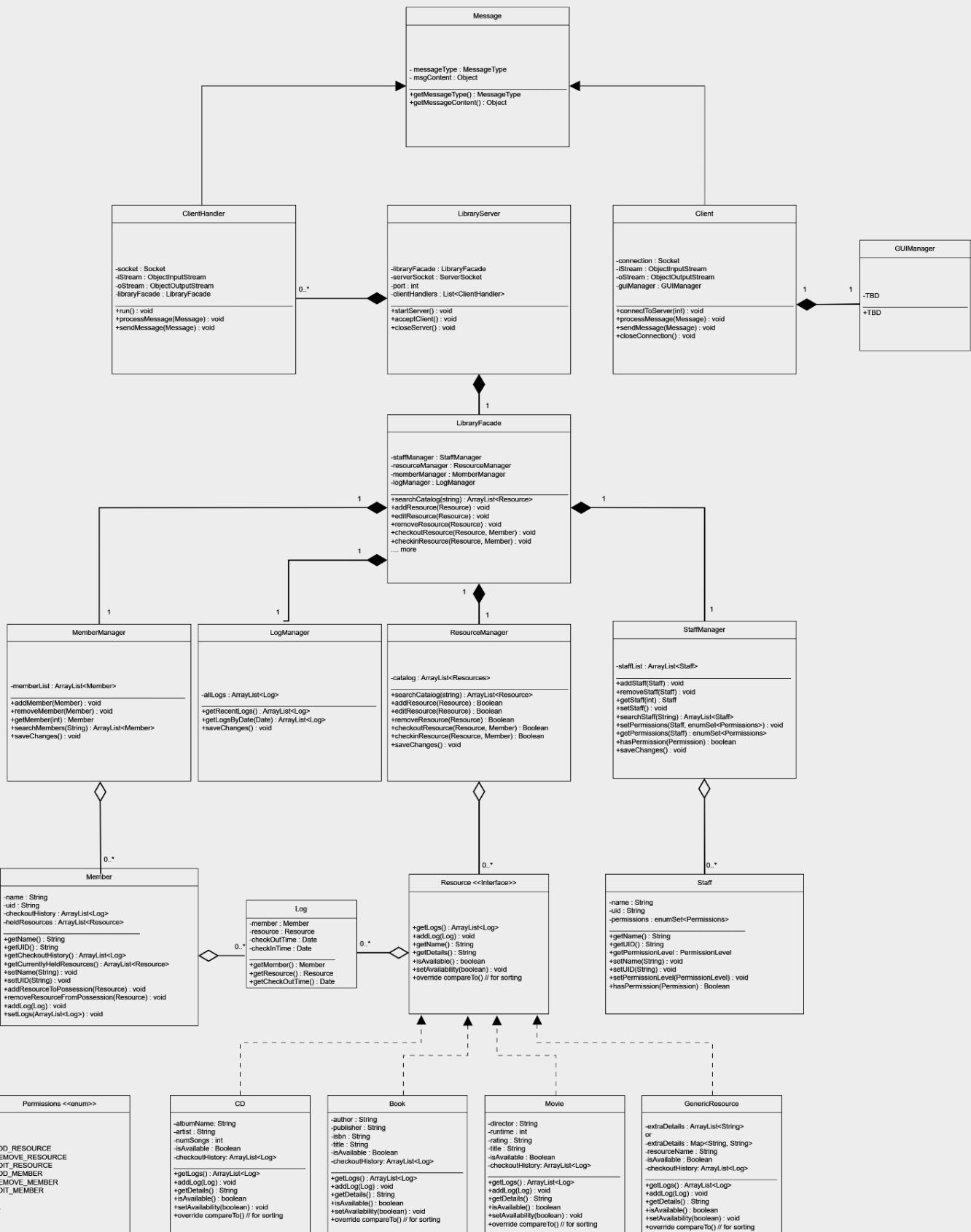


## *Use Case Document*

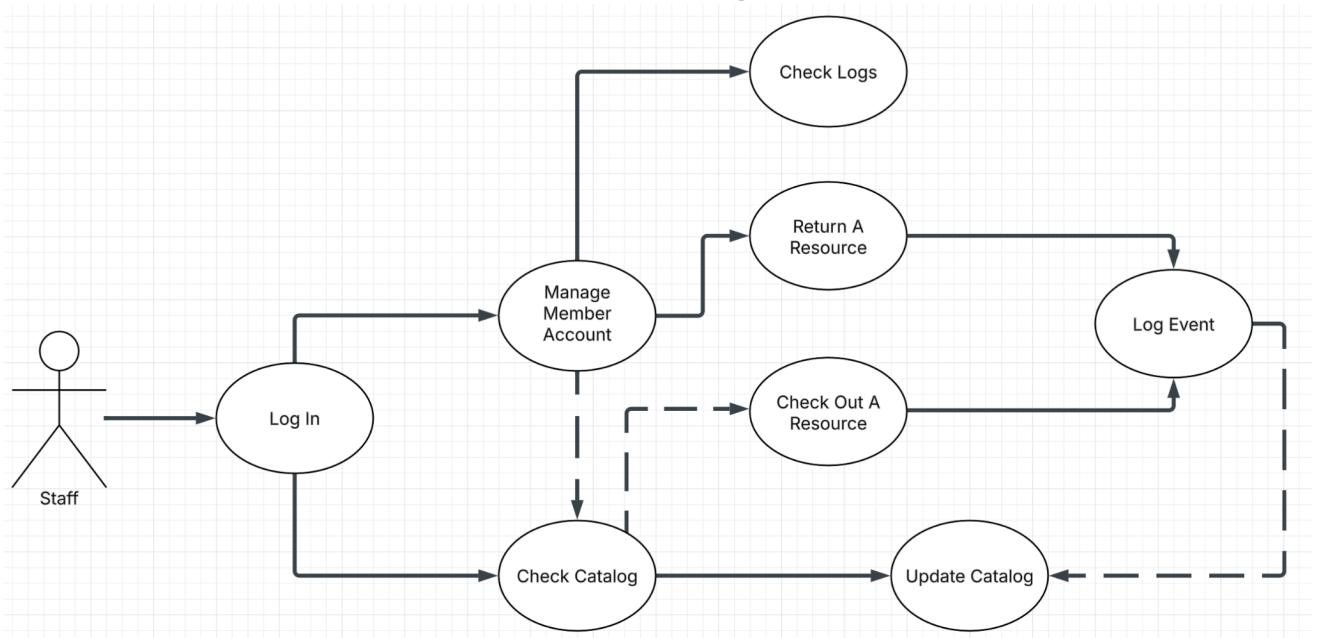
*Hugo Hernandez, Salvador Barriga, Ralph Tchaya, Ryan Hair & Nathan Siu*

## Revision History

Date	Revision	Description	Author
<u>09/11/2025</u>	1.0	Checking in/out book, Staff check logs use cases	
<u>09/19/2025</u>	1.1	Staff Checks Logs and Update Catalog use cases	Nathan S.
<u>09/23/2025</u>	1.2	Completed Remaining Use Cases	Salvador
<u>10/27/2025</u>	1.3	Edited Basic Flow of Update Catalog	Nathan S.
<u>10/28/2025</u>	1.4	Edited Sequence Diagram of Update Catalog	Nathan S.
<u>10/31/2025</u>	1.5	Include UML Class Diagram	Ryan



# Use Case Diagram



## Returning a Resource

Use case id: 01

**Use case name:** Returning a resource

**Relevant requirements:**

- 3.1.3.1
- 3.1.3.2
- 3.1.4.2

**Primary actor:**

- Staff

**Pre-condition:**

- Media item is checked out
- Media item exists in the catalog
- Staff logged in

**Post-condition:**

- The system updates the status of the media item to +1 copy available
- The system updates the member's account
- The system logs the event

