UE22CS341A: Software Engineering Case Study

Unit 2 Deliverable

Project Plan for Online Retail Database Management System

1. Introduction

The project aims to develop an Online Retail Database Management System that will help manage inventory, customer orders, product details, and transactional records. The platform will ensure secure and efficient handling of retail data for online operations, providing a user-friendly interface for retail companies. The ORDMS project is designed to address the unique needs of small to medium-sized online retailers by integrating all critical operational functions into a single, user-friendly platform. This system aims to streamline processes, reduce manual workloads, and provide real-time insights, thereby enabling retailers to focus on growing their businesses rather than managing complex IT infrastructures.

2. Deliverables of the Project

- Functional Online Retail Database Management System (v1.0) : The core system will include features to manage retail operations, including inventory management, order processing, customer records, and transaction logs.
- User Interface for customer and admin interaction: User Registration and Login, Product browsing and search, Shopping cart and wishlist, Product management, customer and order management, and analytics dashboard.
- Secure API for payment gateway and shipping provider integration: The system will feature an API layer that allows seamless communication with third-party services such as payment gateways and shipping providers
- Comprehensive system and customer reports: The system will offer extensive reporting capabilities to support both operational and strategic decisions
- Compliance with data protection and e-commerce regulations: Ensuring compliance with relevant legal and regulatory frameworks is a core deliverable.

3. Process Model

The project will follow an **Agile development model**, allowing iterative improvements and faster release cycles. Sprint planning will be conducted every two weeks, focusing on delivering key functional modules and integrating feedback after each sprint. The Agile methodology ensures a customer-focused approach, frequent delivery of working features, and the ability to pivot based on stakeholder input. Below is a detailed breakdown of how the Agile model will be applied throughout the project's lifecycle:

3.1. Sprint Planning and Execution

- **Sprint Duration**:
    - Each sprint will last **two weeks**, providing a structured yet flexible development cycle.
- **Sprint Planning**:
    - At the start of each sprint, a **Sprint Planning meeting** will be held to prioritize the tasks and features to be developed.
    - The **Product Owner**, **Scrum Master**, and the **Development Team** will work together to define and clarify user stories, acceptance criteria, and the definition of done for each feature or task.
    - Tasks will be broken down into manageable units and added to the **Sprint Backlog**.
- **Sprint Goals**:
    - Each sprint will focus on delivering specific **functional modules** that can be tested and demonstrated to stakeholders.

3.2. Iterative Development and Continuous Integration

- **Development Process**:
    - The team will follow a **test-driven development (TDD)** approach where unit tests are written before code implementation to ensure that each feature is correctly developed and fully functional.
- **Incremental Feature Release**:
    - Features developed during each sprint will be released in **incremental stages**.
    - At the end of each sprint, a **shippable product increment** will be delivered and made available for testing or deployment in a **staging environment**.

3.3. Feedback Integration

- **End of Sprint Review**:
    - After each sprint, a **Sprint Review (Demo)** will be conducted where the team presents the completed work to stakeholders.
- **Adaptation Based on Feedback**:
    - Any changes, new requirements, or improvements identified during the sprint review will be added to the **Product Backlog**.

3.4. Daily Standup and Communication

- **Daily Standup Meetings**:
    - The team will participate in **Daily Standup (Scrum)** meetings to discuss:
        - What was accomplished the previous day?
        - What tasks are planned for the current day?
        - Any roadblocks or challenges that need to be addressed.

- **Transparent Communication**:
    - Communication between the development team,and product owner, will remain **open and transparent** throughout the project.
    - Tools like **Jira** will be used to manage the backlog, track progress, and ensure visibility for all team members.

3.5. Retrospective and Continuous Improvement

- **Sprint Retrospective**:
    - After each sprint, a **Sprint Retrospective** will be held to reflect on the sprint and identify areas for improvement.
    - The team will evaluate:
        - What went well during the sprint.
        - What challenges were faced?
        - What can be improved in future sprints?
- **Process Optimization**:
    - The Agile framework encourages **continuous improvement**. By regularly evaluating team performance and process effectiveness, adjustments can be made to improve speed, communication, and the overall quality of deliverables.

