SE Lab -Component Diagram Self-Service Coffee Kiosk System – Architecture Justification

---

Name: Keerthan pv

SRN: PES2UG23CS272

Class: CSE - E

University: PES UNIVERSITY

Question 1: Architectural Choice – State which architectural style you selected

Architecture Selection

We chose the **Layered Architecture** for the Self-Service Coffee Kiosk System depicted in the diagram.

The architecture organizes the system into multiple interacting layers:

- **UI Layer (Kiosk Page)** handles user interactions like finding orders, managing the basket, and validating users.
- **Business Logic Layer (Order Manager)** manages orders, extras, preferences, and invoice generation.
- **Data Layer (Menu Repository)** stores menu information, while
- **Accounts and Payment layers** handle user accounts, payment verification, and transaction confirmations.
  This structure ensures separation of concerns, simplifies maintenance, and aligns with the kiosk's functional and technical requirements.

Question 2: Provide 2 reasons for your choice and justify

Reason 1 – Clear Separation of User Interaction and Business Logic

In the diagram, the **Kiosk Page** focuses only on user-facing tasks like order selection and basket management. Meanwhile, the **Order Manager** handles complex processes like verifying orders, generating invoices, and managing preferences. By isolating the UI from business logic, the system ensures that the touchscreen interface remains fast and

responsive, even when backend operations (like session validation or order checking) take time.

Reason 2 – Hardware and External Service Abstraction for Easier Maintenance

The **Payment** and **Accounts** layers are separated from other components and manage sensitive tasks like transaction verification and user authentication. This isolation means that changes to payment processing (for example, upgrading to a new payment provider) do not affect the UI or order handling logic. The architecture allows each layer to evolve independently, improving maintainability and adaptability.

Question 3: Security Advantage – Explain one way your architecture addresses security concerns

The architecture isolates sensitive operations within the **Payment** and **Accounts** layers. For example, **Transaction Confirmation Service** and **Payment Transaction Verifier** only handle secure payment processes. The **Kiosk Page** and **Order Manager** interact with these layers through controlled interfaces, ensuring that user credentials or payment data are never exposed to UI components or other services. This protects customer information while maintaining functional integrity.

Question 4: Performance Benefit – Describe one performance advantage of your chosen architecture

The layered approach ensures that resource-intensive tasks, such as **Order Payment Verification** and **Transaction Confirmation**, are handled in separate layers from the **Kiosk Page**. This prevents the user interface from being blocked by backend processes. Even if the payment request takes time to process, users experience immediate feedback through UI actions like basket updates or order tracking, enhancing the responsiveness and smooth functioning of the kiosk during peak hours.

Component diagram: