VR Library - Use Case Specifications:

Daoqi Xiong
ID: 25335365

Jiayun Zhang
ID: 25334655

Joshua O'Donnell
ID: 25353241

Keshav Tripathi
ID: 25347551

Priyansh Nayak
ID: 25350660

Rundong Chen
ID: 25336554

Sarvesh Jaiswal
ID: 24361554

Viraj Bhor
ID: 25358463

Xu Gong
ID: 25334562

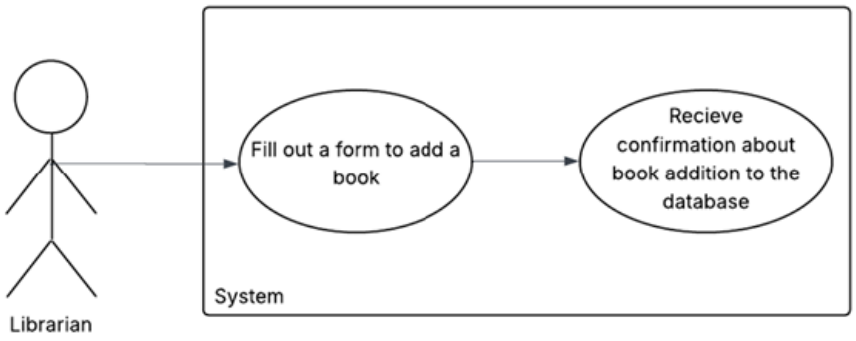
Yokesh Kathiravan
ID: 25341409

Submission Date: October 17th, 2025

1 Backend & Admin

1.1 Manual Book addition to Library Database

Use Case Diagram:



1.1.1 Description

The librarian can visit the system to add a book after filling in a form. The system will send a confirmation to the librarian after the book has been successfully added.

Actors:

- Librarian
- System

Triggers & Inputs

- Triggers: The librarian sends a request to add a book by filling a form.
- Inputs: The form sent by the librarian, which contains the information about the book.

1.1.2 Flow of Events

Basic Flow:

Librarian	Action Number	System
Librarian will fill out registration form.	1	
	2	The system will verify if the book already exists.
	3	If not the system will add the book details in the database.
Librarian will get a confirmation about the status of the book.	4	

Alternative Flow 1:

Librarian	Action Number	System
	1	The book yet to add already exists in the library database.
Librarian will notified that the book already exists	2	

1.1.3 Special Requirements:

- Internet and Database connection required

1.1.4 Preconditions:

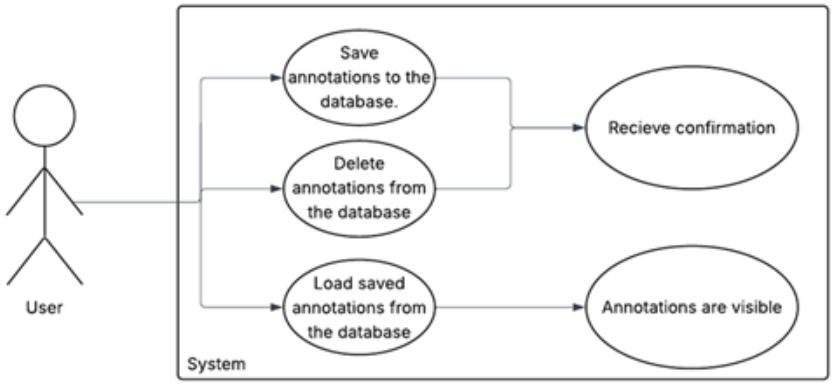
- Authenticated librarian

1.1.5 Postconditions:

- Book entry created with availability status and user register linked.

1.2 Saving & Loading Book annotations

Use Case Diagram:



1.2.1 Description

User can convert annotations (highlight/underline) into prefabs saved with the book and user IDs.
Actors:

- User and System

Triggers & Input

- All the annotations done in the books are automatically saved.
- To load different annotations from different users there will be list of annotations.
- Editing the annotations will automatically overwrite the older one.

1.2.2 Flow of Events:

Basic Flow:

User	Action Number	System
User will annotate something on the book.	1	
	2	The system will save those annotations.

Alternative Flow 1:

User	Action Number	System
User will open a pre-saved annotation.	1	
	2	The system will verify and delete the annotation..

Alternative Flow 2:

User	Action Number	System
User will delete their own saved annotation.	1	
	2	The system will verify and delete the annotation.

1.2.3 Special Requirements:

- Internet and Database connection required

1.2.4 Preconditions:

- Ability to annotate the books.

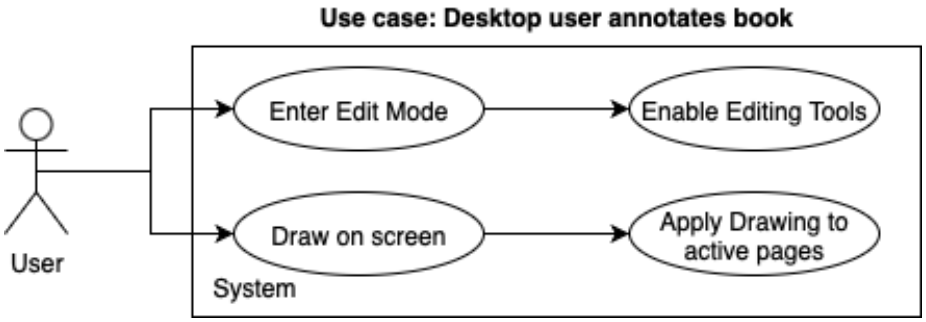
1.2.5 Postconditions:

- Prefab stored in database; can be reloaded later with the same book.

2 General Book Interactions

2.1 Ordering a book

Use Case Diagram:



2.1.1 Description

Ordering a book from the web app to a user's library.

Actors:

- User (any)

Triggers & Input:

- Trigger: Searching for a book on the website
- Input: Book title/author name/doi.

2.1.2 Flow of Events:

Basic Flow:

User	Action number	System
User searches for a book	1	
	2	Search results are presented to the user
User selects their book of choice	3	
	4	Book is added to the user's library
	5	System confirms operation success

Alternative Flow:

User	Action number	System
	2.1	No results are found, user is prompted to try a different search,

2.1.3 Special Requirements:

- Active Internet connection.

2.1.4 Preconditions:

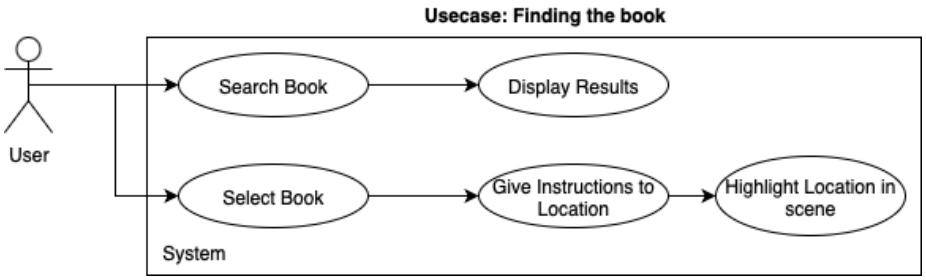
- User is successfully logged into the web-app.
- User is subscribed to the service.

