

**CSI1007 - Software Engineering Principles Laboratory**  
**Assessment – 1**

**Reg.No.:23MIC0059**  
**Name: Jayesh Pani**

**Title of the Project:** Inventory Management System

**Description of the Project:** An Inventory Management System is a software application designed to efficiently track and manage inventory levels, stock, orders, sales, and deliveries. This system enables businesses to automate inventory-related processes, reducing manual errors and optimizing stock management. Key features include real-time inventory tracking, stock alerts, order processing, and detailed reporting. It is particularly useful for streamlining operations in retail, warehouses, and supply chain management, ensuring products are available when needed while minimizing overstock or stockouts. The system can be implemented using technologies like SQL for database management and frameworks like Django, Flask, or MERN for user interface and functionality.

**Scope of the Project:** To develop a secure and user-friendly Inventory Management System (IMS) that automates inventory tracking, streamlines order management, and integrates with external systems to improve operational efficiency.

**Impact of the Developing Inventory Management System Project:**

Economic Impact:

- Reduces operational costs and minimizes waste through automated inventory tracking.
- Increases productivity and profitability by improving order management.
- Enhances competitiveness by providing better service and accuracy.

Social Impact:

- Creates job opportunities in development, support, and training.
- Improves customer satisfaction with better stock availability and faster delivery.
- Increases employee focus on strategic roles, improving workplace satisfaction and safety.

Technological Impact:

- Promotes innovation through modern technologies like real-time tracking and analytics.
- Facilitates system integration with ERP, POS, and suppliers.
- Supports data-driven decision-making with real-time reporting.

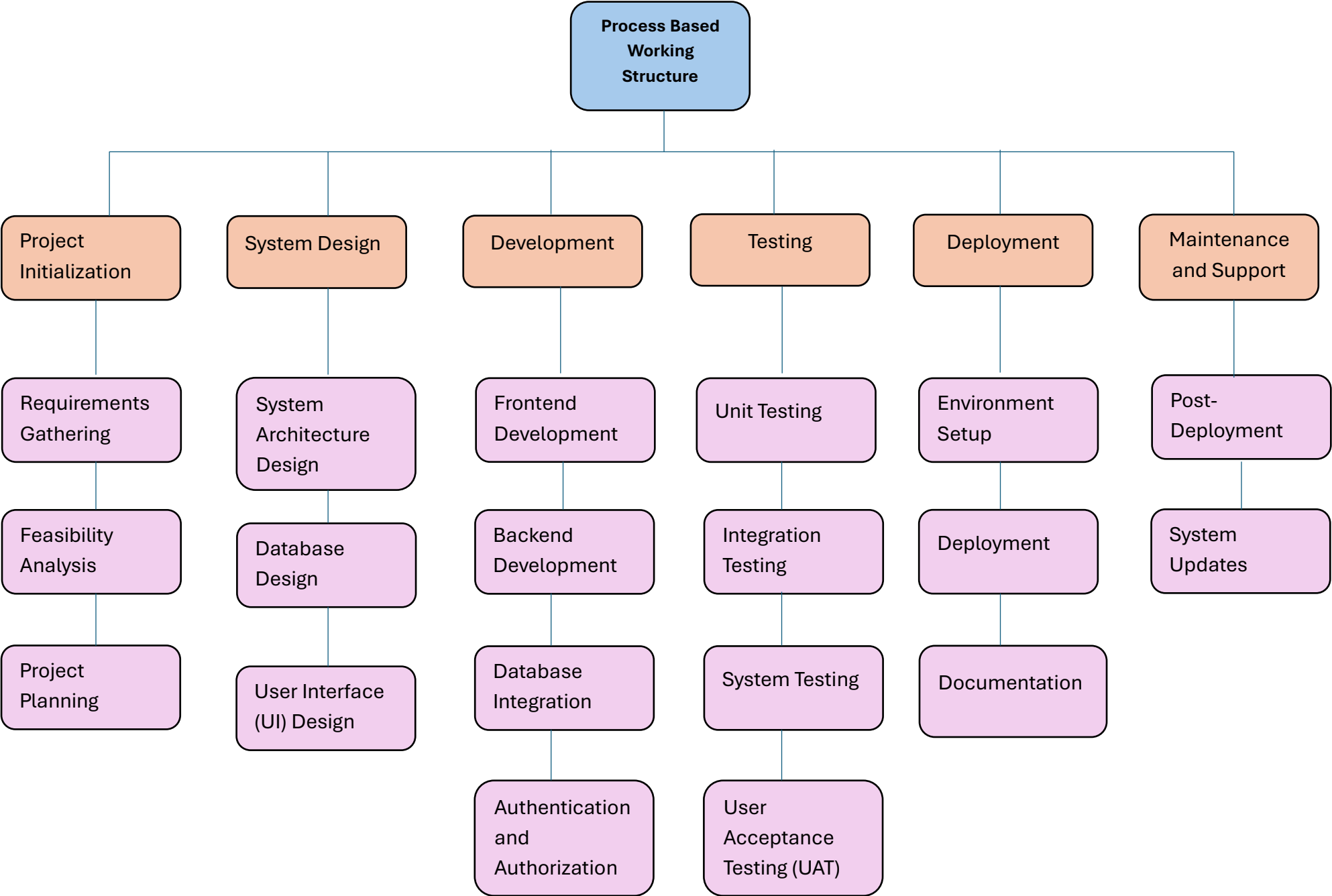
Environmental Impact:

