Requirements Analysis for Inventory System (Task 1)

1. Introduction

- **Purpose**: Describe the purpose of the inventory system (e.g., to manage products, stock, sales, and suppliers efficiently for a business).
- **Scope**: Outline what the system will cover (e.g., product management, stock tracking, sales reporting, low-stock notifications) and any limitations (e.g., no customer-facing features or payment processing).
- **Objective**: Enable efficient inventory management, reduce manual errors, and provide insights into stock levels and sales trends.

2. Stakeholders

- **Users**:
    - **Inventory Manager**: Manages stock levels, adds/updates products, and monitors low-stock alerts.
    - **Sales Staff**: Records sales transactions and views sales reports.
    - **Admin/Owner**: Oversees all operations, generates reports, and manages suppliers.
- **Other Stakeholders**: Suppliers (provide product details), IT team (system maintenance).

3. Key Features

List the core functionalities the system must support:

- **Product Management**:
    - Add, update, or delete product details (e.g., product ID, name, description, price, category).
- **Stock Management**:
    - Track stock levels for each product.
    - Update stock quantities (e.g., after receiving new stock or recording sales).
    - Notify when stock falls below a predefined threshold (e.g., 10 units).
- **Sales Management**:
    - Record sales transactions (e.g., product sold, quantity, date, total amount).
    - Generate sales reports for a specific period (e.g., last 30 days).
- **Supplier Management**:
    - Store supplier details (e.g., supplier ID, name, contact info, associated products).
- **User Management** (optional, if applicable):
    - Manage user roles (e.g., admin, manager, staff) with appropriate access levels.

4. Functional Requirements

Detail specific functionalities with clear requirements:

- **FR1: Add Product**:
    - Users can add a new product with details (e.g., name, description, price, initial stock).
    - Input validation: Ensure price is positive, stock is non-negative.
- **FR2: Update/Delete Product**:
    - Modify product details or remove products no longer offered.
- **FR3: Stock Management**:
    - Update stock levels after sales or restocking.
    - Automatic low-stock notifications (e.g., email or dashboard alert) when stock < threshold.
- **FR4: Sales Tracking**:
    - Record sales with details (e.g., product ID, quantity, date, total).
    - Generate a report summarizing sales over the last 30 days (e.g., total revenue, top-selling products).
- **FR5: Supplier Management**:
    - Add, update, or delete supplier information.
    - Link suppliers to products for restocking purposes.

5. Non-Functional Requirements

- **Performance**: System should handle up to 1,000 products and 100 daily transactions without lag.
- **Usability**: Intuitive interface for non-technical users (e.g., inventory managers).
- **Scalability**: System should accommodate future growth (e.g., additional product categories or users).
- **Security**: Role-based access control to restrict sensitive operations (e.g., only admins can delete products).
- **Reliability**: Ensure data integrity during stock updates and sales recording.

6. Assumptions and Constraints

- **Assumptions**:
    - Users have basic computer literacy.
    - Internet access is available for cloud-based systems (if applicable).
- **Constraints**:
    - Limited to inventory management (no e-commerce or payment integration).
    - Development timeline (as per project requirements).
    - Budget constraints (if specified).

7. Use Case Diagram (Reference)

- **Description**: A use case diagram will be created in Draw.io to visualize interactions between actors (e.g., Inventory Manager, Sales Staff, Admin) and system functions (e.g., Add Product, Update Stock, Generate Sales Report).
- **Actors**:

    - Inventory Manager
    - Sales Staff
    - Admin
    - System (for automated tasks like low-stock notifications)
- **Use Cases**:
    - Manage Products (Add, Update, Delete)
    - Manage Stock (Update, Monitor)
    - Record Sale
    - Generate Sales Report
    - Manage Suppliers
    - Receive Low-Stock Notification
- **Note**: The diagram will be exported as UseCaseDiagram.png and included in the GitHub repo.

8. Data Flow Diagram (Reference)

- **Description**: A data flow diagram (DFD) will be created in Draw.io to show how data moves through the system.
- **Key Processes**:
    - Input: Product details, stock updates, sales data, supplier info.
    - Processing: Validate inputs, update stock levels, calculate sales metrics, trigger notifications.
    - Output: Sales reports, low-stock alerts, updated product/supplier records.
- **Data Stores**:
    - Products (stores product details)
    - Stock (tracks quantities)
    - Sales (records transactions)
    - Suppliers (stores supplier info)
- **External Entities**:
    - Users (Inventory Manager, Sales Staff, Admin)
    - Suppliers (provide restocking data)
- **Note**: The DFD will be exported as DataFlowDiagram.png and included in the GitHub repo.

9. System Workflow

- **Example Workflow**:
    1. Inventory Manager adds a new product to the system.
    2. System validates and stores product data.
    3. Sales Staff records a sale, reducing stock levels.
    4. System checks stock and triggers a low-stock notification if needed.
    5. Admin generates a sales report for the last 30 days.
    6. Inventory Manager updates supplier details for restocking.

10. Risks and Mitigation

- **Risk**: Incorrect stock updates leading to data inconsistencies.
    - **Mitigation**: Implement transaction validation and rollback mechanisms.
- **Risk**: System downtime affecting sales recording.
    - **Mitigation**: Use reliable hosting and regular backups.
- **Risk**: User errors in data entry.
    - **Mitigation**: Provide input validation and clear error messages.

11. Future Enhancements

- Integration with barcode scanners for faster stock updates.
- Mobile app for remote access to inventory data.
- Analytics dashboard for real-time insights into stock and sales trends.

Instructions for Google Docs

1. **Create the Document**:
    - Open Google Docs and create a new document titled “Task 1: Inventory System Requirements Analysis.”
    - Use the structure above, with clear headings and subheadings.
    - Keep the content concise but comprehensive, aiming for 2–3 pages.
2. **Formatting**:
    - Use a professional font (e.g., Arial or Times New Roman, 12pt).
    - Include a title page with your name, project title, and date (June 26, 2025).
    - Use bullet points or numbered lists for clarity in sections like Key Features and Requirements.
    - Add a section at the end referencing the Draw.io diagrams (e.g., “See UseCaseDiagram.png and DataFlowDiagram.png in the GitHub repo for visual representations”).
3. **Export/Alternative**:
    - If submitting as a text file instead (e.g., Task1_Requirements.txt), copy the content into a plain text editor (e.g., Notepad) and maintain clear section headers using dashes or asterisks for readability.
    - Ensure the text file is well-organized and mirrors the Google Docs structure.
4. **GitHub Submission**:
    - Save the Google Docs file as a PDF (Task1_Requirements.pdf) or use the text file (Task1_Requirements.txt).
    - Commit the file to your GitHub repo on the task1 branch, along with the Draw.io diagrams (UseCaseDiagram.png and DataFlowDiagram.png).
    - Push to the task1 branch (git push origin task1) and create a pull request to the main branch, adding the reviewer as specified.

Notes

- **Diagrams**: Use Draw.io (diagrams.net) to create the use case and data flow diagrams. Ensure they are clear and labeled (e.g., actors, processes, data stores). Export as PNG files for submission.
- **GitHub**: Ensure your repo is public or accessible to the reviewer. Double-check that all required files are committed and the pull request is properly set up.
- **Clarity**: The requirements should be specific enough to guide Task 2 (database design) and Task 3 (prototype development). Avoid vague terms; use examples where possible (e.g., “stock threshold = 10 units”).