**High-Level Design (HLD):**

1. **Architecture Overview**:
   * Flipkart operates on a distributed system architecture.
   * Three-tier architecture: Presentation layer (Frontend), Application layer (Backend), and Data layer (Database).
2. **Presentation Layer**:
   * Web-based interface for desktop and mobile.
   * Responsive design for various screen sizes.
   * Use of modern frontend technologies like React.js or Angular for dynamic user experience.
3. **Application Layer**:
   * Microservices Architecture: Different services for catalog management, user management, order processing, payment, recommendation, etc.
   * RESTful APIs for communication between services.
   * Load balancing and service discovery for scalability and fault tolerance.
   * Caching mechanisms (like Redis) for frequently accessed data.
4. **Data Layer**:
   * Relational Database Management System (RDBMS) like MySQL or PostgreSQL for transactional data.
   * NoSQL databases like MongoDB or Cassandra for handling unstructured or semi-structured data like user activity logs, product metadata, etc.
   * Data Warehousing for analytics and business intelligence purposes.
5. **Security**:
   * Secure communication using HTTPS.
   * Authentication and authorization mechanisms (JWT tokens, OAuth) for user identity management.
   * Encryption of sensitive data.
   * Regular security audits and penetration testing.
6. **Scalability and Performance**:
   * Horizontal scaling using containerization (Docker) and orchestration (Kubernetes).
   * Auto-scaling based on traffic patterns.
   * Content Delivery Networks (CDNs) for efficient content delivery.

**Low-Level Design (LLD):**

1. **User Management Service**:
   * Detailed design of user registration, login, and profile management functionalities.
   * User authentication workflows.
2. **Catalog Management Service**:
   * Design of product catalog schema.
   * CRUD operations for products.
   * Search and filtering functionalities.
3. **Order Processing Service**:
   * Order placement, modification, and cancellation workflows.
   * Integration with payment gateway services.
4. **Recommendation Service**:
   * Algorithms for personalized recommendations.
   * Collaborative filtering, content-based filtering, or hybrid approaches.
5. **Payment Service**:
   * Integration with various payment gateways (like Razorpay, Paytm, etc.).
   * Handling payment transactions securely.
6. **Notification Service**:
   * Design of notification delivery mechanisms (email, SMS, push notifications).
   * Integration with messaging services like Twilio or SendGrid.
7. **Logging and Monitoring**:
   * Design of logging infrastructure for tracking system behavior and debugging.
   * Implementation of monitoring tools (like Prometheus, Grafana) for real-time system health monitoring.
8. **Error Handling and Retry Mechanisms**:
   * Design of fault-tolerant mechanisms like retries, circuit breakers, and fallbacks.
   * Graceful degradation under high load or service failures.

This HLD and LLD provide a high-level overview and detailed design considerations for a system like Flipkart. Actual implementations would involve further refinement and optimization based on specific requirements, technologies, and constraints.