Test Plan

3.1. Introduction

3.1.1. Purpose

The purpose of this document is to define the testing approach for the **MOW Garden Website** project. It helps the project team clearly understand the test scope, organization, acceptance criteria, potential risks, and the resources required.

The main contents include:

- An overview of the project and the testing scope.
- A list of requirements and acceptance criteria.
- A detailed testing strategy and test plan.
- Resources, test environment, and project milestones.
- Final deliverables.

3.1.2. Definitions, Acronyms, and Abbreviations

Abbreviation	Description
AT	Acceptance Test
IT	Integration Test
SRS	Software Requirement Specification
QA	Quality Assurance
DB	Database
UAT	User Acceptance Testing
UI	User Interface

3.1.3. References

Title/File name	Author	Effective Date
Business Context – MOW Garden	Project Team	2025-09-25
SRS – MOW Garden	Project Team	2025-09-27
Test Guideline Document	Fsoft	2024-12-01

3.1.4. Background information

MOW Garden is an e-commerce website for selling ornamental plants and gardening accessories. The system allows users to:

- Browse product categories, and search and filter products by type or price.

- View detailed product information (images, descriptions, and prices).
- Add or remove products from the shopping cart.
- Place orders (checkout).
- Receive notifications when an order is successfully placed.
- Administrators (Admin) manage inventory and confirm customer orders.

The objective of testing is to ensure that the system operates correctly, reliably, and securely, while delivering a positive user experience.

3.1.5. Scope of testing

In Scope:

- Functional testing (catalog, cart, checkout, inventory, notifications, and access control).
- User Interface testing (UI/UX).
- Database testing (data consistency and no data loss).
- Basic performance testing (small to medium concurrent load).

Out of Scope:

- International payment processing.
- Integration with external logistics systems.
- Advanced non-functional requirements (enterprise-level scalability).

3.1.6. Constraints

- The testing period is limited to two weeks.
- Human resources are limited (4 team members).
- The system is currently deployed only in a local environment and a demo server; a full production environment is not yet available.

3.1.7. Risk List

Risk	Impact	Mitigation
Lack of a production environment	Test results may not fully reflect real-world conditions	Use mock data and a staging server
Short testing timeframe	Some test cases may be missed	Prioritize critical functional test cases

Limited human resources	Potential delays in testing progress	Assign clear roles and focus on high-priority tasks
Lack of automation tools	Heavy reliance on manual testing	Use basic tools such as Postman and JMeter

3.1.8. Training Needs

The team requires short-term training on the following topics:

- Writing and managing test cases.
- Using Postman for API testing.
- Using JMeter for basic performance testing.
- Bug reporting and issue tracking using GitHub.

3.2. Requirements for Testing

3.2.1. Test Items

The following list identifies the main features and functions selected as testing targets for the **MOW Garden** project:

No.	Name of Function	Outline of Functions	Notes
1	Product Catalog	View product lists, search and filter by category and price. View product details (name, description, price, images).	Includes UI testing and search/filter functionality
2	Shopping Cart	Add/remove products to/from the cart, update quantities, calculate total price.	Tested in both logged-in and guest scenarios
3	Checkout Process	Enter order information, select payment method, and confirm orders.	Includes data validation
4	Inventory Management	Manage stock quantities and update inventory upon order placement.	Admin-side functionality
5	Access Control (User/Admin)	User registration, login, and role-based authorization between User and Admin.	Includes authentication and basic security testing

No.	Name of Function	Outline of Functions	Notes
6	Order Management	View order lists, view order details, and track order status (User & Admin).	Includes data display and order status updates
7	Notification Service	Send notifications when an order is successfully placed.	Push notification testing
8	Product Management	Create, read, update, and delete products and link them to inventory.	Admin authorization applied

3.2.2. Acceptance Test Criteria

To be accepted, the software must meet the following testing criteria:

- **Test coverage** ≥ 90%
- **Successful test coverage** ≥ 95%
- All defined test cases (Unit / Integration / System Tests) must be fully executed
- The number of defects or weighted defects must be ≤ 5 per main module

3.3. Features to Be Tested

The following list defines the items—including use cases, functional requirements, and non-functional requirements—selected as testing targets. These items represent the key components that need to be tested in the **MOW Garden** system.

