



Project: Software Engineering
(DLMCSPSE01)

In M.Sc. in Computer
Science By
Dharak Sanjaybhai Pandadiya

Matriculation No.:

4243201

Date of Submission:

27/08/2025

Development / reflection phase 2 – Project : Online Cake Shop

Contents

| | | |
|-------|--|----|
| 2 | Project Profile | 3 |
| 2.1 | Scope | 4 |
| 2.1.1 | Administrator Panel | 4 |
| 2.1.2 | Customer Portal | 4 |
| 2.1.3 | Guest User Access | 5 |
| 2.2 | Explanation of Design and Implementation Procedure | 5 |
| 2.3 | Risks | 6 |
| 2.4 | Project Plan..... | 7 |
| 2.5 | Design Decisions | 7 |
| 2.6 | Requirements Analysis | 8 |
| 2.7 | Implementation Summary..... | 9 |
| 2.8 | Glossary | 9 |
| 2.9 | System Design..... | 10 |
| 2.10 | Tech Stack | 10 |
| 2.11 | Clarified Machine Learning Function..... | 11 |
| 2.12 | Testing Plan | 11 |
| 2.13 | Success Criteria..... | 13 |
| 2.14 | Hyperlinks | 13 |
| 2.15 | References | 13 |

2 Project Profile

The Online Cake Shop project is a web-based application that improves the process of ordering cakes by overcoming the characteristics of conventional bakeries. The customers have to make cakes selection through the online menu & place orders anytime from anywhere in the world, and purchases can be made online or cash-on-delivery. The system is developed using PHP, MySQL, HTML, CSS, JavaScript, jQuery, and Bootstrap using relationships and constraints through their respective database engines that encapsulate security services for account management and database handling.

It is divided into three main modules for the admin, registered users and guest users so they can manage orders, profiles, and customer communication. With the Online cake shop, the users can automate the order processing of cakes, keep customer records, and create better communication with users and the bakery. The Online Cake Shop project helps provide better customer satisfaction, reduces manual effort from employees, and improves efficient cake delivery service by being timely and reliable.

Customers enjoy 24/7 access, quicker processing, and multi-order management tools, and guest users can browse services and register seamlessly. By automating manual workload, improving order accuracy, and increasing communication between bakery staff and customers, this project improves service quality and allows the bakery to grow its business digitally.

2.1 Scope

The Online Cake Shop project is an online bakery management and e-commerce system that allows users to search for cakes, order cakes online, and select either pickup or delivery as an option. The Online Cake Shop is developed using PHP, MySQL, HTML5, CSS3, JavaScript, Bootstrap and jQuery. The system features a secure and easy-to-use way for users to order cakes but maintains a robust backend for fulfilling orders and manage users. There are three main users—Administrator, Customer and Guest Visitor—and administers to simplify business practices, satisfy customers and improve overall business efficiency.

There are three function scopes:

2.1.1 Administrator Panel

- Manage customer accounts and profiles
- Add, update, and delete cake categories and menu items
- View, confirm, or cancel orders with status updates
- Generate reports on sales, orders, and users
- Monitor customer inquiries, feedback, and subscriptions
- Maintain security through encrypted passwords and session management

2.1.2 Customer Portal

- Register, log in, and manage personal accounts securely
- Browse, search, and filter cakes by type, flavor, or price
- Add cakes to cart and place orders with invoice generation
- Choose payment options (Cash on Delivery or future integration of online payment)
- Track order status and view order history

- Update personal details, shipping addresses, and reset passwords

2.1.3 Guest User Access

- Browse the online cake menu without registration
- View basic information about cakes and bakery services
- Access “About Us” and “Contact Us” pages
- Register to unlock cart, order, and account features

2.2 Explanation of Design and Implementation Procedure

The Online Cake Shop system was developed according to the Agile Scrum methodology that utilized sprints and associated phases. The system was developed over several sprints, each with a specific set of modules (admin, online order-customer, guest, and databases-client order-feedback) and testing.

The back end was developed using PHP, connected to a MySQL database where the database was structured to identify and manage users, cakes (category), orders, payments and customer feedback. Each database table was normalized to ensure at least one level of consistency, accuracy, and reliability. User authentication and session management was used to ensure a secure login and entry into the system without unauthorized access.