3.6. Flexible Project Scope and Adaptability

- **Dynamic Backlog Management**:
    - The project will follow a **flexible scope** that allows changes to the Product Backlog based on customer feedback, business goals, and market conditions.

3.7. Final Delivery and Project Wrap-Up

- **Acceptance Criteria**:
    - Upon completion of all planned sprints, the final deliverables will undergo a comprehensive evaluation against the **acceptance criteria** defined at the beginning of the project.
- **Deployment**:
    - After final testing and acceptance, the system will be deployed to the **production environment**. Post-launch support will be provided to ensure a smooth transition.

4. Organization of Project

- **Frontend Development Team:** Builds user interfaces using HTML/CSS.
- **Backend Development Team:** Handles database design and logic with Python and MySQL.
- **Testing Team:** Ensures functionality and security.

- **Project Management:** Coordinates timelines, team communication, and resource allocation.
- **Scrum Master:** Ensuring that the team adheres to the Agile Scrum framework and operates efficiently while minimizing obstacles.

5. Standards, Guidelines, Procedures

- **Code Standards**: Follow industry-standard coding practices and security guidelines OWASP.
- **Documentation**: All code and design changes will be thoroughly documented.
- **Compliance**: Adherence to GDPR, PCI-DSS, and other e-commerce regulations. Ensuring compliance with **PCI-DSS** is essential for handling payment data to protect customer information and avoid costly fines or reputational damage in a breach.
- **Testing Guidelines:** Implement unit testing using unittest or pytest.
- **Version Control:** Manage code with Git for tracking changes and collaboration.

6. Management Activities

- **Sprint Reviews**: Weekly meetings to review progress and adjust priorities.
- **Stakeholder Meetings**: Bi-weekly review sessions with business stakeholders for feedback.
- **Risk Reviews**: Monthly reviews of project risks and mitigation strategies.

7. Risks

- **Third-Party API Dependency**: Potential delays in integrating payment and shipping APIs.
- **Scalability Concerns**: Risk of system performance degradation during peak transaction volumes.
- **Compliance Issues**: Risk of non-compliance with evolving data protection laws.

8. Staffing

- **Project Manager**: 1
- **Front-End Developers**: 2 (both the team members)
- **Back-End Developers**: 2(both the team members)
- **Database Administrator**: : 1 for managing MySQL configurations and ensuring data integrity.
- **Quality Assurance (QA)**: 1 for testing and validating the functionality and performance of the system.

9. Methods and Techniques

- **Development Tools**: HTML, CSS, Javascript React.js for front-end, Python, Django for back-end, MySQL for database.
- **Collaboration Tools**: Jira for task tracking, and GitHub for version control.
- **Agile Methodology:** For iterative development and feedback integration.
- **ER diagrams** for designing database schema

10. Quality Criteria/Assurance

- **Unit Testing**: Minimum 90% code coverage.
- **Integration Testing**: Ensure smooth communication between modules and APIs.
- **User Acceptance Testing (UAT)**: Customers and retail managers will test the system before the final release.

11. Work Packages

- **Product Management Module**: 2 weeks
- **Order Management Module**: 1 week
- **Customer Management Module**: 1 week
- **Report Generation Module**: 1 week
- **Payment Processing Module**: 1 week

12. Resources

- **Software**: AWS Cloud, Azure or Google Cloud for hosting, payment gateway API access.
- **Hardware**: Servers with a backup and disaster recovery setup.
- **Secure Protocols:** Use of HTTPS for web traffic and SSL/TLS for database connections.

13. Budget and Schedule

- **Total Budget**: $0(as this is an academic project and does not use cost-intensive frameworks or tools)
- **Project Duration**:
- Requirements Analysis: 1 week
- Design: 1 week
- Implementation: 4 weeks
- Testing: 2 weeks
- Deployment: 1 weeks

14. Change Control Process

- **Request Submission**: Any change request is submitted via Jira for review.
- **Impact Assessment**: Business analysts and the project manager evaluate the impact on scope, budget, and schedule.
- **Approval**: Changes must be approved by the product owner before being implemented.

15. Delivery Means

- **Staging Environment**: All features will be tested in a staging environment.
- **Production Deployment**: Final deployment to AWS after UAT approval.