**Basic flow or main scenario**

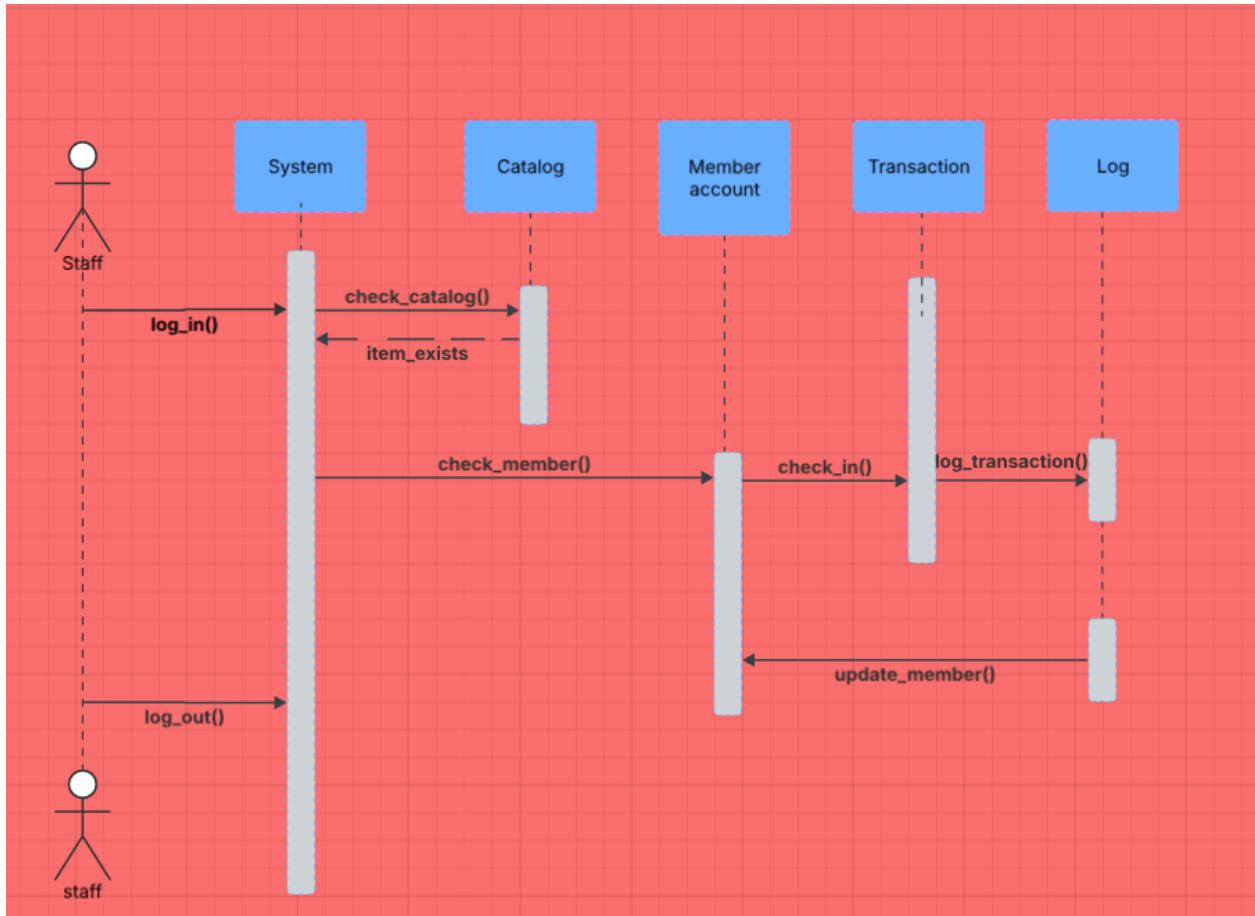
- Staff logs in
- Staff selects “check in” from main menu
- Staff enters a search for the matching item
- Staff selects the desired result from the search results
- System displays the borrower name and resource for confirmation
- Upon confirmation, system updates the relevant logs and profile information
- System displays a completion message and pressing any key returns to the main menu

**Extensions: (alternate scenarios):****Exceptions:**

- Staff checks in a book that was not returned. Later on, when the book is requested to be checked out, it's not available.
- Staff checks in a book under another book ID

**Related use cases:**

- Updating catalog
- Checking catalog



## Borrowing a Resource

**Use case id:** 02

**Use case name:** Checking out a resource

**Relevant requirements-**

- 3.1.4.2

**Primary actor:** Staff

**Pre-condition:**

- Staff is logged into the system
- Member has an active account and no overdue penalties that block borrowing
- Resource is available

**Post-condition:**

- Resource is marked as “borrowed”
- Due date is recorded in the system
- System logs the event

