E-Commerce Site and Mobile Application for Automobile Spare Parts Manufacturer using SDLC V-Model

Project Overview:

The project involves designing and developing an e-commerce website and a mobile application for a manufacturer of automobile spare parts. The manufacturer will have control over product uploads, pricing, and will facilitate online payments through Razor-pay. This project will follow the SDLC V-Model, emphasizing verification and validation at each phase.

SDLC V-Model Phases:

1. Requirements Analysis:

Verification:

Goal: Gather detailed requirements from the manufacturer.

Output: Requirement Specification Document.

Activities: Conduct meetings with stakeholders to understand requirements for product management, pricing control, and payment integration.

Validation: Review requirements with stakeholders to ensure completeness and clarity.

2. System Design:

Verification:

Goal: Design the overall system architecture.

Output: High-Level Design (HLD) Document.

Activities: Define system components, data flow, and interactions between the website, mobile application, and backend server.

Validation: Review design with stakeholders to ensure it meets requirements.

3. Architecture Design:

Verification:

Goal: Detail the system's technical architecture.

Output: Low-Level Design (LLD) Document.

Activities: Define detailed design for the database schema, API endpoints, and integration points for Razor-pay.

Validation: Conduct design reviews and walkthroughs to ensure technical accuracy and feasibility.

4. Module Design:

Verification:

Goal: Design individual modules or components.

Output: Module Design Specifications.

Activities: Create design documents for key modules such as User Authentication, Product Management, Shopping Cart, and Payment Processing.

Validation: Validate designs through peer reviews and stakeholder feedback.

5. Coding:

Verification:

Goal: Implement the design into code.

Output: Source Code.

Activities: Develop frontend using React.js/Angular.js for the website and Flutter/React Native for the mobile app. Develop backend using Node.js/Express.js or Python/Django/Flask. Integrate Razorpay for payment processing.

Validation: Conduct code reviews and unit testing for each module.

Corresponding Testing Phases:

6. Unit Testing:

Verification:

Goal: Test individual components or modules.

Activities: Write and run unit tests for each module to ensure they function correctly.

Validation: Validate that each unit meets its design specifications and requirements.

7. Integration Testing:

Verification:

Goal: Test interactions between integrated modules.

Activities: Perform integration tests to ensure modules work together as expected.

Validation: Validate the system's behaviour as a whole, ensuring seamless interaction between the website, mobile app, and backend.

8. System Testing:

Verification:

Goal: Test the complete system.

Activities: Conduct system testing to verify that the entire application meets the specified requirements.

Validation: Validate end-to-end functionality, including user registration, product management, shopping cart, and payment processing.

9. Acceptance Testing:

Verification:

Goal: Validate the system against user requirements.

Activities: Conduct user acceptance testing (UAT) with stakeholders.

Validation: Ensure the final system meets business needs and user expectations.

Implementation Details:

Technology Stack:

Frontend (Website): HTML, CSS, JavaScript, React.js or Angular.js

Mobile Application: Flutter or React Native

Backend: Node.js with Express.js or Python with Django/Flask

Database: MongoDB or MySQL/PostgreSQL

Payment Gateway: Razor-pay

Development Process using V-Model:

Requirement Analysis:

Activities: Gather requirements for product upload, pricing control, and payment integration through meetings and workshops.

Output: Requirements Specification Document.

Validation: Review and approval from stakeholders.

System Design:

Activities: Create system architecture diagrams and HLD document.

Output: High-Level Design Document.

Validation: Design review with stakeholders.

Architecture Design:

Activities: Detail database schema, API design, and module interactions.

Output: Low-Level Design Document.

Validation: Technical review and approval.

Module Design:

Activities: Design individual modules and create detailed design documents.

Output: Module Design Specifications.

Validation: Peer reviews and stakeholder feedback.

Coding:

Activities: Develop the frontend, backend, and integrate Razorpay.

Output: Source Code.

Validation: Code reviews and unit testing.

Testing Phases:

Unit Testing:

Activities: Write and run unit tests for each module.

Output: Test Results.

Validation: Ensure individual modules function correctly.

Integration Testing:

Activities: Test interactions between modules.

Output: Integration Test Results.

Validation: Verify module interactions.

System Testing:

Activities: Conduct end-to-end system testing.

Output: System Test Results.

Validation: Ensure the system meets requirements.

Acceptance Testing:

Activities: Perform UAT with stakeholders.

Output: Acceptance Test Results.

Validation: Confirm the system meets business needs.

Timeline:

Requirement Analysis: 2 weeks

System Design: 2 weeks

Architecture Design: 3 weeks

Module Design: 2 weeks

Coding: 6 weeks

Testing Phases: 5 weeks

Deployment: 1 week

Maintenance and Support: Ongoing

Conclusion:

Following the SDLC V-Model ensures that each phase is thoroughly verified and validated before moving to the next, resulting in a high-quality e-commerce website and mobile application that meets the manufacturer’s needs for product management and payment processing.