DESENVOLVIMENTO DE SOFTWARE SEGURO

Project – Phase 1

M1B
António Fernandes 1190402
Carla Barbosa 1200928
Carlos Rodrigues
Jorge Almeida
Nuno Figueiredo

pbs@isep.ipp.pt

27/04/2025

Content

Table of Figures

Table index

Introduction

The first phase of the DESOFS project focuses on the secure planning and design of the Library Online Rental System.

This web service aims to provide users with a simple, secure platform to browse, rent, suggest, and review books online.

Throughout this phase, we have gathered and documented the functional, non-functional, and security requirements, established the domain and logic views, and outlined the main user stories. Threat modelling has been conducted using the STRIDE methodology, supported by DFDs, abuse cases, attack trees, and qualitative and quantitative risk analyses.

The application architecture and design were built according to secure software development lifecycle (SSDLC) principles, ensuring defense-in-depth, least privilege, secure data handling, and rigorous authentication and access control mechanisms.

Future phases will implement and validate these designs through secure development practices, security testing, and compliance verification with OWASP ASVS Level 2 requirements.

Analysis/Requirements

Domain Model

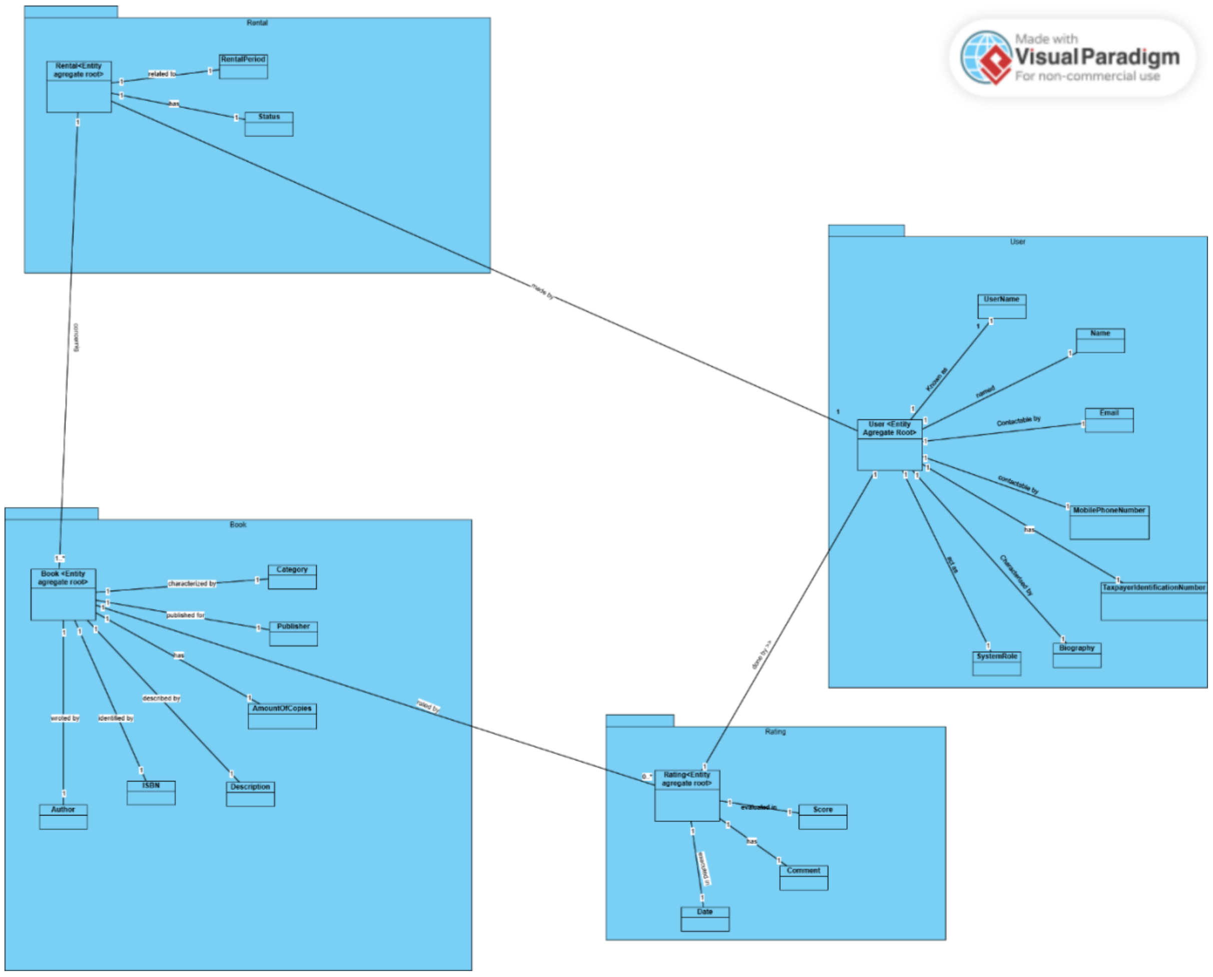


Figure 1 - DDD Domain model

Logic View – Level 1

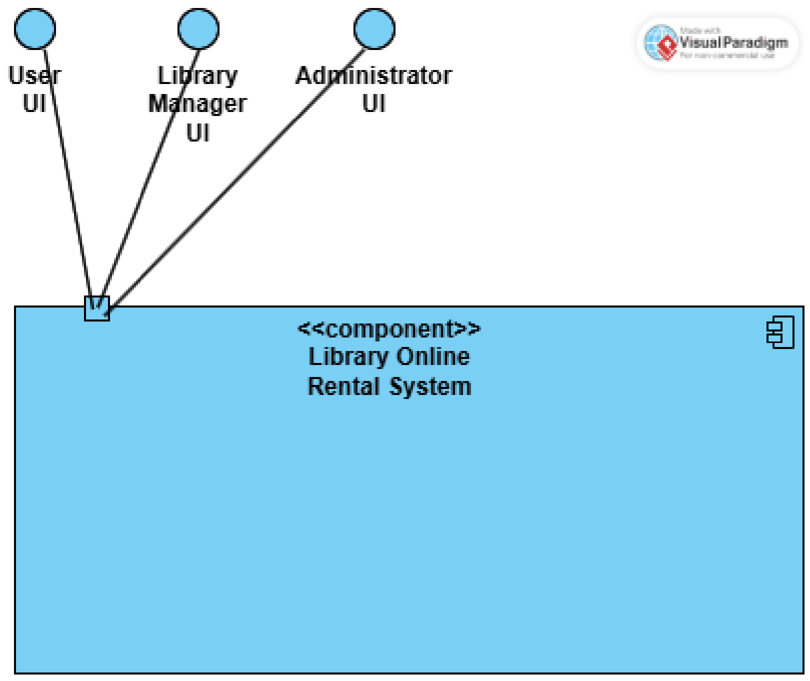

User UI
Library Manager UI
Administrator UI
<<component>>
Library Online Rental System
Made with Visual Paradigm For non-commercial use

Figure 2 - Logic View - Level 1

Logic View – Level 2

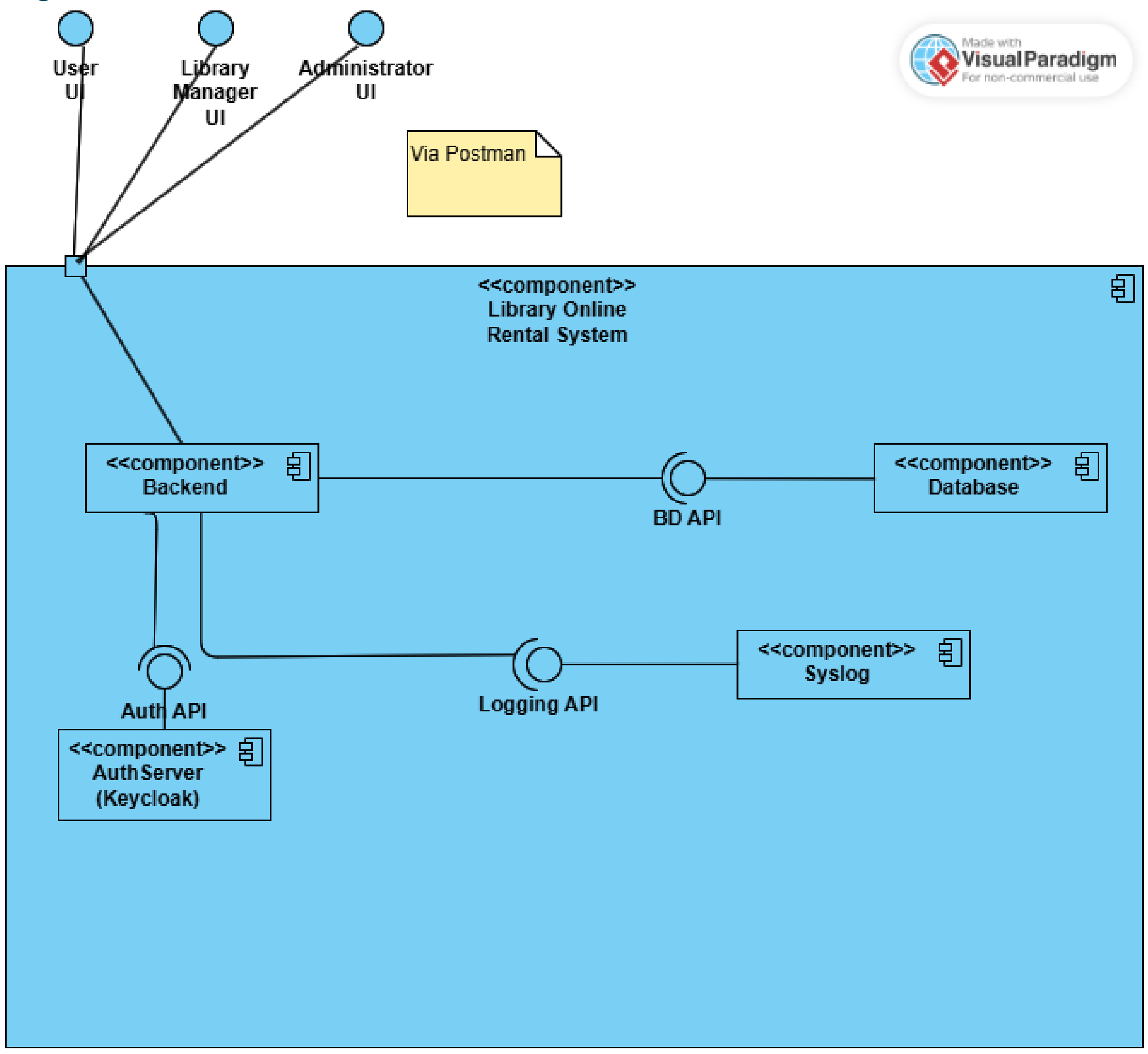


Figure 3 - Logic View - Level 2

User Stories

This section details the user stories.

1. **As a registered user**, I want to view and edit my profile information. I must not be able to view or edit other users' profiles.
2. **As a registered user**, I want to rent available books to enjoy reading. I must not be able to rent books that are not available or rent on behalf of other users.
3. **As a registered user**, I want to suggest new books to be added to the library. I must not be able to submit invalid or malicious data in the suggestion.
4. **As a registered user**, I want to rate and review books I have rented. I must not be able to rate books that I haven't rented.
5. **As an unauthenticated user**, I want to browse the book collection and use the advanced search with filters by category, author, and publisher. I must not be able to perform actions restricted to authenticated users, such as renting books or submitting suggestions.
6. **As a user**, I want to log in to access restricted functionalities. I must not be able to access features that require higher privileges than those assigned to me.
7. **As an administrator**, I want to manage user data and access permissions. Only the administrator should be able to perform these actions.
8. **As a library manager**, I want to manage books and rental records to ensure proper library operations. I must not be able to access user account data or modify access levels (only administrators can do this).
9. **As a library manager**, I want to update or remove books from the system. I must not be able to remove books that are currently rented out.
10. **As a library manager**, I want to register the return of rented books. I must not be able to register the return of books that have not been rented.

Functional Requirements

User

- **FR1.** The user shall be able to register and authenticate within the application.
- **FR2.** The user shall be able to change their profile anytime (password and MFA-associated email address). When the MFA email is changed, a notification must

be sent to the registered email address associated with the account to alert the user of this change.

- **FR3.** The user shall be able to view the list of available books.
- **FR4.** The user shall be able to search for books by title, author, publisher, or category.
- **FR5.** The user shall be able to view detailed information about a specific book.
- **FR6.** The user shall be able to rent a book if copies are available. There shall be a limit on the number of books a user can rent at the same time.
- **FR7.** The user shall be able to view their personal rental history.
- **FR8.** The user shall be able to rate books they have previously rented. Each user may rate a book only once per rental.
- **FR9.** The user shall be able to suggest new books for the library to acquire.

Library Manager

- **FR10.** The manager shall be able to add, update, or remove books from the catalogue.
- **FR11.** The manager shall be able to view and manage active rentals.
- **FR12.** The manager shall be able to approve or reject book suggestions from users.

Administrator

- **FR13.** The administrator shall be able to create, edit, or remove user accounts.
- **FR14.** The administrator shall be able to assign roles to users (user, manager, admin).
- **FR15.** The administrator shall be able to view audit logs and system activity.

Non-Functional requirements

Security

- **NFR1.** All communication between client and server must be encrypted (HTTPS).
- **NFR2.** User passwords must be securely hashed (e.g., using bcrypt).
- **NFR3.** The system must enforce role-based access control (RBAC).
- **NFR4.** The API must be protected against common threats (e.g., SQL Injection, XSS, CSRF).
- **NFR5.** All sensitive actions must be logged for auditing purposes.
- NFR6. All users must authenticate using two-factor authentication (code via email).

Performance & Scalability

- **NFR6.** The API should respond in under 500ms for 95% of requests.
- **NFR7.** The system should support at least 100 concurrent users without performance degradation.
- **NFR8.** The database should be horizontally scalable for read operations.

Maintainability & Deployment

- **NFR9.** The codebase must follow clean architecture principles with clear separation of concerns.
- **NFR10.** The system must support CI/CD pipelines with automated testing.
- **NFR11.** The deployment should be containerised (e.g., Docker) and support multiple environments (dev, staging, prod).
- **NFR12.** At least two backup copies must be stored in separate online locations.
- **NFR13.** At least one copy of the backup must be stored offline.

Usability & Availability

- **NFR12.** The API must follow RESTful conventions and return meaningful error messages.
- **NFR13.** The system must maintain 99% uptime availability (internal SLA).
- **NFR14.** API documentation must be accessible through Swagger/OpenAPI.

Security Requirements

Authentication and Access Control

- **SDR1.** User authentication must implement Multi-Factor Authentication (MFA) via a code sent by email, using Keycloak as the identity and access management system.
- **SDR2.** The application must lock the user account after three failed login attempts, as a preventive measure against brute force attacks.
- **SDR3.** User passwords must contain at least 8 characters, including uppercase and lowercase letters, numbers, and special characters.
- **SDR4.** Upon registration, the application must send a verification email to the user to validate their identity.
- **SDR5.** The system must implement Role-Based Access Control (RBAC), applying the principle of least privilege.
- **SDR6.** Authentication-related error messages must not be overly detailed to prevent the exposure of sensitive information to potential attackers.

Data Protection

- **SDR7.** All sensitive data, both at rest and in transit, must be encrypted, ensuring secure communication using the TLS protocol.
- **SDR8.** Passwords must be securely hashed using algorithms such as bcrypt.
- **SDR9.** Collected personal data must be treated confidentially and used exclusively for their intended purposes, in compliance with the General Data Protection Regulation (GDPR).

Input Validation & Error Handling

- **SDR10.** The application must validate user-submitted data, rejecting invalid values (e.g., nulls or incorrectly formatted data).
- **SDR11.** For the new book suggestion field, the application must only accept plain text, which must be validated and sanitized to prevent code injection, including SQL Injection and Cross-Site Scripting (XSS).

Logging and Auditing

- **SDR12.** All logs of sensitive actions must be generated and stored in the system, with a copy also sent to an external logging server.
- **SDR13.** The system must perform three backups of the logs, with two stored in separate online systems and one in an offline location.

Use Cases Diagram

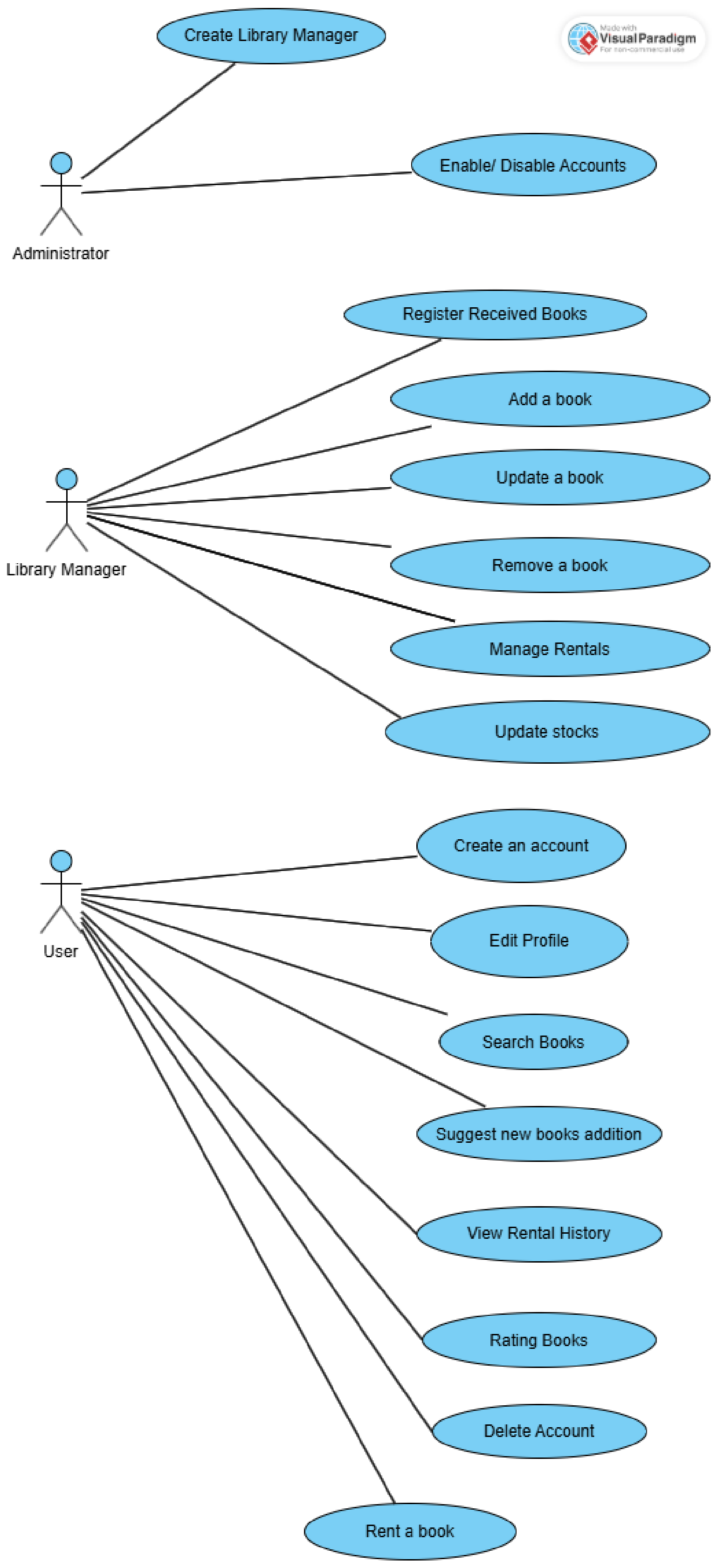


Figure 4 - Use Cases Diagram

Physical View Diagram

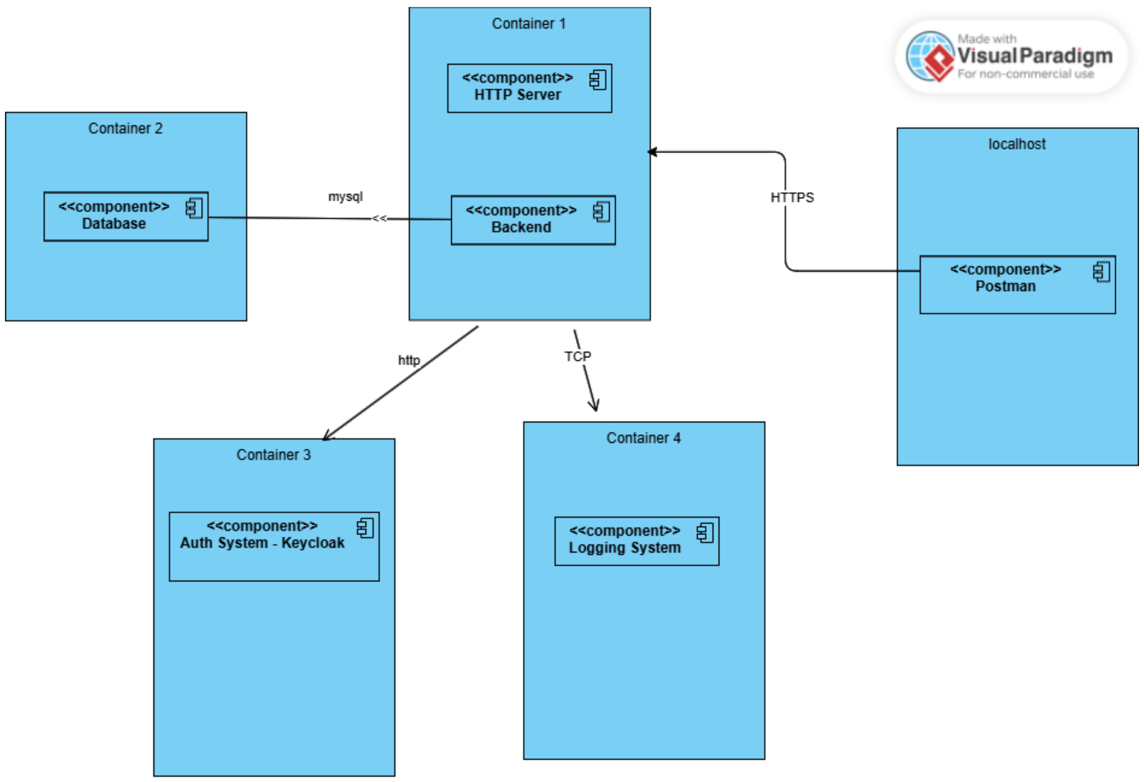


Figure 5 - Physical View Diagram

Design

Threat modelling

Application: Library Online Rental System

Application Version: 1.0

Description:

This project objective is to develop a webservice to support the operation of a book library website. The service is provided free of charge to promote reading habits among users.