- Reduces waste and resource consumption by optimizing stock levels.
- Contributes to sustainability by minimizing overproduction and storage.
- Promotes paperless operations, reducing waste and environmental footprint.

### **Stakeholders of the Project:**

1. Project Sponsor: Provides financial resources and support for the project.
2. Project Manager: Oversees the planning, execution, and delivery of the project.
3. Software Developers: Responsible for coding and developing the IMS system.
4. UI/UX Designers: Design the user interface and ensure a user-friendly experience.
5. Business Analysts: Analyze business requirements and translate them into technical specifications.
6. End Users: Employees or business stakeholders who will use the IMS daily (e.g., inventory managers, sales staff).
7. IT Support Team: Provides ongoing technical support and maintenance after deployment.
8. External System Integrators: Integrate the IMS with ERP, POS, and supplier systems.
9. Suppliers: External parties who will interact with the system for stock replenishment and order processing.
10. Customers: Beneficiaries of improved inventory management, leading to better products and services.

# Process Based Working Structure:-



## 1. Project Initialization

- 1.1 Requirements Gathering
  - o Identify key stakeholders.
  - o Conduct interviews or surveys to understand needs.
  - o Document functional and non-functional requirements.
- 1.2 Feasibility Analysis
  - o Evaluate technical feasibility.
  - o Assess cost and time constraints.
- 1.3 Project Planning
  - o Develop project timeline.
  - o Allocate team roles and responsibilities.
  - o Prepare project budget.

## 2. System Design

- 2.1 System Architecture Design
  - o Define system architecture (e.g., client-server, web-based, standalone).
  - o Choose technology stack (e.g., programming languages, database, frameworks).
- 2.2 Database Design
  - o Identify entities and relationships.
  - o Create an Entity-Relationship Diagram (ERD).
  - o Design database schema.
- 2.3 User Interface (UI) Design
  - o Create wireframes or mockups.
  - o Design user-friendly interfaces for different roles (e.g., admin, staff).

## 3. Development

- 3.1 Frontend Development
  - o Implement UI for product addition, update, and removal.
  - o Create dashboards and reports.
- 3.2 Backend Development
  - o Develop API endpoints for inventory operations.
  - o Implement business logic (e.g., stock updates, reorder levels).
- 3.3 Database Integration
  - o Connect backend to the database.
  - o Implement CRUD operations for inventory data.

- 3.4 Authentication and Authorization
  - o Implement user login and role-based access control.

## 4. Testing

- 4.1 Unit Testing
  - o Test individual components (e.g., database queries, APIs).
- 4.2 Integration Testing
  - o Verify interactions between frontend, backend, and database.
- 4.3 System Testing
  - o Test complete system for functionality, performance, and security.
- 4.4 User Acceptance Testing (UAT)
  - o Gather feedback from stakeholders.
  - o Make necessary adjustments based on user feedback.