2.1.5 Postconditions:

- User's library updated to include the ordered book.

2.2 Use Case: Finding a book in the library

Use Case Diagram:



2.2.1 Description:

Guiding the user to a book's location within the library.

Actors:

- User (any)

Triggers & Inputs:

- Trigger: Searching for a book within the library.
- Input: Book title/author name/doi.

2.2.2 Flow of Events:

Basic Flow:

User	Action number	System
Search Book	1	
	2	Display search results to user
Select desired book from results	3	
	4	Give instructions on navigating to location
	5	Highlight book location in scene.

Alternative Flows:

User	Action number	System
	2.1	No search results, user is prompted to try a different search.

2.2.3 Special Requirements:

- Active Internet connection.

2.2.4 Preconditions:

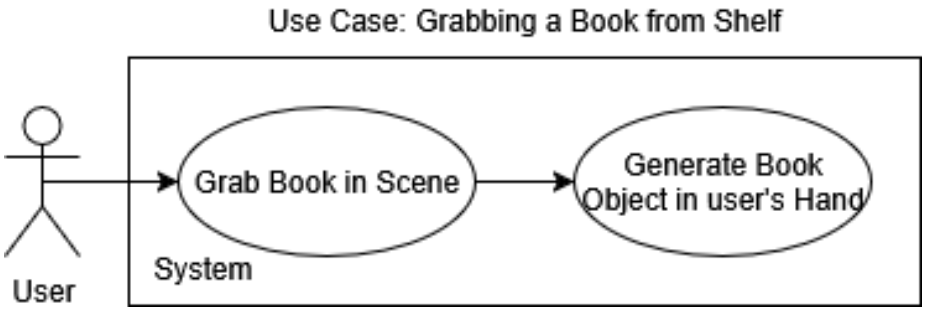
- User is logged in & subscribed to the system.
- User is in an active session.

2.2.5 Postconditions:

- Highlighted book area in scene.

2.3 Use Case: Grabbing a book

Use Case Diagram:



2.3.1 Description:

Providing a copy of a book the user has selected to read.

Actors:

- User (any)

Triggers & Inputs:

- Trigger: User grabs a book in the library.
- Input: Interaction position.

2.3.2 Flow of Events:

Basic Flow:

User	Action number	System
User grabs a book in scene	1	
	2	Generate book object in user's hand

Alternative Flows:

User	Action number	System
	2.1	No book at grab location. No book is generated.

2.3.3 Special Requirements:

- Books populated into scene at library instance launch → any book grabbed is available.
- Active Internet connection

2.3.4 Preconditions:

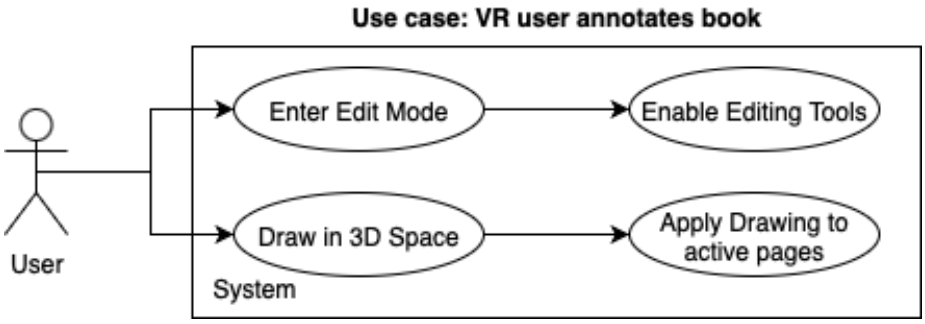
- User is logged in & subscribed to the service.
- User is in an active library session.

2.3.5 Postconditions:

- Digital copy of the selected book present in scene to be used at will.

2.4 Use Case: VR User annotates book

Use Case Diagram:



2.4.1 Description:

VR User adding notes to books in scene.

Actors:

- VR User

Triggers & Inputs:

- Triggers: Entering edit mode.
- Input: None

2.4.2 Flow of Events:

Basic Flow:

VR User	Action number	System
User enters edit mode	1	
	2	Make editing tools available to user
User draws in 3D space	3	
	4	Flattened drawing applied to active pages

No alternative flows to mention.

2.4.3 Special Requirements:

- Active Internet connection

2.4.4 Preconditions:

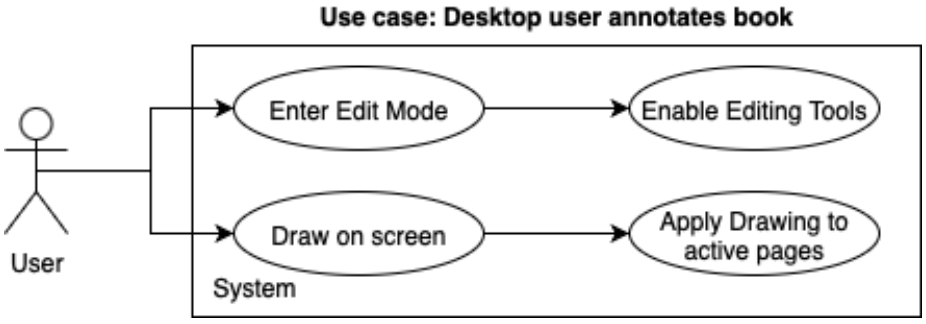
- User is logged in and subscribed to the service.
- User is in an active session.
- There is a book copy present in session

2.4.5 Postconditions:

- Edited book has annotations visible on the given pages.

2.5 Use Case: Desktop User annotates book

Use Case Diagram:



2.5.1 Description:

Desktop user adding notes to books in scene.

Actors:

- Desktop User

Triggers & Inputs:

- Triggers: Entering edit mode.
- Input: None

2.5.2 Flow of Events:

Basic Flow:

Desktop User	Action number	System
User enters edit mode	1	
	2	Make editing tools available to user
User draws on screen	3	
	4	Drawing applied to active pages

No alternative flows to mention.

2.5.3 Special Requirements:

- Active Internet connection

2.5.4 Preconditions:

- User is logged in and subscribed to the service.
- User is in an active session.
- There is a book copy present in session

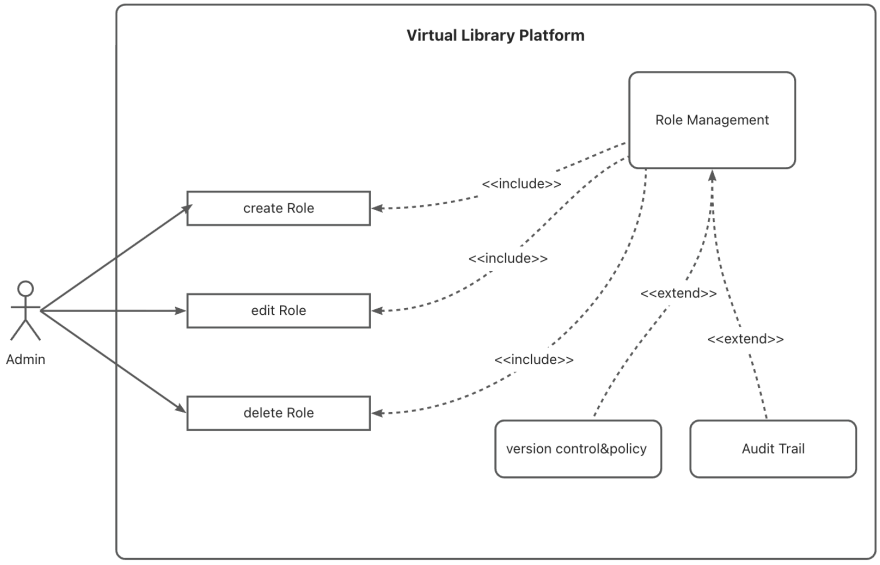
2.5.5 Postconditions:

- Edited book has annotations visible on the given pages.