The service is composed of a web application programming interface using REST, developed in the ASP.net framework. To support the web application data, a MySql relational database is used.

The proposed system enables users to browse a comprehensive collection of books, apply advanced search capabilities and filter options based on categories, authors, and publishers. The user can rent or suggest new book additions to the library as well as provide ratings. The rented books must be collected at the library and are not sent over delivery service.

The library manager is responsible for maintaining data, including books and rentals, the admin is responsible for maintaining user and accessing data.

External Dependencies

They are fundamental for the safe and efficient operation of the API, so it is important to identify and analyse them.

Table 1- External Dependencies

ID	Description
1	**Keycloak Identity Provider** – Used for user authentication and role-based access control (RBAC). The backend application communicates with Keycloak to handle login, token management, and user sessions.
2	**Keycloak Database** – Stores user credentials, MFA configurations, and role mappings. Although not accessed directly by the backend.
3	**MySQL Relational Database** – Stores business data such as books, rentals, suggestions, ratings, and audit logs. Access is performed via secure connections.

4	**Email Server (SMTP)** – Used to send MFA codes, verification emails, and security notifications (e.g., profile changes, password updates).
5	**Docker** – Used for deployment and containerization of the backend services and supporting components (e.g., database, Keycloak).
6	**CI/CD Pipeline Server** – Automates testing, builds, and deployment processes. Interacts with the source code repository and Docker environment.
7	**Firewall and Network Security** – Ensures that only necessary ports (e.g., 443 for HTTPS) are open. All other traffic is filtered according to security policies.
8	**HTTPS/TLS Certificates** – Ensure encrypted communication between clients and the server. All API endpoints must enforce HTTPS.
9	**External Logging Server** – Receives logs of sensitive or critical actions from the backend application for auditing and monitoring purposes.
10	**Backup Storage Services** – At least two geographically separate online locations and one offline medium used to store encrypted backups of data and logs.
11	**.NET Runtime Environment** – Required to run the backend application
12	**Third-party Libraries** – Possible external code libraries used in development. Must be updated regularly to avoid known vulnerabilities.

Trust Levels

This section details the various *trust levels* of our API.

Table 2 - Trust Levels

ID	Name	Description
1	**Anonymous User**	A non-authenticated visitor. Can browse the book catalogue and perform book searches but cannot access any functionality that requires login.
2	**Authenticated User**	A registered user who can log in to the system, manage their profile, rent books, view rental history, rate books, and suggest new books (titles).
3	**Library Manager**	A staff member responsible for managing the book catalogue and rental records. Can add, update, or remove books and process user suggestions. Cannot manage user roles or access audit logs. Operates with limited privileges according to API logic.
4	**Administrator**	A user with full administrative privileges in the system, including database. Can manage user accounts, assign roles,

		and access system-level audit logs and activity records. Cannot rent books or interact with the catalogue as a regular user.

The keycloack database is not managed manually by any employee. This management is done internally by keycloak.

Entry Points

Table 3 - Entry Points

ID	Name	Description	Trust Levels	Data Received
1	**HTTPS Port**	All web traffic is served over TLS/HTTPS to ensure secure communication.	All user roles	Encrypted HTTP requests
1.1	**Login Page**	Web interface where users enter credentials to authenticate.	Anonymous User	Email, password
1.2	**User Profile Page**	Allows authenticated users to view and update their profile.	Authenticated User, Manager, Admin	Name, email, optional photo, MFA email, password
1.3	**Book Catalogue Page**	Allows users to browse and filter books.	All user roles	Search parameters (title, author, category, etc.)
1.4	**User Registration Page**	Allows new users to create an account.	Anonymous User	Name, email, password, optional photo, date of birth

1.5	**Book Rental Page**	Interface to request a book rental.	Authenticated User	Selected book ID, rental request details
1.6	**Suggestion Form**	Allows users to suggest new books.	Authenticated User	Book title, author, and optional notes
2.1	**Manager Dashboard**	Interface for library managers to manage books and rentals.	Library Manager	Book info (title, author, price, stock), rental status
2.2	**Admin Dashboard**	Interface for administrators to manage users and roles.	Administrator	User info (name, role, email), system activity filters
3	**API Root**	All frontend interactions are handled through RESTful API calls over HTTPS.	All user roles	JSON payloads via secure HTTP requests
3.1	**/api/auth/login**	Endpoint to authenticate users and issue JWT tokens.	Anonymous User	Email, password
3.2	**/api/auth/logout**	Terminates active sessions.	Authenticated User, Manager, Admin	Auth token (header)
3.3	**/api/user**	Retrieves authenticated user’s account details.	Authenticated User, Manager, Admin	Auth token
3.4	**/api/user/{id}**	Retrieves or updates account details for the specified user.	Authenticated User (self only), Admin	User ID in URL, updated fields in JSON

3.5	**/api/books**	Retrieves list of books with filter options.	All user roles	Query parameters
3.6	**/api/books/{id}**	Retrieves detailed information about a specific book.	All user roles	Book ID
3.7	**/api/rentals**	Creates a new rental if a book is available.	Authenticated User	Book ID, user ID (inferred from session)
3.8	**/api/suggestions**	Submits a book suggestion for review.	Authenticated User	Book suggestion details (plain text)
3.9	**/api/manage/books**	CRUD operations for managing book data.	Library Manager	Book metadata in JSON
3.10	**/api/manage/rentals**	Views and updates rental records.	Library Manager	Rental data
3.11	**/api/admin/users**	Manages user accounts and roles.	Administrator	User account data
3.12	**/api/admin/logs**	Retrieves audit logs for system activity.	Administrator	Log filters

Exit Points

Table 4 – Exit Points

ID	Name	Description	Data Sent	Potential Vulnerabilities
1	**HTTPS Port**	All server responses are returned over encrypted HTTPS connections.	Encrypted HTTP responses	TLS misconfiguration, outdated cipher suites

1.1	**Login Response**	Authentication result sent after login attempt.	Success or error message, JWT token (if successful)	Token leakage, error message disclosure
1.2	**Logout Response**	Confirmation that the user has been logged out.	Success message	None (low risk)
1.3	**User Profile Response**	Sends current or updated user profile information.	User data (name, email, MFA info, etc.)	Sensitive data exposure, broken access control
1.4	**Book Catalogue Results**	Returns filtered or full list of available books.	Book summaries (title, author, category, availability)	Information disclosure
1.5	**Book Details Response**	Send detailed information for a specific book.	Full book metadata (title, author, publisher, description, etc.)	Excessive data exposure
1.6	**Rental Confirmation**	Confirms that a book was successfully rented.	Success message, rental ID, due date	Enumeration or info leak via predictable rental IDs
1.7	**Rental History Response**	Sends list of past rentals for the authenticated user.	List of books with rental dates and statuses	Privacy violation (accessing others data)
1.8	**Book Rating Acknowledgment**	Confirms a submitted rating or review.	Success message	Lack of validation, repeated submission

1.9	**Suggestion Acknowledgment**	Confirms that a new book suggestion has been submitted.	Success or error message	Injection via suggestion feedback
2.1	**Book Management Result**	Result of a book creation, update, or deletion.	Success or error message	Broken access control, injection
2.2	**Rental Management Response**	Provides rental information or update confirmation.	List of active rentals or confirmation of changes	Unauthorized rental changes
2.3	**Suggestion Review Response**	Result of approving or rejecting user suggestions.	Status message	Disclosure of internal workflow
3.1	**User Management Result**	Response from user creation, update, or deletion.	Success or error message	Account tampering, role abuse
3.2	**Role Assignment Confirmation**	Confirms user role changes.	Success message	Privilege escalation
3.3	**Audit Log Output**	Returns filtered audit logs and system activity data.	List of log entries with timestamps, actions, and user IDs	Log injection, data leakage, broken access control

Assets

This section details the relevant business assets for threat modeling.

Table 5 - Assets

ID	Name	Description	Trust Levels
1	**Stored Data**	All persistent data maintained by the application, including user accounts, books, rental records, and suggestions.	
1.1	**User Credentials**	Managed by Keycloak. Includes email, hashed password, and MFA email. Used for authentication and RBAC.	Authenticated User, Library Manager, Administrator, Keycloak
1.2	**User Personal Data**	Includes name, email, birth date, and address. Considered sensitive under GDPR.	Authenticated User, Library Manager, Administrator
1.3	**Rental Records**	Includes book ID, user ID, rental date, due date, and rental status. Managed in the backend.	Authenticated User, Library Manager
1.4	**Book Data**	Information about books (title, author, category, description, availability, etc.). Stored and managed in the backend.	All user roles
1.5	**Suggestions**	Suggested book titles submitted by users for potential acquisition.	Authenticated User, Library Manager
1.6	**Audit Logs**	Logs of sensitive operations (e.g., user management, rental activity, login attempts). Stored locally and externally.	Administrator
2	**System Services**	Key services required for the platform's operation, including the backend, database, and authentication system.	
2.1	**Keycloak (Authentication)**	External service used for authentication and issuing access tokens (JWT). Verifies user identity and roles.	All user roles

2.2	**MySQL Database**	Stores all persistent backend data: user credentials, book records, rental history, etc.	Administrator, Library Manager
2.3	**Email Service (SMTP)**	Sends MFA codes, account verification emails, and notifications about changes to user accounts.	Authenticated User, Library Manager, Administrator
3	**Sessions and Tokens**	Active login sessions and access tokens (JWT). Used to control and verify user access to protected endpoints.	Authenticated User, Library Manager, Administrator
4	**Deployment Infrastructure**	Docker containers, CI/CD pipelines, and servers used to build, deploy, and host the system.	
4.1	**Docker Containers**	Backend and related services (e.g., Keycloak, MySQL) are containerized for easy deployment and scalability.	Administrator
4.2	**CI/CD Pipeline**	Automates the testing, building, and deployment of the backend. Ensures smooth and secure deployment cycles.	Administrator
5	**Backups**	Encrypted backups of the MySQL database stored in multiple locations (two online, one offline) to ensure data recovery.	Administrator

Data Flow Diagrams

General DFD

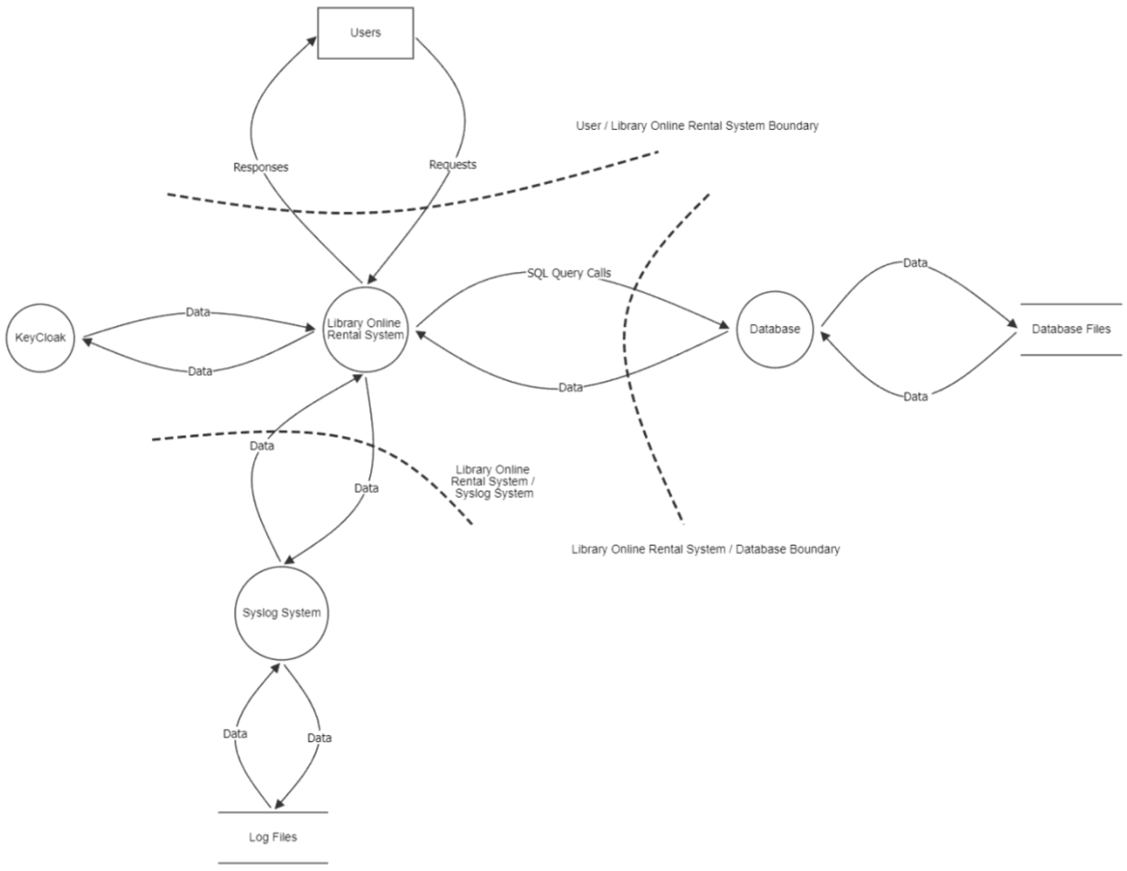

Users
Responses
Requests
User / Library Online Rental System Boundary
KeyCloak
Data
Data
Library Online Rental System
SQL Query Calls
Data
Database
Data
Data
Database Files
Data
Data
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 6 - General DFD

User – Creating Account

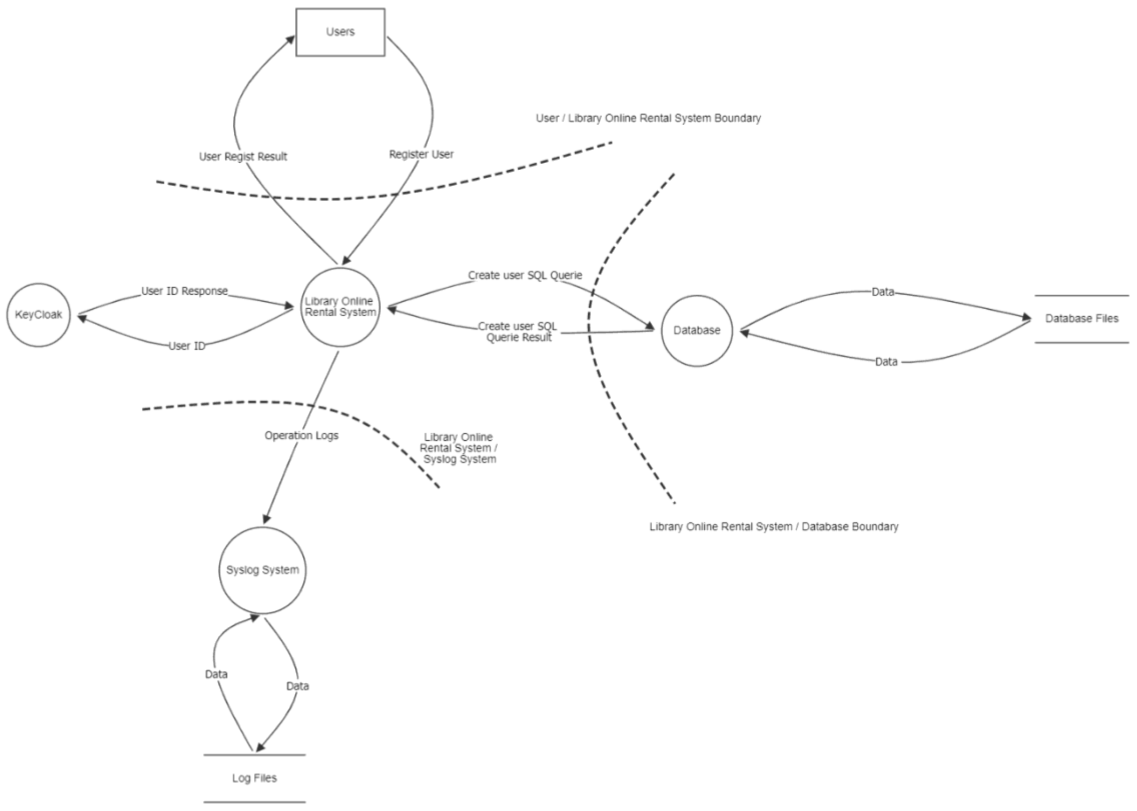

Users
User Regist Result
Register User
User / Library Online Rental System Boundary
KeyCloak
User ID Response
User ID
Library Online Rental System
Create user SQL Querie
Create user SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 7 - DFD - Creation of the user account

User - Updating Profile

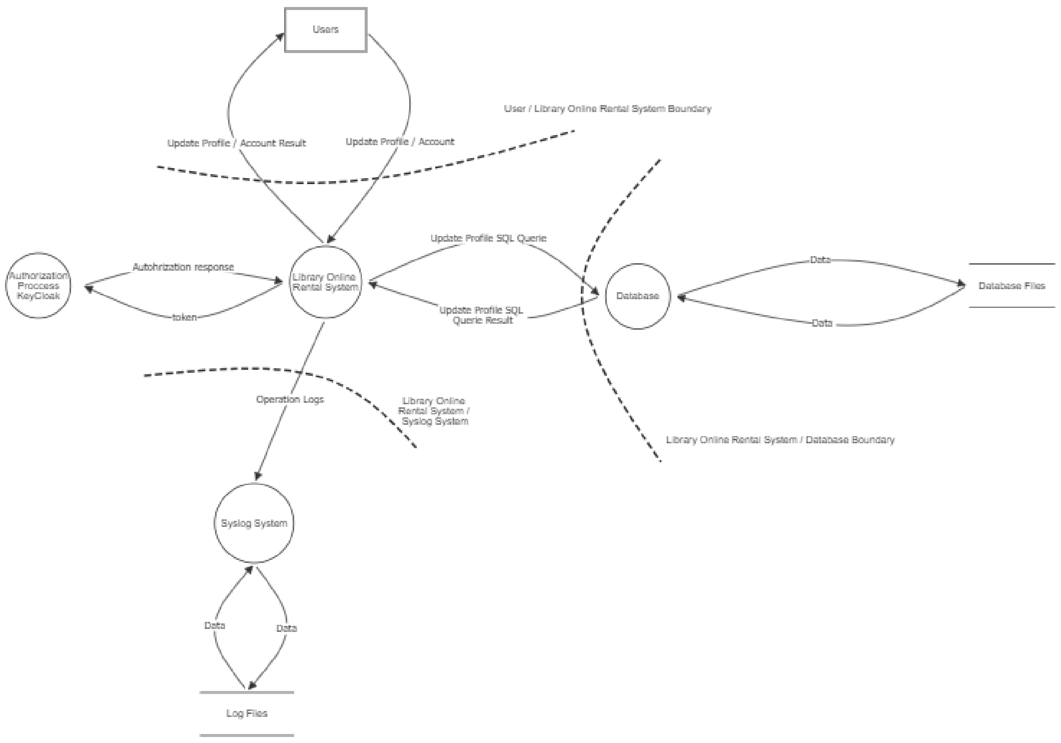

Users
User / Library Online Rental System Boundary
Update Profile / Account Result
Update Profile / Account
Update Profile SQL Querie
Autohrization response
Authorization Proccess KeyCloak
token
Library Online Rental System
Update Profile SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 8 - DFD - Update of the user profile