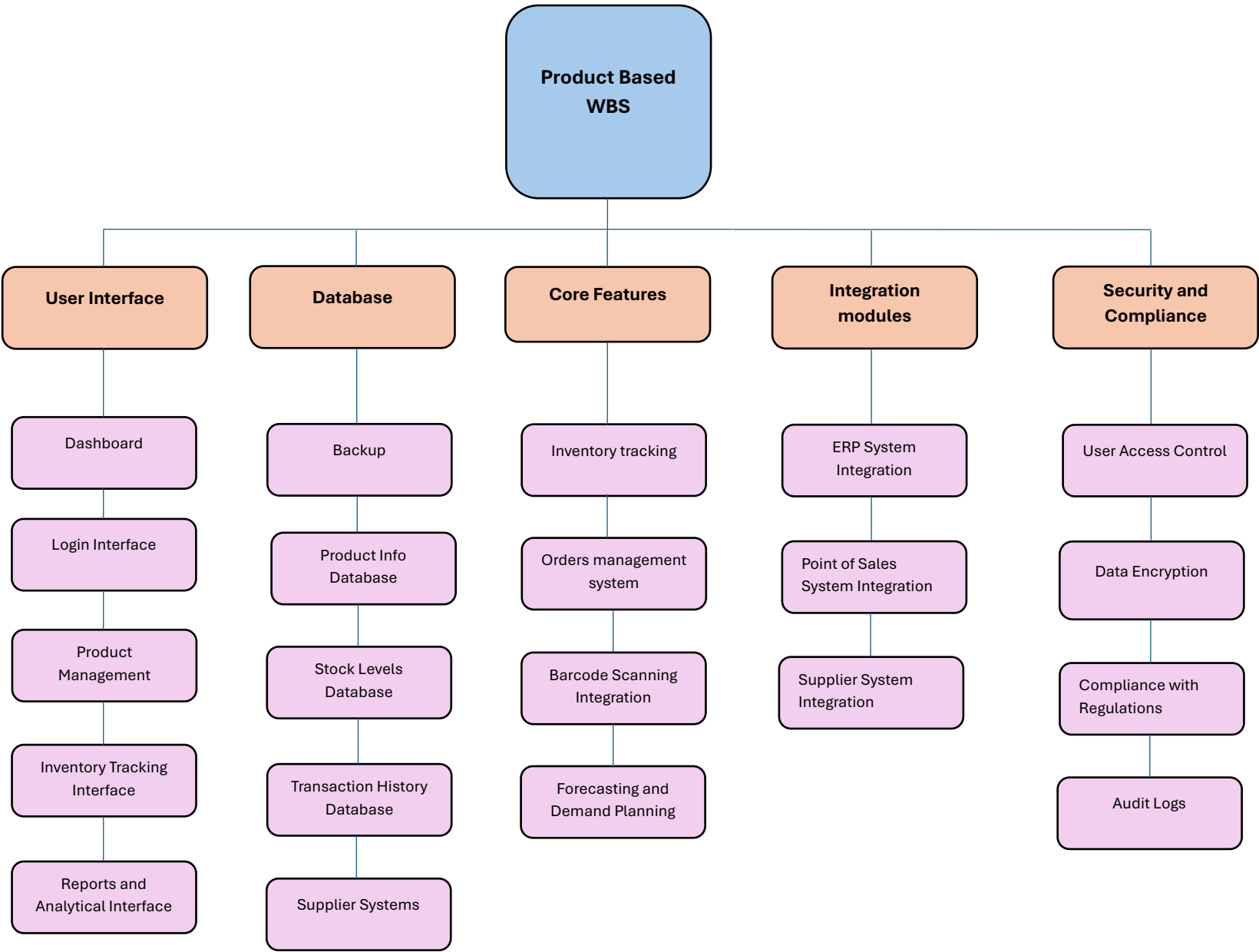
## 5. Deployment

- 5.1 Environment Setup
  - o Set up production server or cloud environment.
  - o Configure database and application.
- 5.2 Deployment
  - o Deploy the system to the production environment.
  - o Perform initial data migration if necessary.
- 5.3 Documentation
  - o Prepare user manuals.
  - o Document system architecture and codebase.

