**LibCore Library Management System (LMS) - Use Case Specifications**

**Use Case ID: UC001**

**Use Case Name: User Registration**

**Relevant Requirements:** SR3  
**Primary Actor:** Library Member  
**Pre-conditions:** The user must not already have an account.  
**Post-conditions:** A new user account is created, and login credentials are issued.

**Basic Flow or Main Scenario:**

1. The user opens the **LibCore LMS application in Eclipse IDE**.
2. The user navigates to the **registration form**.
3. The user enters personal details (**name, address, contact number, and unique membership ID**).
4. The system validates the entered information.
5. The system creates a new user account **and stores the details in a structured text file**.
6. The system sends confirmation and login credentials to the user.

**Extensions or Alternate Flows:**

* If a **membership ID is already in use**, the system notifies the user and prompts re-entry.
* If **required fields are left empty**, an error message is displayed.

**Exceptions:**

* **File system failure** prevents storing the new user.
* **Invalid input formatting** causes a failed registration attempt.

**Use Case ID: UC002**

**Use Case Name: User Login**

**Relevant Requirements:** SR4  
**Primary Actor:** Library Member, Library Staff  
**Pre-conditions:** The user must have a registered account stored in the text file.  
**Post-conditions:** The user gains access to system functionalities.

**Basic Flow or Main Scenario:**

1. The user **opens the LibCore LMS application in Eclipse IDE**.
2. The user enters their **username and password**.
3. The system **reads the stored credentials from the text file**.
4. If valid, the **user is granted access**.
5. If invalid, **an error message is displayed**.

**Extensions or Alternate Flows:**

* If the password is incorrect multiple times, the **account is temporarily locked**.
* If the user forgets their password, they can **request a reset**, which updates the text file.

**Exceptions:**

* **Text file missing or corrupted** prevents authentication.
* **Invalid user credentials formatting causes a login failure**.

**Use Case ID: UC003**

**Use Case Name: Search for Resources**

**Relevant Requirements:** SR6  
**Primary Actor:** Library Member  
**Pre-conditions:** User must be logged in.  
**Post-conditions:** The system displays search results based on user query.

**Basic Flow or Main Scenario:**

1. The user enters a **search query (title, author, category)**.
2. The system **parses the resource text file** to find matching results.
3. The system **displays available resources**.

**Extensions or Alternate Flows:**

* If no matching resources are found, the **system notifies the user**.

**Exceptions:**

* **File read failure prevents search execution**.

**Use Case ID: UC004**

**Use Case Name: Borrow Resource**

**Relevant Requirements:** SR7, SR8  
**Primary Actor:** Library Member  
**Pre-conditions:** The user must be logged in and the resource must be available.  
**Post-conditions:** The system updates the resource status to "Checked Out" and records the loan transaction.

**Basic Flow or Main Scenario:**

1. The user selects an **available resource**.
2. The system verifies the **user’s borrowing limit** by reading from the text file.
3. The system **updates the resource status in the text file**.
4. The system **records the transaction in a structured text log**.
5. The system **notifies the user of the due date**.

**Extensions or Alternate Flows:**

* If the **user has overdue books, borrowing is restricted**.
* If the **resource is reserved by another user, borrowing is not permitted**.

**Exceptions:**

* **File system failure prevents the transaction from being recorded**.

**Use Case ID: UC005**

**Use Case Name: Return Resource**

**Relevant Requirements:** SR13  
**Primary Actor:** Library Member  
**Pre-conditions:** The user must have a borrowed resource.  
**Post-conditions:** The resource status is updated to "Available".

**Basic Flow or Main Scenario:**

1. The user selects the **resource to return**.
2. The system **verifies the transaction using the text file**.
3. The system **updates the resource status to "Available"**.
4. The system **records the return transaction in a structured text log**.
5. The system **notifies the user of any overdue fees**.

**Extensions or Alternate Flows:**

* If the **resource is damaged, the system prompts a damage fee**.

**Exceptions:**

* **File system failure prevents status update**.

**Use Case ID: UC006**

**Use Case Name: Overdue Notification**

**Relevant Requirements:** SR9  
**Primary Actor:** System  
**Pre-conditions:** A resource must be overdue.  
**Post-conditions:** The user is notified about the overdue resource.

**Basic Flow or Main Scenario:**

1. The system **checks the text file for overdue resources**.
2. The system **sends notifications by updating a notification log**.
3. The system **updates the notification log file**.

**Extensions or Alternate Flows:**

* If the **user does not respond, additional reminders are logged**.

**Exceptions:**

* **File corruption prevents notifications from being logged**.