User – Delete Account

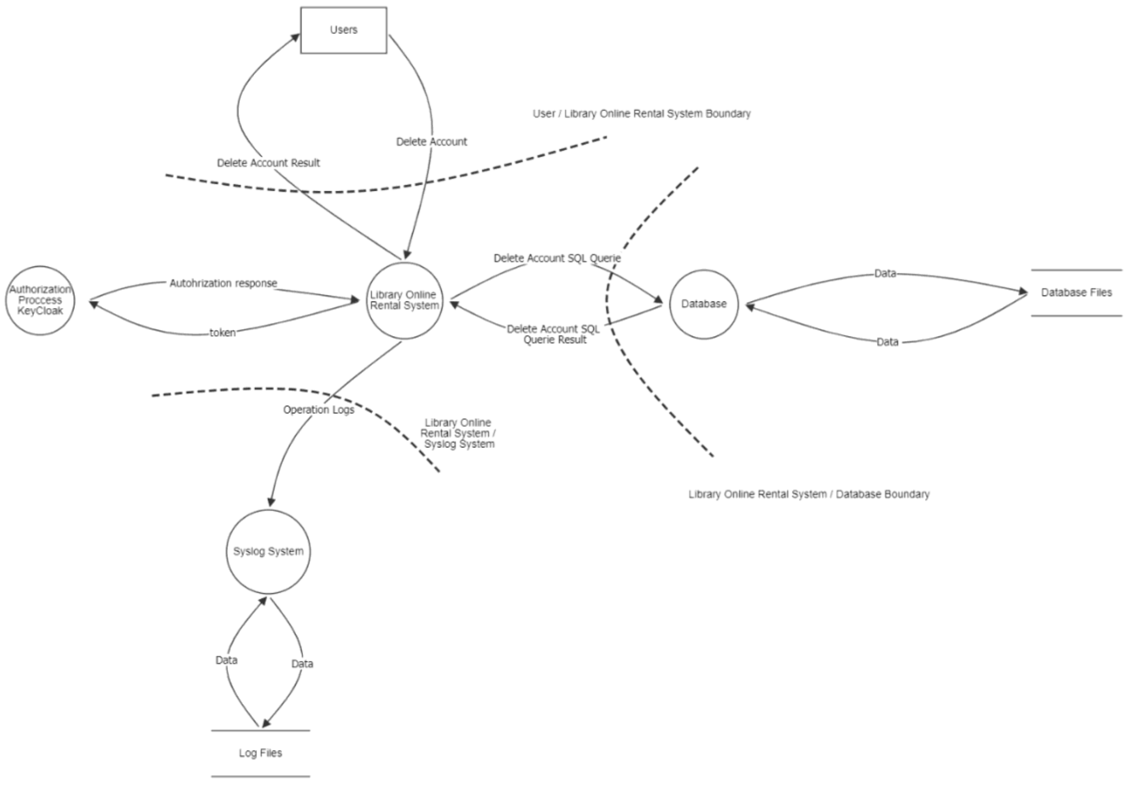
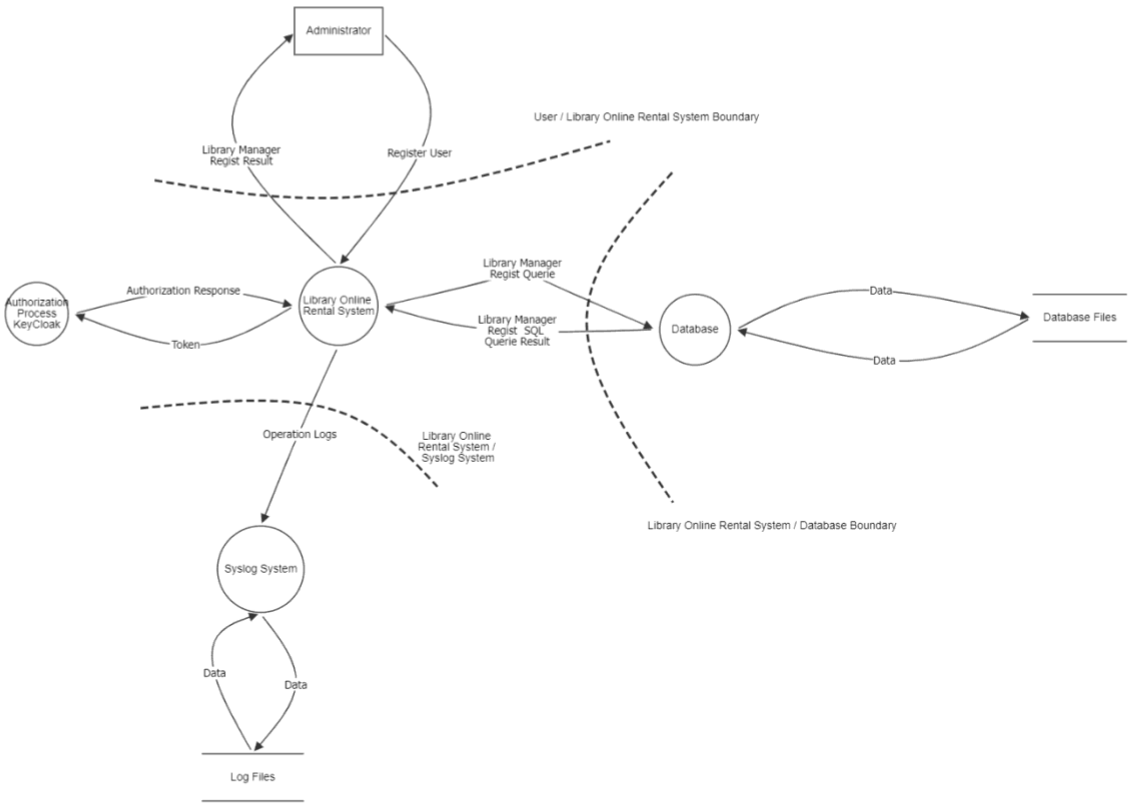

Users
User / Library Online Rental System Boundary
Delete Account
Delete Account Result
Authorization Proccess KeyCloak
Autohrization response
token
Library Online Rental System
Delete Account SQL Querie
Delete Account SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 9 - DFD - Deleting the user account

Administrator - create library manager account



Administrator
Library Manager Regist Result
Register User
User / Library Online Rental System Boundary
Authorization Process KeyCloak
Authorization Response
Token
Library Online Rental System
Library Manager Regist Querie
Library Manager Regist SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 10 - DFD - Creation of the Library Manager Account

Library Manager - create book

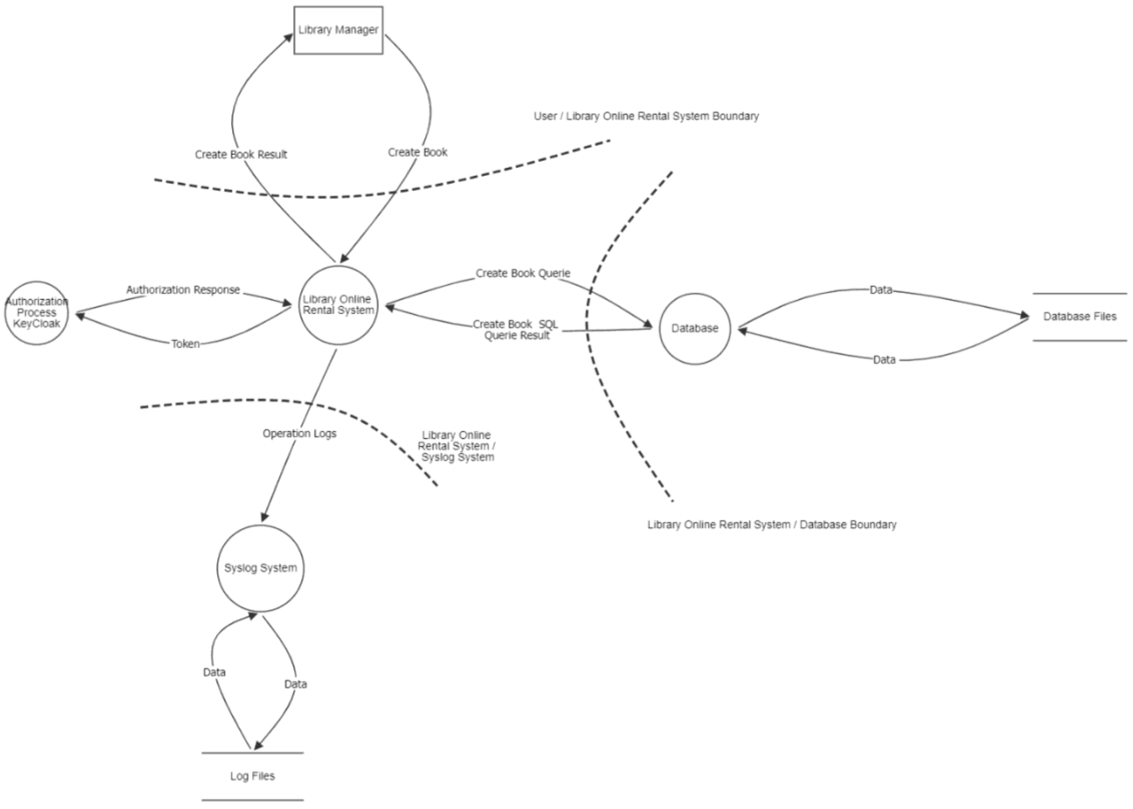
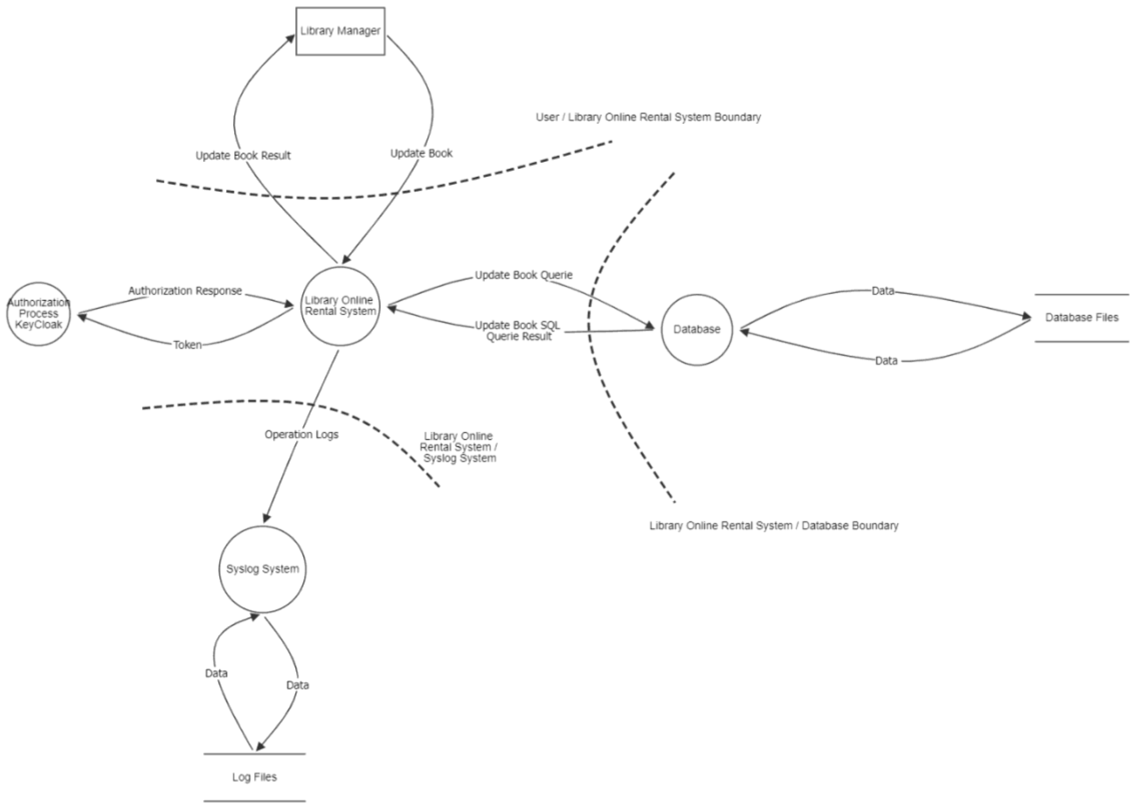

Library Manager
Create Book Result
Create Book
User / Library Online Rental System Boundary
Authorization Process KeyCloak
Authorization Response
Token
Library Online Rental System
Create Book Querie
Create Book SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 11 - DFD Create Book

Library Manager - update book



Library Manager
Update Book Result
Update Book
User / Library Online Rental System Boundary
Authorization Process KeyCloak
Authorization Response
Token
Library Online Rental System
Update Book Querie
Update Book SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 12 - DFD - Update Book

Library Manager - delete book

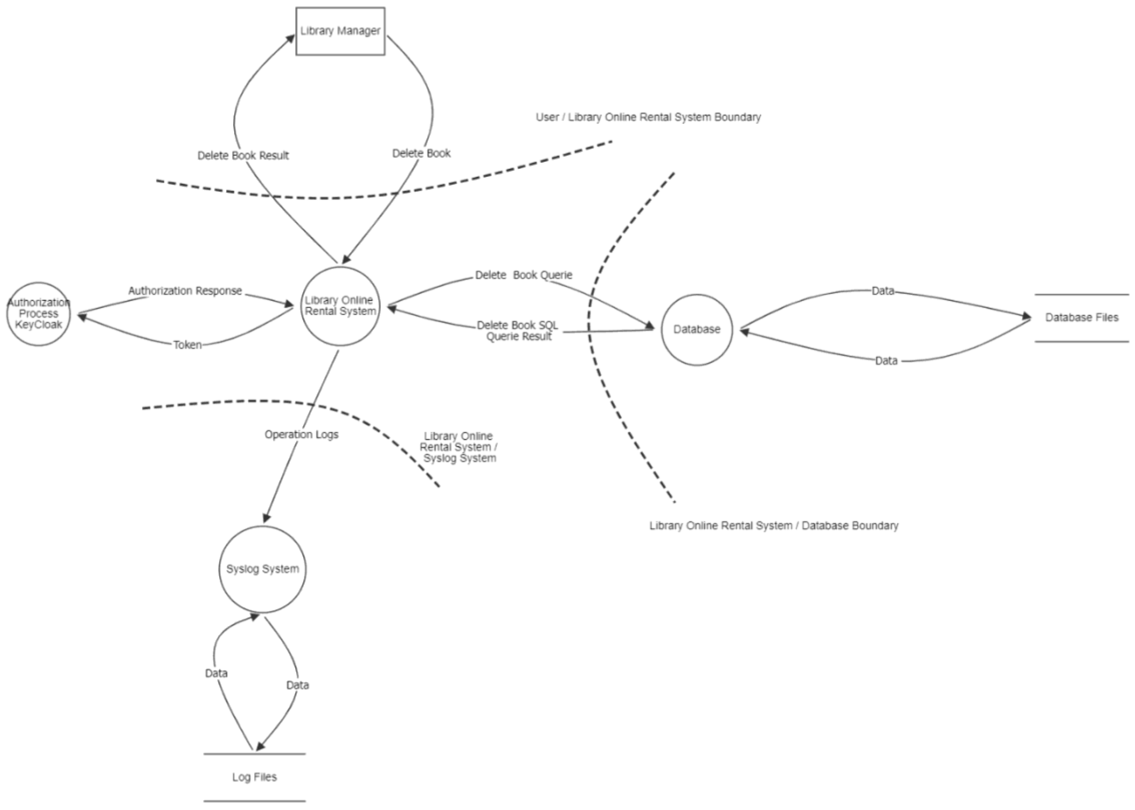

Library Manager
Delete Book Result
Delete Book
User / Library Online Rental System Boundary
Authorization Process KeyCloak
Authorization Response
Token
Library Online Rental System
Delete Book Querie
Delete Book SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 13 - DFD - Delete Book

Library Manager – manage rentals

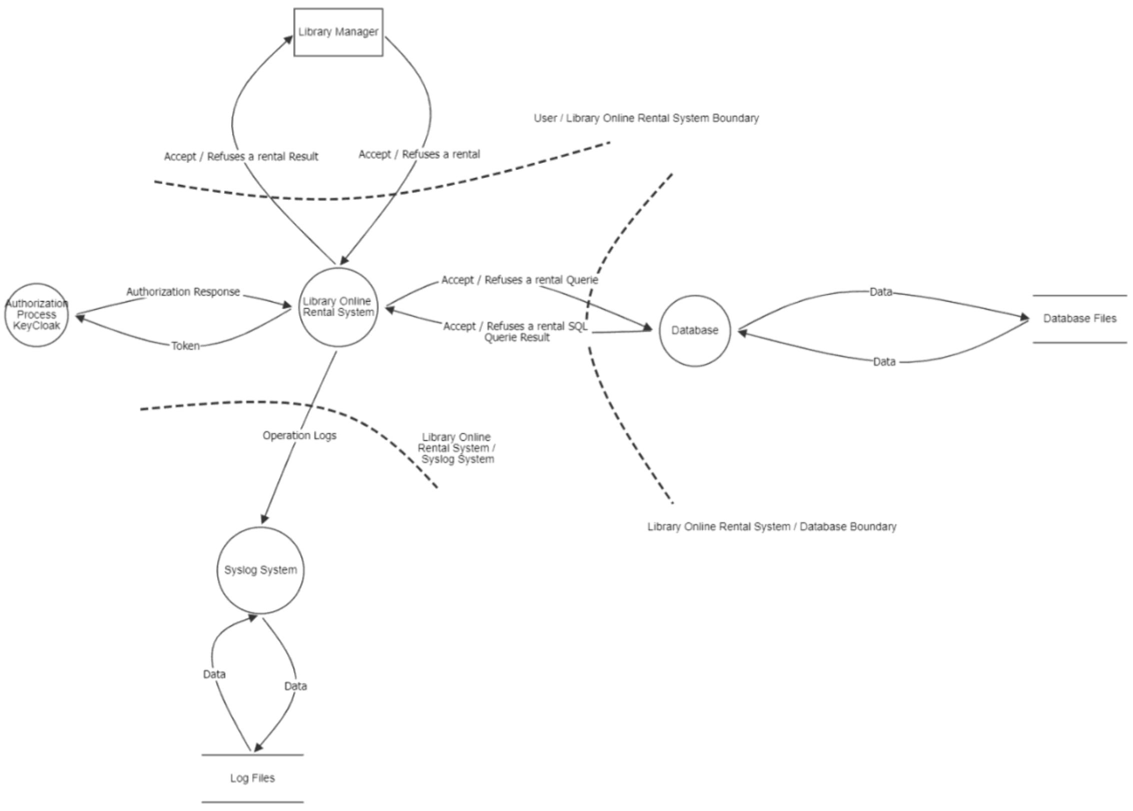
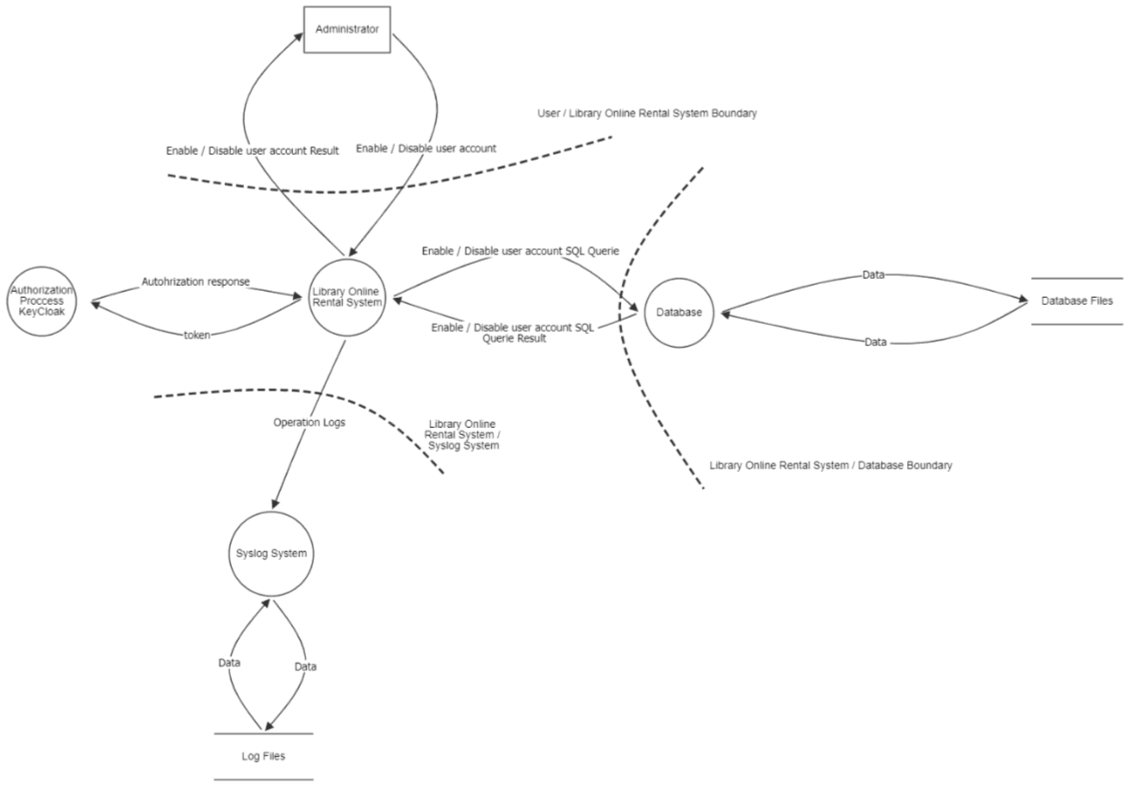

