

TEST STRATEGY FOR MOMONDO WEBSITE PROJECT

Table of Contents

INTRODUCTION.....	3
PURPOSE.....	3
SYSTEM OVERVIEW	3
SCOPE OF TESTING	3
<i>In-scope</i>	4
<i>Out of scope</i>	4
APPROACH TO TESTING	4
<i>Principles and Application</i>	4
<i>TEAM - Planned Iterative</i>	5
<i>TEST PHASE 1 - First Sprint</i>	6
<i>TEST PHASE 2 - Second Sprint</i>	7
TEST ENVIRONMENTS REQUIREMENTS	8
TEST DATA REQUIREMENTS	8
TESTING TOOLS AND TECHNIQUES.....	9
<i>Requirements and Use Case Management</i>	9
<i>Test Management and Defect Tracking</i>	9
TESTING ROLES AND RESPONSIBILITIES	9
TEST MANAGEMENT	10
DEFECT MANAGEMENT.....	10
<i>Defect Management Process</i>	11
TEST SCHEDULE	11
<i>TEST PHASE 1</i>	11
<i>TEST PHASE 1</i>	12
REFERENCED DOCUMENTS.....	12

Introduction

The objective of the Momondo Website Project Test Strategy is to check if the functionality of the implemented features is working properly and can be used by the website's users.

Purpose

This Test Strategy will provide a high-level view of how testing will be completed for the Momondo Website Project.

Test Cases will be designed and run to test every feature listed in the system plan.

Performance, security, and load reliability will be included as part of the system tests and will be performed by an external QA team.

Performance and API testing will start as soon as the system has achieved stability.

Before bug fixes are reintroduced into the test, they must first successfully pass the unit testing, and, if necessary, integration testing.

All the test cases must be documented and run in test cycles.

95% of all test cases must pass.

All the test cases for the flight search functionality must pass.

All Medium and High severity defects must be fixed before release.

Code coverage must be 100% (including Integration and Unit Testing).

System Overview

The System Under Test (SUT) consists of the following newly created components:

- A homepage of the website, located at <https://www.momondo.com>
- A flight search results page, located at <https://www.momondo.com/flight-search>
- Features and functionality for the above web pages.

Scope of Testing

The scope of the testing is to check the functionality of the Momondo Website.

In-scope

The types of testing that are in-scope are:

1. Static Testing
 - a. Requirements
 - b. Architecture
 - c. Code Reviews
2. Unit Testing
3. Functional Testing
4. Non-functional Testing
 - a. Browser compatibility
 - b. Mobile compatibility
5. Regression Testing (where required)

Out of scope

The type of tests that are out of the project scope is:

1. Non-functional Testing
 - a. Performance
 - b. Security Testing
2. User Acceptance Testing (this will be performed in the development's environment but will be planned and executed by the client.)
3. Operational Acceptance Testing

Approach to testing

The Testing approach is using a requirement-based strategy and it features two key components:

- Test Preparation: In this stage, the team will review each available document and starts working on the functional and UI Test Cases.
- Test Execution: The team will execute the created Test Cases and document the actual results and raise defects.

Principles and Application

Principle

It is important to understand the risks to the business of any system that is developed. Some parts of the system will be critical to the running of the business and some will not. By understanding the risk profile, we can tune the amount & types of testing we complete. This

can then give a realistic scale of testing for each project and therefore we can support the estimates we put forward.

The software testing principles impacting our project include:

- Testing should show the defects obtained after the execution of the Test Cases
- Exhaustive testing is impossible in our project because of the number of key combinations the user can make.
- Early Testing is cost-efficient and should be implemented in the first stages of the process.
- Defects Cluster together and result in more bugs in the later stages of the process.
- The software test data needs to be reviewed regularly to revise the test cases to avoid the “pesticide paradox”
- The Testing Approach depends on the software development
- The application must fulfill the specified client requests i.e., flight search results, and filtering.

Application

The application in testing is a flight search website that allows the users to find better deals than the ones that they are getting from the official airline's websites by combining different airlines, partners, etc.

TEAM - Planned Iterative

Testing is a core discipline within Development Company’s development framework (TEAM – The Development Company Adaptive Model), with engagement at each of the key stages of development as shown in the diagram below.