**Basic flow or main scenario:**

1. Staff searches for the member's account.
2. Staff selects the resource to loan
3. The system checks resource availability
4. The system records the loan transaction and assigns a due date

### Extensions

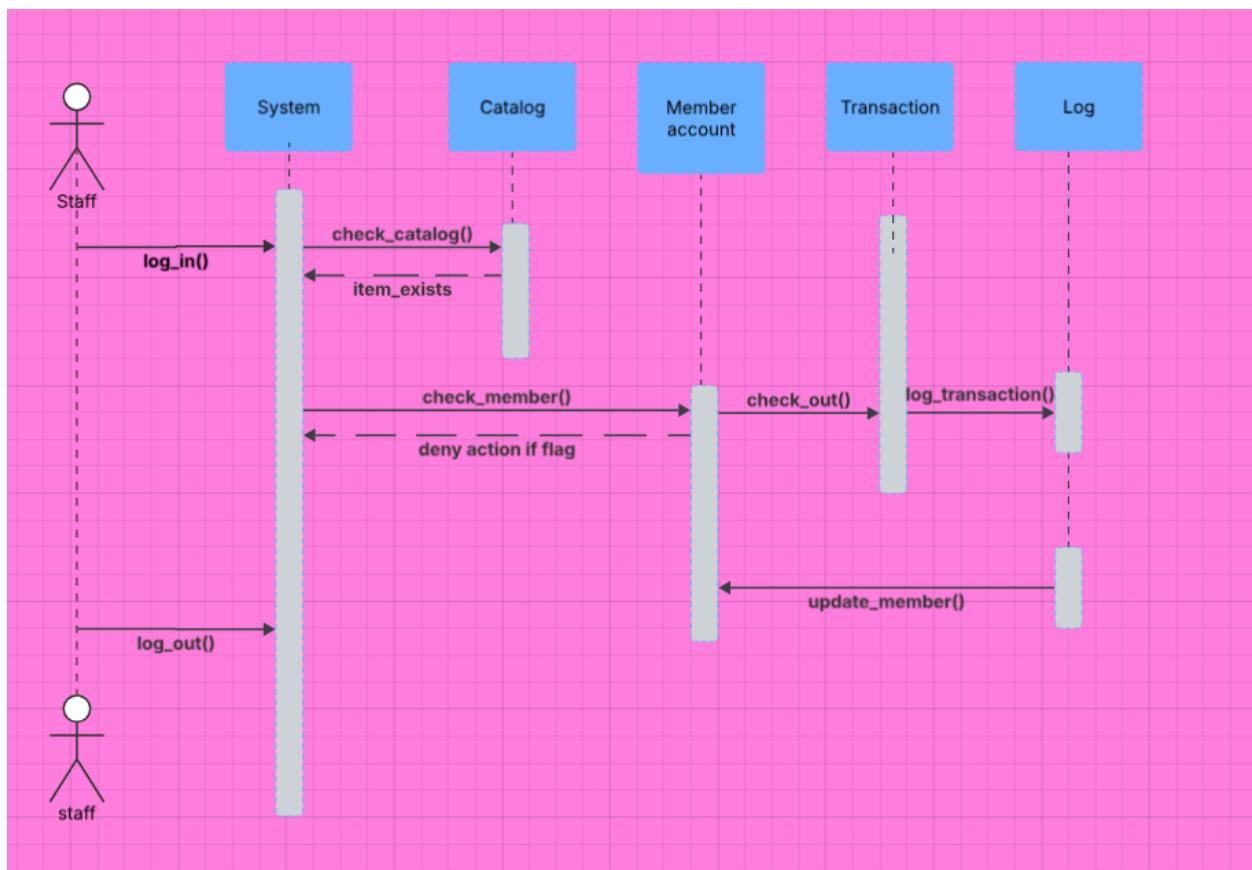
- Updating catalog

### Exceptions

- If requested item does not exist, system indicates “no items match the search”
- If requested item is already checked out, system indicates “item already checked out”
- If user is flagged for any reason, the system indicates the warning

