**Exercise 6: Online Bookstore - Exception Handling in REST Controllers**

Business Scenario:

Implement a global exception handling mechanism for the bookstore RESTful services.

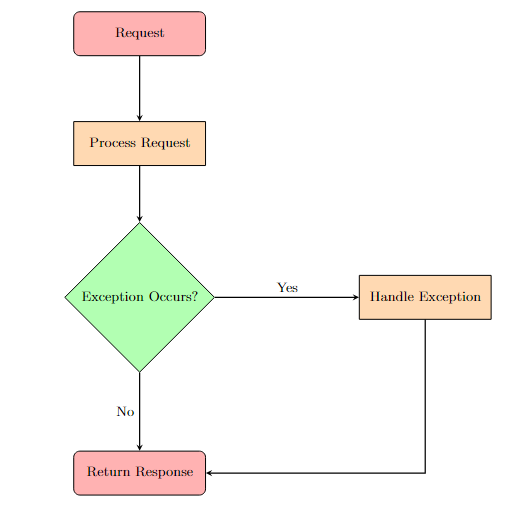
To enhance the robustness of the RESTful services in the Online Bookstore application, a global exception handling mechanism needs to be implemented. This will ensure that exceptions are consistently managed and appropriate HTTP status codes are returned to clients.

**Key Points**

**1. Global Exception Handler**

* **Purpose**: To handle exceptions thrown by any REST controller across the application and provide a standardized response.
* **Implementation**:
  + **Class**: GlobalExceptionHandler
  + **Annotation**: @ControllerAdvice - This annotation indicates that the class provides global exception handling across all controllers.
  + **Methods**:
    - **handleAllExceptions**:
      * **Purpose**: To handle any generic exception not specifically handled by other methods.
      * **Parameters**: Exception ex - The exception object.
      * **Return Type**: ResponseEntity<String> - Returns a response with the exception message and an HTTP status code of 500 Internal Server Error.
  + **HTTP Status Codes**:
    - **500 Internal Server Error**: Used for unhandled exceptions to indicate that the server encountered an unexpected condition.

**FLOWCHART :**



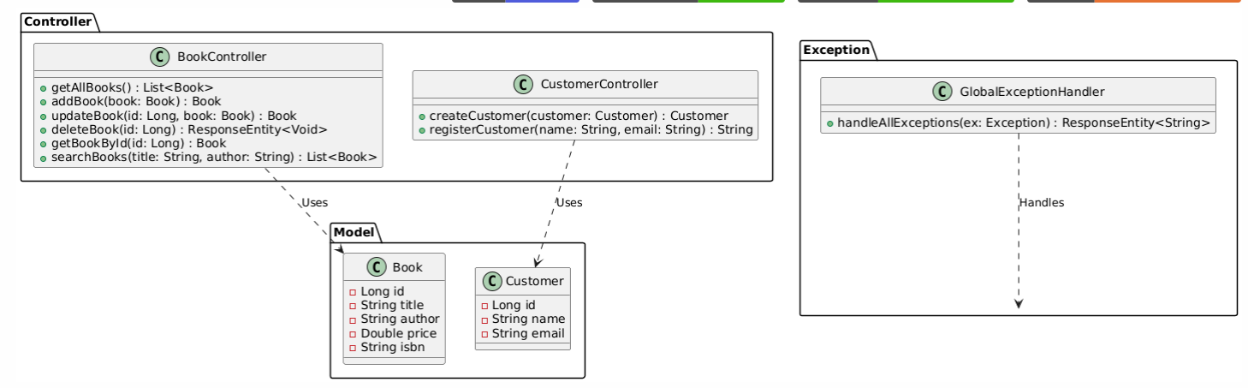
**Explanation:**

1. **Node Distance**: Adjusted the node distance to 2.5cm and added yshift and xshift to better position the nodes and avoid overlapping arrows.
2. **Start Node**: Represents the incoming request.
3. **Process Request**: Represents processing of the request.
4. **Exception Occurs?**: Decision node to check if an exception occurred.
5. **Handle Exception**: Node where the exception is handled if one occurs.
6. **Return Response**: Final node where a response is returned to the client.

**Arrows**:

* **Request** to **Process Request**.
* **Process Request** to **Exception Occurs?**.
* **Exception Occurs?** to **Handle Exception** if "Yes", and to **Return Response** if "No".
* **Handle Exception** to **Return Response** with a right-angle connection to ensure clarity.

**CLASS DAIGRAM :**



**Explanation:**

1. **Book**: Represents a book entity with attributes like id, title, author, price, and isbn.
2. **Customer**: Represents a customer entity with attributes like id, name, and email.
3. **BookController**: Manages CRUD operations for Book entities and exposes RESTful endpoints.
4. **CustomerController**: Manages operations related to Customer entities and exposes RESTful endpoints.
5. **GlobalExceptionHandler**: Handles exceptions globally and provides a response.

**Relationships**:

* BookController and CustomerController use Book and Customer classes respectively.
* GlobalExceptionHandler handles Exception class.