## 6. Maintenance and Support

- 6.1 Post-Deployment Support
  - o Fix bugs reported by users.
  - o Provide technical support.
- 6.2 System Updates
  - o Add new features as required.
  - o Optimize system performance over time.<sup>1</sup>

# Product Base Working Structure:-



## 1. User Interface (UI)

- 1.1. Dashboard
- 1.2. Login and Authentication System
- 1.3. Product Management Interface
- 1.4. Inventory Tracking Interface
- 1.5. Reports and Analytics Interface

## 2. Database

- 2.1. Product Information Database
- 2.2. Stock Levels Database
- 2.3. Transaction History Database
- 2.4. Supplier/Customer Database
- 2.5. Backup and Recovery

## 3. Core Features

- 3.1. Inventory Tracking
  - 3.1.1. Real-time Stock Updates
  - 3.1.2. Alerts for Low Inventory
- 3.2. Order Management
  - 3.2.1. Purchase Order Processing
  - 3.2.2. Sales Order Processing
- 3.3. Barcode Scanning Integration
- 3.4. Forecasting and Demand Planning

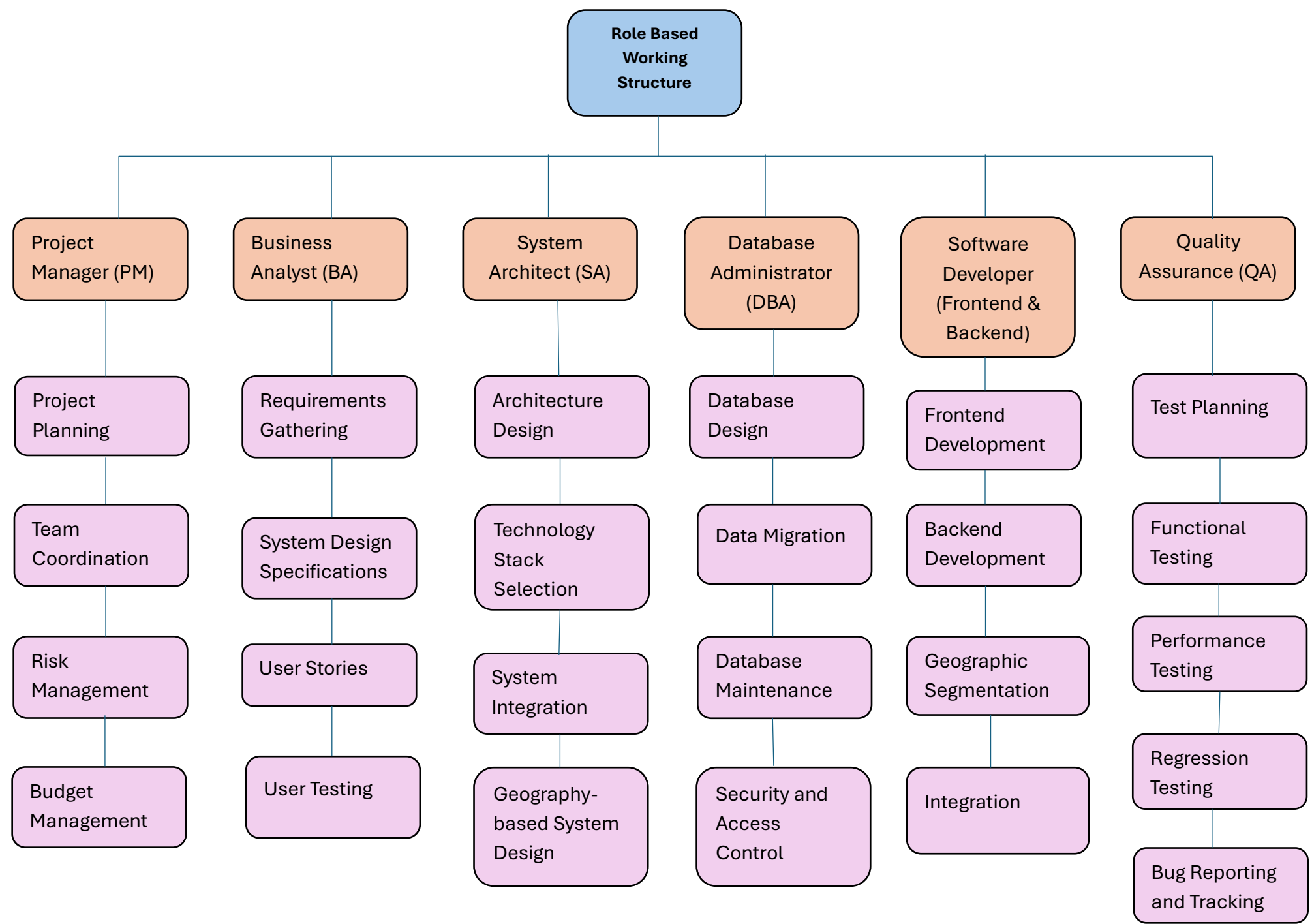
## 4. Integration Modules

- 4.1. ERP System Integration
- 4.2. Point of Sale (POS) System Integration
- 4.3. Supplier System Integration

