**LOW LEVEL DOCUMENT**

**Amazon App**

Low-Level Design is a phase in the software development process where detailed system components and their interactions are specified. It involves converting the high-level design into a more detailed blueprint, addressing specific algorithms, data structures, and interfaces.

**Amazon App Components:**

1. User Authentication and Profile Management:

* Authentication Service: Handles user authentication (login, logout, registration) and generates authentication tokens.
* Profile Service: Manages user profiles, including personal information, addresses, payment methods, etc.
* Database: Stores user credentials, profile information, and session data.

2.Product Search and Browsing:

* Product Service: Stores and retrieves product information (name, description, price, reviews, etc.).
* Search Service: Provides search functionality based on product attributes and user queries.
* Category Service: Manages product categories and subcategories.
* Database: Stores product details, category information, and search indices.

3.Online Shopping Cart and Checkout:

* Cart Service: Manages shopping carts for users, including adding/removing items, updating quantities, etc.
* Checkout Service: Handles the checkout process, including order summary, shipping options, and payment processing.
* Inventory Service: Tracks available inventory and updates quantities when orders are placed.
* Database: Stores cart contents, order details, and inventory status.

4.Order Tracking and Management:

* Order Service: Manages orders, including order placement, status tracking, and history.
* Notification Service: Sends notifications to users regarding order status updates.
* Shipping Service: Coordinates with shipping providers to manage delivery logistics.
* Database: Stores order details, shipping information, and notification preferences.

5.Customer Support and Feedback:

* Support Service: Handles customer inquiries, complaints, and support tickets.
* Feedback Service: Manages customer feedback, including ratings, reviews, and suggestions.
* Analytics Service: Analyzes feedback data to improve services and products.
* Database: Stores support tickets, feedback submissions, and analytics data.

**Technology Stack:**

**Backend**: Microservices architecture using technologies like Java/Spring Boot, Node.js, or Python/Flask.

**Database**: Relational databases (e.g., MySQL, PostgreSQL) for structured data and NoSQL databases (e.g., MongoDB, Redis) for high-volume, low-latency data.

**Authentication**: JSON Web Tokens (JWT) for authentication and authorization.

Search: Elasticsearch or Apache Solr for search functionality.

**Messaging**: Apache Kafka or RabbitMQ for inter-service communication.

Infrastructure: Containerization with Docker and orchestration with Kubernetes for scalability and deployment.

**Monitoring**: Prometheus and Grafana for monitoring service health and performance.