



**Aim:** To learn and understand the way of developing the software by classical methods of software engineering. Planning and monitoring, testing, validating of the project using tools and prepares a document for the same by using the concept of software engineering



# Books/ Journals/ Websites referred:

1. Roger Pressman, Software Engineering: A practitioners Approach, McGraq Hill, 2010 ,6th edition
2. Ian Somerville , Software Engineering , Addison Wesley,2011,9th edition

3 <http://en.wikipedia.org/wiki/Software_requirements_specification>



ONLINE PERFUME STORE – LE MIELE MISTS

***Prepared by:***

Rishil Desai – 16010122039

Rohit Deshpande – 16010122041

Jeet Gada – 16010122053

Kunj Gohil – 16010122060

**1.0 INTRODUCTION**

The *Le Miele Mists* Online Perfume Store allows users to browse through a curated selection of perfumes, view detailed product descriptions, add products to their cart, and complete purchases through an integrated payment gateway. The system includes functionalities such as user account management, search and filter options, personalized recommendations, order tracking, and integration with third-party logistics providers for shipment tracking. This test plan will ensure that each functionality is thoroughly tested to provide a smooth and enjoyable shopping experience for users.

**2.0 OBJECTIVES AND TASKS**

The primary objectives of this test plan are:

* To define the scope and approach for testing the *Le Miele Mists* online store.
* To assign roles and responsibilities for each phase of testing.
* To establish a comprehensive testing structure to identify and resolve issues related to functionality, performance, and security before the system’s official launch.

**2.2 Tasks**

The main tasks included in this test plan are:

* **Requirement Analysis**: Identify and analyze functional and non-functional requirements.
* **Test Case Design**: Create test cases for each functionality (product search, filtering, checkout, etc.).
* **Unit Testing**: Test each individual module, such as the product search or shopping cart.
* **System and Integration Testing**: Test the interactions between modules, such as from product selection to checkout.
* **User Acceptance Testing (UAT)**: Confirm with end-users that the system meets user expectations.
* **Defect Logging and Tracking**: Record, track, and fix issues identified during testing.
* **Regression Testing**: Ensure that updates do not impact existing functionalities.

**3.0 SCOPE**

This test plan covers all major functionalities of the *Le Miele Mists* online perfume store. The scope includes:

* **Functional Testing**: For product search, filtering, cart management, checkout, and order tracking.
* **Non-Functional Testing**: For system performance under high load, security testing of user data, and response time.
* **Interface Testing**: Validation of interactions with external systems like payment gateways and shipping providers.

To achieve this scope, we will:

* Use automation tools for regression testing.
* Simulate multiple users for stress testing.
* Test on various devices and browsers for compatibility.

**4.0 TESTING STRATEGIES**

**4.1 Unit Testing**

* **Definition**: Verify the functionality of individual modules (e.g., product search).
* **Participants**: Developers will conduct unit tests, with QA team reviewing test cases.
* **Methodology**: Each module will be tested independently to confirm it performs as expected.

**4.2 System and Integration Testing**

* **Definition**: Test to ensure all modules work together seamlessly.
* **Participants**: QA team.
* **Methodology**: Simulate a complete user flow, from browsing and adding items to the cart to checking out. Testing will confirm data integrity across modules.

**4.3 Performance and Stress Testing**

* **Definition**: Evaluate system performance under normal and peak loads.
* **Participants**: Performance testing team.
* **Methodology**: Simulate high user traffic and multiple transactions to assess system stability and response times.

**4.4 User Acceptance Testing (UAT)**

* **Definition**: Ensure the platform meets user requirements and expectations.
* **Participants**: Selected group of end-users or representatives.
* **Methodology**: Users will test scenarios like purchasing a product or tracking an order to confirm the system meets their needs.

**4.5 Batch Testing**

* **Definition**: Ensure batch processes, such as bulk order processing, function correctly.
* **Participants**: QA and database admin team.
* **Methodology**: Run batch processes to ensure performance and reliability.