3 Authentication and Session Management

3.1 Define Global Role Catalog

Use Case Diagram:



3.1.1 Description

Defines and maintains the list of global roles (Admin, Librarian, Member) and their permissions to ensure consistency and traceability.
Actors:

- Admin
- System

Triggers & Input

- Triggers: Admin opens the *Role Management* interface
- Input: create, edit, or delete a role Event

3.1.2 Flow of Events:

Basic Flow:

Admin	Action Number	System
Admin navigates to “Role Management.”	1	
Admin adds or edits a role definition.	2	
	3	System validates role name and permission set.
	4	System saves the configuration and creates a version stamp.
	5	System logs the change in the audit trail.

3.1.3 Special Requirements:

Integrate with Authorization Engine and Log System; support rollback of role versions.

3.1.4 Preconditions:

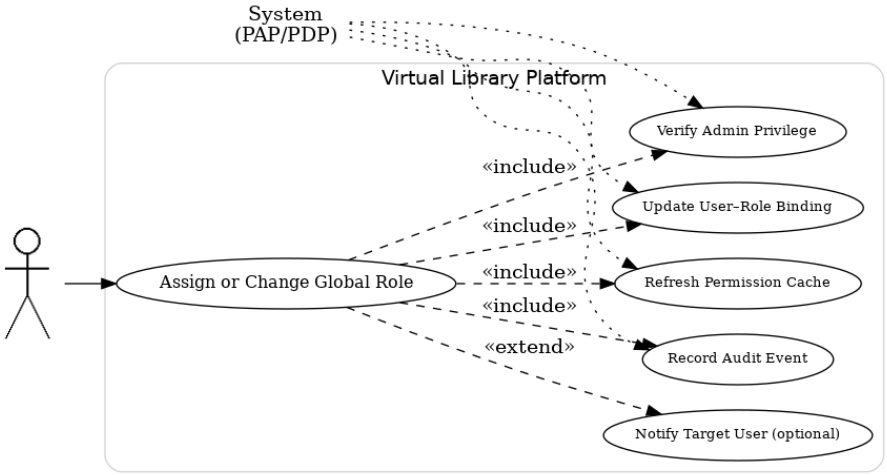
- Admin logged in with management privileges.

3.1.5 Postconditions:

- Role catalog updated and available for future authorization checks.

3.2 Assign or Change Global Role

Use Case Diagram:



3.2.1 Description

Allows Admin to promote or demote a user's global role (e.g., Member → Librarian).
Actors:

- Admin
- Users
- System

Triggers & Input

- Triggers: Admin selects a user and assigns a new role via dashboard.
- Input: Change Into

3.2.2 Flow of Events:

Basic Flow:

Admin	Action Number	System
Admin opens user list.	1	
Admin selects target user and new role.	2	
	3	System verifies Admin's privilege to make the change.
	4	System updates user-role binding in database.
	5	System notifies affected user (optional).
	6	System records the event and refreshes permission cache.

3.2.3 Special Requirements:

Changes must propagate to active sessions or take effect at next login.

3.2.4 Preconditions:

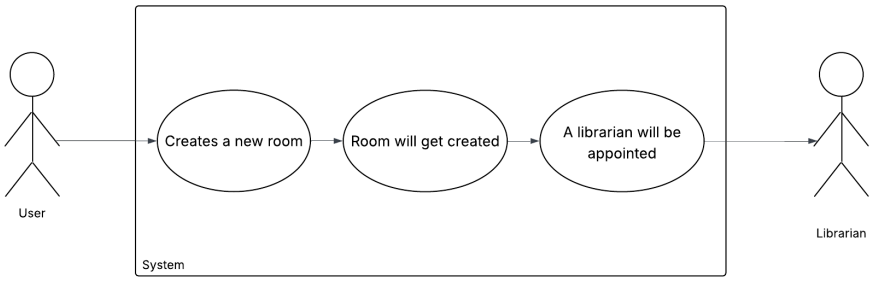
- Admin has access to role management functions.

3.2.5 Postconditions:

- User's global role updated; new permissions active system-wide.

3.3 Room and Scoped Role Binding

Use Case Diagram:



3.3.1 Description

When a user creates a room, the system assigns them as the Host and establishes room-level scoped roles.
Actors:

- Member/Admin/Librarian
- System

Triggers & Input

- Triggers: User clicks "Create Room" and sets configuration .
- Input: configuration info (name, visibility, capacity)

3.3.2 Flow of Events:

Basic Flow:

Admin	Action Number	System
User selects "Create Room."	1	
User inputs room details and submits.	2	
	3	System validates settings and policy.
	4	System creates room and binds creator → Host, scope = roomId .
	5	System issues join token and logs creation event.

3.3.3 Special Requirements:

Integrate with Session Management and Authorization modules.

3.3.4 Preconditions:

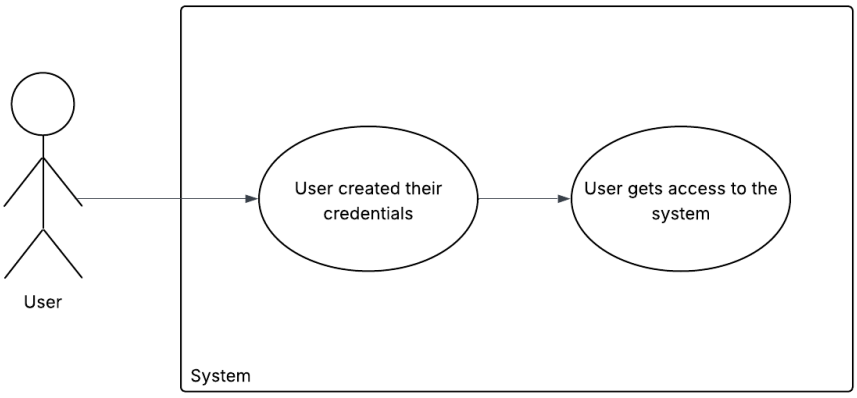
- User has "Create Room" permission in global role.

3.3.5 Postconditions:

- Room created; creator becomes Host; audit record stored.

3.4 Sign In

Use Case Diagram:



3.4.1 Description

Creates a new authentication session after successful user login or single sign-on (SSO). The system issues access and refresh tokens to maintain secure user authentication.
Actors:

- Member / Librarian / Admin / Guest
- System

Triggers & Input

- Triggers: User submits login credentials or completes SSO authentication.
- Input: Login info

3.4.2 Flow of Events:

Basic Flow:

User	Action Number	System
Provides credentials or SSO token.	1	
	2	Validates credentials and optional MFA.
	3	Generates Access Token (AT) and Refresh Token (RT).
	4	Binds tokens to device and session context.
	5	Logs session.create with userId, device, and issue time.

