

UniStyle E-Commerce Test Plan

Test Plan ID TP-UNISTYLE-01

Version	v1.0														
Date	2025-10-19														
Project Name	UniStyle, E-Commerce Web App														
Test Manager	Garam Yoon														
Test Objectives	This test plan focuses on early defect detection and acceptance criteria validation for prioritized user stories.														
Test Scope	<p>API Testing</p> <ul style="list-style-type: none">- Authentication & JWT- Product CRUD- Cart & order services- Payment integrations <p>System Testing</p> <ul style="list-style-type: none">- Core user-facing features- Role-based access via UI- Form validation & navigation <p>User Acceptance Testing</p> <ul style="list-style-type: none">- End-to-end user scenarios- Usability & UX validation- Business acceptance criteria <p>Unit Testing</p> <ul style="list-style-type: none">- Out of QA scope (handled by development)														
Test Strategy	<table><thead><tr><th>Testing Approach</th><th>Purpose</th></tr></thead><tbody><tr><td>Black-box</td><td>Validate system behavior via inputs/outputs</td></tr><tr><td>Requirement-based</td><td>Validate prioritized requirements and AC</td></tr><tr><td>Risk-based</td><td>Focus on high-impact user stories</td></tr><tr><td>Exploratory</td><td>Find negative and edge cases</td></tr><tr><td>Manual</td><td>Verify new features</td></tr><tr><td>Automated</td><td>Enable regression and early detection</td></tr></tbody></table>	Testing Approach	Purpose	Black-box	Validate system behavior via inputs/outputs	Requirement-based	Validate prioritized requirements and AC	Risk-based	Focus on high-impact user stories	Exploratory	Find negative and edge cases	Manual	Verify new features	Automated	Enable regression and early detection
Testing Approach	Purpose														
Black-box	Validate system behavior via inputs/outputs														
Requirement-based	Validate prioritized requirements and AC														
Risk-based	Focus on high-impact user stories														
Exploratory	Find negative and edge cases														
Manual	Verify new features														
Automated	Enable regression and early detection														
Success Criteria	<ol style="list-style-type: none">1. All selected user stories are covered by testing2. All related business rules are validated3. All identified defects are logged and reported4. Regression tests for stable features are automated														