3.3.1. Functionality

3.3.1.1. Product Catalog

- Verify that the system can correctly display the complete list of products retrieved from the database.

- Verify that users can filter products by category, price, or stock status.

- Verify that users can search for products using keywords.

- Verify that the product list is automatically updated when inventory changes occur.

3.3.1.2. Product Details

- Verify that the system displays accurate product information, including name, description, price, images, and remaining quantity.

- Verify that users can view multiple images of the same product.

- Verify that out-of-stock products are clearly indicated and cannot be added to the cart.

- Verify that when an Admin updates product information, the product detail page is updated automatically.

3.3.1.3. Shopping Cart

- Verify that users can add products to the cart from the product list or product detail page.

- Verify that users can update product quantities in the cart.

- Verify that users can remove products from the cart.

- Verify that the cart total price is calculated correctly and displayed in the correct format.

- Verify that if a product is removed or becomes out of stock, the cart status is automatically updated.

- Verify that the system correctly handles adding the same product multiple times.

3.3.1.4. Checkout Process

- Verify that users can enter shipping and payment information.

- Verify that the system validates all required input data (name, phone number, address).

- Verify that orders are stored in the database and displayed in the order history.

- Verify that users receive notifications after successful order placement.

- Verify that inventory quantities are reduced correctly after checkout.

- Verify that order status is updated after payment (e.g., “Confirmed”, “Shipping”).

3.3.1.5. Inventory Management

- Verify that inventory quantities are automatically updated after each order.

- Verify that out-of-stock products cannot be added to the cart.

- Verify that Admin users can update product information (price, stock, description).

- Verify that inventory is updated correctly when orders are completed or canceled.

- Verify that inventory changes are reflected in real time on the product detail page.

3.3.1.6. Access Control

- Verify that the system correctly enforces authorization between buyers and admins.

- Verify that users can only perform actions appropriate to their assigned roles.

- Verify that Admin users can add, edit, and delete products, while buyers can only purchase products.

- Verify that users can update their personal information.

- Verify that the “Forgot Password” functionality works correctly.

- Verify that Admin users can lock or unlock user accounts.

- Verify that users can update personal details such as name, address, and phone number.

3.3.1.7. Notification Service

- Verify that users receive notifications after successful order placement.

- Verify that Admin users can send promotional or order status notifications to users.

- Verify that notifications are displayed correctly on the user interface.

3.3.1.8. Order Management

- Verify that users can view their order list, including order ID, creation date, total amount, and status.

- Verify that users can view detailed information for each order (shipping info, items, invoice, status).

- Verify that order status is displayed correctly according to progress (Pending → Processing → Shipping → Completed).

- Verify that Admin users can view all customer orders.

- Verify that Admin users can update order status (e.g., from "Processing" to "Completed").

- Verify that users receive notifications when Admin updates order status.

3.3.1.9. Product Management

- Verify that Admin users can add new products with complete information (name, description, price, images, stock).

- Verify that Admin users can edit product information.

- Verify that Admin users can delete products from the catalog.

- Verify that Admin users can assign products to inventory.

- Verify that the system prevents duplicate product names or IDs.

- Verify that when Admin users update or delete products, related data in **Product Catalog** and **Inventory** is updated accordingly.

3.3.2. Usability

3.3.2.1. Session Support

- Verify that users remain logged in throughout the working session.

- Verify that cart data is preserved when users reload the page or temporarily leave the site.

- Verify that the system automatically logs users out after a defined period of inactivity for security purposes.

- Verify that when users manually log out, all temporary session data is securely cleared.

3.3.2.2. Display Support

- Verify that the interface is displayed correctly on common browsers (Chrome, Edge, Safari, Firefox).

- Verify that the layout automatically adapts to different screen sizes (desktop, tablet, mobile).

- Verify that fonts, colors, and icons are clearly displayed and do not change across different resolutions.

- Verify that the system supports dark mode or customizable UI settings (if applicable).

3.3.3. Design Constraints

3.3.3.1. Test Environment

- Verify that the system operates stably on the defined test environments (Windows 10/11, macOS).

- Verify that supported browser versions are clearly defined (e.g., Chrome 120+, Edge 110+).

- Verify that the backend (Node.js/Express) and frontend (ReactJS) are successfully deployed in the staging environment.