TEAM defines six dimensions, the core dimensions of Process, Team, and Tools, and the supporting dimensions of Communication, Culture, and People. Each of these is addressed specifically in the testing discipline within the methodology.

There are two distinct software development approaches described by TEAM:

- Agile
- Planned Iterative

This project will be delivered using the Agile methodology.

Planned Iterative Principles impacting testing include:

- Testing is completed by an Independent Testing Group

- Test throughout the Project
- Focus on finding defects early in the SDLC.
- Formal Testing Scope within each iteration (Continuously verifying quality)
- Focus on testing the Architecture first.
- Test Cases derived from Use Cases.
- Tailor process and documentation for the size and complexity of the project

TEST PHASE 1 - First Sprint

Objective

The overall objective of the first sprint is to have a website stable version, capable to resolve the incoming traffic.

The testing team's objective is to build the required Test Cases from the User Stories' Acceptance Criteria and execute them by documenting the results.

Scope

The scope of testing for TEST PHASE 1 is:

- Functional Testing: test each part of the software application, by providing the appropriate input, and verifying the output and user journey against the requirements.
- UI Testing: test the interface to see if it responds to the design requirements.

Test Preparation

The documentation and User Stories are studied to create relevant Test Cases.

In the Test Case development process a proactive approach is used, initiating the process as early as possible to find and fix the defects.

Testers must continuously explore, learn and apply the most efficient approaches for data collection, generation, maintenance, and automation for any types of functional and non-functional testing.

Entry Criteria

- Verify if the test data is available and validated.
- Requirements are defined and approved.
- The test environment is set up.
- The tools and devices are available.
- QA team is prepared with the functionality overview.

Exit Criteria

- All the Test Cases are executed.
- 100% level of requirements coverage has been achieved.
- No Medium and High severity bugs left outstanding.
- Regression testing was performed (if required).

Test Execution

- Execute tests as planned
- Document test results.
- Log defects for failed Test Cases.
- Regression is performed.
- Run as many test cycles as required to complete the sprint.

TEST PHASE 2 - Second Sprint

Objective

The overall objective of the second sprint is to have a stable website version. The users can use the website to their needs without encountering any flaws. The resulted website version will be the foundation to further build upon.

The testing team's objective is to build the required Test Cases from the User Stories' Acceptance Criteria and execute them by documenting the results.

Scope

The scope of testing for TEST PHASE 2 is:

- Functional Testing: test each part of the software application, by providing the appropriate input, and verifying the output and user journey against the requirements.
- UI Testing: test the interface to see if it responds to the design requirements.
- Integration Testing:

Test Preparation

The documentation and User Stories are studied to create relevant Test Cases.

In the Test Case development process a proactive approach is used, initiating the process as early as possible to find and fix the defects.

Testers must continuously explore, learn and apply the most efficient approaches for data collection, generation, maintenance, and automation for any types of functional and non-functional testing.

Entry Criteria

- Verify if the test data is available and validated.
- Requirements are defined and approved.
- The test environment is set up.
- The tools and devices are available.
- QA team is prepared with the functionality overview.

Exit Criteria

- All the Test Cases are executed.
- 100% level of requirements coverage has been achieved.
- No Medium and High severity bugs left outstanding.
- Regression testing was performed (if required).

Test Execution

- Execute tests as planned
- Document test results.
- Log defects for failed Test Cases.
- Regression is performed.
- Run as many test cycles are required to complete the sprint.

Test Environments Requirements

The test environment will be configured as shown below:

- Production Environment (for the TEST PHASE 1) is not accessible to the general public yet.
- Pre-PROD Environment (for TEST PHASE 2) is accessible only by the internal team and the selected community testers.
- Testing is done using Chrome Browser
- Testing devices:
 - Laptop with internet connection running Windows 10
 - Samsung Galaxy S22 Mobile Phone, connected to the internet, running Android OS

Test Data Requirements

The data used for testing are:

- Airports' IATA codes and access to the internal airport's database
- Provided Google Account

Based on the combination of the provided data and the results of the test execution, the obtained data must be collected and documented for further reference.

