Set A

**BRAC UNIVERSITY**

**Department of Computer Science and Engineering**

Examination: Quiz 2 Duration: 30 min

Semester: Fall 2024 Full Marks: 15

**CSE 470: Software Engineering**

| Name: | ID: | Section: |
| --- | --- | --- |

The bustling Modern City Library implemented a robust Library Management System to handle its growing collection of 20,000 books and its 5,000 active members. The system revolves around several key entities: **Library**, **Book**, **Member**, **Transaction**, and **Librarian**, each playing a pivotal role in ensuring smooth operations.

At the heart of the system is the **Library**, which acts as a central hub. Its attributes include libraryName, address, books, members, and transactions. The library provides functions such as searching books and showing a list of books available for borrowing.

The **Book** class represents each book in the library's collection. It has attributes like title, author, ISBN, and availability. A book can be marked as borrowed or returned to help in tracking whether a book is in circulation. The **Book** class is related compositionally to the **Library** in a way that the books cannot exist outside the library's domain.

Members of the library are represented by the **Member** class, which aggregates into **Library**. Each member has attributes like memberID, name, contactInfo, and membershipStatus. Members can borrow books and return them. The **Member** class depends on **Transaction** to record the borrowing and returning activities.

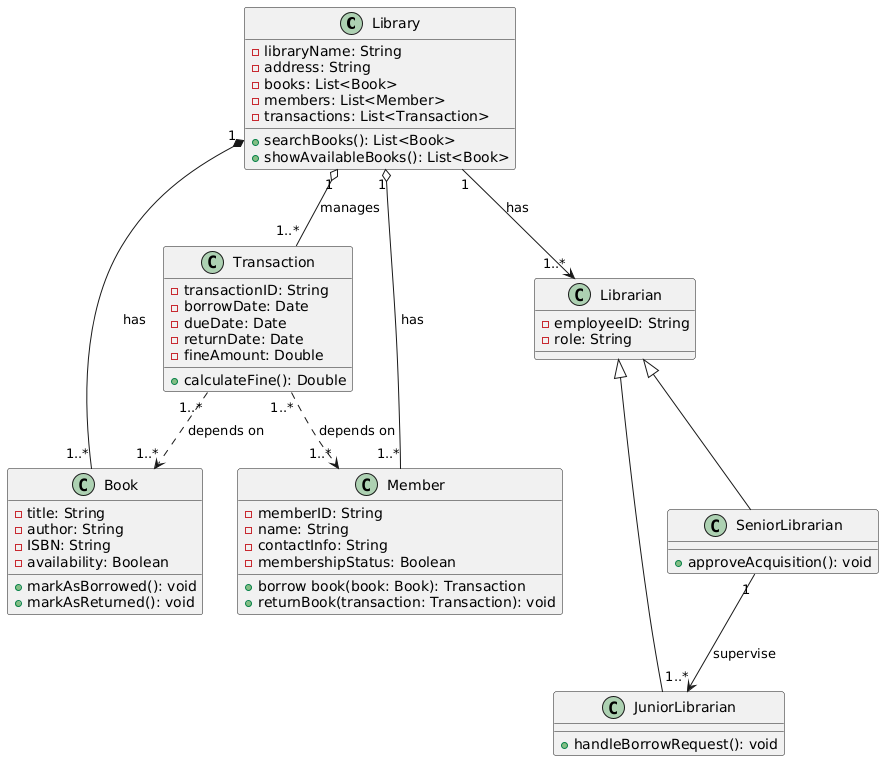
The **Transaction** class links books and members through attributes such as transactionID, borrowDate, dueDate, returnDate, and fineAmount. It determines fines for overdue returns. Transactions aggregate into **Library** but depend on **Book** and **Member** for their operation, highlighting the relationships between borrowing and returning activities.

The **Librarian** class oversees library operations, generalized into **Junior Librarian** and **Senior Librarian**. Common attributes include employeeID and role. Junior Librarian focuses on basic tasks like handling borrow requests, while Senior Librarian performs advanced functions like approving acquisitions and supervising junior librarians.

This tightly integrated system ensures that books, members, and transactions are managed seamlessly while enabling efficient communication between the different entities. This dependency between classes ensures data consistency and operational efficiency, making the Modern City Library a model of digital transformation in public services.

1. Design a UML class diagram from the above scenario. [10]
2. Write down three functional and two non-functional requirements from the given scenario. [5]

**Solution:**



Set B

**BRAC UNIVERSITY**

**Department of Computer Science and Engineering**

Examination: Quiz 2 Duration: 30 min

Semester: Fall 2024 Full Marks: 15

**CSE 470: Software Engineering**

| Name: | ID: | Section: |
| --- | --- | --- |

Green Valley Hospital, a leading healthcare provider, implemented an advanced Hospital Management System (HMS) to efficiently manage its operations. The system involves several key entities: **Hospital, Doctor, Patient, Appointment, MedicalRecord, Admin, Nurse, and Staff.**

At the core of the system is the **Hospital**, which serves as the central hub. The hospital has a list of doctors, patients, appointments, and medical records. It provides functions such as finding doctors based on their specialty and listing available slots for appointments.

The **Doctor** class represents the medical professionals employed by the hospital. Each doctor has attributes like doctorID, name, specialization, availability, and contactInfo. Doctors can schedule appointments for patients and update their availability. The **Hospital** aggregates the Doctor class, as doctors can exist independently but are an essential part of the hospital's operations.

**Patients** are central to the system, represented by the Patient class. Each patient has attributes like patientID, name, contactInfo, and medicalHistory. Patients can request appointments, and view their medical history.

The **Appointment** class links doctors and patients. It stores information about the appointment's date, time, and status. Patients can schedule appointments, while doctors also can manage appointments. The **Appointment** class depends on both the **Doctor** and **Patient** classes to function, as both are required to schedule an appointment.

The **MedicalRecord** class holds vital health data, a list of treatment details, and visit dates. Each patient has one or more medical records, which are updated as needed. The **Patient** class compositionally the MedicalRecord, meaning that the existence of a medical record is tightly bound to the patient.

The **Admin** class oversees administrative tasks such as adding new patients, and generating medical bills. Admins have common staff attributes like name, role, and employeeID which they inherit from the **Staff** class. The **Nurse** class also inherits from the **Staff** class, representing other hospital staff members who assist in patient care. Nurses can view patient records and administer medication.

This tightly integrated system ensures that the hospital operates smoothly, with clear dependencies and relationships between patients, doctors, staff, and medical records. This dependency between classes ensures data consistency and operational efficiency.

1. Design a UML class diagram from the above scenario. [10]
2. Write down three functional and two non-functional requirements from the given scenario. [5]

**Solution:** [add the multiplicity according to your intuition]