- Verify that the database system (MongoDB) operates normally and data is stored and retrieved accurately.

3.3.3.2. Character Support

- Verify that the system supports Vietnamese characters, special characters, currency symbols, and emojis.

- Verify that data containing special characters does not cause display or database errors.

- Verify that the system correctly handles Unicode input (e.g., names, addresses, product descriptions).

3.3.4. Interfaces

- Verify that page layouts (Home, Product List, Product Detail, Cart, Checkout, Admin Dashboard) are properly organized.

- Verify that all buttons, links, and UI elements respond correctly to user interactions (hover, click, focus).

- Verify that input forms display clear validation messages when data is missing or invalid.

- Verify that system notifications (toast messages, alerts) clearly indicate success or failure.

- Verify that the Admin interface allows intuitive management of products, orders, and inventory.

- Verify that backend APIs provide accurate data to the frontend, return valid JSON responses, and use correct HTTP status codes.

3.4. Features Not to Be Tested

The following list includes features, functions, or requirements that are **not included in the testing scope** of the current release. These features may be considered in future development phases or extended versions.

3.4.1. Functional Features

- Third-Party Online Payment Integration (Payment Gateway Integration):
Integration and transaction processing with payment services such as PayPal, MoMo, ZaloPay, or VNPay have not been implemented and are therefore not tested in this phase.

- Product Recommendation System:
Features that recommend products based on user behavior or purchase history are not included in the current testing scope.

- Advanced User Management Features:
Advanced user management functions such as password recovery, two-factor authentication (2FA), and social login (Google, Facebook) have not been implemented and are not tested in this phase.

3.4.2. Non-Functional Features

- Advanced Performance Testing (Stress Testing):
Performance testing under extremely high concurrent user loads (over 10,000 simultaneous users) is not conducted in this phase and will be considered in future stages.

- Scalability Testing:
Scalability tests involving server clustering, load balancing, or enterprise-level infrastructure are outside the current testing scope.

- Advanced Security Testing:
In-depth security testing such as penetration testing or advanced injection attack simulations is not performed in this phase.

3.5.1.1. Functional Testing

Functional Testing aims to ensure that all system functions operate correctly according to the specified requirements.
For the **MOW Garden** system, this testing focuses on core functionalities such as product browsing, searching, adding items to the cart, checkout, payment, account management, and inventory management.

Item	**Description**
Test Objective	Verify that core system functionalities (view products, search, add to cart, checkout, account management, inventory management) work correctly according to business requirements.
Technique	- Execute test cases based on use cases and functional requirements - Use valid and invalid test data - Verify input validation and business rules - Tools: Postman, Selenium, JUnit
Completion Criteria	- All functional test cases are executed - Critical defects are fixed - No blocking defects remain
Special Considerations	Special focus on critical flows such as adding to cart without login, canceling orders during processing, and temporary cart storage

3.5.1.2. Business Cycle Testing

Business Cycle Testing ensures that the entire business workflow operates smoothly from start to end without interruption.
In **MOW Garden**, this includes the flow: product selection → add to cart → checkout → inventory update → order management.

Item	**Description**
Test Objective	Ensure that core business processes (purchase → payment → inventory update) operate smoothly
Technique	- Perform end-to-end testing - Verify data flow across modules (Product → Cart → Order → Inventory)
Completion Criteria	All core business workflows are completed successfully without errors
Special Considerations	Simulate real scenarios such as order cancellation, payment failure, and out-of-stock situations

3.5.1.3. User Interface Testing

User Interface Testing ensures that the web interface is displayed correctly, is easy to use, and responds accurately to user interactions.

Item	**Description**
Test Objective	Ensure the UI is displayed correctly, responsive, and user-friendly
Technique	Cross-browser testing (Chrome, Firefox, Edge, Safari), responsive testing (desktop, tablet, mobile), exploratory testing, Selenium
Completion Criteria	No Major or Blocker UI defects; at least 95% of target pages pass responsive testing

Item	Description
Special Considerations	Test under slow network conditions, large image loading, and font fallback behavior

3.5.1.4. Security and Access Control Testing

This testing ensures that the system is secure, only authorized users can access protected resources, and user data is protected.

