CSCI 4210: Introduction to Software Engineering

University of New Orleans
Department of Computer Science

Capstone Project: Touchless Kiosk System

Dev-Cycle 3:
Feature Expansion & Optimization

Cycle Overview

The Feature Expansion and Optimization Cycle focuses on adding new high-priority features and enhancing system performance. This cycle builds on the refined MVP by expanding functionality and optimizing resource management, ensuring a robust, user-friendly system.

Key Objectives

- **Introduce New Features:** Implement additional functionalities identified in user feedback and requirements analysis.
- **Enhance Performance:** Optimize system responsiveness and efficiency for a smoother user experience.
- **Update Testing and Documentation:** Revise testing procedures and documentation to reflect added features and optimizations.

Cycle Tasks

Part 1: Feature Implementation

Objective: Add high-priority features to the system, enhancing the user experience and expanding system capabilities.

1. **Implement Additional Functionalities:**
   - Integrate new features as identified in the product roadmap and user feedback.
   - Ensure seamless interaction between added features and existing components.
2. **Develop Enhanced User Interfaces:**
   - Update UI elements to support new features, maintaining a consistent design.
   - Improve user interface responsiveness and interactivity.
3. **Expand Device Integration (if applicable):**
   - Add support for additional input/output devices or sensors if required by new features.
   - Validate new integrations through testing to ensure reliability.

Part 2: System Optimization

Objective: Optimize performance to handle the expanded feature set, ensuring smooth and efficient operation.

1. **Optimize Resource Management:**
   - Review system performance and identify resource-intensive processes.
   - Implement optimizations to improve speed and reduce latency.
2. **Enhance System Scalability:**
   - Modify architecture to accommodate anticipated increases in user load.
   - Ensure system remains stable and responsive under higher demand.
3. **Implement Advanced Error Handling:**
   - Expand error handling mechanisms to cover new features and integrations.
   - Log errors effectively for improved debugging and support.

Part 3: Testing and Documentation Update

Objective: Ensure all new and existing features are thoroughly tested, and update documentation to reflect changes.

1. **Conduct Feature and Integration Testing:**
    - Test newly implemented features to verify functionality.
    - Perform integration tests to ensure compatibility with existing components.
2. **Revise API and System Documentation:**
    - Update API documentation with endpoints for new features.
    - Ensure user documentation reflects added features and any UI changes.
3. **Generate Optimization Report:**
    - Document performance improvements and resource optimization strategies.
    - Include recommendations for future performance monitoring.

Deliverables

The following deliverables will be tracked via GitHub Issues and GitHub Projects, allowing for transparent progress updates and collaborative management:

- **Feature Implementation Summary:** GitHub Issues for each new feature, with detailed descriptions, associated tasks, and assignment of team members. Use the `Feature Expansion` label for easy identification.
- **Optimization Report:** Document summarizing optimizations made, challenges encountered, and areas for future improvement. Post in the repository under the `Reports` folder.
- **Updated API and User Documentation:** Revised documentation reflecting new features and optimizations, tracked as GitHub Issues under the `Documentation Update` label.
- **Testing Report:** Comprehensive report of tests conducted, including feature and integration tests. Document all test cases and results, tagging this deliverable with the `Testing Report` label.

Each deliverable will be organized and tracked in the GitHub Project board under appropriate categories (e.g., `Backlog`, `In Progress`, `Completed`), ensuring a structured and visible workflow for all team members.

Grading Rubric for Dev-Cycle 3

This rubric evaluates key aspects of the Feature Expansion and Optimization cycle, assessing feature implementation, system optimization, testing, documentation, and project management.

Category	Description	Points
Feature Implementation	Code for each new feature meets requirements, integrates smoothly with existing components, and enhances system functionality effectively. Demonstrates seamless interaction between added and pre-existing components.	25%
System Optimization	Implements effective performance optimizations to improve speed, scalability, and resource management. Includes error handling and resource management strategies.	25%
Testing and Documentation	Conducts thorough testing of new features, integration points, and optimizations. Provides comprehensive and clear documentation updates, covering both API and user-facing changes.	20%
Code Review and Collaboration	Completes peer code review sessions focused on optimization, providing a summary of findings, improvements made, and unresolved issues.	15%
Project Management	Maintains updated project board and task tracking, with clear labels, statuses, and assignments for all Cycle 3 tasks. Demonstrates active management of GitHub Issues and Projects.	15%
Total		**100%**

Detailed Breakdown

- **Feature Implementation (25%):** New features should meet identified user needs and integrate seamlessly with the MVP. Demonstrates smooth interactions and compatibility with existing components.
- **System Optimization (25%):** Optimizations should improve the system's overall performance, with efficient resource management, scalability, and robust error handling.
- **Testing and Documentation (20%):** Tests should thoroughly cover new features and optimized areas, with clear and updated documentation of changes and newly introduced APIs.
- **Code Review and Collaboration (15%):** Summarize peer reviews with highlights on code improvements, optimizations, and any unresolved issues or recommendations.
- **Project Management (15%):** Maintain the GitHub Project board with organized tasks and updates, providing clear labeling and assignments, ensuring consistent tracking.

Appendix: Agile Development Roadmap

This Agile Development Roadmap provides an outline of each development cycle in the project, highlighting the objectives and activities central to each phase.

0. **Discovery and Planning Cycle** – This phase centers on requirements gathering, stakeholder analysis, and initial roadmap creation. It aligns with Agile's discovery phases, where understanding user and system needs is essential for setting a clear development path.

1. **MVP Development and Integration Readiness Cycle** – This cycle emphasizes initial development, focusing on isolated component building and preparing for integration. The goal is to establish a minimal product foundation that can operate independently but is ready for further integration.

2. **User Feedback and Refinement Cycle** – After the MVP is complete, this phase collects user feedback, allowing for testing insights to guide iterative improvements. This cycle ensures the product evolves to meet real user needs effectively.

3. **Feature Expansion and Optimization Cycle** – Building on the MVP, this phase introduces high-priority features and optimizations. It expands the product's functionality and enhances performance, creating a more robust and refined system.

4. **Deployment and Postmortem Analysis Cycle** – The final phase focuses on deploying the application, delivering a structured presentation, and conducting a post-mortem to reflect on the project's journey and lessons learned.

Each cycle represents an integral stage in Agile development, from planning to deployment, ensuring continuous alignment with user needs and system goals.