**4.6 Automated Regression Testing**

* **Definition**: Confirm that updates don’t disrupt existing functionality.
* **Participants**: Automation testing team.
* **Methodology**: Automated scripts will run after each significant change to verify stability.

**4.7 Beta Testing**

* **Participants**: Selected group of actual users.
* **Methodology**: Release the system to a limited audience for feedback and bug identification before full launch.

**5.0 HARDWARE REQUIREMENTS**

* **Computers**: Desktops or laptops with at least 8GB RAM and 250GB storage.
* **Servers**: A reliable hosting environment to handle web traffic and database operations.
* **Network Equipment**: To simulate different internet conditions.

**6.0 ENVIRONMENT REQUIREMENTS**

**6.1 Server**

* **Operating System**: Windows/Linux.
* **Database**: MySQL or other relational database systems.
* **Security**: Implement role-based access control to protect user data.

**6.2 Client Side**

* **Web Browsers**: Chrome, Firefox, Safari, and Edge for compatibility.
* **Network Conditions**: Simulate different speeds to assess performance.

**7.0 TEST SCHEDULE**

* **Unit Testing**: 2 weeks.
* **System & Integration Testing**: 3 weeks.
* **UAT**: 1 week.
* **Final Beta Testing**: 1 week (adjustable as needed).

**8.0 CONTROL PROCEDURES**

**Problem Reporting**

* **Procedure**: All issues identified will be logged in a bug tracking system (e.g., Jira), including:
  + Issue description
  + Steps to reproduce
  + Severity and priority
  + Assigned developer for resolution
  + Status updates until resolution

**Change Requests**

* **Procedure**: Any changes to features or functionality will go through a Change Control Board (CCB) and require approval from key stakeholders before implementation.

**9.0 FEATURES TO BE TESTED**

**Product Search and Filter:**

* **Objective**: Verify search accuracy based on keywords, categories, and price range.
* **Test Cases**:
  + Valid search with existing products.
  + Invalid search with non-existent products.
  + Boundary cases (e.g., price filter extremes).

**Shopping Cart Management:**

* **Objective**: Ensure items can be added, updated, and removed from the cart.
* **Test Cases**:
  + Adding and removing items.
  + Adjusting quantity and validating price updates.
  + Checking stock availability before checkout.

**Payment Processing:**

* **Objective**: Confirm transaction handling and notifications.
* **Test Cases**:
  + Successful transaction with valid card details.
  + Handling of declined transactions.
  + Confirmation emails sent after payment.

**10.0 FEATURES NOT TO BE TESTED**

**Integration with Shipping Providers**:

* **Scope**: While system-level validation for shipment tracking will be tested, direct integration with third-party logistics providers will not be included.

**User Profile Security**:

* **Scope**: While user registration and basic profile functionalities will be tested, detailed security audits on user authentication are out of scope.

**11.0 RESOURCES/ROLES & RESPONSIBILITIES**

**Test Manager**:

* Oversee the test plan, coordinate efforts, and ensure timely reporting.

**QA Engineers**:

* Design and execute test cases, track defects, and conduct regression testing.

**Developers**:

* Resolve defects identified during testing and perform unit testing on new features.

**Stakeholders**:

* Approve change requests and prioritize features.

**12.0 SCHEDULES**

**Deliverables:**

* **Test Plan**: Due by [insert date].
* **Test Cases**: Due by [insert date].
* **Test Incident Reports**: Ongoing during testing.
* **Test Summary Reports**: Due at the end of the testing phase.

**13.0 SIGNIFICANTLY IMPACTED DEPARTMENTS**

* **Customer Support**: Participate in UAT and assist in gathering feedback.
* **Operations**: Coordinate deployment and ensure smooth user transition.
* **Development Team**: Resolve defects and make changes based on feedback.

**14.0 DEPENDENCIES**

* **Test Environment**: Access to a stable test environment is essential.
* **Data Availability**: Realistic sample data for transactions and product testing.
* **Resource Availability**: Ensure QA Engineers and Developers are available.

**15.0 RISKS/ASSUMPTIONS**

**Risks**:

* Delay in environment setup may impact timelines.
* Insufficient test data could limit testing accuracy.

