Library Management System Report

Comprehensive Overview of Classes and Features

Mohamad Imad Kassem

EECE332 Project

- Table of Contents
  - Overview of Classes
  - Explanation of Dynamic Behavior
  - Features and Functionalities
  - OOP Concepts
  - Sample Outputs

Overview of Classes

The Library Management System consists of Java classes that represent an element or a function in the system. Below is a breakdown of classes, where there's a brief description about each one.

- **Admin**
  Represents administrator of a library. Admin can delete, add, update books, rules configuration, view all users, view logs, and usage statistics viewing.
- **Student**
  Represents student (user type). Has an id, and can borrow books, reserve resources, and view borrowing history.
- **Professor**
  Represents professor (user type). Has a rank, Same as Student but with more permissions like renewing books.
- **Book**
  Represents a book resource. It tracks availability, book renewal status, borrowing history, and reservation list of a book through a queue.
- **Library**
  Represents a library within a faculty. Manages books, users, meeting rooms, and electronic resources. Can search for books, provide list of overdue books, and generate statistics about most popular books and most frequent borrowers.
- **Faculty**
  Represents a faculty. Tracks associated libraries and borrowing rules, where it manages borrowing rules for resource types and user roles.
- **MeetingRoom**
  Represents a reservable meeting room resource in the library, and tracks reservation duration, where it prevents overbooking.
- **PC**
  Represents a PC resource in the library's electronic resources. It tracks usage duration and automatically releases resources after the allowed time.
- **Tablet**
  Represents a tablet resource in the library's electronic resources.
- **ElectronicResources**
  Abstract base class for electronic resources like PCs and tablets.

- **User**
  Abstract base class for Student and Professor. Tracks borrowing history and penalties. It also tracks number of items borrowed by a user, along with a penalty system for late returns, and prevents further borrowing during penalty period.
- **BorrowingRecord**
  Represents a record of a borrowing transaction, including dates and overdue status. It automatically determine if a borrowing record is overdue. It tracks due dates, borrow dates, and return dates for each record.
- **Rules**
  Setting rules for borrowing durations and renewal options.

Some Program features

- **Penalty System**
  Users receive penalties for late returns, preventing them from borrowing new books for a specified period.
- **Renewal Feature**
  Professors can renew borrowed books once, extending the due date.
- **Overdue Books Tracking**
  The system identifies overdue books and applies penalties accordingly.
- **Multiple Copies of Books**
  Books have multiple copies, and the system manages availability per copy.
- **Logging System**
  All significant actions (borrowing, returning, reservations, renewals) are logged for administrative review.
- **Notifications**
  Users are notified when a reserved book becomes available.
- **Faculty-Specific Rules**
  Borrowing rules are configurable per faculty, resource type, and user role.
- **Dynamic Borrowing Rules**
  The system allows admins to set and modify borrowing durations and permissions.
- **User Restrictions**
  Users can only interact with libraries belonging to their own faculty.

Dynamic Code Behavior

The system employs dynamic behavior through inheritance and polymorphism. For example:

- The `User` class serves as a parent class for `Student` and `Professor`. The system treats users generically while applying specific behaviors for students and professors.
- The `ElectronicResources` class is a parent class for `PC` and `Tablet`. This abstraction enables handling of different electronic resources while maintaining unique attributes and methods for each type.

This behavior makes code more flexible and scalable for management.

Features and Functionalities

The system includes the following features, demonstrating its capabilities and dynamic behavior:

- Borrowing and returning books, with validation of user permissions and overdue handling.
- Reservation system for books, with a waiting list for popular resources.
- Usage statistics generation for administrators, highlighting popular books and frequent borrowers.
- Support for electronic resource usage (e.g., PCs, tablets), with time limits and availability checks.
- Reservation and management of meeting rooms for students and professors.

OOP Concepts

- Private and protected Attributes, and public getters/setters
- Inheritance: User class serves as superclass for Student and Professor
- Abstraction: Electronic Resources: abstract class for PC and Tablet
- Method overriding: Student and Professor override methods from User
- Generics: collections in library class use generics to ensure type safety and avoid casting

```java
public class user {

    //attributes
    protected String name;
    protected String address;
    protected int phoneNumber;
    protected String emailAddress;
    protected faculty faculty;
    private int borrowCount;
    private List<book> reservedBooks;
    private Date penaltyEndDate;

    //borrowing history
    private List<borrowingRecord> borrowingHistory;
```

```java
//getters
public String getname() {return this.name;}
public String getaddress() {return this.address;}
public int getphoneNumber() {return this.phoneNumber;}
public String getemailAddress() {return this.emailAddress;}
public faculty getfaculty() {return this.faculty;}

//setters
public void setName(String nm) {this.name = nm;}
public void setAddress(String adr) {this.address = adr;}
public void setPhoneNumber(int nbr) {this.phoneNumber = nbr;}
public void setEmailAddress(String email) {this.emailAddress = email;}
public void setFaculty(faculty fty) {this.faculty = fty;}
```

```java
public class student extends user{
```

```java
public abstract class electronicResources {
```

```java
public class pc extends electronicResources{
```

```java
public class library {

    //attributes
    private String name;
    private String location;
    private faculty faculty;
    private List<book> books;
    private List<user> users;
    public List<meetingRoom> meetingRooms;
    public List<electronicResources> electronicResources;
    private List<String> logs = new ArrayList<>();
```

Sample Outputs:

- Adding books, meeting rooms, electronic resources

```
Book added: Introduction to Algorithms
Book added: Design Patterns
Meeting room added: Conference Room 1A
Meeting room added: Conference Room 1B
Electronic resource added: Dell
Electronic resource added: Apple
```

- Setting borrowing rules for book, meeting rooms, electronic resources with students and professors

```
Set borrowing duration for book and student: 3 days at faculty Engineering
Set borrowing duration for book and professor: 7 days at faculty Engineering
Set renewal option for book and student: not allowed at faculty Engineering
Set renewal option for book and professor: allowed at faculty Engineering
```

- Search

```
--- Students and Professors Search for Books ---

Engineering Faculty:

Science Faculty:

Arts Faculty:
- Found Book: Head First Design Patterns by Eric Freeman

--- End of Search Results ---
```

- Borrowing

```
--- Students and Professors Borrow Books Multiple Times ---
Book borrowed: Design Patterns by Alice
Book borrowed: Artificial Intelligence: A Modern Approach by Bob
Book borrowed: Head First Design Patterns by Dr. Williams
Book not available or user not authorized!
User Dr. Smith added to reservation list for book: Design Patterns
Book not available or user not authorized!
User Bob added to reservation list for book: Artificial Intelligence: A Modern Approach
```

- Returns

```
--- Simulating Returns and Overdue Books ---
Late penalty applied to Alice until Fri Dec 06 15:15:03 EET 2024
Late return! Alice cannot borrow books until Fri Dec 06 15:15:03 EET 2024
Book returned: Design Patterns by Alice
Notified user: Dr. Smith that book is now available.
Late penalty applied to Dr. Smith until Fri Dec 06 15:15:03 EET 2024
Late return! Dr. Smith cannot borrow books until Fri Dec 06 15:15:03 EET 2024
```

- Statistics

```
--- Generating Usage Statistics for Engineering Library ---
Total Number of Borrowings: 2

Popular Books:
- Introduction to Algorithms borrowed 1 times
- Design Patterns borrowed 1 times

Frequent Borrowers:
- Alice borrowed 1 times
- Dr. Smith borrowed 1 times

Overdue Books:
```