NAME: RAJARSHI GHOSH

ROLL: 2023000218

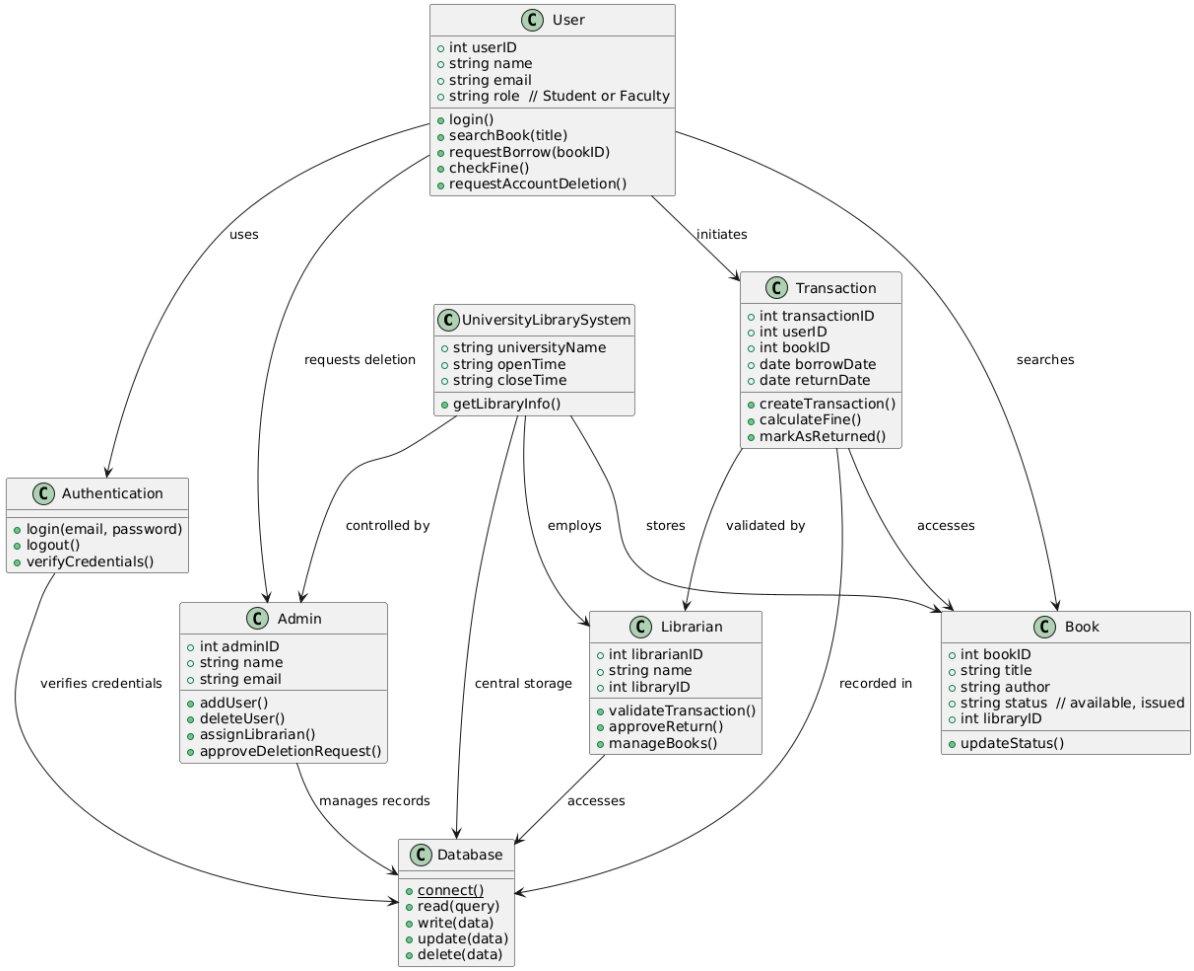
Date: 23-07-25

University Library System (ULS)

This document presents the development plan of the University Library System (ULS) following the **Software Development Lifecycle (SDLC)** methodology. The system is intended to streamline the management of library resources, improve user experience, and support the university's library operations.

Class Diagram Documentation: University Library System

This document outlines the software classes, including their attributes, methods, and relationships, for the University Library System as depicted in the provided diagram.