Library Manager
Accept / Refuses a rental Result
Accept / Refuses a rental
User / Library Online Rental System Boundary
Authorization Process KeyCloak
Authorization Response
Token
Library Online Rental System
Accept / Refuses a rental Querie
Accept / Refuses a rental SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 14 - DFD - Managing Rentals

Administrator – activate or disable user accounts



Administrator
User / Library Online Rental System Boundary
Enable / Disable user account Result
Enable / Disable user account
Enable / Disable user account SQL Querie
Authorization Proccess KeyCloak
Autohrization response
token
Library Online Rental System
Enable / Disable user account SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 15 - DFD - Activate or Deactivate Users Accounts

User – search books

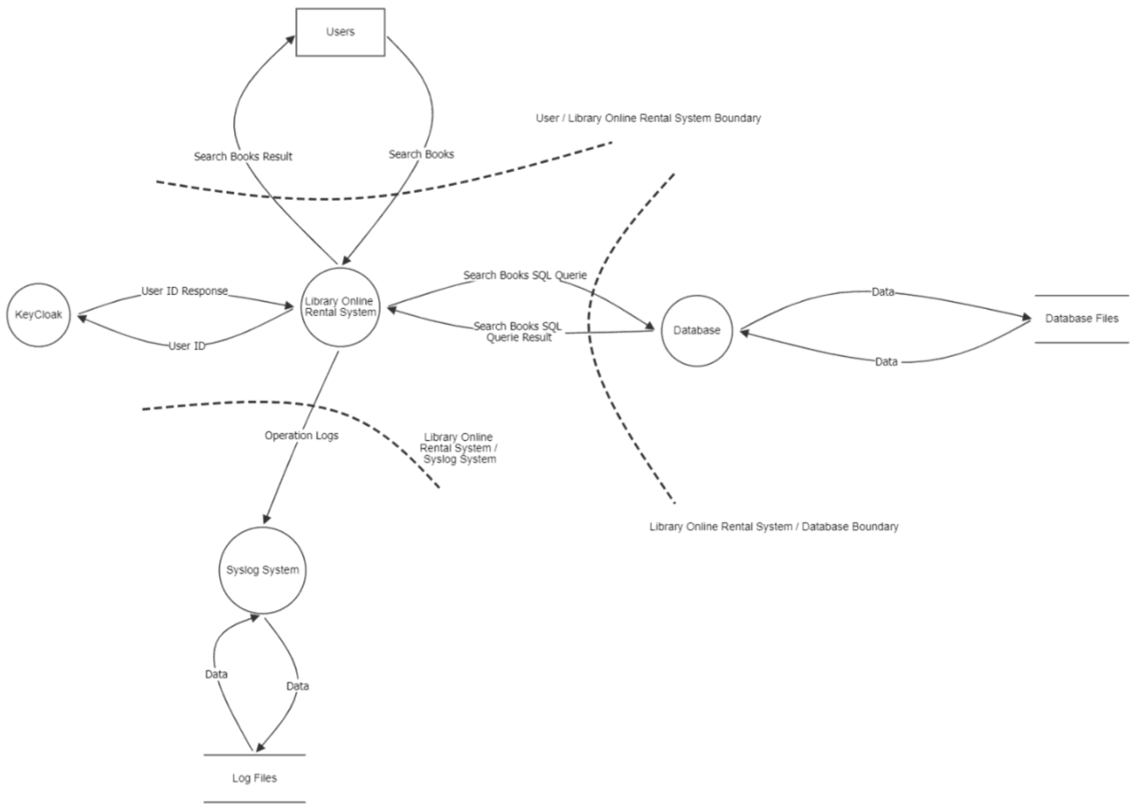

Users
User / Library Online Rental System Boundary
Search Books Result
Search Books
KeyCloak
User ID Response
User ID
Library Online Rental System
Search Books SQL Querie
Search Books SQL Querie Result
Database
Data
Data
Database Files
Library Online Rental System / Database Boundary
Operation Logs
Library Online Rental System / Syslog System
Syslog System
Data
Data
Log Files

Figure 16 - DFD - Searching for Books

User – suggest new books insertion

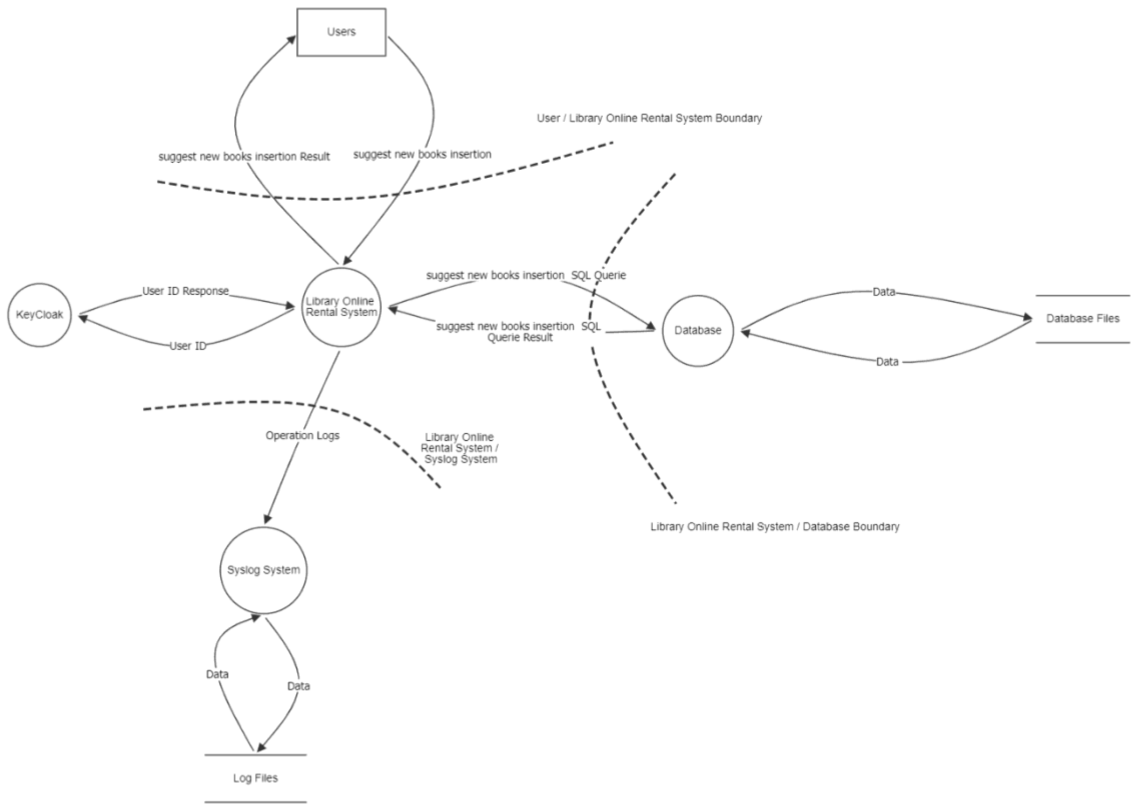
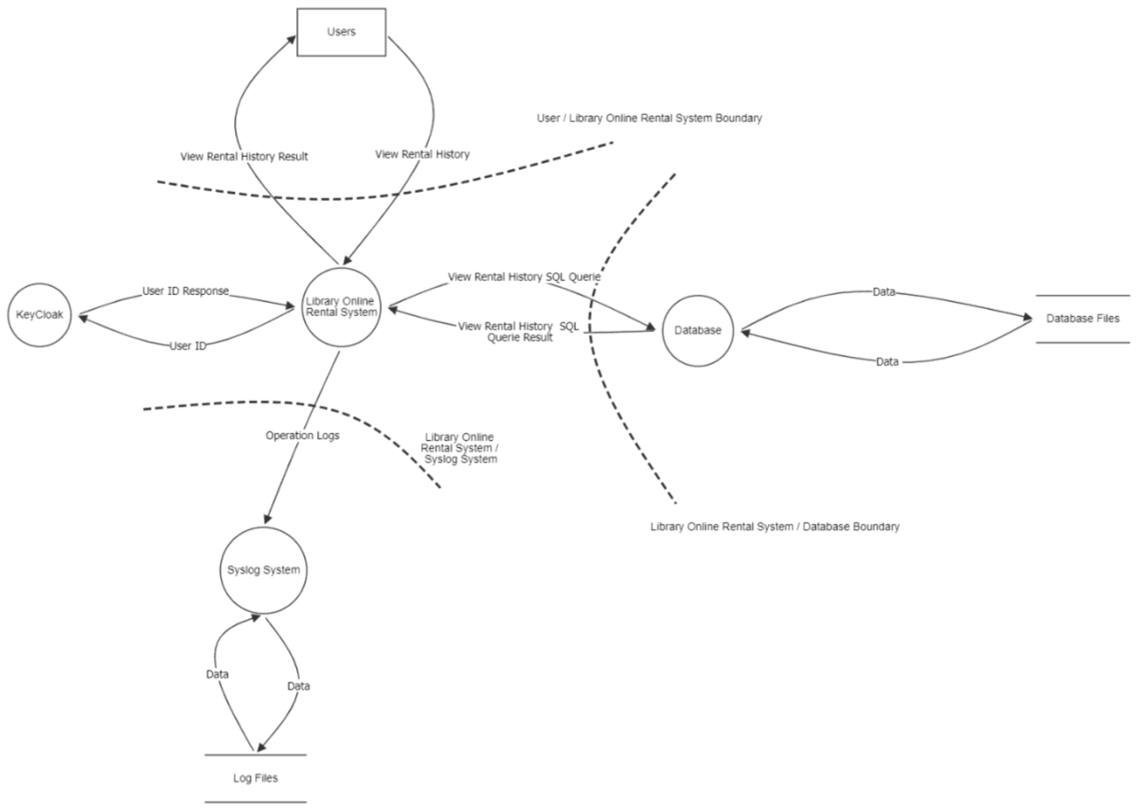

Users
suggest new books insertion Result
suggest new books insertion
User / Library Online Rental System Boundary
KeyCloak
User ID Response
User ID
Library Online Rental System
suggest new books insertion SQL Querie
suggest new books insertion SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 17 - DFD - Suggest new Books insertion

User – View Rental History



Users
View Rental History Result
View Rental History
User / Library Online Rental System Boundary
KeyCloak
User ID Response
User ID
Library Online Rental System
View Rental History SQL Querie
View Rental History SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 18 - DFD - View Rental History

User – Rating Books

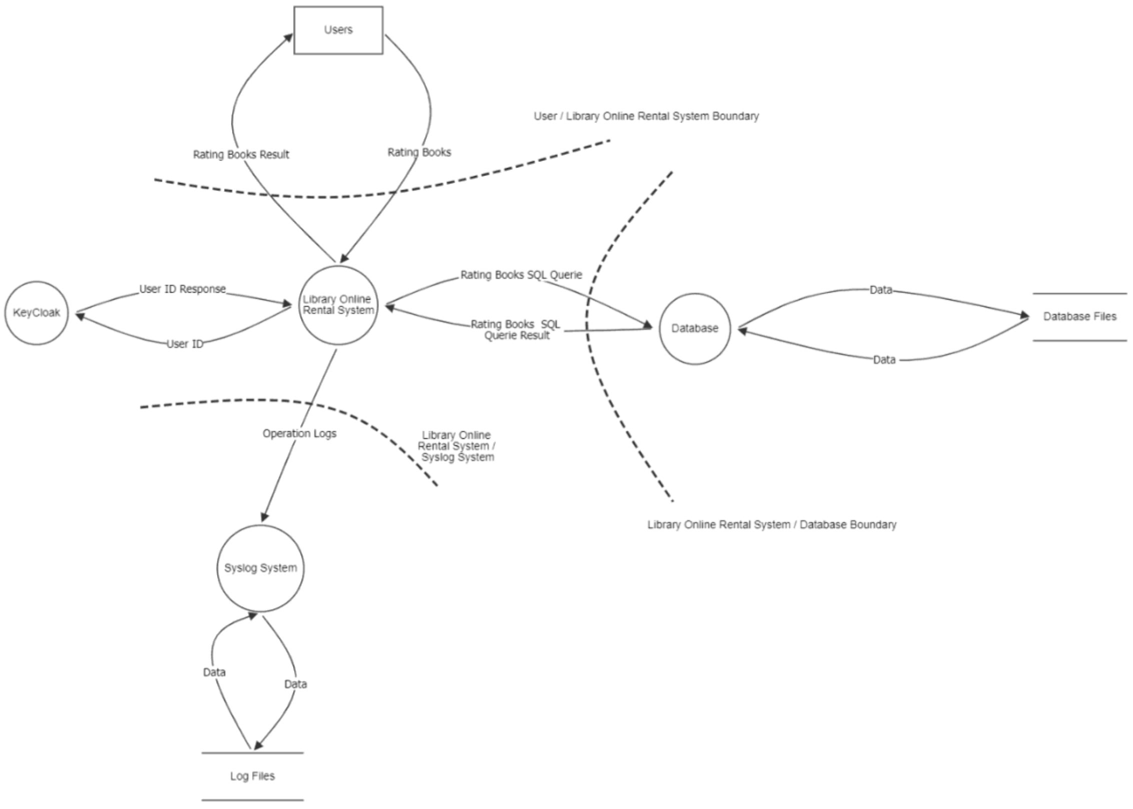

Users
Rating Books Result
Rating Books
User / Library Online Rental System Boundary
KeyCloak
User ID Response
User ID
Library Online Rental System
Rating Books SQL Querie
Rating Books SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 19 - DFD - Rating Books

User – Rent a Book

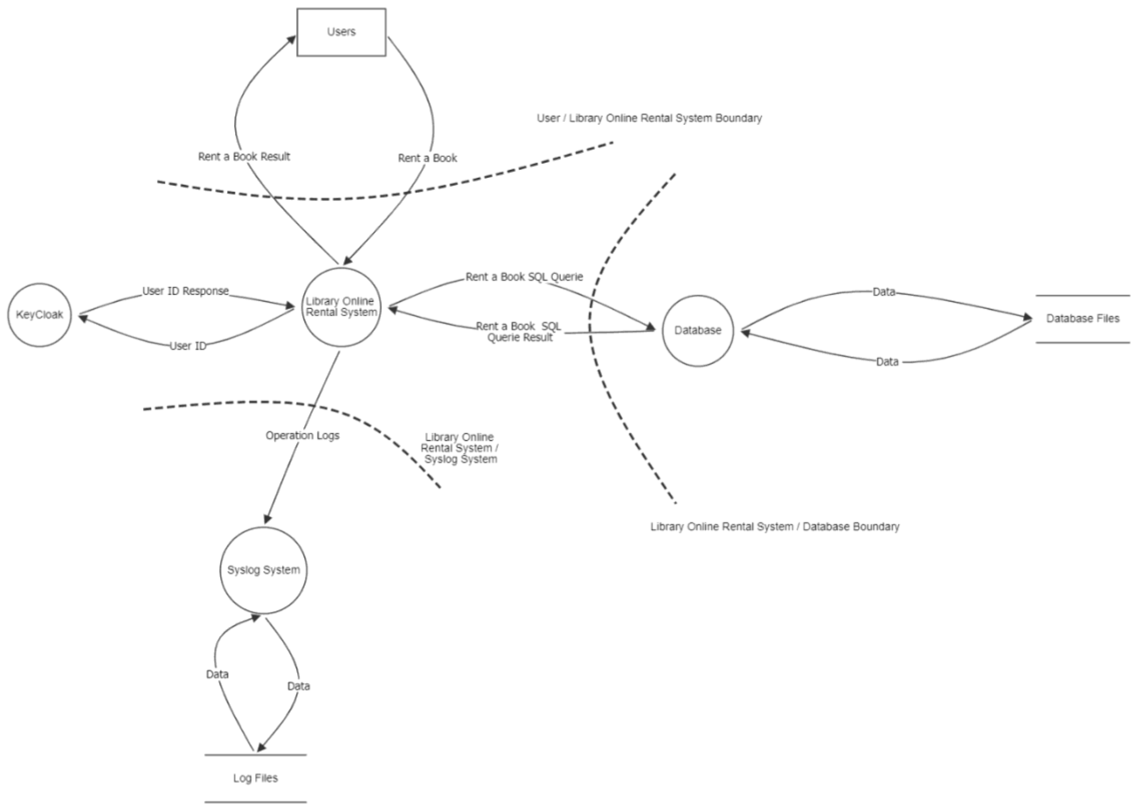

Users
Rent a Book Result
Rent a Book
User / Library Online Rental System Boundary
KeyCloak
User ID Response
User ID
Library Online Rental System
Rent a Book SQL Querie
Rent a Book SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 20 - DFD - Rent a Book

Rental Expiration Notification

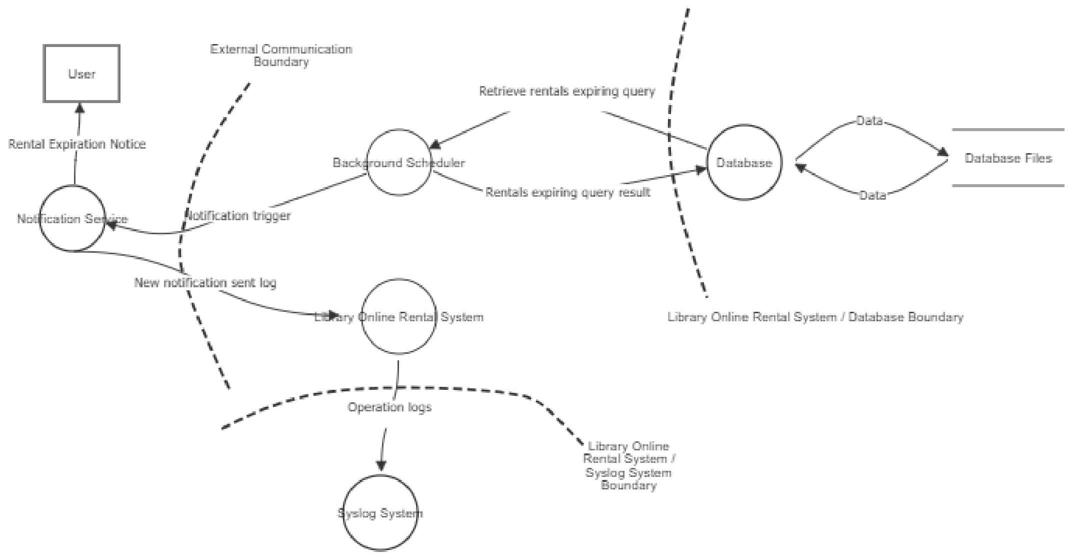

User
External Communication Boundary
Retrieve rentals expiring query
Rental Expiration Notice
Background Scheduler
Database
Data
Data
Database Files
Rentals expiring query result
Notification Service
Notification trigger
New notification sent log
Library Online Rental System
Library Online Rental System / Database Boundary
Operation logs
Library Online Rental System / Syslog System Boundary
Syslog System

