**Requirement Gathering and Analysis Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 04-07-2024 |
| Team ID | SWTID1719929609 |
| Project Name | Project - BOOKNEST |
| Maximum Marks |  |

**Solution Architecture for BOOKNEST:**

* **Find the best tech solution to solve existing business problems.**

**Business Problems:** Identify challenges such as efficient book catalog management, secure transactions, and personalized recommendations.

**Technology Evaluation:** Assess options (e.g., React.js, Node.js, PostgreSQL) based on scalability, security, and cost-effectiveness.

* **Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders**.

**Component Diagram:** Visualize the frontend (web app), backend (microservices), and database interactions.

**Quality Attributes:** Set expectations for performance, security, and reliability.

**Behavioural Models:** Used UML diagrams to depict user interactions

* **Define features, development phases, and solution requirements.**

**Key Features:**

* + Book Purchase: Users can browse and buy books.
  + Cart and Wishlist: Users can add books to their cart or wish list.
  + Order Tracking: Users can check the status of their orders.
  + Seller Portal: Sellers manage inventory and full- fill orders.
  + Admin Portal: Admins oversee the entire system.

**Solution requirements:**

**User Requirements:**

* + **Book Purchase:** Users should be able to browse books, view details, and make purchases.
  + **Cart and Wishlist:** Users can add books to their cart or wishlist.
  + **Order Tracking:** Users should track the status of their orders.

**Seller Portal Requirements:**

* + **Inventory Management:** Sellers can add, update, or remove books from the catalog.
  + **Order Fulfillment:** Sellers process orders placed by users.

**Admin Portal Requirements:**

* + **Full Control:** Admins have access to all features (user management, order status changes).
  + **Monitoring:** Admins can monitor system health and performance.

**Security Requirements:**

* + **Authentication:** Secure login for users, sellers, and admins.
  + **Authorization:** Role-based access control (user, seller, admin).
  + **Data Encryption:** Protect sensitive data (e.g., user passwords).

**Performance Requirements:**

* + **Responsiveness:** Fast loading times for book listings and order tracking.
  + **Scalability:** Handle increasing user traffic efficiently.
  + **Caching:** Cache frequently accessed data (e.g., book details).

**Development phases**

**Web Application (MERN Stack):**

* + **Frontend (React.js):** User interface for browsing, purchasing, and managing orders.
  + **Backend (Node.js, Express):** Handles user requests, authentication, and business logic.
  + **Database (MongoDB):** Stores book data, user profiles, and order details.
* **Provide specifications according to which the solution is defined, managed, and delivered.**

**User Portal:**

* + Allows user to add/remove books in cart and wish list, manages orders, allows users to buy and view books .

**Seller Portal:**

* + Allows sellers to add/remove books, manage inventory, and process orders.

**Admin Portal:**

* + Provides full system control, including order status management.

**Architecture Patterns:**

* + **Microservices (Backend):** Divide functionality (e.g., user management, order processing) into separate services.
  + **RESTful APIs:** Define clear API contracts for communication between frontend and backend.
  + **Authentication :** Secure user logins and sessions.
  + **Authorization (Role-Based Access Control):** Control access based on user roles (user, seller, admin).