Class Descriptions

User

Represents an end-user of the library, such as a **Student** or **Faculty** member.

- **Attributes:**
    - int userID: Unique identifier for the user.
    - string name: The user's full name.
    - string email: The user's email address.
    - string role: The user's designated role (e.g., Student).
- **Methods:**
    - login(): Initiates the session for the user.
    - searchBook(title): Searches the catalog for a specific book.
    - requestBorrow(bookID): Makes a request to borrow a book.
    - checkFine(): Checks the user's account for outstanding fines.
    - requestAccountDeletion(): Submits a request for account removal.
- **Relationships:**
    - **Uses** the Authentication class for logging in.
    - **Initiates** a Transaction when borrowing a book.
    - **Requests deletion** from the UniversityLibrarySystem.
    - **Searches** for a Book.

Librarian

Represents a library staff member responsible for managing transactions and books.

- **Attributes:**
    - int librarianID: Unique identifier for the librarian.
    - string name: The librarian's name.
    - int libraryID: The ID of the library branch where the librarian works.
- **Methods:**

    - validateTransaction(): Approves and processes a borrowing transaction.
    - approveReturn(): Processes the return of a book.
    - manageBooks(): Handles the addition or removal of books from the collection.
- **Relationships:**
    - **Is employed by** the UniversityLibrarySystem.
    - **Validates** a Transaction.
    - **Accesses** the Database to manage records.

Admin

Represents a system administrator with high-level control over the system and user management.

- **Attributes:**
    - int adminID: Unique identifier for the admin.
    - string name: The admin's name.
    - string email: The admin's email address.
- **Methods:**
    - addUser(): Creates a new user account.
    - deleteUser(): Removes a user account.
    - assignLibrarian(): Assigns a librarian to a library.
    - approveDeletionRequest(): Approves a user's request for account deletion.
- **Relationships:**
    - **Controls** the UniversityLibrarySystem.
    - **Manages records** in the Database.

UniversityLibrarySystem

The main class representing the library system's core entity and information.

- **Attributes:**
    - string universityName: The name of the host university.

    - string openTime: The library's daily opening time.
    - string closeTime: The library's daily closing time.
- **Methods:**
    - getLibraryInfo(): Retrieves general information about the library.
- **Relationships:**
    - **Employs** the Librarian.
    - **Is controlled by** the Admin.
    - A User **requests deletion** from it.

Book

Represents a single physical or digital book in the library's catalog.

- **Attributes:**
    - int bookID: Unique identifier for the book.
    - string title: The title of the book.
    - string author: The author's name.
    - string status: The current availability (e.g., "available", "issued").
    - int libraryID: The ID of the home library branch.
- **Methods:**
    - updateStatus(): Changes the book's status (e.g., after being borrowed or returned).
- **Relationships:**
    - Is **searched** by a User.
    - Is **accessed by** a Transaction.

Transaction

Represents a record of a book being borrowed.

- **Attributes:**
    - int transactionID: Unique identifier for the transaction.

    - int userID: The ID of the borrowing user.
    - int bookID: The ID of the borrowed book.
    - date borrowedDate: The date the book was checked out.
    - date returnDate: The calculated due date for the book.
- **Methods:**
    - createTransaction(): Generates a new record for a loan.
    - calculateFine(): Determines the fine if the book is overdue.
    - markAsReturned(): Updates the transaction upon the book's return.
- **Relationships:**
    - Is **initiated by** a User.
    - Is **validated by** a Librarian.
    - **Accesses** the Book class for details.

Authentication

A dedicated class to handle user login, logout, and credential verification.

- **Attributes:** None specified.
- **Methods:**
    - login(email, password): Manages the user login sequence.
    - logout(): Manages the user logout sequence.
    - verifyCredentials(): Validates user credentials.
- **Relationships:**
    - Is **used by** the User.
    - **Verifies credentials** against the Database.

Database

Represents the back-end data storage system.

- **Attributes:** None specified.
- **Methods:**

    - connect(): Opens a connection to the database.
    - read(query): Retrieves data.
    - write(data): Inserts new data.
    - update(data): Modifies existing data.
    - delete(data): Removes data.
- **Relationships:**
    - Is **accessed by** the Librarian.
    - Is **managed by** the Admin.
    - Used by Authentication to **verify credentials**.