# 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 objective of this Test Plan is to validate the Login functionality and Dashboard module of the VMO application. This includes verifying user authentication, proper redirection after login, and the correct display of dashboard elements.

## 2. Scope

The following features will be tested:  
- Login Page: Username/Password entry, error handling, successful login  
- Dashboard Page: Menu navigation, widgets, data display, and overall UI layout  
  
The following types of testing will be performed:  
- Manual Testing  
- Functional Testing  
- UI/UX Testing  
- Regression Testing  
  
Testing will be conducted on the following environments:  
- Windows 10 (Chrome, Firefox, Edge)  
- MacOS (Safari)  
- Android (Chrome)  
- iOS (Safari)

## 3. Inclusions

- Login functionality (Valid and Invalid Login)  
- Dashboard Navigation  
- Dashboard Data Display and Widgets  
- User Profile Menu on Dashboard

## 4. Exclusions

- Advanced admin configurations  
- Third-party integrations  
- Email notifications or external API calls

## 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  
- iPhone: Safari

## 6. Defect Reporting Procedure

Defects will be logged using JIRA.  
The following team members will be responsible for defect triaging:  
- Frontend: Devesh  
- Backend: Sonal  
- DevOps: Prajeeth  
  
Defects will be categorized based on severity and priority, and reported to developers along with screenshots, logs, and steps to reproduce.

## 7. Test Strategy

The first step is to create test scenarios and test cases for the various features in Scope. While developing test cases, we'll use a number of test design techniques.

o Equivalence Class Partition

o Boundary Value Analysis

o Decision Table Testing

o State Transition Testing

o Use Case Testing

We also use our expertise in creating Test Cases by applying the below:

o Error Guessing

o Exploratory Testing

• We prioritize the Test Cases

Step 2: Our testing procedure when we receive a request for testing:

• First, we'll conduct smoke testing to see if the various and

important functionalities of the application are working.

• We reject the build, if the Smoke Testing fails and will wait for the stable

build before performing in depth testing of the application functionalities.

• Once we receive a stable build, which passes Smoke Testing, we perform

in depth testing using the Test Cases created.

• Multiple Test Resources will be testing the same Application on Multiple

Supported Environments simultaneously.

We then report the bugs in bug tracking tool and send dev. management

the defect found on that day in a status end of the day email.

As part of the Testing, we will perform the below types of Testing:

o Smoke Testing and Sanity Testing

o Regression Testing and Retesting

o Usability Testing, Functionality & UI Testing

• We repeat Test Cycles until we get the quality product.

Step3 – We will follow the below best practices to make our Testing better:

• Context Driven Testing – We will be performing Testing as per the context

of the given application.

• Shift Left Testing – We will start testing from the beginning stages of the

development itself, instead of waiting for the stable build.

• Exploratory Testing – Using our expertise we will perform Exploratory

Testing, apart from the normal execution of the Test cases.

• End to End Flow Testing – We will test the end-to-end scenario which

involve multiple functionalities to simulate the end user flows.

## 8. Test Schedule

The following table outlines the planned 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 Submission | 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

- Requirements finalized and shared with QA  
- Test Cases reviewed and approved  
- Test Environment available and stable

### Exit Criteria

- All planned test cases executed  
- All critical and major defects resolved  
- Test Summary Report shared

## 11. Risks and Mitigations

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

## 12. Approvals

The following deliverables require approval before proceeding:  
- Test Plan  
- Test Scenarios  
- Test Cases  
- Test Summary Report

Testing will only continue to the next steps once these approvals are done.