INCREMENTAL SDLC MODEL FOR A KIOSK-BASED PRE-ORDER, DIGITAL PAYMENT, AND TABLE RESERVATION MANAGEMENT SYSTEM FOR A FAST-FOOD CHAIN

Planning

- Define the main goal: To develop an integrated self-service kiosk system that allows customers to pre-order meals, make digital payments, and reserve tables, streamlining the operations of a fast-food chain and enhancing the customer experience.

- Identify stakeholders: Customers, Cashiers/Restaurant Staff, Store Managers, Franchise Owners, IT Developers, and Payment Gateway Providers.

- Assess feasibility: Ensure the project is cost-effective against the expected increase in order efficiency, technically possible with available kiosk hardware, and beneficial by reducing wait times and labor costs.

- Deliverable: Project plan and scope.

Requirement Analysis

Functional Requirements:

- Pre-Order Module: Display menu items and real-time availability, allow customers to place an order and view order count, generate a unique order ID.

- Table Reservation Module: Allow customers to select and reserve an available table, display table availability, and trace their selected table.

- Digital Payment Module: Integrate multiple payment methods (e.g., credit card, e-wallet) for seamless checkout.

- Feedback Module: Provide an interface for customers to submit feedback.

- Admin Monitoring: Monitor and verify payments, monitor order status (finished, cancelled) and make emergency alterations, and manage table statuses (mark as occupied/available).

-**Non-Functional Requirements:**

- Compatibility: Must run on designated kiosk hardware (touchscreen).

- Performance: Fast response time for order processing and payment verification.

- Usability: Simple, intuitive, and visually appealing touchscreen UI for customers and a clear dashboard for admins.

- Security: PCI-DSS compliance for secure payment processing and data encryption.

- Integration: Must integrate with existing POS/Kitchen Display Systems (KDS) and payment gateway APIs.

- Deliverable: Software Requirement Specification (SRS).

-System Design

- Architecture Design: Client-Server architecture. Kiosk (Client) application communicates with a central backend server which integrates with the database, payment gateways, and KDS.

- UI/UX Design:
- Customer Kiosk: Wireframes for a step-by-step flow: Welcome -> Pre-order/Reservation -> Menu -> Cart -> Payment -> Confirmation (with Order ID).
- Admin Panel: Wireframes for a dashboard showing order queue, payment status, and a visual table management layout.

- Database Design: Tables for Users, Menu Items, Orders, Order Items, Transactions, Tables, and Feedback.

- Technology Stack:
- Kiosk Frontend: Plotter

- Backend: JavaScript

- Database: MySQL

- Deliverable: Design Document, Wireframes, ERD.

-Development

- Build the application in functional increments using Agile sprints:

- Increment 1: Core Pre-Order Module (Menu, Cart, Order ID Generation).

- Increment 2: Digital Payment Module integration with payment gateways.
- Increment 3: Table Reservation & Management Module.
- Increment 4: Admin Monitoring Dashboard and Feedback Module.
- Deliverable: Initial working prototype (e.g., a functional pre-order and payment loop).

-Testing

- Unit Testing: Test individual modules (e.g., payment processing, order calculation, table status update).
- Integration Testing: Test the interaction between modules (e.g., placing an order updates the kitchen display, a successful payment creates a transaction record).
- System Testing: Test the complete end-to-end system functionality under load.
- User Acceptance Testing (UAT): Have actual store staff and managers test the Admin Panel and a select group of customers test the kiosk in a pilot store.
- Deliverable: Test Report, Bug Fixes.

-Deployment

- Deploy the kiosk software to hardware units in pilot store locations.
- Provide comprehensive installation guides and training sessions for restaurant staff on using the Admin Panel for monitoring and management.
- Conduct a pilot run to ensure all integrated systems (KDS, payment) work smoothly in a live environment.
- Deliverable: Deployed system in a real environment.

-Maintenance

- Provide regular updates for bug fixes, security patches, and performance improvements.

- Monitor system performance and gather user feedback for future feature enhancements (e.g., loyalty program integration, advanced analytics).

- Add new enhancements based on business needs.

- Deliverable: Updated, reliable, and evolving system.