The front-end was developed using HTML5, CSS3, Bootstrap 5, Javascript, jQuery, and AJAX to enable a responsive, interactive online ordering experience. The cake shop website allows customers to browse cakes, filter cakes by category and determine which cakes they would like to purchase, add items to a shopping cart, and finally place an order. Dynamic validation was applied to the client's order with powered screen alerts to reduce errors when completing forms.

The order processing workflow was developed so once a client placed an order, client details, and payment processing were automatically stored in the database and displayed in the Admin Dashboard, where the administrator could update the order status to confirm, being prepared, delivered, or canceled.

Version Control using GitHub was used to keep track of events, e.g., registration module, cart system, feature branch, and admin panel where people reviewed code and completed pull requests.

2.3 Risks

- **Data Security Risk:** Customers may have their credentials and their order information compromised. → July 2: The risk can be reduced by using password encryptions, session management and secure SQL queries.
- **Payment Risk:** Methods of payment currently available to customers are limited to Cash on Delivery which poses a risk for either an order cancellation or potential fraud. → We will be adding secure online payment gateways as a future improvement.
- **System Downtime:** Server disruptions or problems with hosting may create disruptions when placing orders. - Mitigated with the use of safe work practices, back up management, and reputable hosts.
- **User Error Risk:** Customers can provide incorrect order details or incorrect shipping address. - Mitigated by doing client-side validations and order confirmations.
- **Scalability Risk:** As the number of customers grows, database performance may degrade. → Addressed by optimizing queries and planning for future database scaling.
- **Cross-Browser Compatibility:** Some features may not be displayed uniformly across all browsers. → Mitigated through responsive design testing and Bootstrap framework support.

2.4 Project Plan

1. **Requirement Analysis** – Gather functional and non-functional requirements, identify stakeholders (admin, customer, guest), and define system modules such as order management, payment, and user authentication.
2. **System Design** – Develop database schema for users, cakes, orders, and feedback; design client–server architecture; and create UML diagrams (use case, activity, sequence) to represent workflows.
3. **Implementation** – Build the backend using PHP and MySQL, develop the frontend with HTML5, CSS3, Bootstrap, JavaScript, jQuery, and AJAX, and integrate modules for admin, customer, and guest access.
4. **Testing & Validation** – Conduct unit testing, integration testing, and user acceptance testing to validate registration, login, cart, order, and admin functionalities; ensure secure sessions and data handling.
5. **Deployment & Documentation** – Deploy the application on a local or hosted server (XAMPP/WAMP), prepare user and admin manuals, and provide detailed technical documentation for developers.

2.5 Design Decisions

- **MVC Architecture:** Adopted Model–View–Controller structure to separate concerns and improve maintainability.
- **PHP with MySQL:** Chosen as backend stack for secure session handling and efficient relational database management.
- **Bootstrap 5 & jQuery:** Used for responsive, mobile-friendly design and simplified DOM manipulation.
- **Session-Based Authentication:** Implemented for secure user login, role-based access, and order tracking.
- **Modular Database Schema:** Separate tables for users, cakes, orders, and feedback to ensure normalization and data integrity.

- **AJAX Integration:** Enabled dynamic cart updates and smoother user experience without full page reloads.
- **Cash-on-Delivery First Approach:** Selected as the initial payment method due to simplicity, with scope for future online payment gateway integration.

2.6 Requirements Analysis

Functional Requirements:

- Allow customers to register, log in, and manage their accounts.
- Display cake categories, products, and details in the menu.
- Enable customers to add items to cart and place orders.
- Provide administrators with the ability to add, update, and delete products, categories, and user accounts.
- Generate invoices and maintain order history for customers.
- Allow administrators to view reports on sales, orders, and customer data.
- Provide feedback and inquiry submission options for customers.
- Guest users can browse the menu and view information without registration.

Non-Functional Requirements:

- **Performance:** Order placement and page load response time < 2s.
- **Scalability:** Support 500+ concurrent users without service disruption.
- **Availability:** Backend uptime of 99% with secure database connectivity.
- **Security:** Encrypt user passwords and validate input to prevent SQL injection.
- **Usability:** Responsive design to ensure compatibility across mobile, tablet, and desktop devices.
- **Maintainability:** Modular design allowing easy updates and feature additions.