Figure 21 - DFD - Rental Expiration Notification

Library Manager – Update Stocks

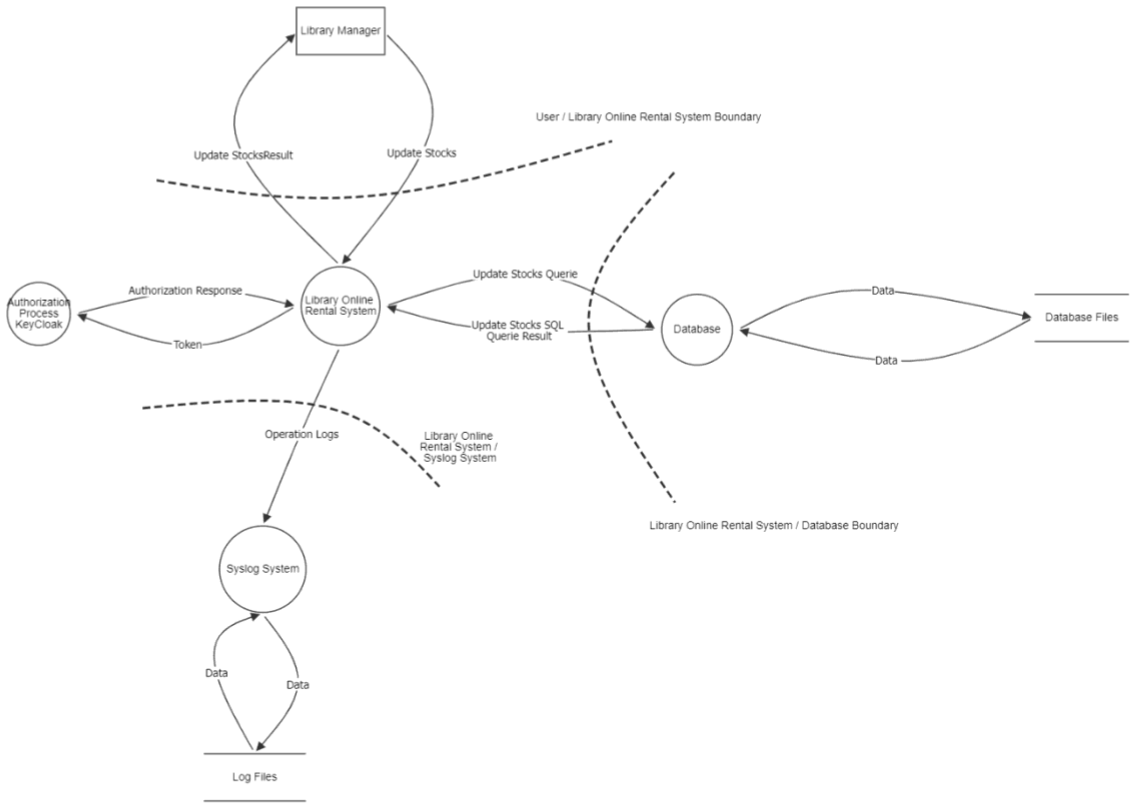
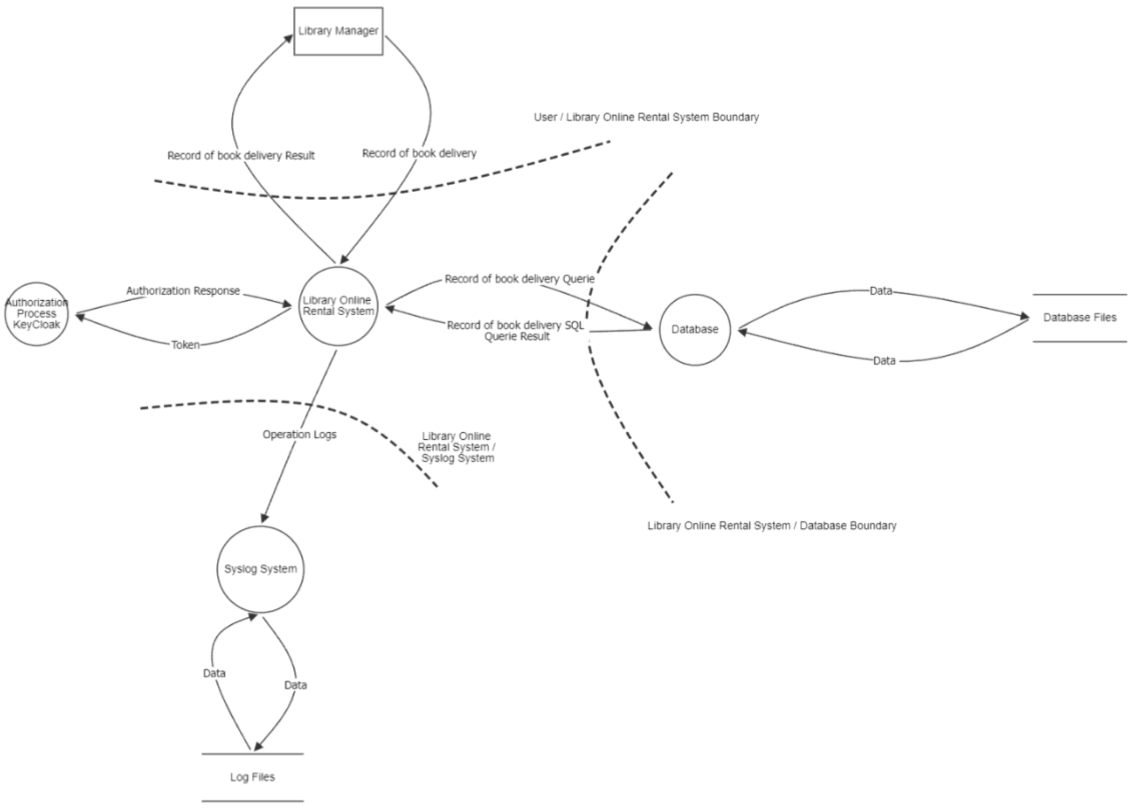

Library Manager
Update StocksResult
Update Stocks
User / Library Online Rental System Boundary
Authorization Process KeyCloak
Authorization Response
Token
Library Online Rental System
Update Stocks Querie
Update Stocks SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figure 22 - DFD – Update Stocks

Library Manager – Register Received Books



Library Manager
User / Library Online Rental System Boundary
Record of book delivery Result
Record of book delivery
Authorization Process KeyCloak
Authorization Response
Token
Library Online Rental System
Record of book delivery Querie
Record of book delivery SQL Querie Result
Database
Data
Data
Database Files
Operation Logs
Library Online Rental System / Syslog System
Library Online Rental System / Database Boundary
Syslog System
Data
Data
Log Files

Figura 1 - DFD - Library Manager – Register Received Books

Determine and Rank Threats

Categorization

Threat classification is essential for understanding and categorizing the risks that the application faces. Using the STRIDE method, it is feasible to recognize a wide variety of threats and thus classify them.

Table 6 - Stride Explanation

STRIDE	Property Violated	Identified Threats
Spoofing	**Authentication**	Pretending to be something or someone other than yourself.
Tampering	**Integrity**	Modify something on disk, network, memory, or anywhere else.
Repudiation	**Non-repudiation**	Claiming that you did not do something or that we are not responsible. It may or may not be true.
Information Disclosure	**Confidentiality**	Provide information to someone not authorized to access it.
Denial of Service	**Availability**	Flood the system with requests, exhausting resources.
Elevation of Privilege	**Authorization**	Allowing someone to do something that they are not allowed to do.

Analysis

Analysis is the identification of threats and involves analyzing each aspect of the application.

Stride

STRIDE is applied to each DFD so that it is possible to analyse each element.

User - create account

Table 7 - STRIDE: User - create account

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1 : An attacker may attempt to register a fake identity, such as using an email that resembles an admin or library manager account.
Tampering	Tampering Threat 1: Malicious users could tamper with form data (e.g., injecting a role value) to escalate their privileges.
Repudiation	Repudiation Threat 1: A user could deny having registered if there is no log of the action.
Information Disclosure	Information Disclosure Threat 1: System messages (e.g., “email already in use”) might leak internal information.
Denial of Service	Denial of Service Threat 1: Attackers could flood the system with account creation requests, exhausting resources.
Elevation of Privilege	Elevation of Privilege Threat 1: A user might try to self-assign higher privileges during account creation.

User - Updating Profile

Table 8 - STRIDE - User - Updating Profile

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: An attacker could try to reuse or forge a valid token to impersonate another user and edit their profile.
Tampering	Tampering Threat 1: A user might manipulate the form data (e.g., change email, phone number or inject code in names).
Repudiation	Repudiation Threat 1: A user might deny having updated their profile, especially if the data is sensitive (e.g., change of MFA or email).

Information Disclosure	Information Disclosure Threat 1: Responses or error messages could unintentionally reveal internal logic (e.g., whether a field is already in use or why an update failed).
Denial of Service	Denial of Service Threat 1: An attacker could search butood the endpoint with repeated update requests, overloading the system.
Elevation of Privilege	Elevation of Privilege Threaother user could try to update fields they shouldn't have access to (e.g., their own role or user status).

User – Delete Account

Table 9 - STRIDE: User – Delete Account

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: An attacker could forge or steal a valid token to impersonate another user and delete their account.
Tampering	Tampering Threat 1: Manipulating the request payload could redirect the deletion to another user's account.
Repudiation	Repudiation Threat 1: A user could deny that they requested account deletion.
Information Disclosure	Information Disclosure Threat 1: Deletion result or confirmation messages may reveal sensitive system info (e.g., "account X does not exist").
Denial of Service	Denial of Service Threat 1: Attackers could flood the deletion endpoint with requests, possibly resulting in service degradation or unintended deletions if improperly validated.
Elevation of Privilege	Elevation of Privilege Threat 1: A low-privilege user might attempt to delete accounts beyond their access level (e.g., admins).

Administrator - create library manager account

Table 10 - STRIDE: Administrator - create library manager account

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: An attacker could try to spoof admin credentials or hijack a session to register new library managers.
Tampering	Tampering Threat 1: Admin input could be manipulated in transit if not protected (e.g., register user with unintended role).
Repudiation	Repudiation Threat 1: An administrator might deny that they created a new library manager account.
Information Disclosure	Information Disclosure Threat 1: Error responses (e.g., "username already exists") might leak details about registered users.
Denial of Service	Denial of Service Threat 1: A malicious admin could intentionally overload the system by spamming account creation.
Elevation of Privilege	Elevation of Privilege Threat 1: An admin could attempt to create a user with unauthorised access level (e.g., another admin or system-level role).

Library Manager - create book

Table 11 - STRIDE: Library Manager - create book

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: A non-authorised user might forge a token or escalate privileges to impersonate a library manager.
Tampering	Tampering Threat 1: Malicious input could modify book metadata (e.g., script injection in title/author fields).
Repudiation	Repudiation Threat 1: A manager could deny creating or modifying a book entry.
Information Disclosure	Information Disclosure Threat 1: Server messages might disclose DB schema or validation logic (e.g., “book already exists”).
Denial of Service	Denial of Service Threat 1: Attackers could spam book creation requests (e.g., automated bots filling the catalogue).
Elevation of Privilege	*No relevant threats identified.* Book creation is properly scoped to the Library Manager role, and there are no admin-restricted books in the system.

Library Manager - update book

Table 12 - STRIDE: Library Manager - update book

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: A malicious user could attempt to spoof a library manager to gain access to update book records.
Tampering	Tampering Threat 1: Payload manipulation could result in unintended changes to critical fields (e.g., pricing, book classification).
Repudiation	Repudiation Threat 1: A manager could deny making changes if logs are incomplete.
Information Disclosure	Information Disclosure Threat 1: Update error responses could leak DB schema or validation logic (e.g., "book ID not found").
Denial of Service	Denial of Service Threat 1: A malicious manager could send repeated updates to overload the system or cause race conditions.
Elevation of Privilege	*No relevant threats identified.* Book management is properly scoped to the Library Manager role, and there are no admin-restricted books in the system.

Library Manager - delete book

Table 13 - STRIDE: Library Manager - delete book

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: An attacker might impersonate a library manager to delete important books.
Tampering	Tampering Threat 1: The request payload could be altered to delete unintended records or bypass checks.
Repudiation	Repudiation Threat 1: A manager could deny having deleted a book.
Information Disclosure	Information Disclosure Threat 1: Responses might reveal internal logic (e.g., "book ID not found" or "already deleted").

Denial of Service	Denial of Service Threat 1: Repeated deletion requests (even if invalid) could overload DB or disrupt the system.
Elevation of Privilege	*No relevant threats identified.*

Library Manager – manage rentals

Table 14 - STRIDE : Library Manager – manage rentals

STRIDE	**Identified Threats**
Spoofing	Spoofing **Threat 1:** A malicious user might spoof a library manager and approve/deny rentals of other users.
Tampering	Tampering **Threat 1:** Interception or manipulation of the request could alter the target rental (e.g., modify user ID or book ID).
Repudiation	Repudiation **Threat 1:** A manager could later deny having accepted or refused a rental.
Information Disclosure	Information Disclosure **Threat 1:** Responses may reveal sensitive user data (e.g., who has rented a book or declined a rental).
Denial of Service	Denial of Service **Threat 1:** A malicious manager or attacker could flood the system with decision requests, overloading DB or locking records.
Elevation of Privilege	*No relevant threats identified.* The action of approving or rejecting rentals is limited to the Library Manager role and does not escalate privileges beyond their authorized scope.

Administrator – activate or disable user accounts

Table 15 - STRIDE: Administrator – activate or disable user accounts

STRIDE	**Identified Threats**
Spoofing	Spoofing **Threat 1:** An attacker could impersonate an administrator and disable user accounts.
Tampering	Tampering **Threat 1:** Request manipulation could disable the wrong account (e.g., change user ID in payload).

Repudiation	Repudiation **Threat 1:** An admin could deny having disabled or re-enabled a user account.
Information Disclosure	Information Disclosure **Threat 1:** Error messages might expose account status or internal logic (e.g., “account already disabled”).
Denial of Service	Denial of Service **Threat 1:** Repeated requests could disable many users or affect service availability.
Elevation of Privilege	Elevation of Privilege Threat 1: An administrator might try to affect accounts outside their scope (e.g., other admins or system accounts).

User – search books

Table 16 - STRIDE: User – search books

STRIDE	**Identified Threats**
Spoofing	Spoofing Threat 1: *No relevant threats identified.* This is a read-only action and typically does not modify resources or access control.
Tampering	Tampering **Threat 1:** An attacker might manipulate search parameters to inject SQL or bypass filters.
Repudiation	Repudiation **Threat 1:** A user could deny having performed certain searches (e.g., for sensitive content).
Information Disclosure	Information Disclosure **Threat 1:** Search results might return restricted or sensitive books (e.g., hidden collections, admin-only items).
Denial of Service	Denial of Service **Threat 1:** Attackers could automate massive search queries to overload the system.
Elevation of Privilege	*No relevant threats identified.* Users cannot escalate privileges by using the search feature.

User – suggest new books insertion

Table 17 - STRIDE: User – suggest new books insertion

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: *No relevant threats identified*. Users are suggesting not altering protected data, and no privilege-sensitive action is triggered.
Tampering	Tampering Threat 1: Users might submit invalid or malicious input (e.g., scripts or SQL) in suggestion fields.
Repudiation	Repudiation Threat 1: A user might deny having submitted a suggestion, especially if it contains offensive or manipulated content.
Information Disclosure	Information Disclosure Threat 1: System messages could unintentionally reveal data like suggestion table structure or internal validation mechanisms.
Denial of Service	Denial of Service Threat 1: A malicious user could flood the system with a high volume of suggestions to overload storage or disrupt moderation flows.
Elevation of Privilege	*No relevant threats identified*. The user cannot change roles or gain access through this functionality.

User – View Rental History

Table 18 - STRIDE: User – View Rental History

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: A user might try to access another user's rental history by manipulating session tokens.
Tampering	Tampering Threat 1: *No relevant threats identified*. The user is only retrieving data, not altering it.

Repudiation	Repudiation Threat 1: *No relevant threats identified*. Since this is a read-only action, and no user-generated content is involved, denial of access is less relevant.
Information Disclosure	Information Disclosure Threat 1: Rental history might contain personal or sensitive data (e.g., user preferences, book topics).
Denial of Service	Denial of Service Threat 1: A user could attempt to flood the system with frequent requests to slow down or crash the service.
Elevation of Privilege	*No relevant threats identified*. Users cannot escalate privileges via this process.

User – Rating Books

Table 19 - STRIDE: User – Rating Books

STRIDE	**Identified Threats**
Spoofing	Spoofing Threat 1: A user might try to spoof another user to submit ratings on their behalf.
Tampering	Tampering Threat 1: Users may tamper with the payload to rate books they haven't read or submit invalid values (e.g., 0–100 stars).
Repudiation	Repudiation Threat 1: A user may deny having rated a book, especially in cases of abuse (e.g., low ratings as retaliation).
Information Disclosure	Information Disclosure Threat 1: Returned data or error messages could reveal book statistics or internal logic.
Denial of Service	Denial of Service Threat 1: automated bots could mass-rate books to manipulate reputation or overcharge the system.
Elevation of Privilege	*No relevant threats identified*. Book rating does not give access to restricted data or roles.

User – Rent a Book

Table 20 - STRIDE: User – Rent a Book

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: An attacker could impersonate another user and rent books using their identity.
Tampering	Tampering Threat 1: A user may manipulate request data (e.g., to rent restricted books, alter book IDs or rental duration).
Repudiation	Repudiation Threat 1: A user might deny renting a specific book, possibly avoiding penalties or disputes.
Information Disclosure	Information Disclosure Threat 1: Rental results might reveal internal book metadata or expose other users' rental actions.
Denial of Service	Denial of Service Threat 1: An attacker could spam rental requests to lock books or exhaust system/database resources.
Elevation of Privilege	No relevant threats identified.

Rental Expiration Notification

Table 21 - STRIDE: Rental Expiration Notification

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: An attacker could attempt to impersonate the notification service or user to redirect or intercept notices.
Tampering	Tampering Threat 1: Manipulation of the expiration query or data could cause false notifications or silence legitimate ones.
Repudiation	Repudiation Threat 1: The system might lack evidence that a notification was sent.
Information Disclosure	Information Disclosure Threat 1: If not properly secured, notifications could leak user or book data (e.g., email content or SMS).

Denial of Service	Denial of Service Threat 1: Attackers could overload the scheduler or notification system to delay or block reminders.
Elevation of Privilege	Elevation of Privilege Threat 1: *No relevant threats identified*. The scheduler does not grant access or modify roles; only fetches based on time logic.

Library Manager – Update Stocks

Table 22 - STRIDE: Library Manager – Update Stocks

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: A malicious user could impersonate a library manager to update stock values.
Tampering	Spoofing Threat 1 :The payload may be tampered with to inject incorrect values (e.g., inflate or reduce stock counts).
Repudiation	Repudiation Threat 1: A manager might deny having made specific stock changes, especially after errors or fraud.
Information Disclosure	Information Disclosure Threat 1: Stock metadata might be exposed se o sistema devolver mensagens detalhadas de erro ou confirmação.
Denial of Service	Denial of Service Threat 1: A user with bad intentions can make multiple updates with high frequency, overloading the system.
Elevation of Privilege	**No relevant threats identified.** Stock update operations are scoped to the Library Manager role only.

Library Manager – Register Received Books

Table 23 - Library Manager – Register Received Books

STRIDE	Identified Threats
Spoofing	Spoofing Threat 1: A malicious user could impersonate a library manager and falsely register book returns.
Tampering	Tampering Threat 1: An attacker could manipulate the request payload to falsely modify rental or stock data.
Repudiation	Repudiation Threat 1: A library manager might deny having recorded the return of a book.
Information Disclosure	Information Disclosure Threat 1: Responses might leak sensitive information about rentals or users.
Denial of Service	Denial of Service Threat 1: Attackers could flood the system with fake return registrations to disrupt operations.
Elevation of Privilege	No relevant threats identified.