The two sprints are linked. The relationship between them will be tested and will be the foundation for the other features that need to be implemented in the future.

Testing Tools and Techniques

The following tools will be used for testing:

- Jira – as a management tool
- Zephyr Scale as an addon for Jira to help the QA team on Test Cases
- Atlassian Confluence for any documentation of the project.
- Postman for API testing
- Chrome Version 101.0.4951.67 (Official Desktop Build)
- Chrome Version 101.0.4951.61 (Latest Android 12 Mobile Build)

All testing will be done using the Blackbox testing techniques.

Requirements and Use Case Management

The user stories will be developed in Jira using the Scrum board.

Test Management and Defect Tracking

Test Cases will be developed in Jira using the Zephyr Scale extension. The test cycles will be created using Zephyr Scale. The defects will be reported to the Jira board.

Testing Roles and Responsibilities

The following table shows the testing roles for the project, together with the individuals involved in the testing effort.

Activity	Responsibility	Name
Test Plan Creation	Test Manager	RAUL RITI delegated to CLAUDIA HENT
Test Phase Plan Creation	Test Manager	RAUL RITI delegated to CLAUDIA HENT

Test Management	Test Manager	RAUL RITI
Test Preparation, Execution & Results	QA Engineer	CLAUDIA HENT
Test Defect Submission	QA Engineer	CLAUDIA HENT
Test Summary Reporting	Test Manager	RAUL RITI delegated to CLAUDIA HENT
Test Completion Reporting	Test Manager	RAUL RITI delegated to CLAUDIA HENT
Test Environment Deployment	Release Manager	Internal Development Team

Test Management

Overall responsibility for the Testing Project and the day-to-day Test Management will be on the Test Manager, Raul Riti.

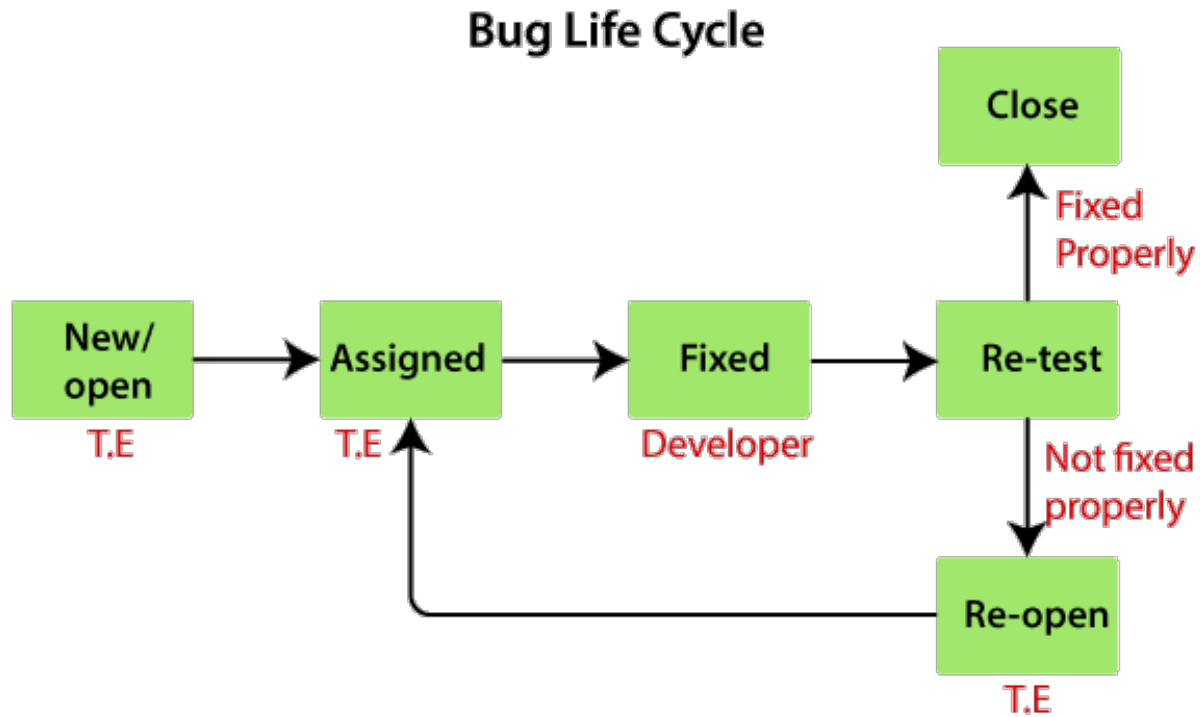
One of the key roles of the Test Management Team is to work closely with the Project Manager, Technical Lead & Lead Business Analyst to ensure that testing is integrated and executed efficiently. On that basis, the Test Management Team will attend the weekly Project Managers meeting with the Technical Team Lead and the Lead Business Analyst where all actions, issues, and risks are reviewed and progressed.