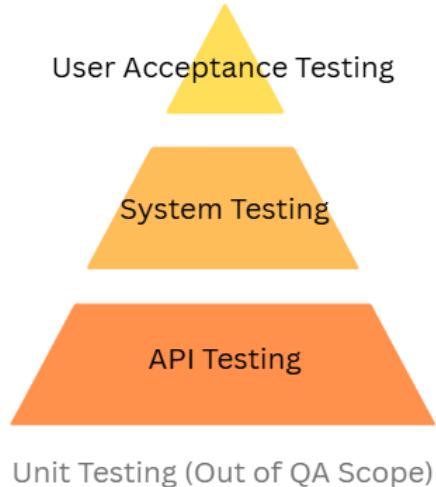


Figure 1: Test Scope Pyramid

Risk Assessment	<ol style="list-style-type: none"> 1. Schedule constraints may impact test planning and execution. 2. Gaps between estimated and actual effort may affect testing efficiency. 3. Changes introduced during development may influence test prioritization and coverage. 																		
Resources	<table border="1"> <thead> <tr> <th>Role</th><th>Name</th><th>Responsibilities</th></tr> </thead> <tbody> <tr> <td>QA</td><td>Garam</td><td>API testing (Postman)</td></tr> <tr> <td>Engineer</td><td>Yoon</td><td>System testing (Selenium)</td></tr> <tr> <td></td><td></td><td>Manual User Acceptance Testing (UAT)</td></tr> <tr> <td></td><td></td><td>Regression Testing Automation</td></tr> </tbody> </table>	Role	Name	Responsibilities	QA	Garam	API testing (Postman)	Engineer	Yoon	System testing (Selenium)			Manual User Acceptance Testing (UAT)			Regression Testing Automation			
Role	Name	Responsibilities																	
QA	Garam	API testing (Postman)																	
Engineer	Yoon	System testing (Selenium)																	
		Manual User Acceptance Testing (UAT)																	
		Regression Testing Automation																	
Dependencies	<table border="1"> <thead> <tr> <th>Dependency</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Testable User Stories</td><td>Selected user stories must be implemented to a testable state before execution</td></tr> <tr> <td>Approved Acceptance Criteria</td><td>Acceptance criteria for selected user stories must be clearly defined and agreed upon</td></tr> <tr> <td>Backend API Availability</td><td>Required APIs must be deployed and accessible</td></tr> <tr> <td>Database Readiness</td><td>Database schema and required test data must be available</td></tr> <tr> <td>Authentication & Authorization</td><td>JWT authentication and role-based access control must be correctly implemented</td></tr> <tr> <td>Third-Party Services</td><td>External services (e.g., payment gateways, image storage) must be available in test/sandbox mode</td></tr> <tr> <td>Dependency Stability</td><td>No critical dependency failures that prevent normal system behavior</td></tr> <tr> <td>Test Data Availability</td><td>Required test data must be prepared for execution</td></tr> </tbody> </table>	Dependency	Description	Testable User Stories	Selected user stories must be implemented to a testable state before execution	Approved Acceptance Criteria	Acceptance criteria for selected user stories must be clearly defined and agreed upon	Backend API Availability	Required APIs must be deployed and accessible	Database Readiness	Database schema and required test data must be available	Authentication & Authorization	JWT authentication and role-based access control must be correctly implemented	Third-Party Services	External services (e.g., payment gateways, image storage) must be available in test/sandbox mode	Dependency Stability	No critical dependency failures that prevent normal system behavior	Test Data Availability	Required test data must be prepared for execution
Dependency	Description																		
Testable User Stories	Selected user stories must be implemented to a testable state before execution																		
Approved Acceptance Criteria	Acceptance criteria for selected user stories must be clearly defined and agreed upon																		
Backend API Availability	Required APIs must be deployed and accessible																		
Database Readiness	Database schema and required test data must be available																		
Authentication & Authorization	JWT authentication and role-based access control must be correctly implemented																		
Third-Party Services	External services (e.g., payment gateways, image storage) must be available in test/sandbox mode																		
Dependency Stability	No critical dependency failures that prevent normal system behavior																		
Test Data Availability	Required test data must be prepared for execution																		
Deliverables	<table border="1"> <thead> <tr> <th>Test Phase</th><th>Deliverables / Activities</th></tr> </thead> <tbody> <tr> <td>Before Testing</td><td> Prioritized test cases Requirement Traceability Matrix (RTM) Test scripts Test data </td></tr> <tr> <td>During Testing</td><td> Test case and script execution Defect logging and tracking Test progress monitoring </td></tr> <tr> <td>After Testing</td><td> Test execution reports Bug reports Test metrics: <ul style="list-style-type: none"> - Requirement / user story coverage - Defect severity distribution - Defect detection rate </td></tr> </tbody> </table>	Test Phase	Deliverables / Activities	Before Testing	Prioritized test cases Requirement Traceability Matrix (RTM) Test scripts Test data	During Testing	Test case and script execution Defect logging and tracking Test progress monitoring	After Testing	Test execution reports Bug reports Test metrics: <ul style="list-style-type: none"> - Requirement / user story coverage - Defect severity distribution - Defect detection rate 										
Test Phase	Deliverables / Activities																		
Before Testing	Prioritized test cases Requirement Traceability Matrix (RTM) Test scripts Test data																		
During Testing	Test case and script execution Defect logging and tracking Test progress monitoring																		
After Testing	Test execution reports Bug reports Test metrics: <ul style="list-style-type: none"> - Requirement / user story coverage - Defect severity distribution - Defect detection rate 																		