3.4.3 Special Requirements:

- Tokens must follow JWT standard.
- Optional MFA required for admin roles.

3.4.4 Preconditions:

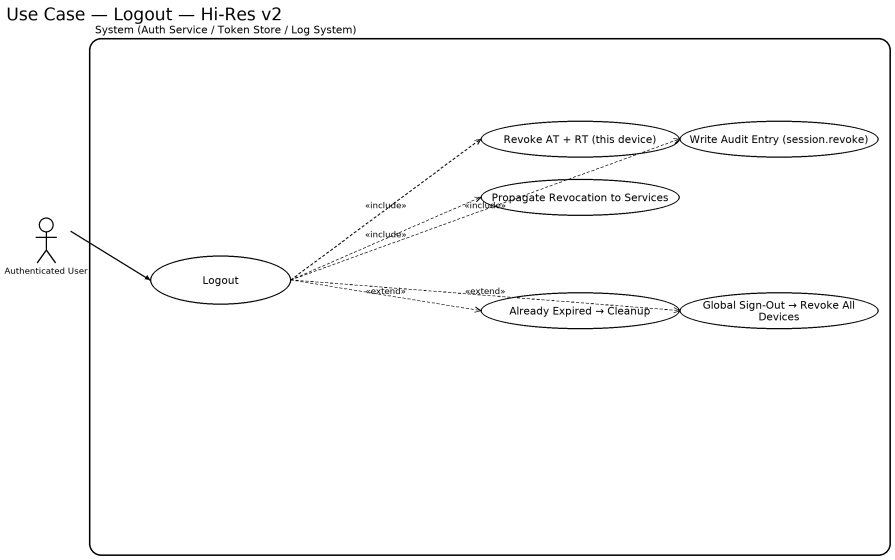
- Operates on the Authentication microservice of the Virtual Library backend, connected to the RBAC and Log System modules.

3.4.5 Postconditions:

- Session established;
- AT and RT issued and stored securely.

3.5 Logout

Use Case Diagram:



3.5.1 Description

Terminates the active session on the current device by invalidating its tokens and clearing cookies or storage data.
Actors:

- Authenticated User
- System

Triggers & Input

- Triggers: User selects “Logout” from account menu.
- Input: logout event

3.5.2 Flow of Events:

Basic Flow:

Admin	Action Number	System
Clicks “Logout.”	1	
	2	Server revokes AT and RT for this device.
	3	Clears local cookies and session storage.
	4	Emits session.revoke for audit.

3.5.3 Special Requirements:

Revocation should propagate instantly across all microservices.

3.5.4 Preconditions:

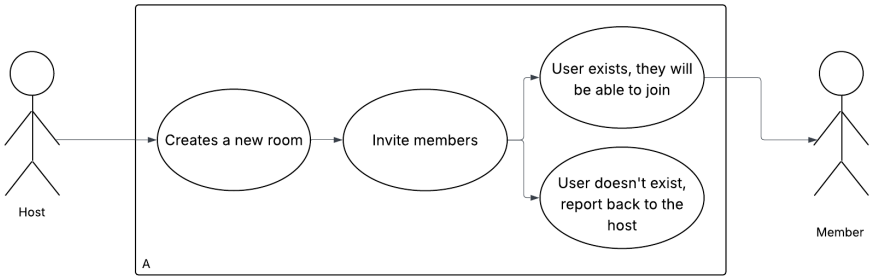
- User signed in with valid AT/RT.

3.5.5 Postconditions:

- Device session terminated; user redirected to login screen.

3.6 Create Room Session

Use Case Diagram:



3.6.1 Description

Creates a scoped Room Session (RS) for an authenticated or invited user when joining a room.
Actors:

- Member / Guest
- System

Triggers & Input

- Triggers: User sends join request with valid Access Token or Invite Token.
- Input: Join Room Event

3.6.2 Flow of Events:

Basic Flow:

Admin	Action Number	System
Sends join request.	1	
	2	Validates token and checks room capacity.
	3	Creates RS record with user's role, roomId, and sessionId.
	4	Attaches user to real-time channel and returns session snapshot.
	5	Logs room.session.create.

3.6.3 Special Requirements:

Must integrate with Authorization Component and Presence Service.

3.6.4 Preconditions:

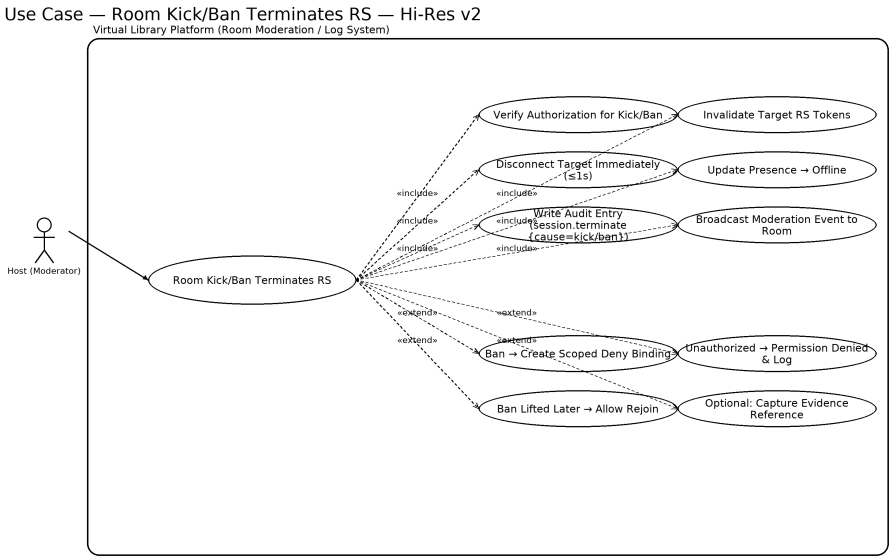
- User authenticated or invited; room exists.

3.6.5 Postconditions:

- Room session established; join recorded.

3.7 Room Kick/Ban Terminates RS

Use Case Diagram:



3.7.1 Description

Terminates a participant's Room Session when they are kicked or banned by a moderator.
Actors:

- Host
- Target Participant
- System

Triggers & Input

- Triggers: Moderator issues a "Kick" or "Ban" action toward a participant.
- Input: Moderator Event

3.7.2 Flow of Events:

Basic Flow:

Host	Action Number	System	Participant
Moderator performs kick/ban action.	1		
	2	System verifies authorization.	
	3	Invalidates participant's RS.	
	4	Disconnects participant from room.	
	5	Logs session.terminate with cause = kick/ban.	
	6		Kicked/Banned

3.7.3 Special Requirements:

Immediate RS termination required (¡1 second latency).

