General Project Guidelines

Project Scope:

- Your project should focus on implementing the core functionalities of the assigned system. Ensure you understand the problem statement thoroughly and plan your approach accordingly.
- The system must incorporate object-oriented design principles, including classes, objects, inheritance, encapsulation, polymorphism, and abstraction.

User Roles & Access Control:

- Each project should include at least three distinct user roles (e.g., Admin, Staff, Customer) with different levels of access and permissions.
- Implement a user authentication and authorization system, ensuring that users can log in and access features relevant to their roles.

Code Structure:

- Organize your code using packages and classes in a modular fashion. Separate functionality into different layers (e.g., business logic, data access, presentation).
- Follow SOLID principles and design patterns where applicable to maintain clean and scalable code.

Database Integration:

- Your project should integrate with a relational database for storing persistent data (e.g., users, transactions, inventory).
- Design and implement appropriate tables for different entities, ensuring proper relationships (e.g., one-to-many, many-to-many).
- Use JDBC (or an ORM framework) to interact with the database for CRUD operations.

UI & User Interaction:

- Implement a simple graphic user interface for user interaction. Make sure the UI is intuitive and user-friendly.
- Users should be able to navigate between different functionalities easily, and actions (e.g., viewing data, updating records) should be clear.

Exception Handling & Validation:

- Implement exception handling for errors (e.g., invalid input, database connectivity issues) to prevent crashes and ensure graceful error messages are displayed to users.
- Validate all user inputs to ensure data integrity (e.g., required fields, correct formats).

Testing:

- Write unit tests for the core components of your application to ensure the correctness of your logic.
- Test each user role separately to ensure that permissions are correctly enforced.

Version Control:

- Use Git for version control. Make sure to commit your code regularly and write meaningful commit messages.
- Ensure that your project is well-documented with README files, installation steps, and usage instructions.

Documentation:

- Document your code using comments, explaining the purpose of each class and method.
- Provide a brief description of the project's architecture, including class diagrams if necessary, to explain the overall structure of your code.

Team Collaboration (if applicable):

- If working in teams, assign tasks and ensure clear communication among members.
- Use collaborative tools (e.g., GitHub, Trello) to track progress and merge code effectively.