Attack Trees

Denial of Service (DoS) – Registration Flooding

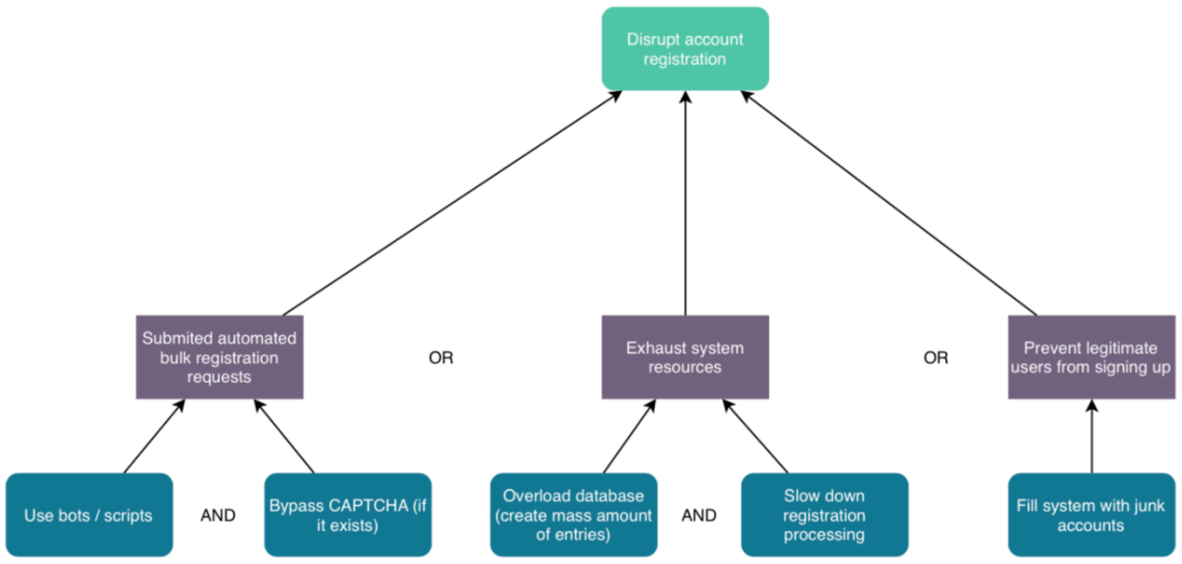

Disrupt account registration
Submited automated bulk registration requests
OR
Exhaust system resources
OR
Prevent legitimate users from signing up
Use bots / scripts
AND
Bypass CAPTCHA (if it exists)
Overload database (create mass amount of entries)
AND
Slow down registration processing
Fill system with junk accounts

Figure 23- Attack trees – Disrupt account registration

Brute Force - Unauthorized access to user accounts

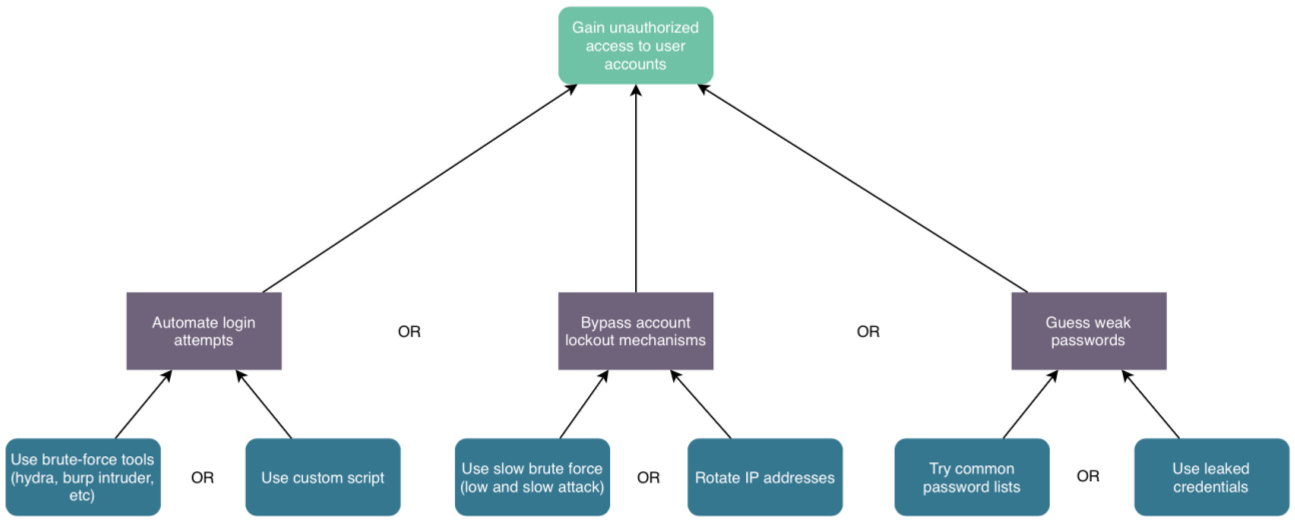

Gain unauthorized access to user accounts
Automate login attempts
OR
Bypass account lockout mechanisms
OR
Guess weak passwords
Use brute-force tools (hydra, burp intruder, etc)
OR
Use custom script
Use slow brute force (low and slow attack)
OR
Rotate IP addresses
Try common password lists
OR
Use leaked credentials

Figure 24 - Attack trees - Gain unauthorized access to user accounts

Broken Access Control – Authentication privilege Escalation

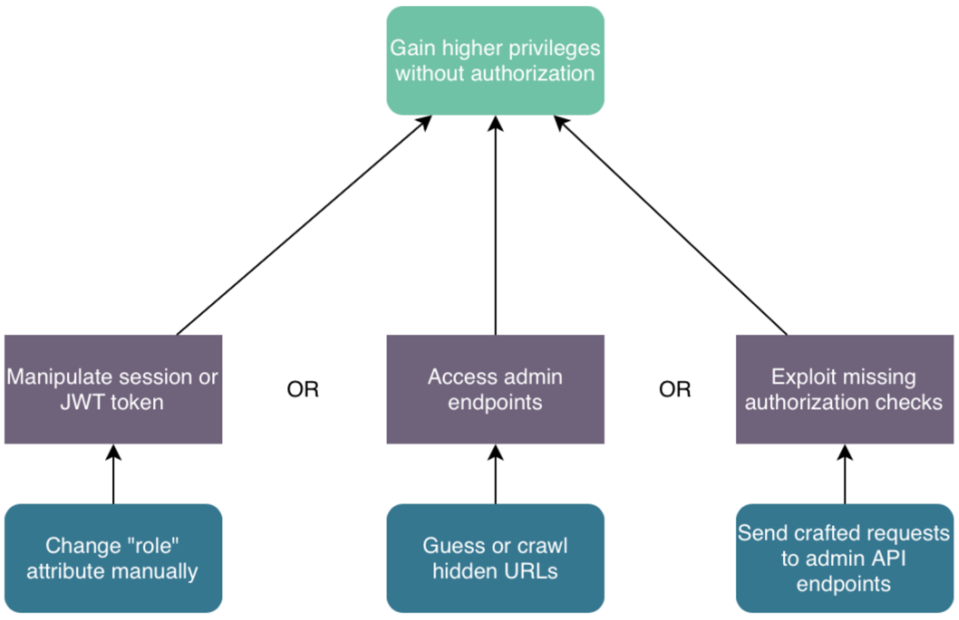

Gain higher privileges without authorization
Manipulate session or JWT token
OR
Access admin endpoints
OR
Exploit missing authorization checks
Change "role" attribute manually
Guess or crawl hidden URLs
Send crafted requests to admin API endpoints

Figure 25 - Attack trees - Gain higher privileges without authorization

Broken Access Control - Book Management

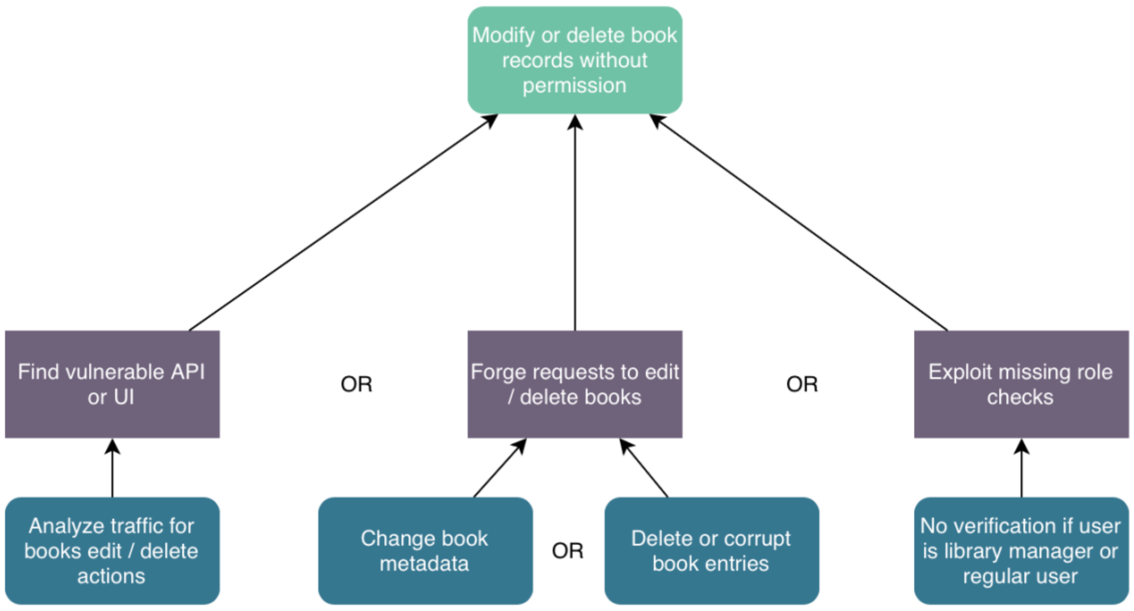

Modify or delete book records without permission
Find vulnerable API or UI
OR
Forge requests to edit / delete books
OR
Exploit missing role checks
Analyze traffic for books edit / delete actions
Change book metadata
OR
Delete or corrupt book entries
No verification if user is library manager or regular user

Figure 26 - Attack trees - Modify book records without permission

Insecure Direct Object References– Rental History Access

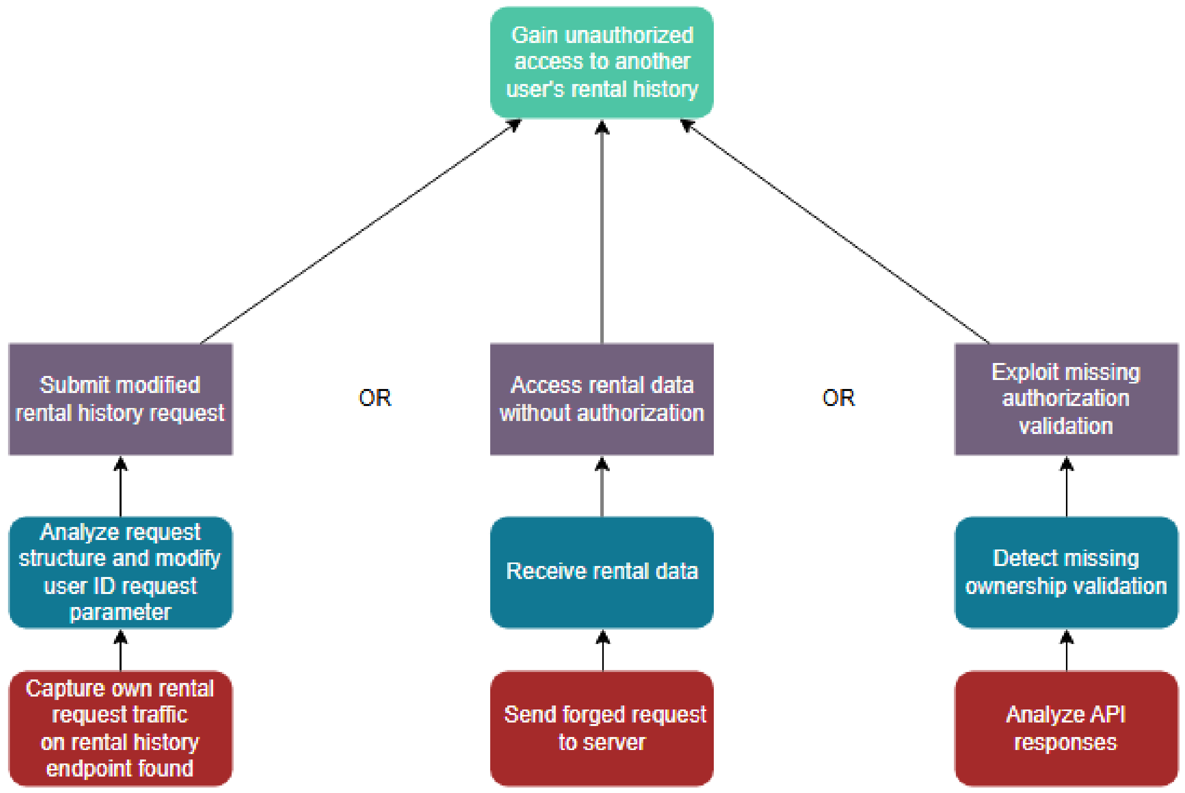

Gain unauthorized access to another user's rental history
Submit modified rental history request
OR
Access rental data without authorization
OR
Exploit missing authorization validation
Analyze request structure and modify user ID request parameter
Receive rental data
Detect missing ownership validation
Capture own rental request traffic on rental history endpoint found
Send forged request to server
Analyze API responses

Figure 27 - Attack trees - Unauthorized access to another user's rental history

Code Injection – Profile Update

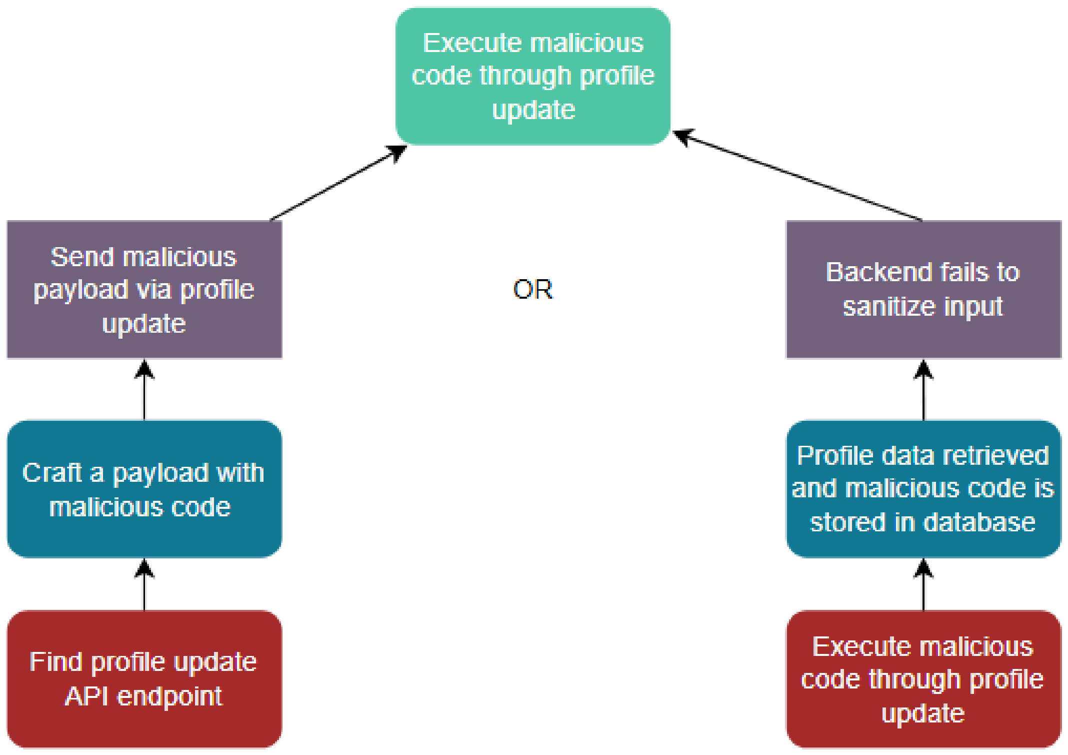

Execute malicious code through profile update
Send malicious payload via profile update
OR
Backend fails to sanitize input
Craft a payload with malicious code
Profile data retrieved and malicious code is stored in database
Find profile update API endpoint
Execute malicious code through profile update

Figure 28 - Attack trees - Execute malicious code through profile update

Broken Object Level Authorization – Modify user suggestion

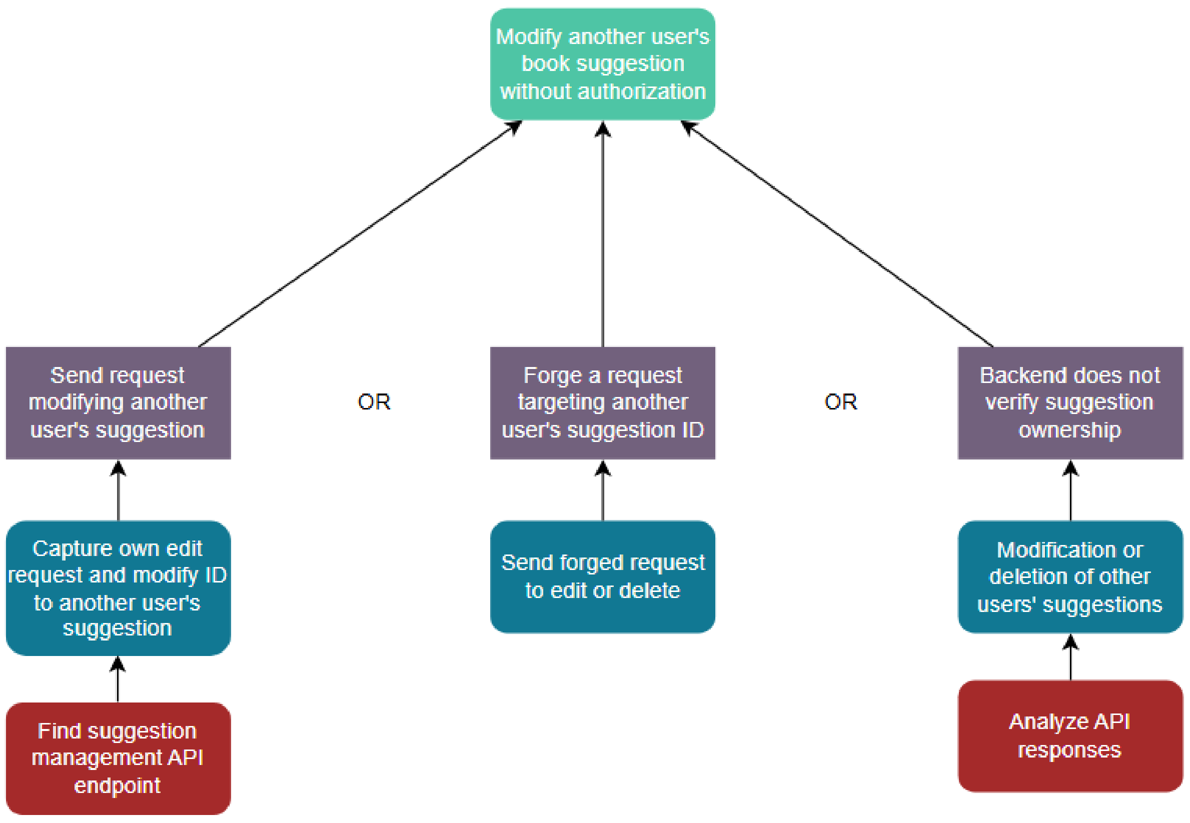

Modify another user's book suggestion without authorization
Send request modifying another user's suggestion
OR
Forge a request targeting another user's suggestion ID
OR
Backend does not verify suggestion ownership
Capture own edit request and modify ID to another user's suggestion
Send forged request to edit or delete
Modification or deletion of other users' suggestions
Find suggestion management API endpoint
Analyze API responses

Figure 29 - Attack trees - Modify another user's book suggestion without authorization

Mass Assignment – Profile Update

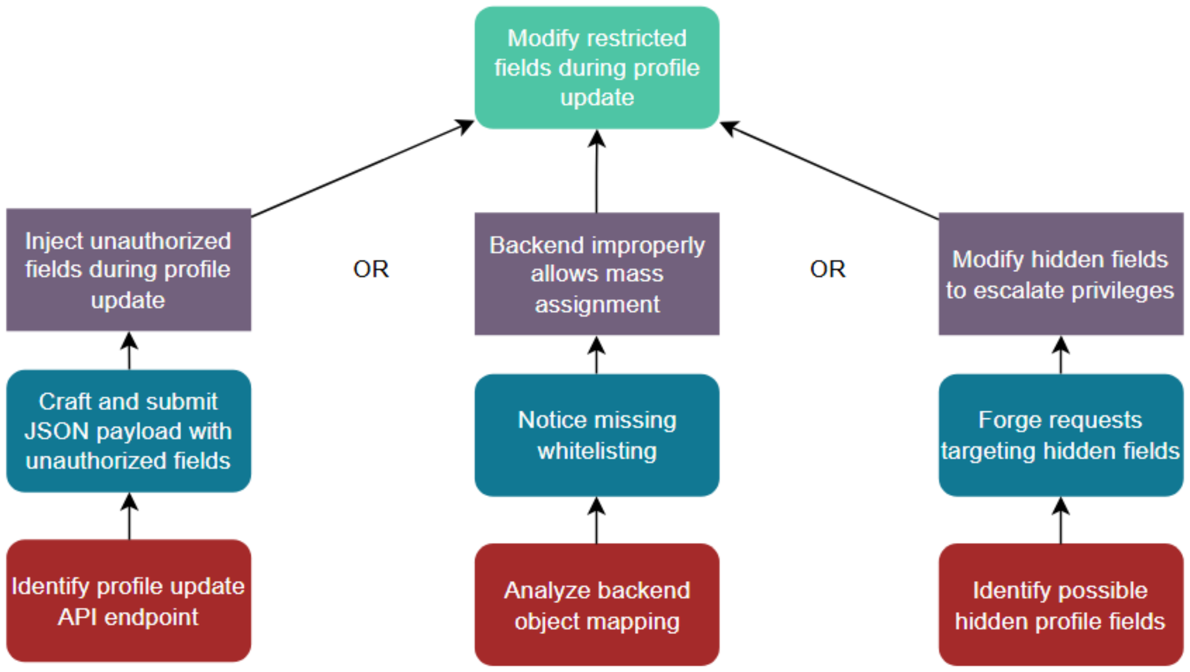

Modify restricted fields during profile update
Inject unauthorized fields during profile update
OR
Backend improperly allows mass assignment
OR
Modify hidden fields to escalate privileges
Craft and submit JSON payload with unauthorized fields
Notice missing whitelisting
Forge requests targeting hidden fields
Identify profile update API endpoint
Analyze backend object mapping
Identify possible hidden profile fields