3.7.4 Preconditions:

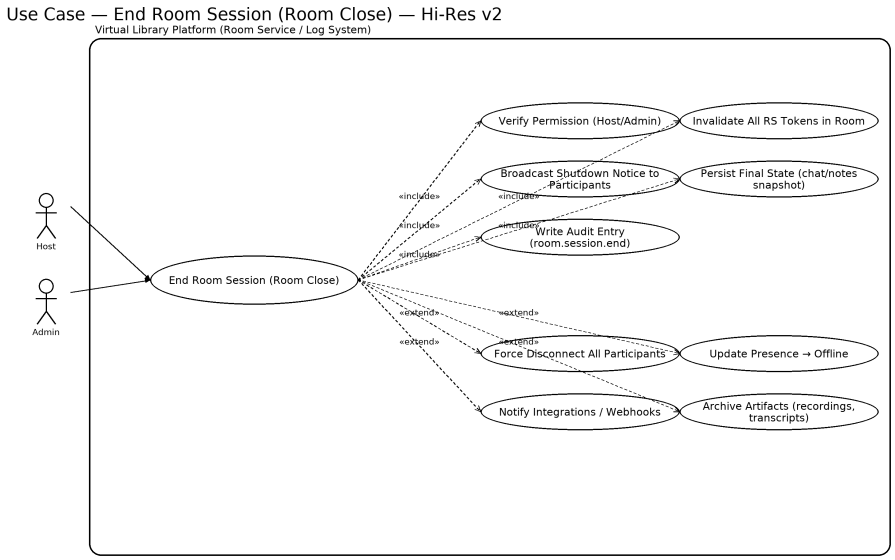
- Active RS exists for target participant.

3.7.5 Postconditions:

- RS closed; participant removed; event recorded.

3.8 Room Close

Use Case Diagram:



3.8.1 Description

Gracefully ends all active Room Sessions when a Host or Admin closes the room.
Actors:

- Host / Admin
- System

Triggers & Input

- Triggers: Host clicks “End Room”; system broadcasts closure signal.

3.8.2 Flow of Events:

Basic Flow:

Host	Action Number	System	Participant
Host confirms room closure.	1		
	2	Server invalidates all RS tokens in that room.	
	3	Broadcasts shutdown notice to participants.	
	4	Persists final chat, notes, and logs room.session.end.	
	5		leave room

3.8.3 Special Requirements:

Shutdown must be orderly; all clients notified with consistent state.

3.8.4 Preconditions:

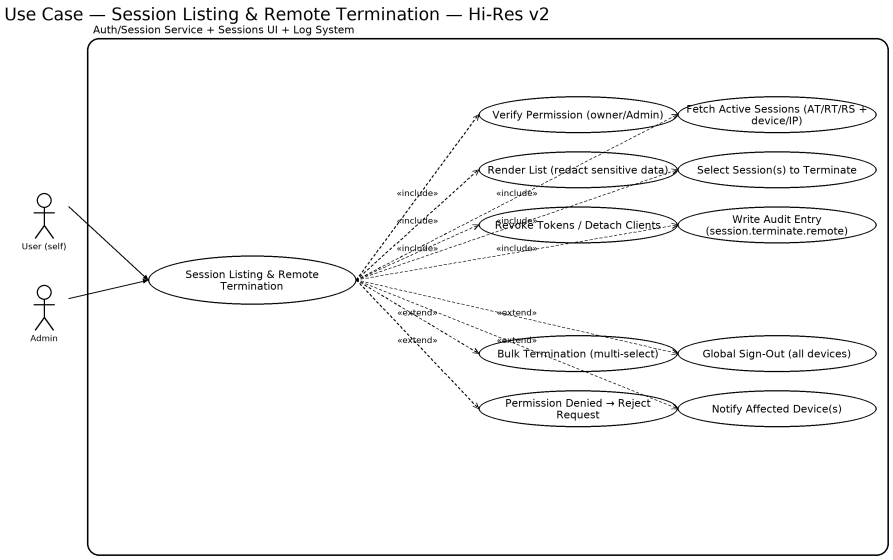
- Active room with at least one session.

3.8.5 Postconditions:

- All RS instances terminated; room marked closed in database.

3.9 Session Listing & Remote Termination

Use Case Diagram:



3.9.1 Description

Allows users or admins to view all active sessions and terminate them individually or remotely.
Actors:

- User (self) / Admin (global)
- System

Triggers & Input

- Triggers: User or Admin opens “Active Sessions” page and requests to end one or multiple sessions.

3.9.2 Flow of Events:

Basic Flow:

Admin	Action Number	System
Opens session list view.	1	
	2	Displays all active AT/RT and RS sessions with device/IP info.
Selects session to terminate.	3	
	4	Server revokes selected tokens and detaches client.
	5	Logs session.terminate.remote.

3.9.3 Special Requirements:

Admin view must redact sensitive data (IP truncated); allow batch termination.

3.9.4 Preconditions:

- User authenticated or admin privileges granted.

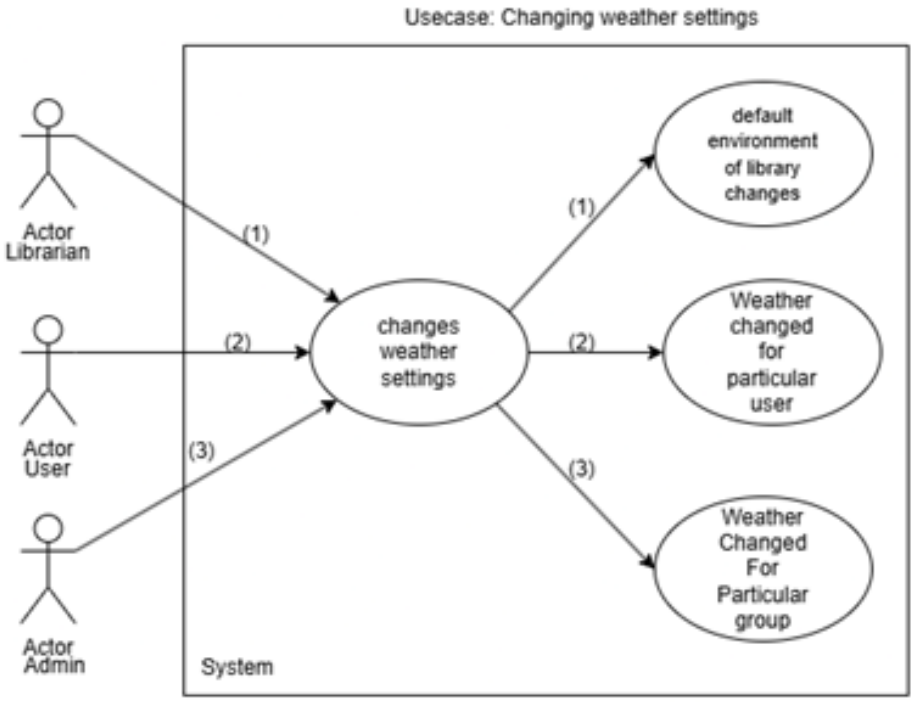
3.9.5 Postconditions:

- Selected sessions revoked; result logged and visible in session history.

4 Settings & Weather System Operation

4.1 Use Case: Weather Adjustment

Use Case Diagram:



4.1.1 Description

Allow the user to manipulate their weather settings.