Item	Description
Test Objective	Verify system security, access control, and protection against attacks
Technique	- Test login/logout and session timeout - Check SQL Injection, XSS, CSRF - Verify admin/user authorization
Completion Criteria	- No authentication bypass - No critical vulnerabilities from OWASP Top 10
Special Considerations	Verify SSL/HTTPS usage and password encryption

3.5.1.5. Performance and Load Testing

Performance and Load Testing evaluates whether the system meets performance and stability requirements under concurrent access.

Item	Description
Test Objective	Ensure the system operates stably under high load (many concurrent users and products)

Item	Description
Technique	- Use JMeter or Locust to simulate 1,000+ concurrent users - Perform Load Test, Stress Test, and Volume Test
Completion Criteria	- Average response time < 3 seconds with 500 concurrent users - System does not crash under load
Special Considerations	Define specific SLA thresholds (response time, maximum concurrent users)

3.5.1.6. Regression Testing

Regression Testing ensures that existing functionalities continue to work correctly after new features are added or bugs are fixed.

Item	Description
Test Objective	Verify that existing functionalities are not affected by changes
Technique	- Re-run previously passed test cases - Use automation tools (Selenium, Jest)
Completion Criteria	- All existing test cases pass - No unexpected defects are introduced
Special Considerations	Integrate regression testing into the CI/CD pipeline

3.5.1.7. Data and Database Integrity Testing

This testing ensures that data is not lost, duplicated, or corrupted during system operations.

Item	**Description**
Test Objective	Ensure product, cart, order, and user data is stored accurately and consistently
Technique	- Perform CRUD operations on the database - Verify transactions during checkout - Validate primary and foreign key constraints - Use SQL scripts and automation
Completion Criteria	- No data loss or duplication - Data integrity maintained across database tables
Special Considerations	Verify rollback behavior when errors occur

3.5.2. Test Stages

Different types of testing are executed at different stages of the testing lifecycle:

- **Unit Test:** Testing individual functions or modules, usually performed by developers.
- **Integration Test:** Testing interactions between multiple modules or APIs.
- **System Test:** Testing the entire system as a complete product.
- **Acceptance Test:** Validating that the system meets business requirements.

Type of Test	**Unit**	**Integration**	**System**	**Acceptance**
Functional Test	✓	✓	✓	✓
Business Cycle Test		✓	✓	✓
User Interface Test		✓	✓	✓
Data & Database Integrity Test	✓	✓	✓	

Type of Test	**Unit**	**Integration**	**System**	**Acceptance**
Performance Test			✓	
Security & Access Control Test		✓	✓	
Regression Test	✓	✓	✓	

- **Functional Testing:** Applied at all stages from Unit to Acceptance.
- **Business Cycle Testing:** Mainly executed from Integration stage onward.
- **UI Testing:** Starts from Integration (Frontend + Backend), then System and Acceptance stages.
- **Database Testing:** Begins at Unit level and continues through Integration and System stages.
- **Performance Testing:** Mainly executed at System Test stage.
- **Security Testing:** Starts from Integration stage (API, authorization, login).
- **Regression Testing:** Continuously executed from Unit → Integration → System to ensure changes do not break existing functionality.

3.5.3. Test Phases

Phase	**Objective**	**Main Activities**	**Responsible**	**Deliverables**
1. Unit Test	Verify that individual functions or small modules work correctly according to logic	- Write test code (Jest, Mocha) - Verify results of each function	Developer	- Unit Test Report - Tested source code
2. Integration Test	Ensure that modules (Catalog, Cart, Checkout, Inventory) are correctly	- Test API integration between Frontend and Backend - Test database	Developer + Tester	- Integration Test Report - Defect Log

Phase	**Objective**	**Main Activities**	**Responsible**	**Deliverables**
	connected and exchange data accurately	transactions - Test API access permissions		
3. System Test	Verify the entire MOW Garden system as a complete product	- Execute comprehensive test cases (functional, UI, security, performance) - Test end-to-end workflow: Product → Cart → Checkout → Order → Inventory	Tester Team	- System Test Result - Defect Report - Test Summary
4. User Acceptance Test (UAT)	Ensure the system meets business requirements and is ready for end users	- Execute test cases based on user scenarios - Verify UI, business processes, and system stability - Collect user feedback	Tester + Client Representative	- UAT Sign-off Document - Acceptance Report
5. Regression Test	Ensure existing functionalities continue to work correctly after bug fixes or updates	- Re-run previously passed test cases - Perform regression	Tester Team	- Regression Report - Updated Defect Log