### Related use case

- Member returns items checked out before checking out more items



## Staff Checks Logs

## **Use case id:** 03

**Use case name:** Checking logs

**Relevant requirements:**

- 3.1.1.3
- 3.1.5.1
- 3.1.5.2
- 3.1.5.3

**Primary actor:** Staff

**Pre-condition:** Staff log in data exists

**Post-condition:** Requested logs display, \**Staff can add notes*\*?

**Basic flow or main scenario:**

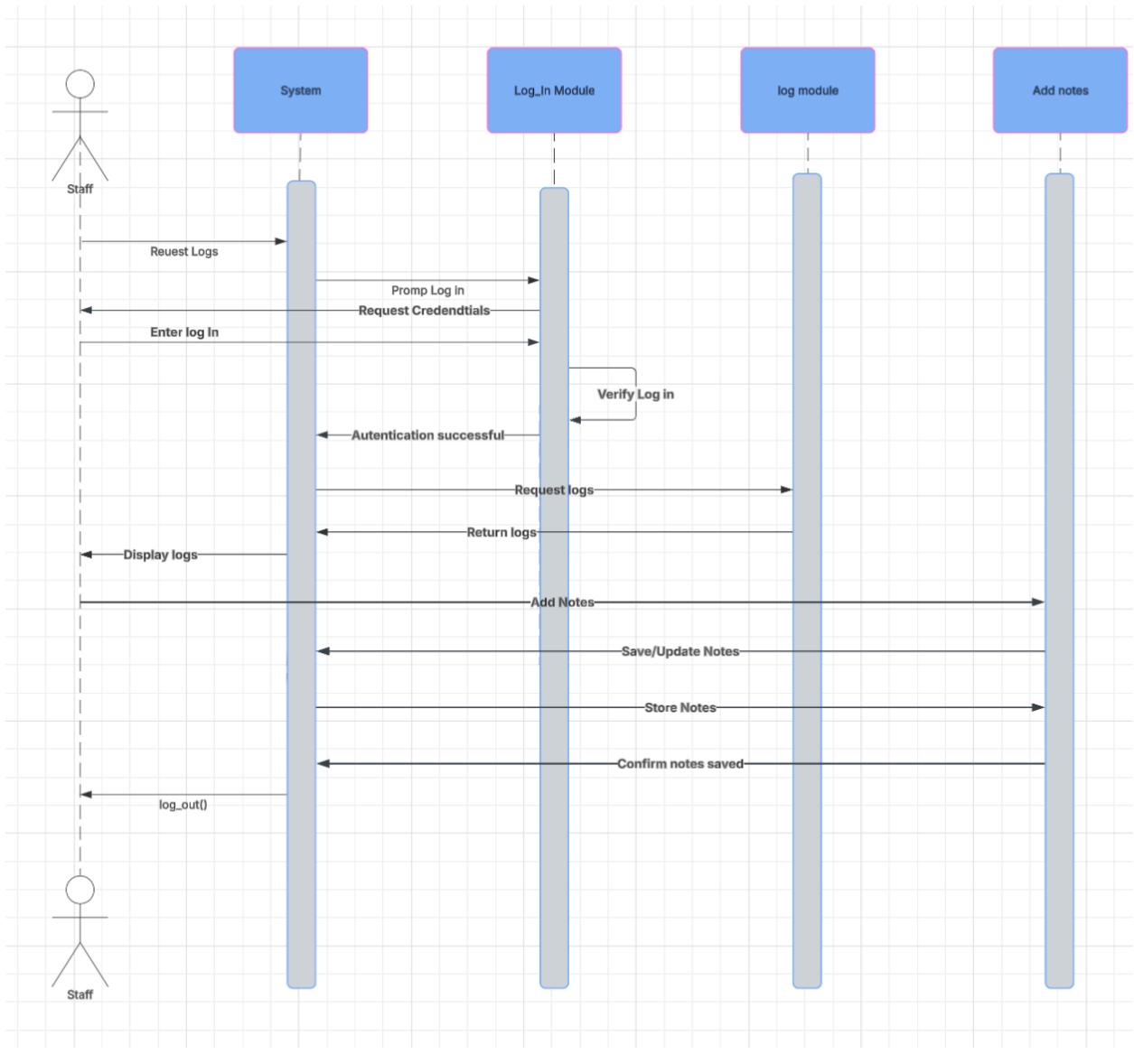
- Staff member logs in
- System verifies credentials
- System displays logs
- Staff member logs out