2.7 Implementation Summary

In line with the Scrum methodology, we organized development in sprints to capture feedback iteratively, and where possible, provide weekly updates. We kept our versioning using GitHub, with feature branches for new work and pull requests for merging into the main repository and tracking progress. We built out the backend modules, and once we built the user flows, integrated the front end and testing workflow.

Key Milestones:

- **Sprint 1–2:** Requirement gathering, database schema design, and environment setup (XAMPP/WAMP with MySQL).
- **Sprint 3–4:** Development of backend modules for user registration, authentication, and admin panel.
- **Sprint 5–6:** Implementation of cake catalog, cart system, and order placement workflow.
- **Sprint 7–8:** Integration of invoice generation, order tracking, and admin reporting features.
- **Sprint 9–10:** Frontend enhancements with Bootstrap and jQuery for responsive design, followed by testing and bug fixing.
- **Sprint 11:** Final deployment on localhost server and preparation of user and developer documentation.

2.8 Glossary

| Term | Definition |
|-----------|--|
| PHP | A server-side scripting language used to build the backend of the Online Cake Shop system. |
| MySQL | A relational database management system used to store user, order, and product details. |
| Bootstrap | A front-end framework used for creating responsive and mobile-friendly web pages. |

| | |
|-------------------|---|
| jQuery | A JavaScript library that simplifies event handling, DOM manipulation, and AJAX interactions. |
| AJAX | Asynchronous JavaScript and XML, used for dynamically updating parts of a web page without reloading. |
| Encryption | The process of securing sensitive information, such as passwords, to prevent unauthorized access. |

2.9 System Design

Architecture:

- **Frontend (HTML5, CSS3, Bootstrap, JavaScript, jQuery, AJAX):** Handles user interface, input validation, dynamic cart updates, and responsive display of cake menus and orders.
- **Backend (PHP):** Manages business logic, user authentication, order processing, and communication between frontend and database.
- **Database (MySQL):** Stores customer details, cake categories, product information, order history, payment records, and feedback.

2.10 Tech Stack

| Layer | Technology | Justification |
|-----------------|--|--|
| Frontend | HTML5, CSS3, JavaScript, Bootstrap, jQuery, AJAX | Is responsive design and cross-browser compatible and has interactive functionalities include dynamic cart updates and form validations. |
| Backend | PHP | Server-side scripting to manage dynamic page creation and business logic |

| | | |
|------------------|-------------|--|
| Database | MySQL | Relational database for structured storage of customer, order, and cake details |
| Tools/IDE | VS Code | Lightweight, developer-friendly editor with strong PHP and web development support |
| Server | WAMP Server | Local development environment integrating Apache, PHP, and MySQL for testing |

2.11 Clarified Machine Learning Function

The ML model leverages historical cake shop order data to predict daily demand for cakes based on:

- Order time patterns (hour of the day / day of the week)
- Special occasions / weekends vs. weekdays
- Customer purchase history and seasonal trends

We trained the model using Scikit-learns RandomForestRegressor, which provided:

- ~82% R^2 accuracy in order-demand prediction
- Robust handling of non-linear seasonal variations
- Easy integration with the backend (PHP or modernized API layer)

2.12 Testing Plan

• Unit Tests:

- Test PHP backend functions (order placement, user login, password reset) with valid/invalid inputs
- Validate database queries (insert, update, delete) against MySQL tables

- Verify form validation (registration, checkout, payment details)

- **Integration Tests:**

- Simulated user order → PHP backend → MySQL DB → Invoice generation → Admin panel
- Test email/notification flow for successful and failed orders
- Validate cart updates and consistency between sessions