**Contingencies**:

* Prepare minimal test data to start testing if delays occur.
* Reschedule as necessary to meet project deadlines.

**16.0 TOOLS**

* **Automation**: Selenium for regression testing.
* **Bug Tracking**: Jira.
* **Test Management**: TestRail (optional).
* **Performance Testing**: JMeter.

**17.0 APPROVALS**

Mrs. Poonam Bhogle:

**18.0 TEST CASES**

Here is an outline of some of the detailed test cases for the major functionalities of the system:

**Product Search and Filtering**

* **Objective**: Ensure users can effectively search and filter perfumes by criteria such as fragrance notes, brand, price range, and user ratings.
* **Test Cases**:
  + Search for a product using a specific keyword.
  + Filter products by price range and verify that only products within that range are displayed.
  + Test sorting options (e.g., sorting by price, popularity, or newest arrivals).
  + Boundary testing on search results (e.g., no results for invalid search terms).
  + Confirm that applying multiple filters works as expected.

**Product Detail Page**

* **Objective**: Validate that the product detail page displays accurate and complete information.
* **Test Cases**:
  + Verify that product descriptions, images, and pricing are correct.
  + Ensure that fragrance notes and ingredients are displayed accurately.
  + Test that the “Add to Cart” button functions as expected.
  + Verify that related product recommendations are relevant to the current product.
  + Check for image zoom functionality, if available.

**Shopping Cart Management**

* **Objective**: Confirm that users can add, update, and remove items from their shopping cart.
* **Test Cases**:
  + Add items to the cart and verify correct quantities and prices.
  + Modify item quantities and check that the cart updates the total price.
  + Remove items and verify the cart adjusts accordingly.
  + Validate that the system displays an error message if an out-of-stock item is added.
  + Ensure that the cart persists even if the user navigates to other pages or logs out and back in.

**Checkout Process**

* **Objective**: Ensure the checkout process is smooth, intuitive, and secure.
* **Test Cases**:
  + Validate that the user can enter shipping and billing information correctly.
  + Test the handling of invalid or incomplete payment information.
  + Check that taxes, shipping charges, and any discounts are calculated correctly.
  + Verify that the user receives an order confirmation email upon successful payment.
  + Confirm the functionality of “Save Address” and “Apply Promo Code” options.

**Payment Processing**

* **Objective**: Ensure that payments are securely processed and confirmed.
* **Test Cases**:
  + Successful transaction with valid card information.
  + Transaction declined with invalid card details or insufficient funds.
  + Validation of security features such as SSL encryption during checkout.
  + Check that users receive error messages for declined payments.
  + Verify that a payment confirmation email is sent after a successful purchase.

**Order Tracking**

* **Objective**: Confirm that users can track the status of their orders.
* **Test Cases**:
  + Verify that order status updates are displayed correctly in the user’s account.
  + Ensure that users can see tracking information once the order is shipped.
  + Check for proper status handling (e.g., “Processing,” “Shipped,” “Delivered”).
  + Test email notifications for order updates.

**19.0 FEATURES NOT TO BE TESTED**

The following features are out of scope for this test plan and will not be tested:

**Third-Party Payment Integration**

* **Scope**: While payment processing is tested at the system level, direct testing of third-party payment providers (e.g., PayPal, Stripe) will not be performed. This includes testing specific provider configurations or troubleshooting provider-side issues.

**User Account Security Audits**

* **Scope**: Basic user registration and login functionality will be tested, but an in-depth security audit (e.g., penetration testing, encryption validation) of the user authentication process is beyond the scope of this testing phase.

**20.0 RISKS/ASSUMPTIONS**

**Risks**

* **Environment Setup Delays**: Any delay in setting up the test environment could impact the testing timeline.
* **Insufficient Test Data**: Lack of adequate sample data could limit comprehensive testing, especially for real-world transaction simulations.
* **Cross-Browser Compatibility Issues**: Unforeseen issues may arise when testing on different browsers and devices.