## 5. Security and Compliance

- 5.1. User Access Control
- 5.2. Audit Logs
- 5.3. Data Encryption
- 5.4. Compliance with Regulations (e.g., GDPR, SOX)

# Role Based Working Structure:-





## 1. Project Manager (PM)

- Project Planning:
  - o Define project scope and objectives
  - o Create project schedule with milestones and deadlines
- Team Coordination:
  - o Assign tasks to team members
  - o Monitor progress and report on project status
- Risk Management:
  - o Identify and mitigate potential project risks
- Budget Management:
  - o Ensure the project stays within budget

## 2. Business Analyst (BA)

- Requirements Gathering:
  - o Collect detailed business requirements from stakeholders (e.g., inventory needs, reporting features)
  - o Analyze current inventory management processes
- System Design Specifications:
  - o Translate business requirements into technical specifications
- User Stories:
  - o Create detailed user stories and use cases
- User Testing:
  - o Assist in testing by validating the system against user needs

## 3. System Architect (SA)

- Architecture Design:
  - o Design the system architecture to support scalability and robustness
- Technology Stack Selection:
  - o Choose appropriate technologies (e.g., database, front-end, back-end frameworks)
- System Integration:
  - o Design how various modules (inventory, order management, etc.) will interact
- Geography-based System Design (if applicable):
  - o Create solutions for managing different locations and regions

## 4. Database Administrator (DBA)

- Database Design:
  - o Create database schema based on inventory and location-based data

- o Implement database normalization and indexing
- Data Migration:
  - o Plan and execute migration from legacy systems (if applicable)
- Database Maintenance:
  - o Set up and monitor database backups, performance, and integrity
- Security and Access Control:
  - o Manage user roles and permissions for secure data access