Figure 30 - Attack trees - Modify restricted fields during profile update

Abuse Cases

Denial of Service (DoS) – Registration Flooding

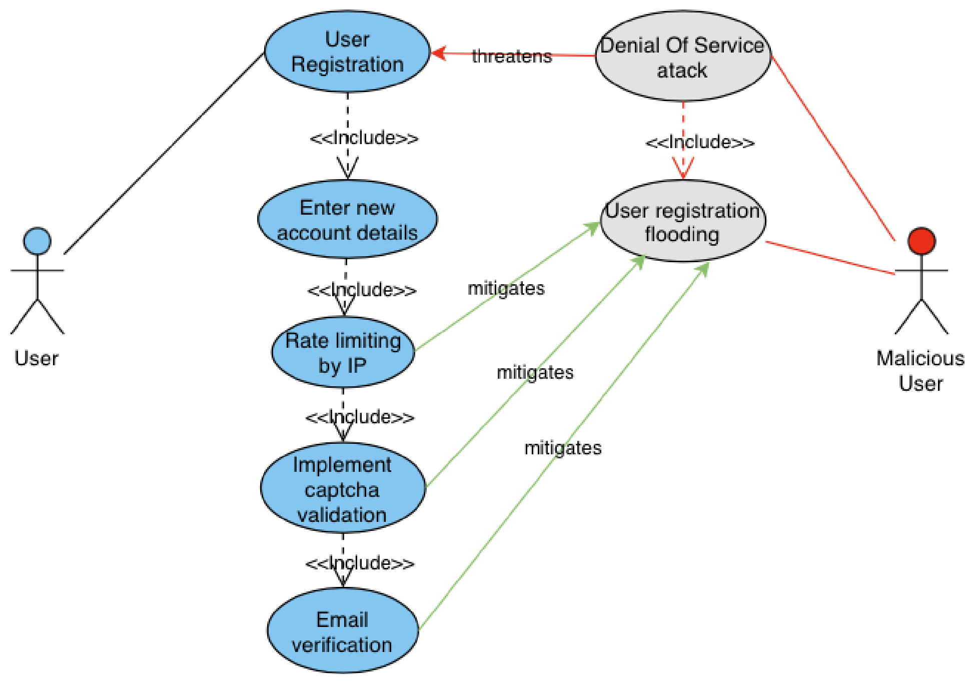


Figure 31- abuse cases – denial of service – user registration flooding

Brute Force – Authentication

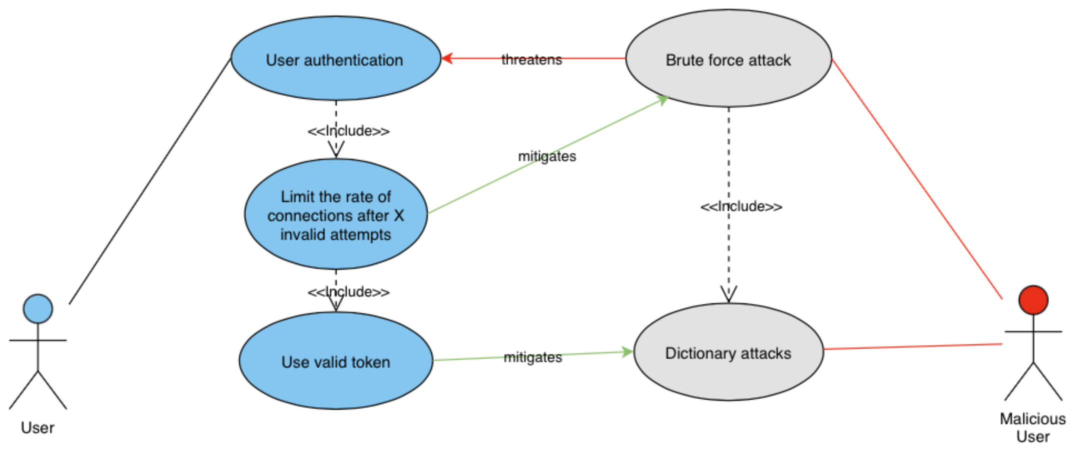


Figure 32 - abuse cases - brute force - authentication

Broken Access Control – Authentication privilege Escalation

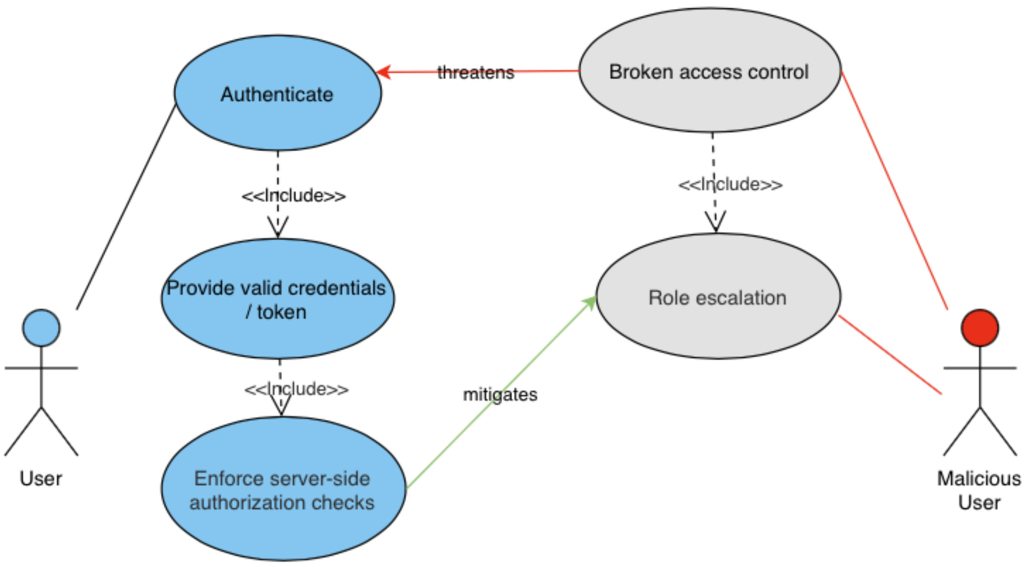


Figure 33 - abuse cases - broken access control – authentication privilege escalation

Broken Access Control - Book Management

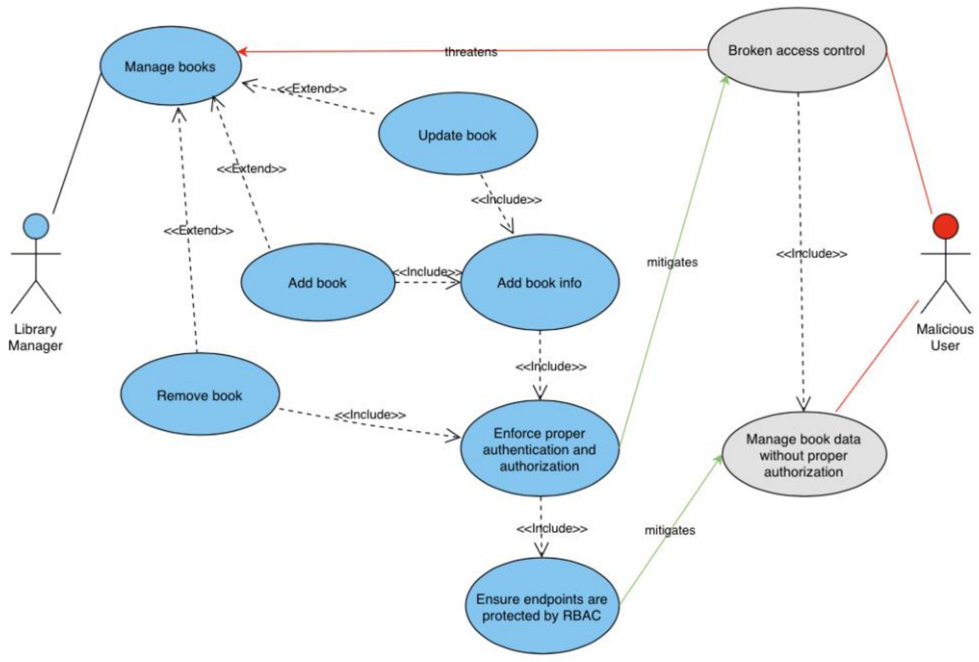


Figure 34 - abuse cases - broken access control - book management

Insecure Direct Object References– Rental History Access

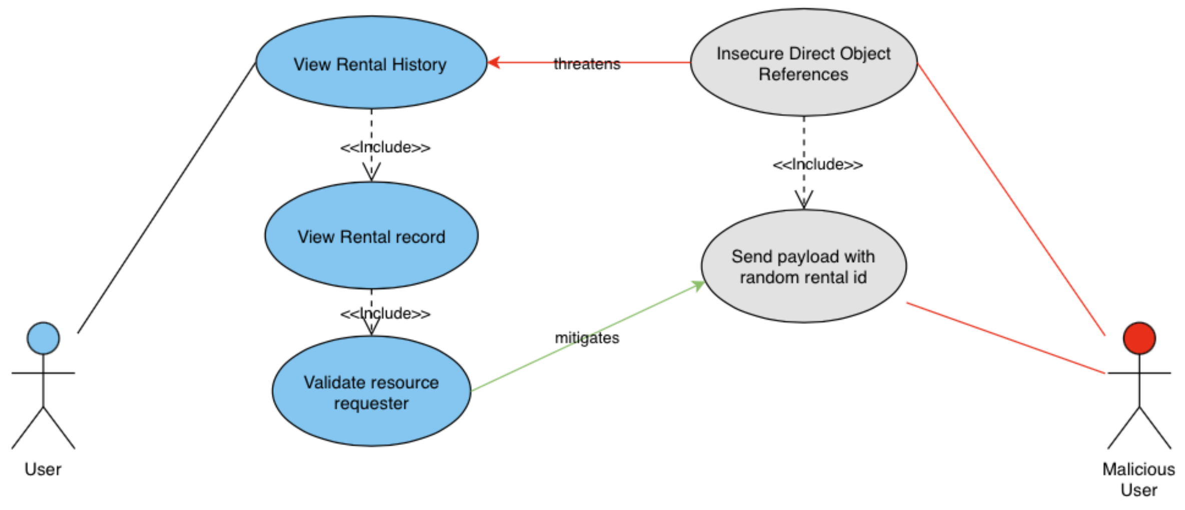


Figure 35 - abuse cases - insecure direct object reference - rental history access

Code Injection – Profile Update

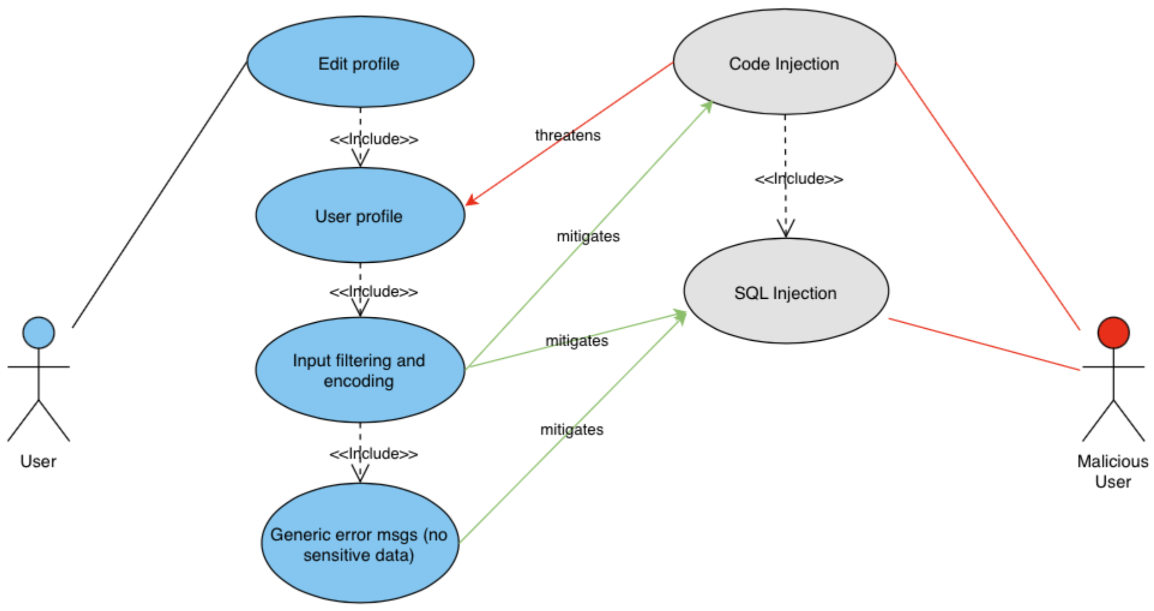


Figure 36 - abuse case - code injection – profile update

Broken Object Level Authorization – Modify user suggestion

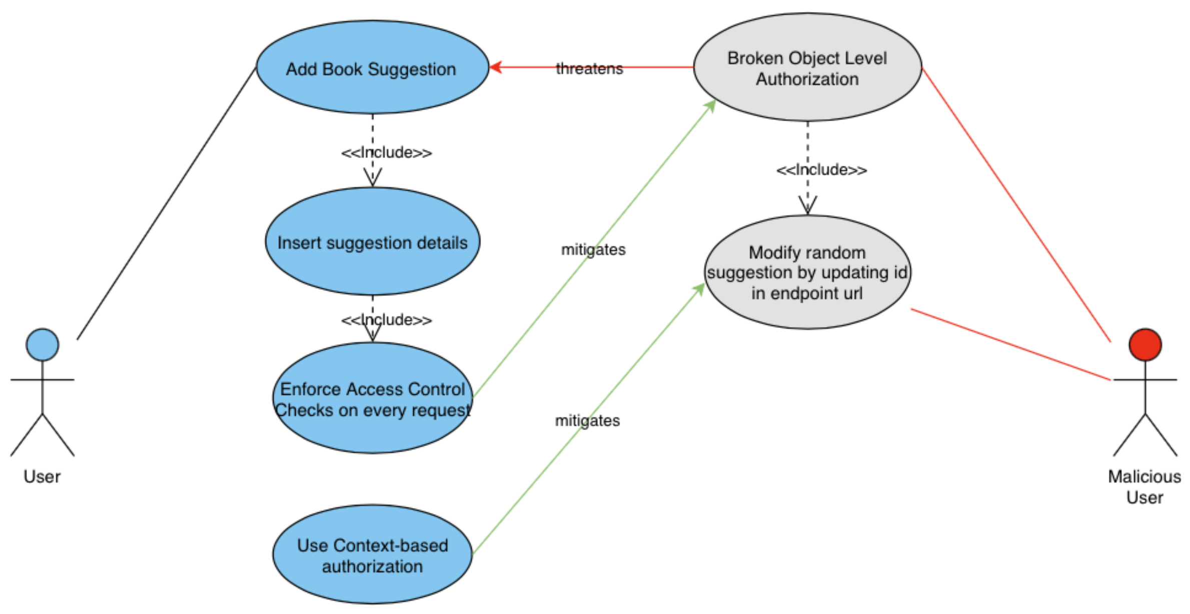

Add Book Suggestion
threatens
Broken Object Level Authorization
<<Include>>
<<Include>>
Insert suggestion details
mitigates
Modify random suggestion by updating id in endpoint url
<<Include>>
Enforce Access Control Checks on every request
mitigates
User
Malicious User
Use Context-based authorization

Figure 37 - abuse cases - broken object level authorization - modify user suggestion

Mass Assignment – Profile Update

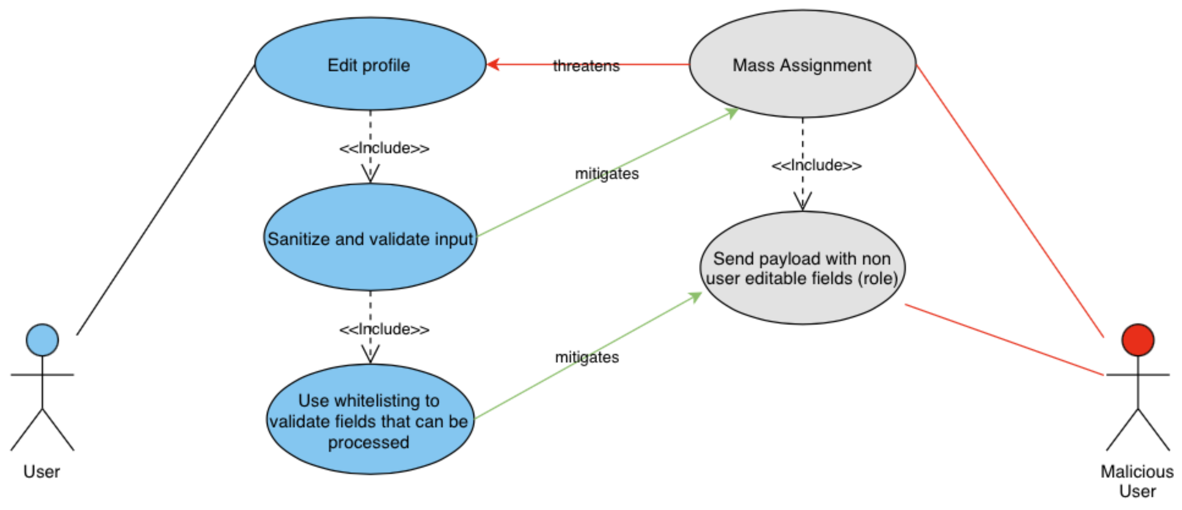

Edit profile
threatens
Mass Assignment
<<Include>>
mitigates
<<Include>>
Sanitize and validate input
Send payload with non user editable fields (role)
<<Include>>
mitigates
Use whitelisting to validate fields that can be processed
User
Malicious User

Figure 38 - abuse cases - mass assignment - profile update

Ranking of Threats

In any security framework, not all threats are created equal. To develop an effective risk mitigation strategy, it is essential to assess and rank threats based on their associated risk factors. This chapter explores how analysing the level of risk posed by identified threats enables the creation of a prioritized list, helping decision-makers determine which threats require immediate attention.

At the core of this analysis lies the concept of risk as the intersection of assets, threats, and vulnerabilities. By understanding how these elements interact, organizations can focus their resources on mitigating the most critical threats first, ensuring a more resilient and secure system.

DREAD

The DREAD model is a structured approach used to assess and prioritize potential security threats. It evaluates five key categories: Damage Potential, Reproducibility, Exploitability, Affected Users, and Discoverability. Each category is scored on a scale from 0 to 10, where higher scores indicate greater risk or concern in that area.

Once all five categories are rated, their values are averaged to determine an overall DREAD score. This score provides a quantifiable way to compare and prioritize threats, helping teams focus on the most critical vulnerabilities first.

