LIBcore

Library
Management
Software

LIBcore

Phase 1

Alejandro Fernandez

Steven Le

Thomas Tran

Kartik Tripathi (KT)

Phase 1 Overview

Requirements:

- User Authentication & Registration, Session Management
- Resource Catalog & Search, Borrowing & Returning Resources
- Loan History & Tracking, Overdue Notifications
- Inventory Management, Role-Based Access Control
- File-Based Storage System, Data Consistency and Logging

Classes:

- User
- Resource
- Loan Transaction
- LoanHistory
- InventoryManagement

Requirements #1 and #2

User Authentication & Registration

- Users must register by providing **name, address, contact number, and unique membership ID**.
- System **validates input**, creates an account, and **stores user data in a structured text file**.
- If an **ID is already in use or a required field is empty**, an error message is displayed.

Session Management

- **User sessions expire** after a predefined period of inactivity.
- Ensures **secure authentication** and prevents unauthorized access.
- **Managed via text file-based session tracking**.

Requirements #3 and #4

Resource Catalog & Search

- Users can search for books, magazines, and digital media by **title, author, category, and availability**.
- System **parses resource text files** for matching results.
- If no matching resources are found, the system **notifies the user**.

Borrowing & Returning Resources

- Users can request to **borrow available resources**.
- System verifies borrowing limits **from the text file**.
- Borrowed books are **marked as "Checked Out" in the text file**.

Requirements #5 and #6

Loan History & Tracking

- The system logs all **loan transactions in a structured text file**.
- Members can view their **borrowing history by reading stored logs**.
- Staff can track **frequently borrowed items and overdue books**.

Overdue Notifications

- The system **checks loan records** to identify overdue resources.
- Overdue books trigger **automatic notifications stored in a log file**.
- Additional reminders are logged **until the book is returned**.

Requirements #7 and #8

Inventory Management

- Staff can **add, edit, and remove resources** from the system.
- Changes are **reflected in the text file storage**.
- Includes tracking of **availability status and loan counts**.

Role-Based Access Control

- Members can only **borrow and return books**.
- Staff can **approve loan requests, manage inventory, and override policies**.
- Enforced using **user roles stored in text files**.

Requirements #9 and #10

File-Based Storage System

- Instead of a traditional database, the system stores **all data in structured text files**.
- Separate files for **users, resources, and transactions**.
- Data is read and written using **structured file I/O operations**.

Data Consistency and Logging

- Each transaction (borrow, return, registration) is **appended to log files**.
- The system **ensures consistency** by locking files during write operations.
- Regular **backups and error-handling mechanisms** prevent data corruption.

Class - User

Attributes:

- userID: String
- name: String
- address: String
- contactNumber: String
- role: UserRole

Methods:

- register(): boolean
- login(): boolean
- logout(): void
- saveToFile(): void
- Setters and getters

Class - Resource

Attributes:

- resourceID: String
- title: String
- author: String
- category: String
- publicationDate: Date
- availability: Boolean

Methods:

- search(title: String): List
- requestBorrow(memberID: String): boolean
- returnResource(): boolean
- saveToFile(): void
- Setters and getters

Class - LoanTransaction

Attributes:

- transactionID: String
- memberID: String
- resourceID: String
- issueDate: Date
- dueDate: Date
- status: LoanStatus

Methods:

- checkout(): void
- returnResource(): void
- logTransaction(): void
- Setters and getters

Class - LoanHistory

Attributes:

- transactionLog: List

Methods:

- viewHistory(memberID: String): List
- viewOverdue(memberID: String): List
- saveToFile(): void
- Setters and getters

Class - InventoryManagement

Attributes:

- resourceList: List

Methods:

- addResource(resource: Resource): void
- editResource(resourceID: String, details: Resource): void
- removeResource(resourceID: String): void
- saveToFile(): void
- Setters and getters

Any Questions or Feedback?

Thank You!