In addition, daily testing update meetings will be undertaken each morning, whereby each tester will be required to state their progress (what they did yesterday, what they intend to do today, and any issues they have preventing them from progressing).

At the end of every Testing Phase, a Testing Review Board Meeting will be arranged. The Testing Review Board (TRB) is a review board working within a project environment assembled from the key stakeholders and decision-makers for the Project. It is designed as a mechanism to review readiness for testing, and testing status, and make formal strategic project decisions on testing and portfolio management.

The Project Team (Test Management, Project Management, Business Analysis, and Development) will also attend a daily Defect Review Board Meeting. This is to be held at the end of each day, whilst Test Execution is in progress. The Defect Review Board (DRB) will assess errors and issues that arise during testing. The key purpose of the DRB is to prevent the testing from becoming distracted by low-priority issues that take them off the critical path. The DRB will carefully assess functional, technical, and data implications, as well as business impact.

Defect Management



Defect Management Process

Once a defect has been found by the QA Team a report must be made and the B.A. or the Dev Team. If the bug does not have a high level of severity and the team decides to work on it in the next sprint the defect will be relocated to the product backlog.

Defects Reports must come with testing proof, screenshots, steps to be reproduced, and a severity level.

Defects/Bugs found after each Sprint will have the following route:

- Discovery – will receive an ID number
- Categorization – will have a status after review by the B.A. or Dev Team
- Fixing the defect – the developing team will fix the defect
- Verification – defect fixed will be tested again
- Defect Closure – after the fixing and verification defect will be closed

Test Schedule

TEST PHASE 1

The following table shows the high-level testing milestones for this page of the project:

Ref No	Stage	Project Milestone	Due date
TM101	Test Preparation Stage	Phase Test Plan document completed.	02.05.2022
TM102	Test Preparation Stage	Test analysis completed on the detailed requirements and technical documentation.	05.05.2022
TM103	Test Preparation Stage	Test Conditions/Cases/Scripts completed and signed off.	09.05.2022
TM104	Test Execution Stage	Execution of Test scripts completed.	13.05.2022

TEST PHASE 1

The following table shows the high-level testing milestones for this page of the project:

Ref No	Stage	Project Milestone	Due date
TM201	Test Preparation Stage	Phase Test Plan document completed.	02.05.2022
TM202	Test Preparation Stage	Test analysis completed on the detailed requirements and technical documentation.	05.05.2022
TM203	Test Preparation Stage	Test Conditions/Cases/Scripts completed and signed off.	16.05.2022
TM204	Test Execution Stage	Execution of Test scripts completed.	20.05.2022

Referenced Documents

The following table identifies the documentation used for developing this Test Strategy:

#	Document	Author	Description
1	Test Plan ¹	Claudia Hent	The document provides information regarding what specific testing will be completed on the Project.
2	Jira Board ²	Project Manager	Jira Board
3	Zephyr Scale - Test Cases ³	Claudia Hent	Test Cases

¹ <https://hent.atlassian.net/wiki/spaces/CH/pages/622600>

² <https://hent.atlassian.net/jira/software/projects/ICHM/boards/6>

³ https://hent.atlassian.net/projects/ICHM?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.kanoah.test-manager_main-project-page#!/design?projectId=10005

4	Zephyr Scale - Test Cycles ⁴	Claudia Hent	Test Cycles
---	--	-----------------	-------------

⁴ https://hent.atlassian.net/projects/ICHM?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.kanoah.test-manager_main-project-page#!/v2/testCycles?projectId=10005