Actors:

- User
- Admin
- Librarian

Triggers & Inputs:

- Be inside the library
- Log in to the member database
- Open settings

- Change Weather settings
- Select weather change option from the drop-down menu.

4.1.2 Flow of Events:

Basic Flow for Private and Public Session:

User	Action Number	System
Enter the library session	1	
Log in to the member database	2	Authentic session
Open settings using the appropriate control	3	Display the Settings Menu
Select weather settings menu	4	Display weather drop-down menu
Player Chooses from 5 options(Sunny, Spring, Rainy, Fall, Snowy)	5	Change the environment and ambiance to the chosen weather settings
	6	Change the audio to fit the new weather conditions
All users in the session view the new environment selected by the host/ The user can view the new selected environment.	7	

4.1.3 Special requirements:

If the weather is changed in a private session, the environment should be synchronized for all users.

4.1.4 Alternative flow 1:

User	Action Number	System
	1	Login Authentication Failed.
	2	User cannot access the library.

4.1.5 Alternative flow 2:

User	Action Number	System
	1	User is not the host of the session.
	2	Unable to change the weather.

4.1.6 Preconditions:

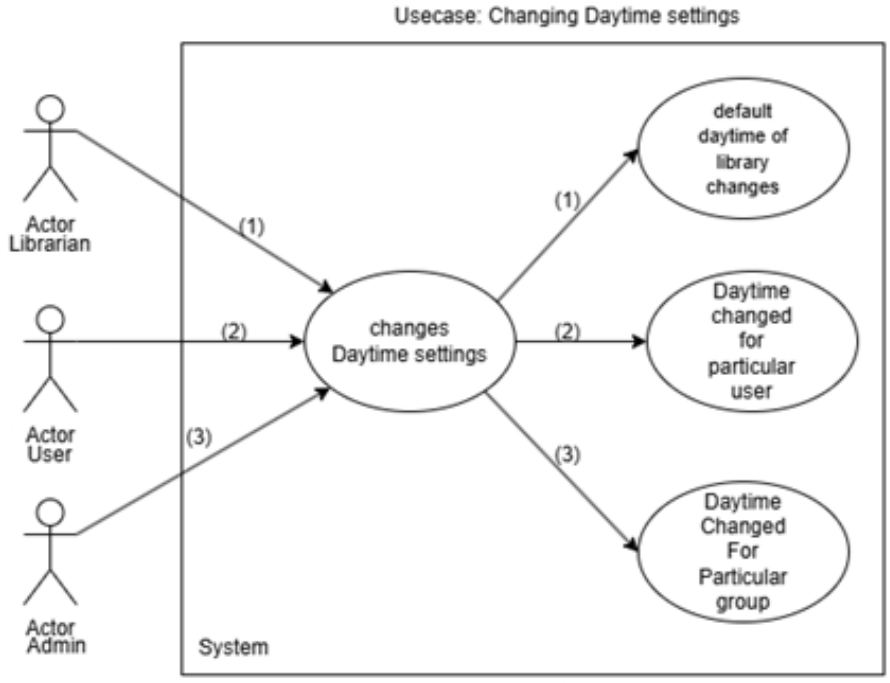
- The user must be logged into the system.
- If the session is private, the user must be a host to change the weather settings.

4.1.7 Postconditions:

- The weather and ambience are successfully changed for the user(s) of the session.

4.2 Use Case: Daytime Adjustment

Use Case Diagram:



4.2.1 Description

Allow the user to change the daytime settings.

Actors:

- User

Triggers & Inputs:

- Be inside the library
- Log in to the member database
- Open settings
- Change daytime settings
- select daytime change option from dropdown menu.

4.2.2 Flow of Events:

Basic Flow for Private and Public sessions:

User	Action Number	System
Enter the library session	1	
Log in to the member database	2	Authentic session
Open settings using the appropriate control	3	Display the Settings Menu
Select daytime settings menu	4	Display daytime dropdown menu
Player Chooses from 4 options(Morning, Afternoon, Evening, Night)	5	Change the environment and ambiance to the chosen daytime settings
	6	Change the audio to fit the new daytime conditions
All users in the session view the new environment selected by the host/ The user can view the new selected environment.	7	

4.2.3 Special requirements:

If the daytime changes in a private session, the environment should be synchronized for all users.

4.2.4 Alternative flow 1:

User	Action Number	System
	1	Login Authentication Failed.
	2	User cannot access the library.

4.2.5 Alternative flow 2:

User	Action Number	System
	1	User is not the host of the session.
	2	Unable to change the daytime.

4.2.6 Preconditions:

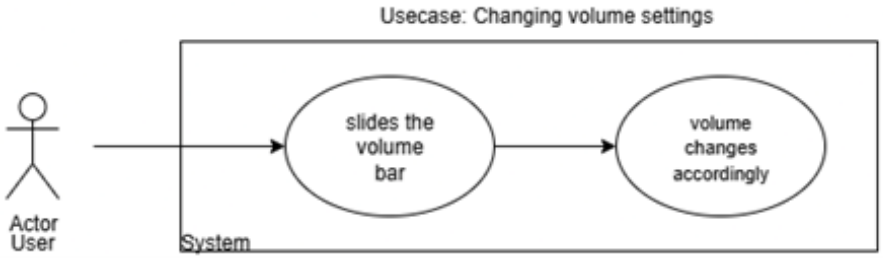
- The user must be logged into the system.
- If the session is private, the user must be a host to change the daytime settings.

4.2.7 Postconditions:

- The weather and ambiance are successfully changed for the user(s) of the session.

4.3 Use Case: Volume Adjustment

Use Case Diagram:



4.3.1 Description

Allow the user to change the volume settings.

Actors:

- User

Triggers & Inputs:

- Be inside the library
- Log in to the member database
- Open settings
- Drag Volume Slider.

4.3.2 Flow of Events:

Basic Flow:

User	Action Number	System
Enter the library session	1	
Log in to the member database	2	Authentic session
Open settings using the appropriate control	3	Display the Settings Menu
Change the volume slider position	4	Change the audio accordingly for the particular user
The player's audio level is changed to their choice.	5	

4.3.3 Preconditions:

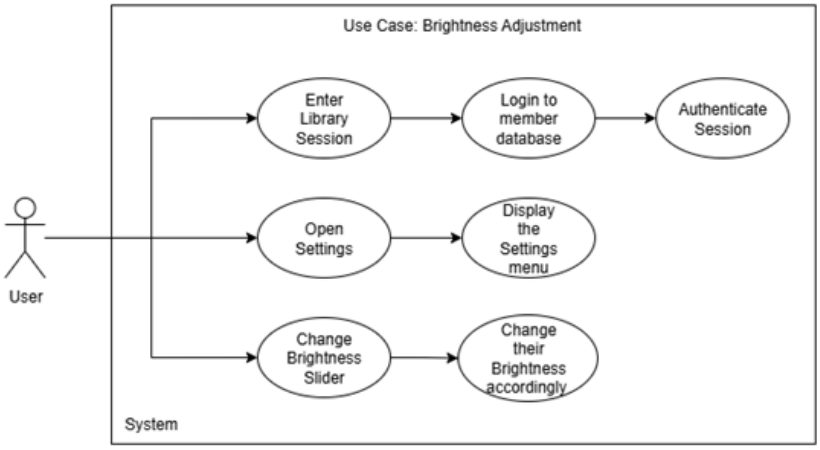
- The user must be logged into the system.