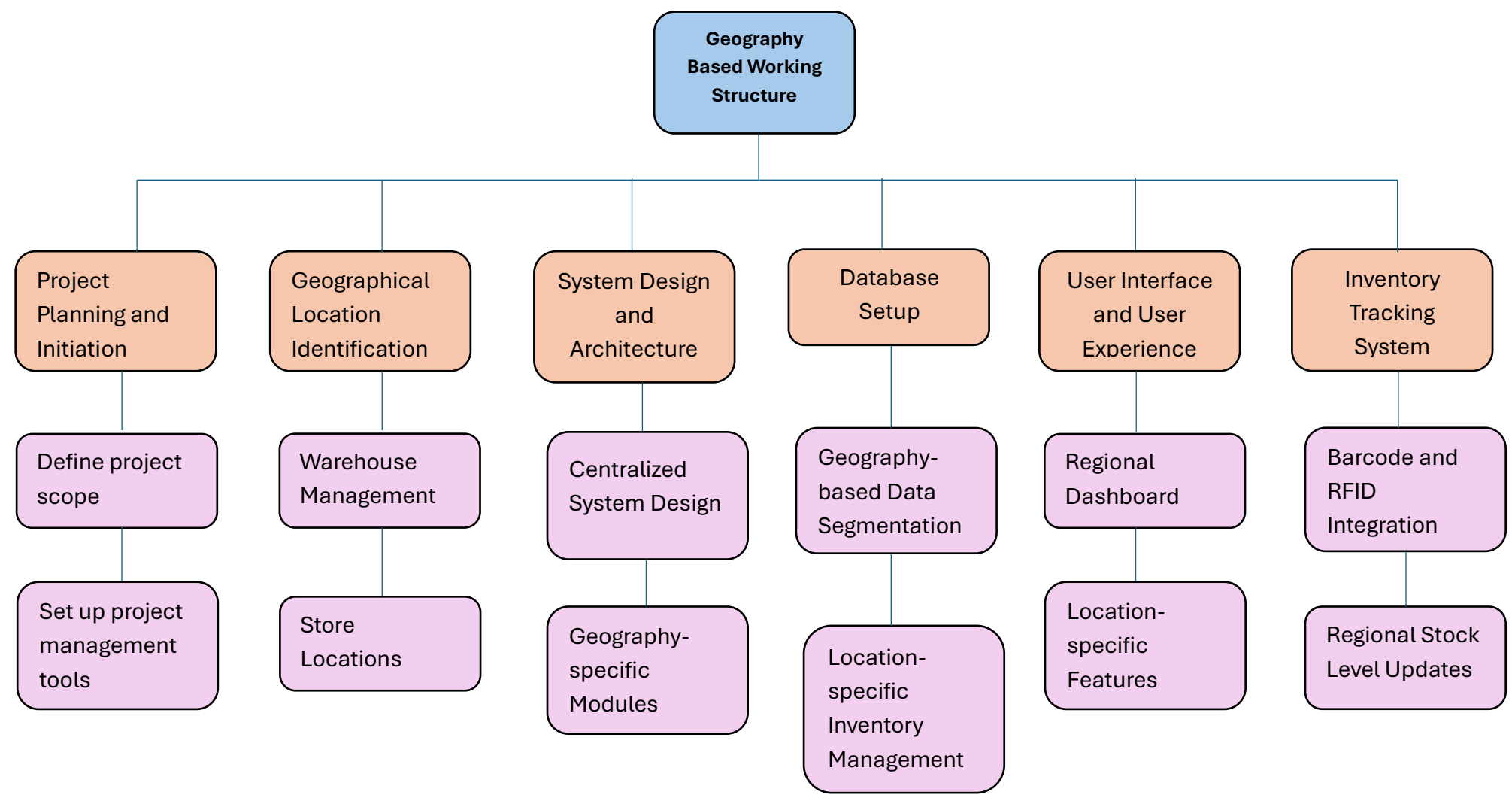
## 5. Software Developer (Frontend & Backend)

- Frontend Development:
  - o Develop user interfaces for inventory tracking, order management, and reports
  - o Implement user authentication and authorization
- Backend Development:
  - o Develop API and business logic for inventory management (stock level updates, order processing)
  - o Implement database interaction layer
- Geographic Segmentation (if applicable):
  - o Implement location-based features, such as region-specific inventory management
- Integration:
  - o Integrate with third-party services (e.g., shipping, ERP systems)

## 6. Quality Assurance (QA)

- Test Planning:
  - o Develop test plans and test cases based on requirements
- Functional Testing:
  - o Test inventory management features like stock tracking, orders, and transfers
- Performance Testing:
  - o Test system performance under heavy loads (e.g., large data volume, multiple users)
- Regression Testing:
  - o Ensure existing functionality is not broken after updates
- Bug Reporting and Tracking:
  - o Document bugs and track their resolution

# Geography Based Working Structure:-



## 1. Project Planning and Initiation

- Define project scope based on geographic locations (warehouses, stores, etc.)
- Set up project management tools and frameworks for remote collaboration if needed.

## 2. Geographical Location Identification

- Warehouse Management:
  - o Identify primary warehouse locations
  - o Identify secondary/backup warehouse locations
- Store Locations:
  - o Identify store fronts, regional offices, or distribution centers

## 3. System Design and Architecture

- Centralized System Design:
  - o Design a centralized system to manage multiple locations
- Geography-specific Modules:
  - o Create modules to handle location-specific inventory (e.g., regional stock, delivery management)

## 4. Database Setup

- Geography-based Data Segmentation:
  - o Design database schema with location-based segmentation (e.g., tables for each region)
- Location-specific Inventory Management:
  - o Implement geographic categorization for inventory (e.g., tracking inventory levels, shipping, and stock counts by region)

## 5. User Interface and User Experience

- Regional Dashboard:
  - o Create regional dashboards to show location-based inventory details
- Location-specific Features:
  - o Customize the user interface for specific warehouses and store locations

## 6. Inventory Tracking System

- Barcode and RFID Integration:
  - o Set up region-specific barcode scanning or RFID for stock management
- Regional Stock Level Updates:
  - o Enable system to update inventory per location