**Assumptions**

* The test environment will be a close replica of the production environment to ensure accurate test results.
* All necessary third-party integrations (e.g., payment gateways, shipment tracking) will be available and accessible during the testing period.

**Contingency Plans**

* If environment setup is delayed, adjust the testing timeline or use a backup environment for initial tests.
* Prepare a basic set of test data to start testing while the complete data set is finalized.

**21.0 TOOLS**

The following tools will be used to facilitate testing activities:

1. **Automation Tools**:
   * **Selenium**: For automated regression testing to verify that existing functionalities remain intact after updates.
2. **Bug Tracking**:
   * **Jira**: For logging, tracking, and managing issues found during testing, enabling clear communication between development and QA teams.
3. **Test Management Tools**:
   * **TestRail** (optional): For managing and organizing test cases, executing tests, and tracking results.
4. **Performance Testing Tools**:
   * **JMeter** (optional): To assess system performance under varying load conditions, especially during peak traffic.

**22.0 SIGNIFICANTLY IMPACTED DEPARTMENTS (SIDs)**

* **Customer Support**: Will assist in User Acceptance Testing (UAT) and gather user feedback post-launch to ensure the system meets user needs.
* **Operations**: Responsible for coordinating deployment and ensuring infrastructure is ready for production.
* **Development Team**: Engaged in addressing and resolving defects identified during testing.

**23.0 DEPENDENCIES**

**Test Environment**

* Availability of a stable test environment, including servers, database access, and third-party integrations, is critical to testing success.

**Data Availability**

* Adequate sample data (e.g., product catalog, customer profiles) must be prepared for effective simulation of real-world scenarios.

**Resource Availability**

* QA Engineers and Developers must be available throughout the testing phase to resolve defects and incorporate feedback in a timely manner.

Assume following test suite is used

| A test suite to verify the "User Login" feature | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | | **TS1** | | | | |
| **Title** | | **Verify "User Login" functionality** | | | | |
| **Description** | | **To test the different scenarios that might arise while an user is trying to login** | | | | |
|  | | | | | | |
| **#** | **Summary** | **Dependenc y** | **Pre-condition** | **Post-conditi on** | **Execution Steps** | **Expected Output** |
| TC1 | Verify that user already registered with the LIS is able to login with correct user ID and password |  | Employee ID *149405* is a registered user of LIS; user's password  is *this\_is\_password* | User is logged in | 1. Type in employee ID as *149405* 2. Type in   password *this\_is\_passwor d*   1. Click on the 'Login' button | "Home" page for the user is displayed |
| TC2 | Verify that an unregistered user of LIS is unable to login |  | Employee ID *149405xx* is not a registered user of LIS | User is not logged in | 1. Type in employee ID as *149405xx* 2. Type in   password *whatever*   1. Click on the 'Login' button | The "Login" dialog is shown with a *"Login failed! Check your user ID and*  *password"* message |
| TC3 | Verify that user already registered with the LIS is unable to login with incorrect password |  | Employee ID *149405* is a registered user of LIS; user's password  is *this\_is\_password* | User is not logged in | 1. Type in employee ID as *149405* 2. Type in   password *whatever*   1. Click on the 'Login' button | The "Login" dialog is shown with a *"Login failed! Check your user ID and password"* message |
| TC4 | Verify that user already registered with the LIS is | TC3 | This test case is executed after execution of TC3 | User is not logged in | 1. Type in employee ID as *149405* | The "Login" dialog is shown with a *"Login failed! Check* |

