**UML Exercise: Design a Library Management System**

**Problem Statement:**

You are required to design a **Library Management System** using UML diagrams. The system should manage books, library members, and the borrowing process. The system should have the following capabilities:

1. **Books:**
   * Each book should have a title, author, ISBN, and publication date.
   * Books can be available in different genres (e.g., Fiction, Non-Fiction, Science, etc.).
   * The library keeps track of the number of copies of each book.
2. **Members:**
   * Library members should have a name, membership ID, contact details, and membership expiration date.
   * Members can borrow books from the library.
   * Each member can borrow up to 5 books at a time.
   * The system should track which books are currently borrowed by each member.
3. **Borrowing Process:**
   * A member can borrow a book if it is available in the library.
   * The borrowing record should include the borrow date and the due date for return.
   * Members should be able to return books, and the system should update the availability of the book.
4. **Fines:**
   * If a member returns a book after the due date, they should be fined.
   * The fine should be calculated based on the number of days overdue.

**Tasks:**

1. **Class Diagram:**
   * Create a class diagram to represent the system, showing the classes, their attributes, methods, and the relationships between them.
2. **Use Case Diagram:**
   * Draw a use case diagram to identify the main actors (e.g., Librarian, Member) and the key use cases (e.g., Borrow Book, Return Book, Add New Book).
3. **Sequence Diagram:**
   * Create a sequence diagram to illustrate the process of a member borrowing a book, including interactions between the member, the book, and the library system.
4. **Activity Diagram:**
   * Design an activity diagram to show the workflow of returning a book, including the steps of checking for overdue fines and updating the book's availability.

**Deliverables:**

* Create your UML diagrams (Class Diagram, Use Case Diagram, Sequence Diagram, and Activity Diagram) with proper annotations.
* Provide a brief explanation of each diagram, describing the design decisions and how the system fulfills the requirements.