# Test Plan (VWO.com)

[**Test Plan**](#_heading=h.30j0zll) **1**

[Objective](#_heading=h.3znysh7) 2

[Scope](#_heading=h.2et92p0) 2

[Inclusions](#_heading=h.tyjcwt) 4

[Test Environments](#_heading=h.3dy6vkm) 6

[Defect Reporting Procedure](#_heading=h.1t3h5sf) 7

[Test Strategy](#_heading=h.4d34og8) 8

[Test Schedule](#_heading=h.2s8eyo1) 9

[Test Deliverables.](#_heading=h.17dp8vu) 10

[Entry and Exit Criteria](#_heading=h.3rdcrjn) 10

[Entry Criteria:](#_heading=h.26in1rg) 10

[Exit Criteria:](#_heading=h.lnxbz9) 11

[Test Execution](#_heading=h.35nkun2) 11

[Entry Criteria:](#_heading=h.1ksv4uv) 11

[Exit Criteria:](#_heading=h.44sinio) 11

[Test Closure](#_heading=h.2jxsxqh) 11

[Entry Criteria:](#_heading=h.z337ya) 11

[Exit Criteria:](#_heading=h.3j2qqm3) 11

[Tools](#_heading=h.1y810tw) 11

[Risks and Mitigations](#_heading=h.4i7ojhp) 11

[Approvals](#_heading=h.2xcytpi) 12

# Test Plan - VMO Login and Dashboard

## 1. Objective

The goal of this Test Plan is to verify the Login and Dashboard functionalities of the VMO application, ensuring smooth user authentication, correct redirection, and proper dashboard display.

## 2. Scope

* Login: Username/Password entry, error handling, and successful login.
* Dashboard: Navigation, widgets, data display, and UI layout.

**Testing types:**

* Manual Testing
* Functional Testing
* UI/UX Testing
* Regression Testing

**Test Environments:**

* Windows 10 (Chrome, Firefox, Edge)
* MacOS (Safari)
* Android (Chrome)
* iOS (Safari)

**3. Inclusions**

* Valid and Invalid Login
* Dashboard Navigation and Widgets
* Data Display and User Profile Menu

**4. Exclusions**

* Advanced admin settings
* Third-party integrations
* Email notifications and external APIs

**5. Test Environments** **Environment URLs:**

* QA: qa.vmo.com
* Pre-Prod: preprod.vmo.com
* UAT: uat.vmo.com
* Production: app.vmo.com

**Supported Devices & Browsers:**

* Windows 10: Chrome, Firefox, Edge
* MacOS: Safari
* Android: Chrome
* iOS: Safari

**6. Defect Reporting Procedure**

* Defects logged in JIRA.
* Assigned to responsible team members (Frontend, Backend, DevOps).
* Prioritized based on severity and reported with necessary details.

**7. Test Strategy** **Steps:**

1. Create test scenarios and test cases using techniques like:
   * Equivalence Partitioning
   * Boundary Value Analysis
   * Decision Table Testing
   * State Transition Testing
   * Use Case Testing
2. Conduct **Smoke Testing** to check basic functionality.
3. Perform in-depth testing once a stable build is available.
4. Execute multiple test cycles with:
   * Smoke and Sanity Testing
   * Regression and Retesting
   * UI and Functionality Testing
5. Follow best practices like:
   * **Context-Driven Testing** (testing based on app context)
   * **Shift-Left Testing** (early-stage testing)
   * **Exploratory Testing** (unscripted testing)
   * **End-to-End Testing** (complete user flow testing)

**8. Test Schedule**

|  |  |
| --- | --- |
| **Task** | **Timeline** |
| Test Plan Creation | Day 1 - Day 2 |
| Test Case Creation | Day 3 - Day 5 |
| Test Execution | Day 6 - Day 12 |
| Summary Report | Day 13 |

**9. Test Deliverables**

* Test Plan Document
* Test Cases
* Test Execution Report
* Defect Report
* Test Summary Report

**10. Entry and Exit Criteria** **Entry Criteria:**

* Finalized requirements
* Approved test cases
* Stable test environment

**Exit Criteria:**

* All test cases executed
* Critical and major defects resolved
* Test Summary Report shared

**11. Risks and Mitigations**

|  |  |
| --- | --- |
| Risk | Mitigation |
| Environment instability | Backup environment access |
| Resource unavailability | Cross-training team members |
| Tight timelines | Prioritize critical scenarios |

**12. Approvals** Required approvals before proceeding:

* Test Plan
* Test Scenarios
* Test Cases
* Test Summary Report

Testing will move to the next phase only after approvals.