| A test suite to verify the "User Login" feature | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | | **TS1** | | | | |
| **Title** | | **Verify "User Login" functionality** | | | | |
| **Description** | | **To test the different scenarios that might arise while an user is trying to login** | | | | |
|  | | | | | | |
| **#** | **Summary** | **Dependenc y** | **Pre-condition** | **Post-conditi on** | **Execution Steps** | **Expected Output** |
|  | unable to login with incorrect password given twice consecutively |  | before executing any other test case |  | 1. Type in   password *whatever2*   1. Click on the 'Login' button | *your user ID and password"* message |
| TC5 | Verify that user already registered with the LIS is unable to login with incorrect password given thrice consecutively | TC4 | This test case is executed after execution of TC4 before executing any other test case | User is not logged in | 1. Type in employee ID as *149405* 2. Type in   password *whatever3*   1. Click on the 'Login' button | The "Login" dialog is shown with a *"Login failed! Check your user ID and password"* message; the security question and input box for the answer are displayed |
| TC6 | Verify that a registered user can login after three consecutive failures by correctly answering the security question | TC5 | This test case is executed after execution of TC6 before executing any other test case. Answer to the security question  is *my\_answer*. | Email sent containing new password. The email is expected to be received within 2 minute. | 1. Type in the answer as *my\_answer* 2. Click on the 'Email Password' button | Login dialog is displayed; an email containing the new password is received |
| TC7 | Verify that a registered user's account is blocked after three consecutive failures |  | Execute the test cases TC3, TC4, and TC5 once again (in order) before executing this test case | User account has been blocked | 1. Type in the answer as *not\_my\_answer* 2. Click on the 'Email Password' button | The message *"Your account has been blocked! Please contact the*  *administrator."* appears |

| A test suite to verify the "User Login" feature | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | | **TS1** | | | | |
| **Title** | | **Verify "User Login" functionality** | | | | |
| **Description** | | **To test the different scenarios that might arise while an user is trying to login** | | | | |
|  | | | | | | |
| **#** | **Summary** | **Dependenc y** | **Pre-condition** | **Post-conditi on** | **Execution Steps** | **Expected Output** |
|  | and answering the security question incorrectly |  |  |  |  |  |

create a Requirements Traceability Matrix (RTM) showing a mapping from individual requirement to test case(s).

| Table 1: A simplified mapping from requirements to test cases | |
| --- | --- |
| **Requirement #** | **Test Case #** |
| R1 |  |
| R2 |  |
| R3 |  |
| R4 |  |

Consider requirements are summarized in the table below

| **#** | **Requirement** |
| --- | --- |
| R1 | New user registration |
| R2 | User Login |
| R3 | Search book |
| R4 | Issue book |
| R5 | Return book |
| R6 | Reissue book |

# Requirements Traceability Matrix (RTM)

The following table outlines the mapping of individual requirements to the corresponding test cases for the Library Information System (LIS). This matrix ensures that all specified functionalities are covered by relevant test scenarios, particularly focusing on the "User Login" feature.

# Requirement #

**Requirement Description**

# Test

**Case #**

# Test Case Description

**R1 New user registration**

# (No relevant test cases provided)

**R2 User Login TC1 Verify that a user already registered can**

# log in with correct credentials.

**TC2 Verify that an unregistered user cannot log in.**

| **Requirement** | **Requirement** | **Test** |
| --- | --- | --- |
| **#** | **Description** | **Case #**  **TC3** |
|  | | **TC4** |
| **TC5** |
| **TC6** |

# Test Case Description

**Verify that a registered user cannot log in with an incorrect password.**

# Verify consecutive failed login attempts with incorrect passwords.

**Verify account security after three consecutive failed login attempts.**

# Verify login after correctly answering the security question.

|  | | **the security question incorrectly.** |
| --- | --- | --- |
| **R3** | **Search book** | **(No relevant test cases provided)** |
| **R4** | **Issue book** | **(No relevant test cases provided)** |
| **R5** | **Return book** | **(No relevant test cases provided)** |
| **R6**  **Summary** | **Reissue book** | **(No relevant test cases provided)** |

**TC7 Verify account blocking after answering**

* R2 (User Login) is comprehensively addressed by multiple test cases (TC1 to TC7), ensuring that various scenarios concerning user login functionality are thoroughly tested.
* R1 (New user registration), R3 (Search book), R4 (Issue book), R5 (Return book), and R6 (Reissue book) currently lack associated test cases in the provided test suite.

This RTM serves as a foundation for further development and enhancement of test cases, aiming to ensure full coverage of the Library Information System’s functionalities. Additional test cases for requirements R1, R3, R4, R5, and R6 should be developed to validate the entire system