Phase	Objective	Main Activities	Responsible	Deliverables
		automation using Selenium		

3.6. Resource

3.6.1. Human Resource

Worker/Doer	Role	Specific Responsibilities	Location
Nguyễn Nhật Hải	Tester	Execute test cases, report defects	SGU – Ho Chi Minh City
Nguyễn Thành Việt	Tester	API testing, integration testing	SGU – Ho Chi Minh City
Nguyễn Thành Trung Hiếu	Tester	UI testing, regression testing	SGU – Ho Chi Minh City
Nguyễn Thành Đạt	Leader	Test planning, task assignment, reporting	SGU – Ho Chi Minh City

3.6.2. Test Management

Test Management Approach:

- Create test plans, assign tasks, and track progress using **Jira / Trello**.
- Conduct daily stand-up meetings to update testing progress.
- Submit weekly progress reports to the Project Manager (PM).

Defect Management:

- Record and track defects using **Jira**.
- Classify defects by severity level: **Critical, Major, Minor**.

Defect Handling Workflow:

Log defect → Assign to Developer → Fix defect → Re-test → Close defect

3.7. Test Environment

3.7.1. Hardware

Laptop/PC (for web testing):

- CPU: Intel Core i5 or higher
- RAM: 8GB or more
- Storage: SSD 256GB
- Quantity: 4 machines
- Usage duration: Entire testing period

Network Devices:

- Wi-Fi Router (IEEE 802.11ac standard)
- Internet connection speed: minimum 50 Mbps
- Quantity: 1 set

3.7.2. Software

Operating Systems:

- Windows 10 Pro (64-bit)
- Ubuntu 20.04 LTS

Web Browsers:

- Google Chrome (latest version)
- Mozilla Firefox (latest version)
- Microsoft Edge (latest version)

Testing Tools:

- Postman (API Testing)
- Selenium WebDriver (Automation Testing)
- JMeter (Performance Testing)

3.7.3. Infrastructure

Purpose	**Tool**	**Vendor / In-house**	**Version**
Defect logging	Jira	Atlassian	Cloud Version
Test case management	TestRail	Gurock Software	7.5
Task management	Trello	Atlassian	Cloud Version
Source code management	GitHub	GitHub Inc.	Cloud Version
Performance testing	JMeter	Apache	5.6
Time tracking (test effort)	Timesheet	In-house	1.2

3.8. Test Milestones

Test Phase	**Activity Description**	**Planned Date**
Test Planning	Complete the Test Plan, define test scope, objectives, resources, and test environment	01/10/2025 – 03/10/2025
Test Design	Analyze requirements, write test cases, prepare test data, and map test cases to user stories	04/10/2025 – 06/10/2025
Test Environment Setup	Set up the testing environment (frontend, backend, database, test accounts)	07/10/2025
Test Execution (System & Integration)	Execute Functional Testing, Business Flow Testing, UI Testing, and Security Testing. Record test results and log defects	08/10/2025 – 13/10/2025
Performance & Regression Testing	Execute performance testing, load testing, and regression testing after bug fixes	14/10/2025 – 15/10/2025

Test Phase	**Activity Description**	**Planned Date**
User Acceptance Testing (UAT)	Client/Product Owner validates system functionality based on business requirements	16/10/2025 – 17/10/2025
Test Closure & Reporting	Consolidate results, evaluate product quality, create Test Summary Report, and document Lessons Learned	18/10/2025

3.9. Deliverables

Deliverable	**Description**	**Responsible**
Test Plan Document	Overall test plan including scope, strategy, and schedule	Test Leader
Test Case Specification	Detailed test cases for each module (Catalog, Cart, Checkout, etc.)	Tester
Test Data Set	Test data used during testing (user data, sample products, sample orders)	Tester
Test Execution Log	Records of test execution status and results for each test case	Tester
Defect Report / Defect Log	List of defects identified during testing	Tester / Test Leader
Performance Report	Performance and load testing results (response time, throughput)	Tester
Test Summary Report	Overall testing results, pass/fail rates, and product quality assessment	Test Leader

Deliverable	Description	Responsible
UAT Sign-off Document	Formal acceptance confirmation from end users	Product Owner / Customer