Low Level Design-LLD for Amazon

LLD, or 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. LLD serves as a guide for developers during coding, ensuring the accurate and efficient implementation of the system’s functionality.

Low-Level Design (LLD) is the process of converting a High-Level Design (HLD) or system architecture into a detailed design that can be implemented by software developers. In the software development life cycle, LLD comes after the completion of the High-Level Design phase and before the actual coding or implementation phase.

**Introduction**

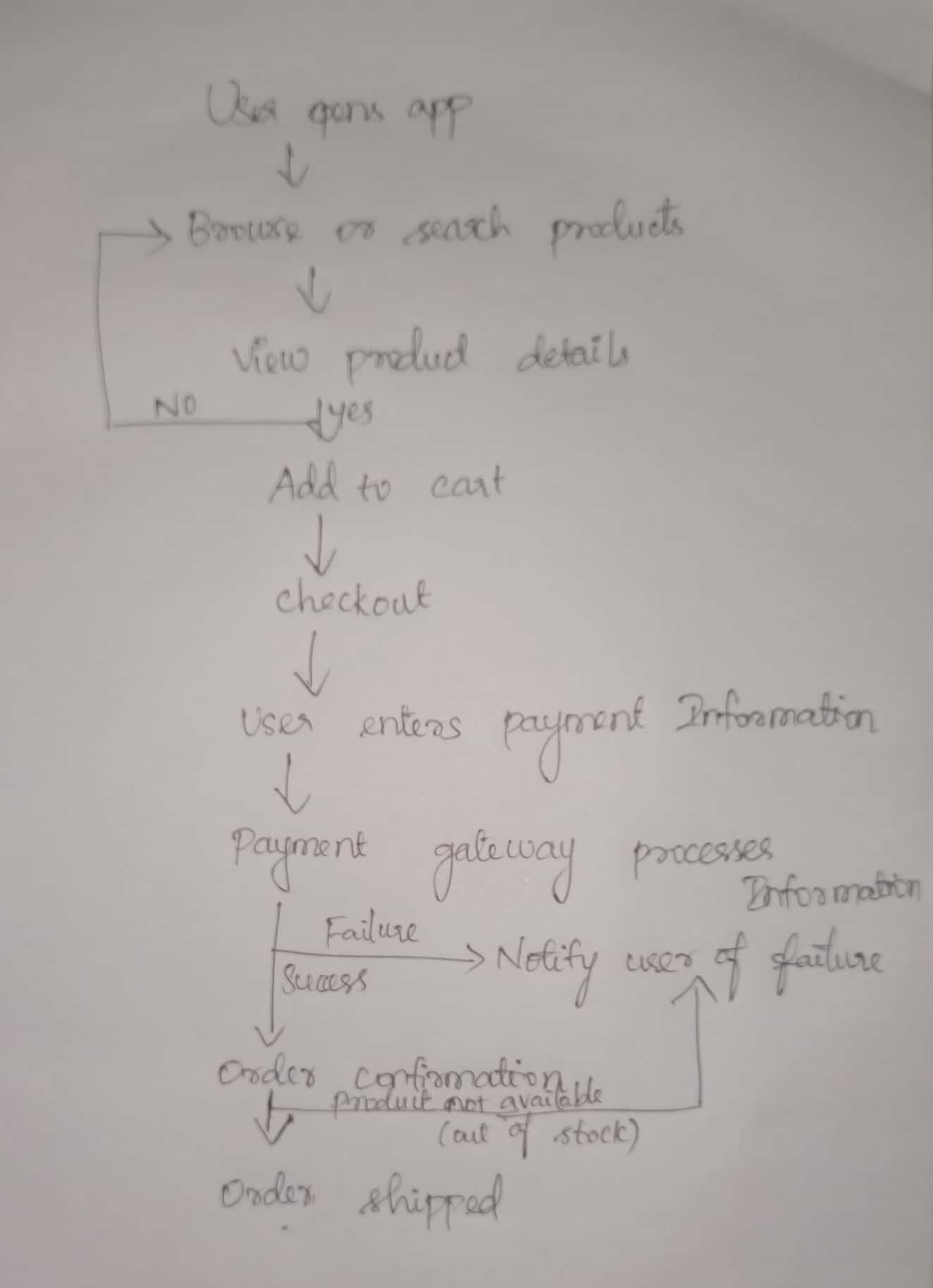
This document outlines the low-level design of a shopping application, detailing the functionalities and technical aspects of its core components.

**2. System Architecture**

The application will employ a client-server architecture with the following components:

* **Mobile App:** The user interface for browsing, searching, and purchasing products.
* **API Server:** Handles communication between the app and backend services.
* **Product Database:** Stores product information, including descriptions, prices, and images.
* **User Database:** Stores user information, including profiles, addresses, and payment methods.
* **Shopping Cart:** Manages the selection and temporary storage of products chosen for purchase.
* **Payment Gateway:** Processes secure online payments.
* **Order Management System:** Tracks and manages order fulfillment.

**3. Flowchart**



**4. Detailed Component Design**

**4.1 Mobile App**

* **User Interface:**
  + Product browsing and search functionalities.
  + Product details screen with information, images, and reviews.
  + Shopping cart screen to view and manage selected items.
  + Secure user checkout with payment processing.
  + User profile management section.
* **Technical Specifications:**
  + Developed using native frameworks (Swift/Kotlin) or cross-platform frameworks (React Native, Flutter).
  + Adheres to platform-specific design guidelines for a consistent user experience.

**4.2 API Server**

* **Functionality:**
  + Receives and processes user requests from the mobile app.
  + Interacts with the product and user databases.
  + Manages shopping cart operations.
  + Handles secure communication with the payment gateway.
  + Provides order tracking information.
* **Technical Specifications:**
  + RESTful API design for efficient data exchange.
  + Secure authentication and authorization mechanisms.

**4.3 Product Database**

* **Data Model:**
  + Stores product information (ID, name, description, price, image, etc.).
  + Maintains inventory levels for each product.
* **Database Management System:** (The choice of database management system will depend on the project's scale and complexity).

**4.4 User Database**

* **Data Model:**
  + Stores user information (name, email, address, payment methods, etc.).
  + Implements secure password hashing for user authentication.
* **Database Management System:**
  + As with the product database, the choice depends on project requirements.

**4.5 Shopping Cart**

* **Functionality:**
  + Allows users to add, remove, and update product quantities.
  + Calculates total cart value based on selected products and quantities.
  + Stores cart information persistently on the device or server-side.

**4.6 Payment Gateway**

* **Integration:**
  + Securely connects with the shopping application via the API server.
  + Processes various payment methods (credit cards, debit cards, etc.).
* **Security:**
  + Complies with industry standards for secure online transactions .

**4.7 Order Management System**

* **Functionality:**
  + Receives confirmed orders from the payment gateway.
  + Manages order fulfillment, including inventory updates and shipping processes.
  + Provides order tracking information to users.

**5. Conclusion**

This low-level design document provides a comprehensive overview of the shopping application's architecture, components, and functionalities.