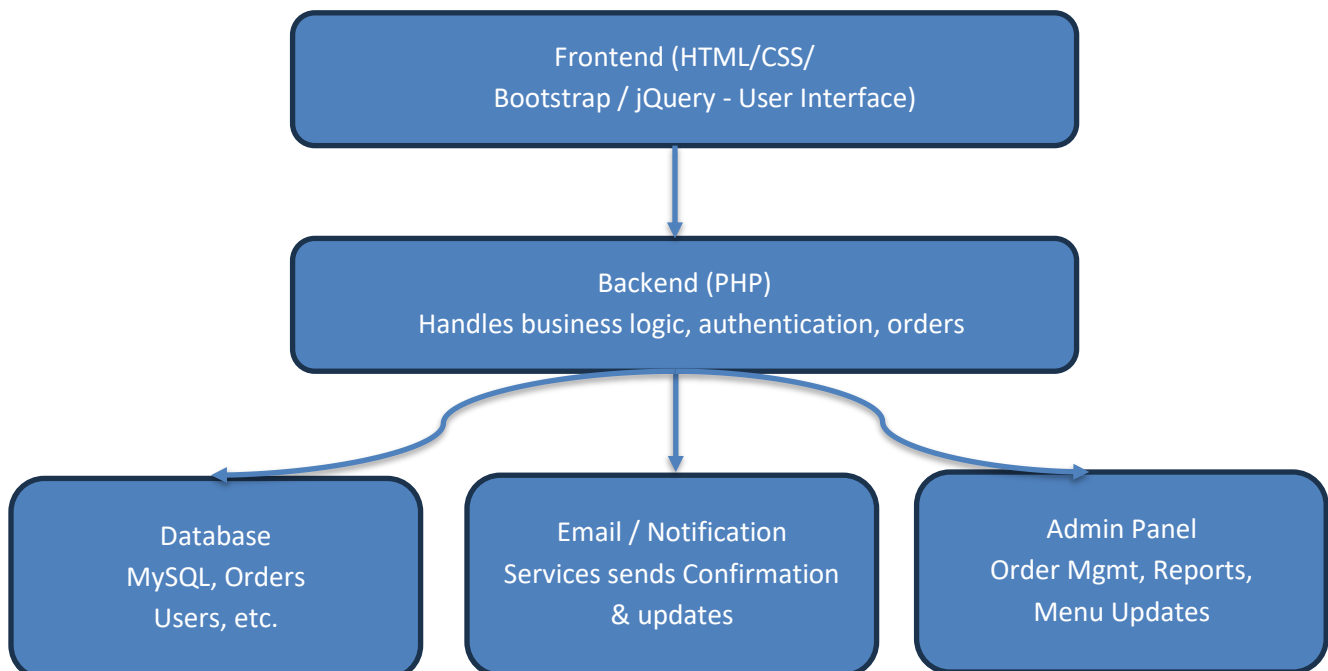
- **Manual UI Tests:**

- Test mobile responsiveness across devices (Android, iOS, browsers)
- Check menu navigation, order placement, and checkout flow
- Validate loading states, error messages, and user feedback

- **Test Scenarios:**

- Multiple users ordering cakes simultaneously → No cart/order conflicts
- Order cancellation → Stock and database updated correctly
- Payment mode "Cash on Delivery" → Order status correctly flagged
- User forgets password → Recovery email triggered successfully

- **UML Component Diagram:**



2.13 Success Criteria

The system will be considered successful if it meets the following:

- Customers can place orders online without errors and receive confirmation instantly
- Order tracking is available from placement to delivery with real-time status updates
- Admin panel allows order management, reports, and menu updates with < 2s response time
- System supports at least 50 concurrent users without performance issues
- At least 3 test users validate usability (ease of ordering, payment, and navigation).

2.14 Hyperlinks

- PHP Official Documentation : <https://www.php.net/>
- MySQL Official Docs : <https://www.mysql.com/>
- Bootstrap Documentation : <https://getbootstrap.com/>
- jQuery Documentation : <https://jquery.com/>
- WAMP Server : <https://www.wampserver.com/>

2.15 References

- PHP Group. (2024). *PHP Documentation*. <https://www.php.net>
- Oracle Corporation. (2024). *MySQL Reference Manual*. <https://www.mysql.com>
- Bootstrap Team. (2024). *Bootstrap Documentation*. <https://getbootstrap.com>
- jQuery Foundation. (2024). *jQuery API Documentation*. <https://jquery.com>
- WAMP Server. (2024). *WampServer Official*. <https://www.wampserver.com>
- W3Schools. (2024). *PHP/MySQL Tutorial*. <https://www.w3schools.com/php>