4.3.4 Postconditions:

- The audio level is successfully changed for the user.

4.4 Use Case: Brightness Adjustment

Use Case Diagram:



4.4.1 Description

Allow the user to manipulate their brightness settings.

Actors:

- User

Triggers & Inputs:

- Be inside the library
- Log in to the member database
- Open settings
- Change brightness slider

4.4.2 Flow of Events:

Basic Flow:

User	Action Number	System
Enter the library session	1	
Log in to the member database	2	Authentic session
Open settings using the appropriate control	3	Display the Settings Menu
Change the Brightness Slider	4	Change the universal brightness accordingly for the particular user
	5	The player's brightness level is changed to their choice

4.4.3 Preconditions:

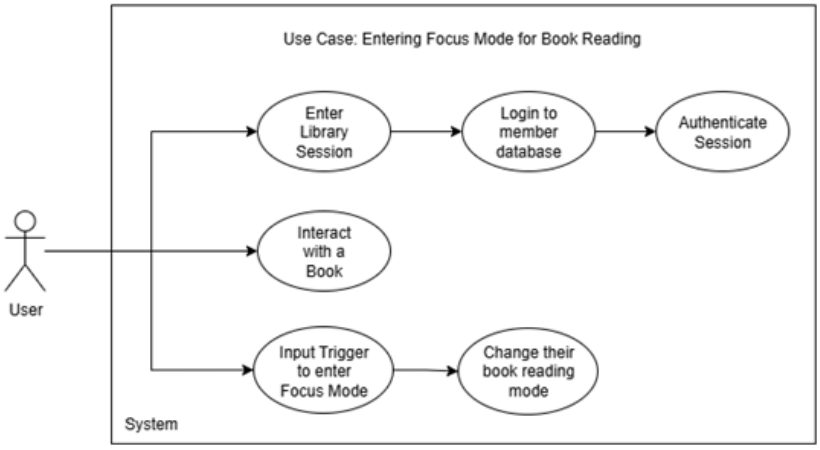
- The user must be logged into the system.

4.4.4 Postconditions:

- The brightness level is successfully changed for the user.

4.5 Use Case: Entering Focus Mode for Book Reading

Use Case Diagram:



4.5.1 Description

Allow user(s) to read books in a focused, full-screen mode instead of as a regular game model.

Actors:

- User

Triggers & Inputs:

- Be inside the library
- Log in to the member database
- Interact with a book
- Input the focus mode assigned key once the book is picked up

4.5.2 Flow of Events:

Basic Flow:

User	Action Number	System
Enter the library session	1	
Log in to the member database	2	Authentic session
Interact with a book (new or already being read)	3	
Input trigger condition to enter focus mode	4	Changes the user's view to feature a full-screen layout of the chosen book
User has entered focus mode	5	

Alternative Flow 1:

User	Action Number	System
	2.1	Login authentication failed.
		Unable to let the user access the library.

Alternative Flow 2:

User	Action Number	System
User has not picked up any book	2.2	
		Unable to change the view to focus mode.

4.5.3 Preconditions:

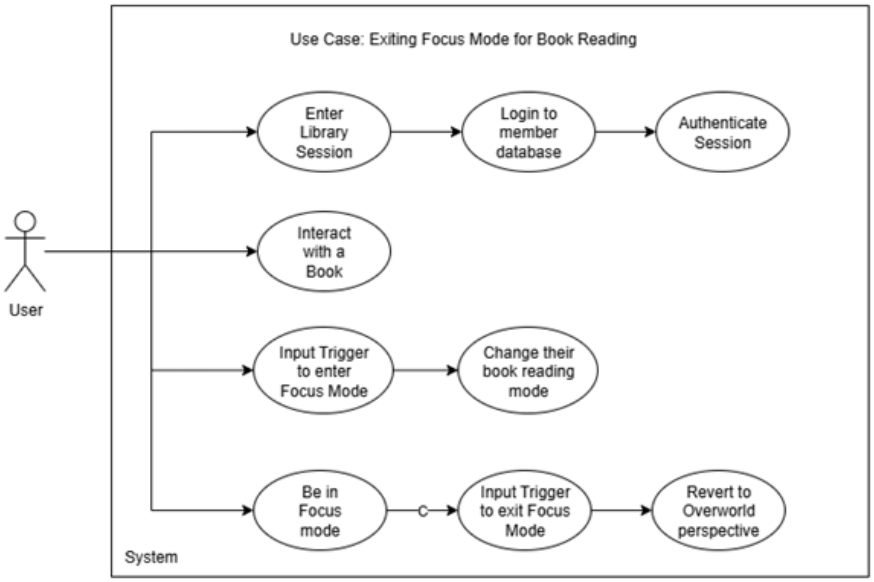
- The user must be logged into the system.

4.5.4 Postconditions:

- The user has entered focus mode to read the book.

4.6 Use Case: Exiting Focus Mode for Book Reading

Use Case Diagram:



4.6.1 Description

Allow user(s) to return from the focus mode to the overworld.

Actors:

- User

Triggers & Inputs:

- Be inside the library
- Log in to the member database
- Interact with the book
- Be in focus mode
- Press the assigned trigger to exit focus mode

4.6.2 Flow of Events:

Basic Flow:

User	Action Number	System
Enter the library session	1	
Log in to the member database	2	Authentic session
Interact with a book (new or already being read)	3	
Input trigger condition to enter focus mode	4	Changes the user's view to feature a full-screen layout of the chosen book
User has entered focus mode	5	
Input trigger condition to exit focus mode	6	Changes the user's view back to the overworld perspective
User has exited focus mode	7	

Alternative Flow 1:

User	Action Number	System
	2.1	Login authentication failed.
		Unable to let the user access the library.

4.6.3 Preconditions:

- The user must be logged into the system.
- The user must already be in focus mode.

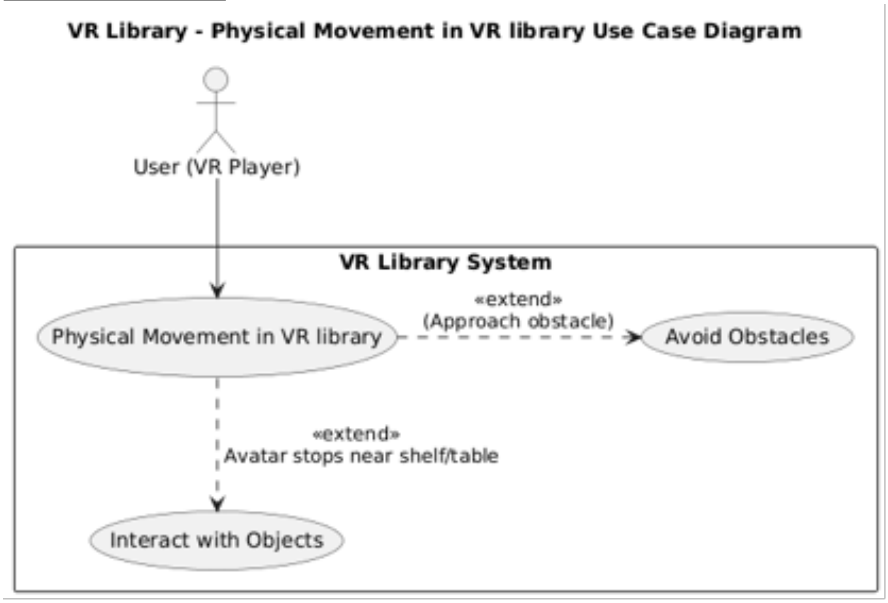
4.6.4 Postconditions:

- The user has exited focus mode.