**Extensions**

- System can deny access for invalid log info -> prompt to re enter log info

**Exceptions**

- Too many failed log in attempts result in system lock
- If the system cannot fetch log in data -> Error prompt out putted



## Update Catalog

**Use case id:** 04

**Use case name:** Update Catalog

**Relevant requirements:**

- 3.1.3.2

**Primary actor:** Staff

**Pre-condition:**

- Staff is logged into the system

**Post-condition:**

- Resource information is updated in the system

**Basic flow or main scenario:**

- Staff navigates to the catalog
- Staff checks catalog
- Staff chooses to add/edit
- System opens up add/edit UI
- Staff adds, updates, or deletes resource(s)
- Catalog is updated

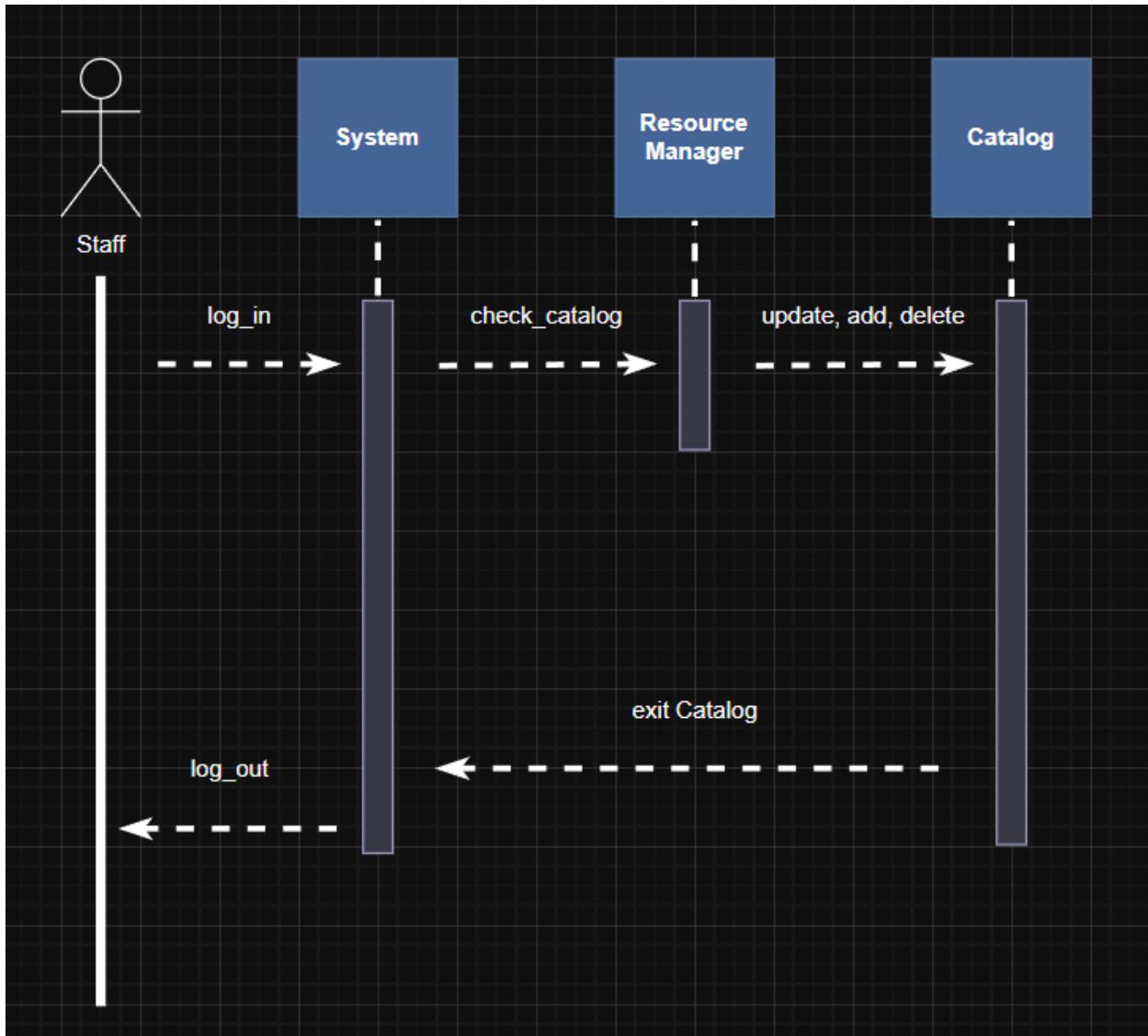
**Extensions:**

**Exceptions:**

- No duplicates
- Borrowed resources can't be deleted

**Related use case:**

- ID01 Returning a Resource
- ID02 Checking Out a Resource



## Catalog Search

Use case id: 05

**Use case name:** Catalog Search

**Relevant requirements:**

**Primary actor:**

- Staff

**Pre-condition:**

- Staff is logged into the application.
- The library catalog database is accessible.

**Post-condition:**

- The staff user is presented with the relevant media information

**Basic flow or main scenario**

- Staff logs in
- Staff selects the “Search Catalog” option from main menu
- Staff enters and submits search criteria
- System returns a list of matching items with their details and current status