Denial of Service (DoS) – Registration Flooding

Table 24 - DREAD - DOS - Registration flooding

DREAD	Question	Score
Damage	How big would the damage be if the attack succeeded? (1- Low; 10- High)	6
Reproducibility	How easy is it to reproduce an attack? (1- Hard; 10- Easy)	9
Exploitability	How much time, effort, and expertise is needed to exploit the threat? (1- Hard; 10- Easy)	8
Affected Users	If a threat were exploited, what percentage of users would be affected. (1- None; 10- All)	7
Discoverability	How easy is it for an attacker to discover this threat? (1- Hard; 10- Easy)	9

Threat score: 7.8 (High)

Brute Force – Authentication

Table 25 - DREAD - Brute force – Authentication

DREAD	Question	Score
Damage	How big would the damage be if the attack succeeded? (1- Low; 10- High)	8
Reproducibility	How easy is it to reproduce an attack? (1- Hard; 10- Easy)	8
Exploitability	How much time, effort, and expertise is needed to exploit the threat? (1- Hard; 10- Easy)	6
Affected Users	If a threat were exploited, what percentage of users would be affected. (1- None; 10- All)	6
Discoverability	How easy is it for an attacker to discover this threat? (1- Hard; 10- Easy)	8
Threat score: 7.2 (High)		

Broken Access Control – Authentication Privilege Escalation

Table 26 - DREAD - Broken access control – Authentication

DREAD	Question	Score
Damage	How big would the damage be if the attack succeeded? (1- Low; 10- High)	9
Reproducibility	How easy is it to reproduce an attack? (1- Hard; 10- Easy)	7
Exploitability	How much time, effort, and expertise is needed to exploit the threat? (1- Hard; 10- Easy)	7
Affected Users	If a threat were exploited, what percentage of users would be affected. (1- None; 10- All)	8
Discoverability	How easy is it for an attacker to discover this threat? (1- Hard; 10- Easy)	6
Threat score: 7.4 (High)		

Broken Access Control - Book Management

Table 27 - DREAD - Broken access control - book management

DREAD	Question	Score
Damage	How big would the damage be if the attack succeeded? (1- Low; 10- High)	7
Reproducibility	How easy is it to reproduce an attack? (1- Hard; 10- Easy)	7
Exploitability	How much time, effort, and expertise is needed to exploit the threat? (1- Hard; 10- Easy)	6
Affected Users	If a threat were exploited, what percentage of users would be affected. (1- None; 10- All)	6
Discoverability	How easy is it for an attacker to discover this threat? (1- Hard; 10- Easy)	6
Threat score: 6.4 (Moderate to High)		

Insecure Direct Object References - Rental History Access

Table 28 - DREAD - Insecure direct object reference - rental history access

DREAD	Question	Score
Damage	How big would the damage be if the attack succeeded? (1- Low; 10- High)	8
Reproducibility	How easy is it to reproduce an attack? (1- Hard; 10- Easy)	8
Exploitability	How much time, effort, and expertise is needed to exploit the threat? (1- Hard; 10- Easy)	7
Affected Users	If a threat were exploited, what percentage of users would be affected. (1- None; 10- All)	7
Discoverability	How easy is it for an attacker to discover this threat? (1- Hard; 10- Easy)	7
Threat score: 7.4 (High)		

Code Injection – Profile Update

Table 29 - DREAD - Code injection - profile update

DREAD	Question	Score
Damage	How big would the damage be if the attack succeeded? (1- Low; 10- High)	9
Reproducibility	How easy is it to reproduce an attack? (1- Hard; 10- Easy)	8
Exploitability	How much time, effort, and expertise is needed to exploit the threat? (1- Hard; 10- Easy)	7
Affected Users	If a threat were exploited, what percentage of users would be affected. (1- None; 10- All)	7
Discoverability	How easy is it for an attacker to discover this threat? (1- Hard; 10- Easy)	6
Threat score: 7.4 (High)		

Broken Object Level Authorization – Modify user suggestion

Table 30 - DREAD - Broken object level authorization - modify user suggestion

DREAD	Question	Score
Damage	How big would the damage be if the attack succeeded? (1- Low; 10- High)	6
Reproducibility	How easy is it to reproduce an attack? (1- Hard; 10- Easy)	8
Exploitability	How much time, effort, and expertise is needed to exploit the threat? (1- Hard; 10- Easy)	7
Affected Users	If a threat were exploited, what percentage of users would be affected. (1- None; 10- All)	6
Discoverability	How easy is it for an attacker to discover this threat? (1- Hard; 10- Easy)	7
Threat score: 6.8 (Moderate to High)		

Mass Assignment – Profile Update

Table 31 - DREAD - Mass assignment - profile update

DREAD	Question	Score
Damage	How big would the damage be if the attack succeeded? (1- Low; 10- High)	8
Reproducibility	How easy is it to reproduce an attack? (1- Hard; 10- Easy)	8
Exploitability	How much time, effort, and expertise is needed to exploit the threat? (1- Hard; 10- Easy)	7
Affected Users	If a threat were exploited, what percentage of users would be affected. (1- None; 10- All)	7
Discoverability	How easy is it for an attacker to discover this threat? (1- Hard; 10- Easy)	7
Threat score: 7.4 (High)		

Qualitative Risk Model

This risk analysis model plays a key role in identifying and prioritizing threats in a qualitative way, so unlike DREAD, no numbers are used. The analysis is done to classify the threats in terms of risk, and then the analysis will be done for each previously identified threat.

Denial of Service (DoS) – Registration Flooding

Table 32 - DoS - registration flooding

Probability	Impact	Risk
Probable	High	Moderate

Brute Force – Authentication

Table 33 - brute force – authentication

Probability	Impact	Risk
Probable	High	Moderate

Broken Access Control – Authentication Privilege Escalation

Table 34 - broken access control – authentication

Probability	Impact	Risk
Probable	High	Severe

Broken Access Control - Book Management

Table 35 - Broken access control - book management

Probability	Impact	Risk
Probable	High	Moderate

Insecure Direct Object References - Rental History Access

Table 36 - Insecure direct object reference - rental history access

Probability	Impact	Risk
Improbable	High	Moderate

Code Injection – Profile Update

Table 37 - Code injection - profile update

Probability	**Impact**	**Risk**
Probable	High	Bigger

Broken Object Level Authorization – Modify user suggestion

Table 38 - Broken object level authorization - modify user suggestion

Probability	**Impact**	**Risk**
Possible	Middle	Moderate

Mass Assignment – Profile Update

Table 39 - Mass assignment - profile update

Probability	**Impact**	**Risk**
Possible	High	Severe

Countermeasures and Mitigations

These are terms used in the context of cybersecurity to describe the actions taken to reduce or neutralize threats and vulnerabilities in information systems. Once the threats and corresponding countermeasures have been identified, it is possible to derive a threat profile with the following criteria:

- Unmitigated threats: Threats that have no countermeasures represent vulnerabilities that can be fully exploited and make an impact.
- Partially mitigated threats: Threats partially mitigated by one or more countermeasures and can only be partially exploited to cause a limited impact.
- Fully mitigated threats: These threats have countermeasures in place and do not expose vulnerabilities.

Below we will describe the implementation of these countermeasures in the premise of this project.

Table 40 - STRIDE Countermeasures

STRIDE	Identified Threats	Countermeasure
Spoofing	Spoofing Threat 1: A user can impersonate a legitimate admin and thus access the user registration system.	Use of Tokens; MFA authentication; Strong Password Policies;
Tampering	Tampering Threat 1: An attacker can try to manipulate the data, such as the role, in order to gain access to other resources on the system.	Data Validation; Principle of Least Privilege; Audit and Monitoring of each role;
	Tampering Threat 2: An attacker may attempt to inject malicious SQL code into the input fields.	Limit database permissions; Monitor Logs; Sanitize entrances;
	Tampering Threat 3: An attacker could try to inject a malicious script.	Code Audit; Sanitize entrances; Apply Security Headers
Repudiation	Repudiation Threat 1:There is a threat that the Admin/User/Manager will deny having made a certain comment.	Peer review; Acceptable Use Policies; Digital Signature;
	Repudiation Threat 2: There is a threat that the user will deny having made a certain recommendation.	

Information Disclosure	Information Disclosure Threat 1: Personal data is confidential and is subject to certain regulations (GDPR). There is the threat where users can access the personal data of other users.	Auditory in the access logs; Data encryption; Privilege Control; MFA authentication; Email filtering; Phishing Simulation Test;
	Information Disclosure Threat 2:There is a threat where attackers can try to gain access to user data through attacks such as phishing, for example.	
	Information Disclosure Threat 3: There is a threat of access to personal data and its disclosure since the filtering is done based on a query to the database.	
	Information Disclosure Threat 4: An attacker can exploit a new user's registration process thus making this form a threat.	
	Information Disclosure Threat 5: SQL Code injection can also pose a threat to Information Disclosure.	
Denial of Service	Denial of Service Threat 1: There is a threat of overloading the system with the attempt to filter for Post search.	Implement request limits for search filtering, such as timeout and request limit per second.
Elevation of Privilege	Elevation of Privilege Threat 1: There is a threat that an attacker will try to elevate your privileges so that you have unauthorized access to restricted system resources	Monitor suspicious activity; Request a second factor for requested change; Log audit; User Access Control;
	Elevation of Privilege Threat 2:An admin/user can use the form used to submit scripts in posts to inject a malicious expression (SQL, for example) in order to change the role of his user or perform actions directly in the database for which he does not have permissions, such as deleting a post that is not his authorship (in the case of an author).	

Secure Design

The secure design of the Library Online Rental System is based on the principles of the Secure Software Development Life Cycle (SSDLC), addressing security from the initial design stage. The system integrates security principles such as defence in depth, least privilege, and secure defaults. This section details the application of these principles to the core system components and workflows.

Authentication and Access Control

- **Role-Based Access Control (RBAC)**: Users are assigned roles (User, Library Manager, Administrator) which determine access to system features and API endpoints. Each role is clearly separated and enforced server-side.
- **Least Privilege**: Each role is only granted permission strictly necessary to perform its tasks.
- **Multi-Factor Authentication (MFA)**: Implemented via email codes during login, ensuring that only the legitimate user can complete the login process.

Data Protection and Privacy

- **Encryption in Transit and at Rest**: All data transmitted between clients and server uses HTTPS/TLS. Sensitive data in the database is securely hashed using bcrypt. Personal information is protected in compliance with GDPR.
- **Secure Backups**: All critical data, including logs, are backed up in encrypted form to two online and one offline location.

Input Validation and Output Handling

- **Whitelisting and Data Type Validation**: All incoming requests are validated using strong data contracts. Fields like name, book suggestions, and search queries are validated and sanitized server-side.
- **Protection Against Injection Attacks**: SQL Injection is mitigated using parameterized queries and ORM frameworks. XSS risks are addressed by accepting only plain text in inputs like book suggestions and applying sanitization rules.
- **Error Handling**: The system avoids leaking technical details in API error messages and returns generic messages for authentication or internal errors.

Secure File and OS Operations

- **Directory/File Isolation**: The system is designed to store any files generated or written by the application in isolated directories, separated from the web root and other publicly accessible paths. This prevents unauthorized file access and exposure through unintended routes.

Logging and Monitoring

- **Audit Logging**: The system is designed to log all sensitive operations, including login attempts, role changes, file interactions, and administrative actions. These logs are recorded locally and are intended to be forwarded to an external log monitoring service as part of the deployment pipeline.

Secure architecture

Secure architecture is based on the principles of defense in depth, where it must ensure that security methods and principles are implemented at all layers, in the part of the infrastructure and operational processes.

We will describe, in stages, the steps necessary to implement a secure architecture in the Library Online Rental System, in addition, the inputs and outputs of each request made by users will be detailed and treated.

Application Layer

The layer closest to the user, responsible for communication between applications and users. This is where data is converted to a user-understandable format and where applications request network services. Therefore, we will point out which steps we will deal with in this layer:

- All client-server communications will be encrypted using https/tls.
- Authentication is managed by Keycloak, introducing the OAuth 2.o and OpenId Connect protocols.
- Role-Based Access Control (RBAC) applies policies and restricted access measures with roles, according to the defined roles (user, Library manager, Administrator).

- Handling of all endpoints to ensure that only correctly formatted and sanitized endpoints are entered.
- The error messages will be generic, where we avoid exposing sensitive system details.

Business Logic Layer

Known as the Business Logic Layer (BLL), it is the part of the software system responsible for implementing the business rules and data processing logic. It acts as an intermediary between the user interface and the data access layer, ensuring that operations are performed according to the company's rules.

- Business rules are applied on the server side, to prevent scalable access or unauthorized operation.
- Sensitive operations (user roles and book manager) Require strict authentication validation and validation of a second user, in addition to storing information in logs.
- Using MFA will be mandatory in login processes.
- Implement good management practices in sessions, including expiration and renewal of passwords and tokens.

Data Access Layer

It refers to a software layer that abstracts the complexity of interacting with the database to the higher layers of the application. DAL makes it easy to manipulate data by allowing the business and presentation layers to focus on business logic and presentation without worrying about the complexities of database access.

- All interactions with MySQL databases will be carried out through secure and encrypted methods.
- Passwords will be stored using encryption hashing methods.
- Sql injection will be prevented with methods of using ORM frameworks, in addition to the parameterized form selects.

Infrastructure Layer

This layer is fundamental for the security of the application, as it covers the points of vulnerability that, if not properly protected, can allow attacks and compromise of the application, where it refers to the physical and logical components that sustain a web application, such as servers, networks, databases and operating systems.

- Backend services are deployed in isolated Docker containers.
- Allow access to Essential Ports, through Firewall.
- Use CI/CD Pipelines where they ensure controlled monitoring and integrated security testing (SAST).
- Logs will be sent to a secure external server, and backups will be encrypted and distributed across multiple locations (online and offline)

External Services

It refers to all services, APIs, and other external components that a web application connects to or relies on. The security of these services is crucial, as they can represent attack vectors, such as authentication failures, data leaks, or vulnerability exploits.

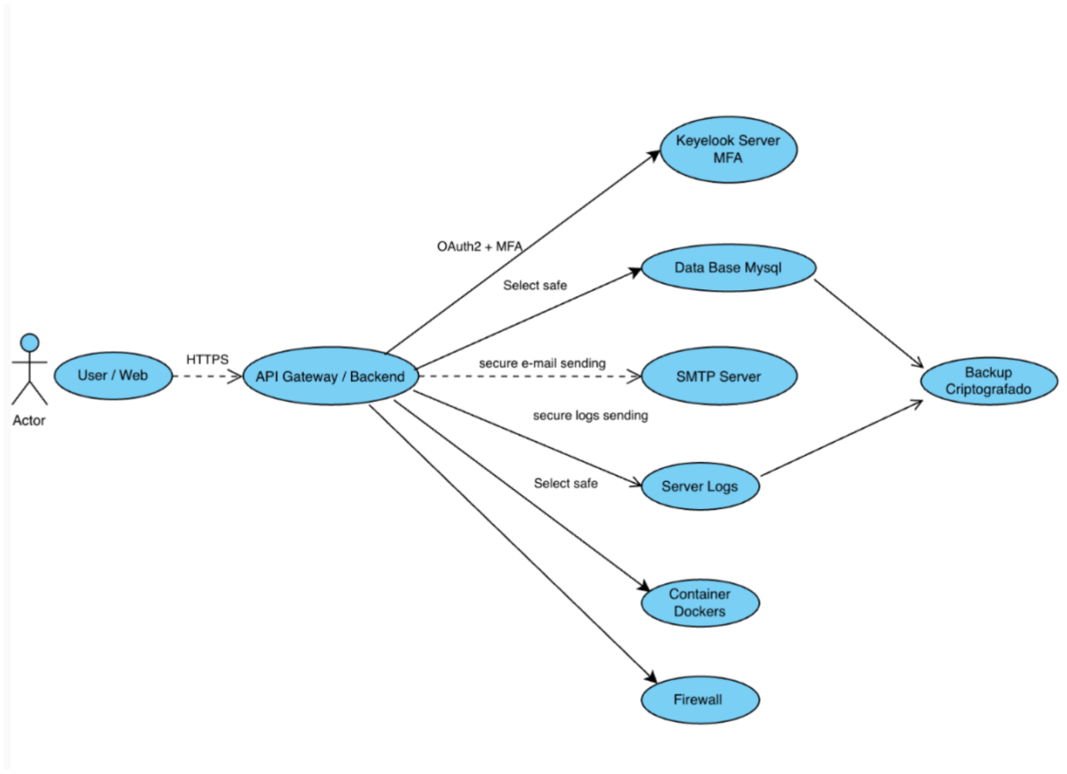
- SMTP services must be protected by sending MFA codes.
- The Keycloak identity management system must store user credentials in addition to access control, i.e. its rules, and MFA settings.
- Use tools to detect and monitor anomalies, such as unauthorized access attempts, in addition to excessive API uses, allowing proactive responses.

Security Controls

It refers to processes, procedures, and measures implemented to mitigate security risks in web applications.

- Mechanisms for rate limiting and throttling, against DDO denial-of-service attacks.
- API endpoints implement access control, using restricted resource-level access, preventing insecure direct object references, the well-known IDOR.
- The application logic includes mitigation against mass assignment and other injection attacks.
- Backup strategies ensure the availability and recovery of system data in the event of data loss or ransomware incidents.

Table 41 - Security architecture



Security test planning

This section defines the planned methodology and strategy to validate the security of the Library Online Rental System.
Since the project is currently in Phase 1, no practical tools or real-world tests have been executed yet.
This document describes the future security validation process based on threat modeling, SDRs, and OWASP ASVS standards.

The following testing approaches will be planned:

- **Static Application Security Testing (SAST)**: Theoretical code review for vulnerabilities.
- **Dynamic Application Security Testing (DAST)**: Future runtime vulnerability scanning.
- **Manual Testing**: Abuse case scenario validation.

- **Threat Modeling Review**: Abuse cases derived from STRIDE and DREAD threats.
- **ASVS Checklist Validation**: Mapping security requirements against OWASP ASVS v4.0.3 Level 2.

Table 42 - Test Cases

Abuse Case	Test Case Description
Denial of Service (DoS)	Simulate a high volume of registration attempts and monitor if the server can detect and block excessive requests.
Brute Force	Perform multiple failed logins attempts to validate the presence of account lockout and rate-limiting mechanisms.
Broken Access Control	Attempt to escalate user privileges (e.g., from User to Manager/Admin) without proper authorization.
Broken Access Control	Attempt to access or modify book management functionalities without having Manager or Admin roles.
Insecure Direct Object References (IDOR)	Attempt to view or modify another user's rental history by manipulating request parameters.
Code Injection	Insert malicious payloads (e.g., SQL injection) into profile update fields and observe application behaviour.
Broken Object Level Authorization	Attempt to modify or delete book suggestions submitted by other users without appropriate authorization.
Mass Assignment	Submit crafted API requests with additional parameters to manipulate unauthorized user properties during profile update.

The system will be considered secure when:

- All abuse cases demonstrate secure behavior (no vulnerabilities found).
- All critical threats identified in threat modeling are mitigated.
- All SDRs are verified against their planned tests.
- Compliance with OWASP ASVS Level 2 is achieved.

Conclusion

The initial analysis and threat modelling of the Library Online Rental System demonstrate a comprehensive understanding of secure software development principles.

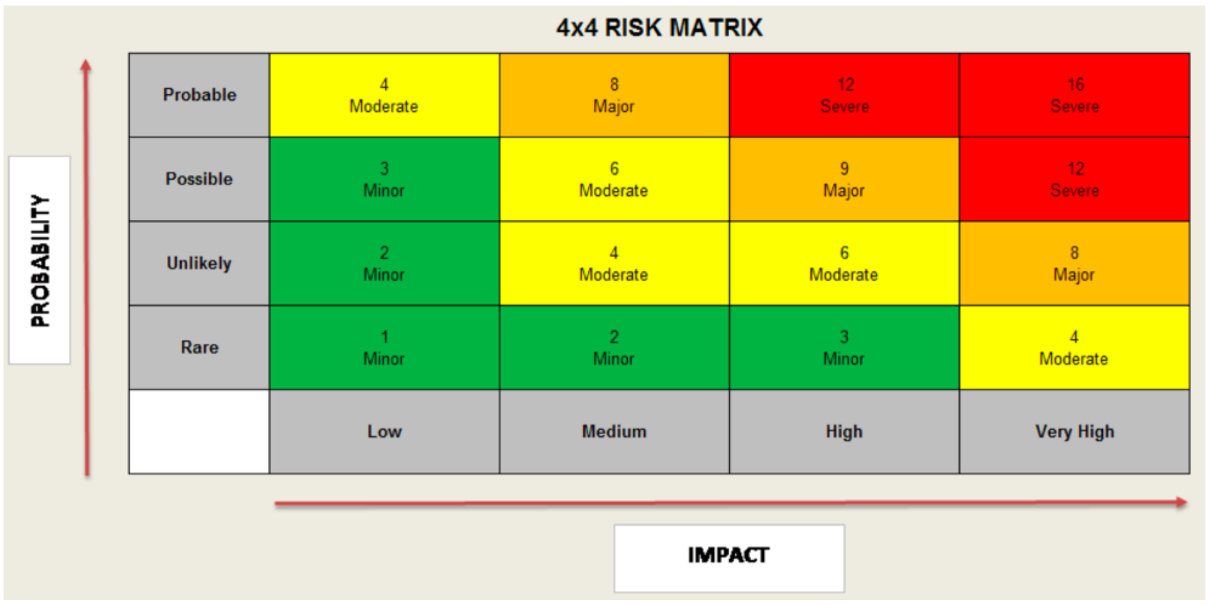
By applying structured methodologies such as STRIDE, DREAD, and Abuse Case modelling, we have identified key risks, designed appropriate mitigations, and built a foundation for secure application architecture.

The layered security architecture covers the application, business logic, data access, infrastructure, and external services. Security requirements are clearly defined to address critical areas like authentication, access control, data protection, and secure coding practices.

Testing strategies based on static, dynamic, and manual analysis have been outlined to validate the resilience of the system against the identified threats.

Moving forward, the next phase will focus on implementing the system according to the defined requirements and performing security testing to ensure compliance with the security objectives set out in this phase.

Attachment A



4x4 RISK MATRIX

PROBABILITY

Probable	4 Moderate	8 Major	12 Severe	16 Severe
Possible	3 Minor	6 Moderate	9 Major	12 Severe
Unlikely	2 Minor	4 Moderate	6 Moderate	8 Major
Rare	1 Minor	2 Minor	3 Minor	4 Moderate
	Low	**Medium**	**High**	**Very High**

IMPACT