Final Project: Basic System Design for Crusty Muse Bakery

Business Scenario

Crusty Muse Bakery wants a simple online ordering system that handles 20-30 orders per day on a small budget.

Task 1: Basic System Analysis

A: Identify Key Stakeholders

Stakeholder	Role
Customers	Browse products, add items to cart, place orders, and receive confirmation emails
Bakery Staff	View incoming orders, update order status, and prepare orders for pickup/delivery
Bakery Manager/Owner	Manage product listings, monitor order volumes, and oversee system operations
Payment Processor	Process customer payments securely and handle transaction verification

B. Identify the System Components

Use Case	Input	Process	Output
Status confirmation email to the customer	Order ID, customer email, order status update	System retrieves order details, formats email message with status information, and sends via email service	Confirmation email delivered to customer's inbox with order status

---

Task 2: Select an SDLC Model

Choice of SDLC Model	**Agile (Iterative)**
Reason	Agile is ideal for Crusty Muse Bakery because it allows for incremental development with frequent releases. Given the small budget and need for simplicity, we can start with core features (product display and ordering) and add features like staff dashboard and email confirmations in later iterations. This approach allows the bakery to start using the system quickly, provide feedback, and make adjustments based on real-world usage. The iterative nature also reduces risk and allows for changes as business needs evolve.

Task 3: Select the Architecture Pattern

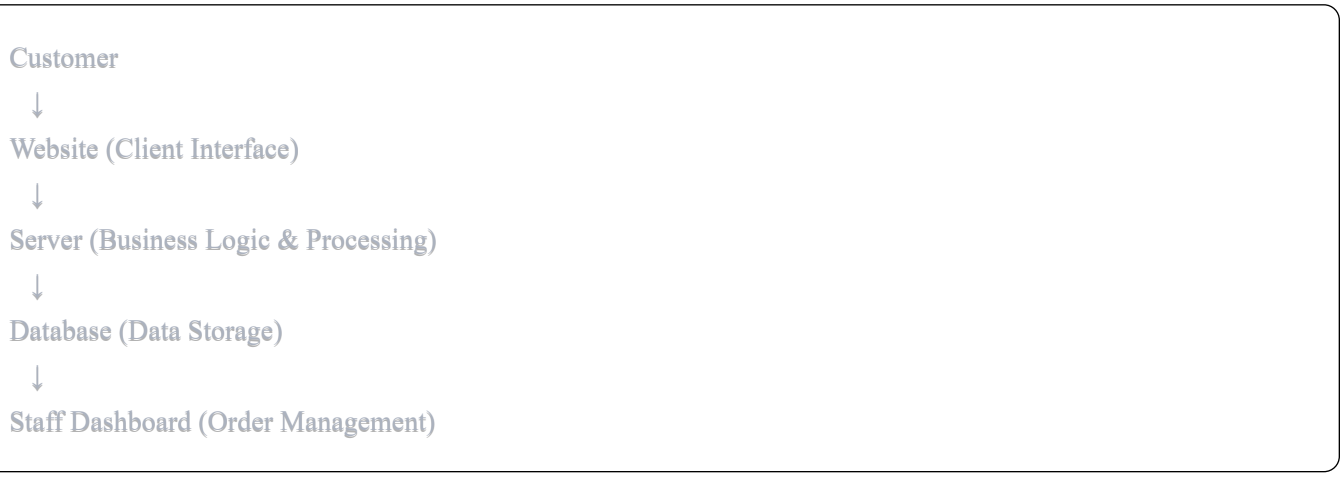
Choice of Architecture Pattern	**2-Tier (Client-Server)**
Reason	The 2-tier architecture is the best fit for Crusty Muse Bakery's needs. With only 20-30 orders per day and a small budget, the simpler client-server model is more cost-effective to build and maintain. The server can handle all business logic, data storage, and processing without the added complexity of separating layers. This architecture is easier for non-technical staff to troubleshoot and requires fewer resources to host. It also meets all requirements (mobile compatibility, order management, email notifications) while keeping development time and costs minimal.

Task 4: Complete a Basic System Diagram

System Flow Components

Number	Component
1	**Customer** (initiates the order by accessing the system)
2	**Website** (displays products and shopping cart interface to customer)
3	**Server** (processes orders, handles business logic, sends confirmation emails)
4	**Database** (stores product information, customer orders, and order status)
5	**Staff Dashboard** (allows bakery staff to view and update order status)

System Flow Diagram



Task 5: Document Your Progress in the Project

Project Summary

Category	Description
Business Problem	Crusty Muse Bakery needs a simple, affordable online ordering system to allow customers to browse and order baked goods while enabling staff to manage orders efficiently. The current manual process is inefficient for handling 20-30 daily orders.
System Goal	Create a mobile-friendly online ordering system that displays products with prices, processes customer orders with cart functionality, allows staff to view and update order status, and automatically sends confirmation emails to customers.
Technical Choices	2-tier client-server architecture with responsive web design for mobile compatibility, integrated email service for notifications, and simple database for storing products and orders.
SDLC Model	Agile (Iterative)
Reason for the Choice	Agile allows incremental development starting with core features, enabling quick deployment and feedback collection. This reduces risk, accommodates budget constraints, and allows for adjustments based on actual bakery operations and user feedback.
Architecture Pattern	2-Tier (Client-Server)
Reason for the Choice	The 2-tier architecture is simpler and more cost-effective for a small business with 20-30 orders per day. It's easier to build, maintain, and troubleshoot while meeting all functional requirements. The reduced complexity aligns with the small budget constraint and limited technical resources.

Summary of Key Design Decisions

This system design prioritizes:

- **Simplicity**: Easy to understand and use for non-technical staff
- **Affordability**: Minimal development and hosting costs
- **Maintainability**: Simple architecture that's easy to troubleshoot and update
- **Scalability**: Can handle current needs with room to grow
- **Mobile compatibility**: Responsive design for customer convenience

All design choices align with the constraints of a small bakery with limited budget and technical resources while meeting all functional requirements for online ordering, order management, and customer communication.