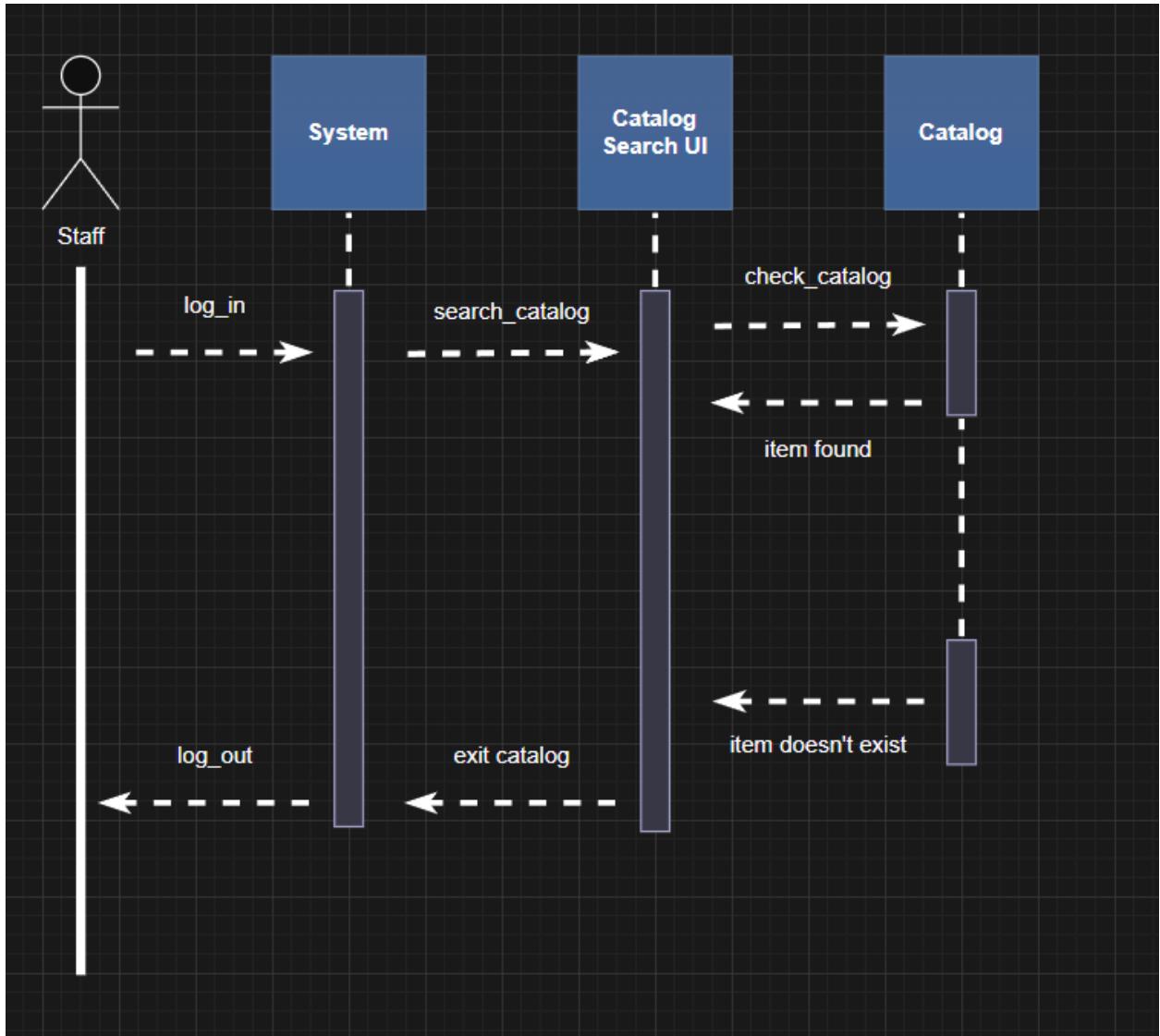
**Extensions: (alternate scenarios)**

**Exceptions:**

- If the search returns no results, the system indicates “no items match the search”

**Related use cases:**

- ID04 Update catalog
- ID02 Check out media



## Manage Member Account

Use case id: 06

**Use case name: Add, Update, and Delete Member Information**

**Relevant requirements:**

- 3.1.6.1

**Primary actor:**

- Staff

**Pre-condition:**

- Staff should be logged in with a valid username and password

- Members should have an account

#### **Post-condition:**

- Member information is updated in the system

#### **Basic flow or main scenario**

- Staff decides to update a member's account info.
- Staff chooses to add a new member, update member account details or delete a member
- The system validates the entered information
- The system is updated

#### **Extensions: (alternate scenarios)**

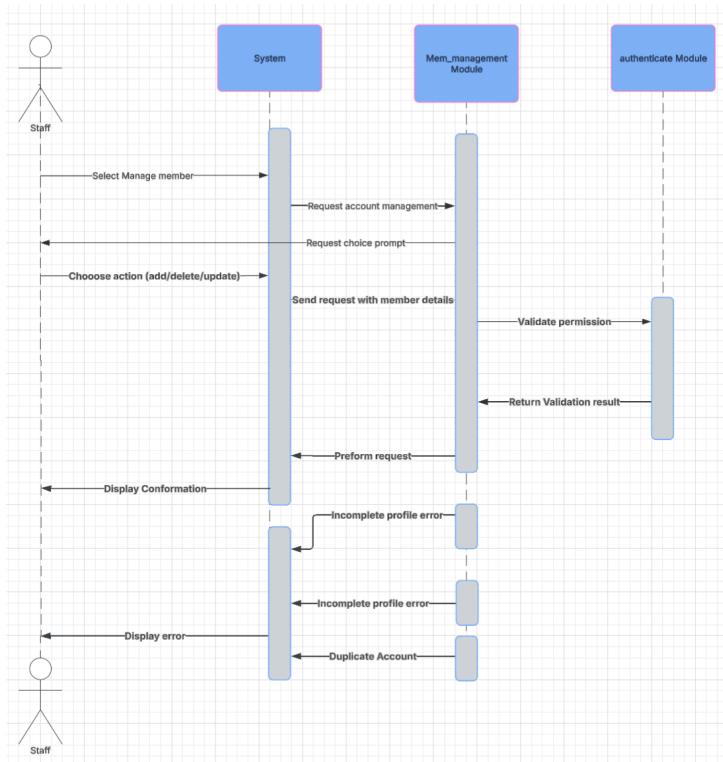
- Adding members with missing information should be flagged as an incomplete profile
- A member profile shouldn't be allowed to be deleted if the member has any resources that haven't been returned

#### **Exceptions:**

- The system shouldn't allow duplicate accounts

#### **Related use cases:**

- ID01 Checking out a resource
- ID02 Returning a resource



# Staff Login

**Use case id:** 07

**Use case name:** Authenticate staff into the system

**Relevant requirements:** 3.1.1.1

**Primary actor:**

- Staff

**Pre-condition:**

- Staff has valid login credentials

**Post-condition:**

- Staff gains access to system features

**Basic flow or main scenario**

- Staff enters username and password
- The system validates credentials against stored records
- The system grants access to the staff member

**Extensions: (alternate scenarios)**

- Multi-factor authentication may be required for security reasons

**Exceptions:**

- Incorrect credentials result in a displayed error message

**Related use cases:**

- ID06: Manage Member Account
- ID04: Update Catalog