5 User (Inter-)Actions

5.1 Physical Movement in VR library

Use Case Diagram:



5.1.1 Description

The goal of this use case is to track the movement of a user in the VR library and limit its access to walk over the object or library limit.

Actors:

- User (VR player)

Triggers & Input

- Triggers: The user physically moves within the boundaries of the library.
- Input: 'A', 'S', 'D', 'W'

5.1.2 Flow of Events:

Basic Flow:

Actor 1	Action Number	System
The user physically walks or moves within designated area.	1	
	2	The VR system continuously tracks the user's position and orientation in real time.
	3	The tracked movement data is translated into corresponding movement inside the virtual library environment.
	4	The user's virtual avatar's position and orientation update seamlessly to reflect real-world movement.

alternative flows 1:

Actor 1	Action Number	System
The user moves towards the obstacle in the virtual library.	1	
	2	As the user approaches the obstacle, the VR system detects the boundary of the obstacle.
	3	The system restricts further forward movement once the user reaches the allowed minimum distance from the obstacle.

alternative flows 2:

Actor 1	Action Number	System
The user's avatar stops in front of the table or shelf.	1	
	2	System enables user interactions like looking, reading, or selecting objects placed on the obstacle, but blocking any attempt to pass through.

5.1.3 Special Requirements:

5.1.4 Preconditions:

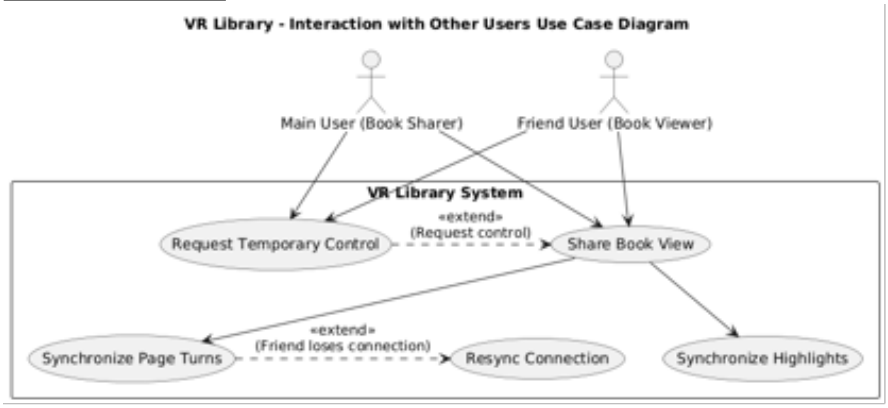
- The VR system is calibrated to the physical play space.
- The VR headset and tracking devices are active and functioning.

5.1.5 Postconditions:

- The user experiences natural and immersive navigation in the VR library by physically moving.
- The virtual environment remains synchronized with the user's physical movement.

5.2 Interaction with Other Users - Sharing Book View in VR Library

Use Case Diagram:



5.2.1 Description

The goal of this use case is to create a flow of sharing a book with friend user and create possible flow for both users.

Actors:

- User (VR player)

Triggers & Input

- Triggers: The main user or friend initiates sharing book access.
- Input:

5.2.2 Flow of Events:

Basic Flow:

Actor 1	Action Number	System
User or friend sends or select the option to share the book.	1	
	2	The system streams the main user's book view state to the friend user(s).
Friend user(s) receive and view the same book content and page as the main user.	3	
The main user turns pages.	4	
	5	Making the friend user(s) view synchronized page turns.
The main user highlights any part of the book.	6	
	7	instantly appear in the friend user's view.

Alternative Flow 1:

Actor 1	Action Number	System
Friends user loses connection	1	
	2	The system attempts to resync when reconnected.

Alternative Flow 2:

Actor 1	Action Number	System
Friend user may request control temporarily to scroll or highlight	1	
	2	The system asks the user to try to grant permissions.
	3	main user grants or denies permission.

5.2.3 Special Requirements:

5.2.4 Preconditions:

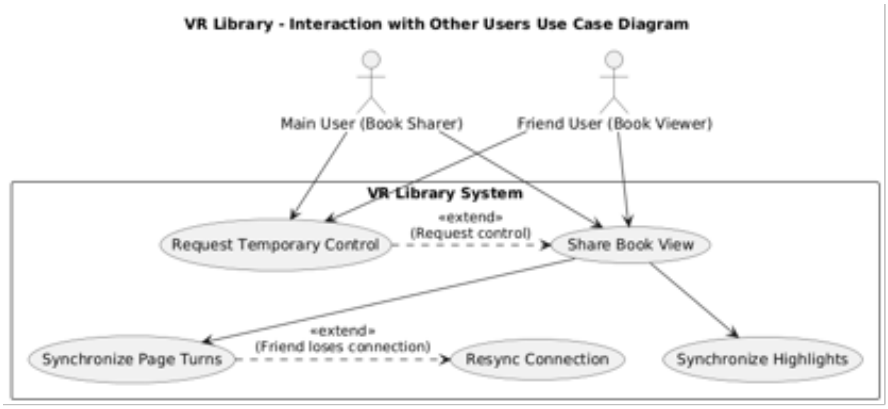
- Both users are logged into the VR library system.
- another user is close enough to request or accept the book sharing action
- The main user has a book open for reading.

5.2.5 Postconditions:

- Book view and highlights are synchronized between users in real-time.
- Users have a shared reading experience, enhancing collaboration.
- The system logs the shared session for history or future reference.

5.3 Interaction with Blackboard

Use Case Diagram:



5.3.1 Description

The goal of this use case is to create a possible action of a user with Blackboard.

Actors:

- Main User (Book Sharer)
- Friend User (Book Viewer)

Triggers & Input

- Triggers: User selects the Blackboard.
- Input:

5.3.2 Flow of Events:

Basic Flow:

Actor 1	Action Number	System
User walks over to white board	1	
	2	The system displays the pointer option in the board.
User clicks on board	3	
	4	The system focus the user screen on the board and marker option pops up.
User edit the Blackboard	5	
	6	Make other users screen synchronized

Alternative Flow 1:

User	Action Number	System
User clicks on board	1	
	2	The system throws the error “white board is currently being used by someone else”

Alternative Flow 2:

User	Action Number	System
User loses connection while editing	1	
	2	The system attempts to resync when reconnected.

5.3.3 Special Requirements:

5.3.4 Preconditions:

- The VR headset and tracking devices are active and functioning.

5.3.5 Postconditions:

- - The system remains synchronized with the user’s markup in Blackboard.