Test Plan for Daraz.pk

# 1. Introduction

## 1.1 Overview of the Project

Daraz.pk is an e-commerce platform enabling users to browse products, manage shopping carts, and complete online purchases.

## 1.2 Objectives of the Test Plan

The primary objectives are:

* **Functionality**: Validate these key features such as the User Interface (UI), Add-to-Cart functionality, Search, Filters, and Reviews & Ratings operate as intended.
* **Usability**: Ensure the application is user-friendly with intuitive navigation.
* **Performance**: Assess the efficiency of the features being tested, such as the responsiveness of the Add-to-Cart and Search functionalities, to ensure a smooth user experience.

# 2. Scope of Testing

## 2.1 Features to be Tested

* **User Interface (UI)**: Evaluate layout, design, and navigation.
* **Add-to-Cart Functionality**: Test item addition and cart management.
* **Search Functionality**: Assess the effectiveness of the search feature.
* **Filters**: Test the functionality of filters.
* **Reviews and Ratings**: Ensure the review and rating systems work correctly.

## 2.2 Features Not to be Tested

## Mobile application features.

* Integration with third-party systems not included in the scope.
* Database Testing, hardware & any other external interfaces.

## 2.3 Type of Testing

**Manual Testing**: Perform manual testing to validate functionality, usability, and performance, including exploratory and scenario-based testing.

## 2.4 Environments

* Browsers: Google Chrome.
* Operating Systems: Windows 10.
* Devices: Desktop computers and laptops.
* Environment URLs
* **QA:** qa.Daraz.pk
* **Pre-Prod:** preprod.Daraz.pk
* **UAT:** uat.Daraz.pk
* **Prod:** app.Daraz.pk

# 3. Test Strategy

## 3.1 Testing Approach

* **Manual Testing**: Execution of test cases manually to validate the application.

## 3.2 Types of Testing

* Functional Testing: Verify the functionality of the application's features such as Add-to-Cart, Search, Filters, and Reviews & Ratings.
* Performance Testing: Assess speed and responsiveness.
* Usability Testing: Ensure user-friendly design & ease of navigation.

## 4. Test Schedule

#### 4.1 Deadlines

* Test Plan Creation: 15 Sep 2024
* Test Case Creation: 16 Sep 2024
* Test Case Execution: 20 Sep 2024
* Summary Reports Submission: 25 Sep 2024

#### 4.2 Sprints

* **Sprint 1**: 1 Sep 2024 to 15 Sep 2024
* **Sprint 2**: 16 Sep 2024 to 30 Sep 2024

### 5. Resource Allocation

#### 5.1 Team Members and Roles

* Testers: Execute test cases and report defects.
* Developers: Fix defects identified during testing.
* Test Lead: Oversee testing process, manage resources.

#### 5.2 Tools and Technologies

* JIRA: Bug tracking.
* Word/Excel: Documentation and reporting.

### 6. Risk Management

#### 6.1 Potential Risks and Issues

* Resource constraints: Limited availability of team members or tools.
* Unforeseen Application Issues: Unexpected bugs or performance problems.

#### 6.2 Mitigation Strategies

* Backup Resource Planning: Prepare additional resources or personnel to handle unexpected absences or shortages.
* Adjust Resource Allocation: Reallocate resources as needed to address emerging issues or shifts in priorities.

### 7. Defect Management

#### 7.1 Process for Reporting and Tracking Defects

* Criteria: Deviations from requirements, user experience issues, and technical errors.
* Reporting Steps:
  + - Use standardized templates.
    - Provide clear reproduction steps.
    - Attach relevant screenshots and logs.
* Triaging:
  + - Assign severity and priority to each defect.
    - Delegate defects to appropriate team members based on expertise.
* Tracking Tools: Use JIRA for defect tracking and management.

#### 7.2 Roles

* Testers: Report defects.
* Developers: Resolve defects.
* Test Lead: Oversee defect management.

#### 7.3 Communication

* Regular updates to stakeholders.

### 8. Test Strategy

#### 8.1 Test Design Techniques

* **Equivalence Class Partitioning:** Divide input data into valid and invalid partitions to reduce the number of test cases.
* **Boundary Value Analysis:** Focus on testing at the boundaries of input ranges to identify edge cases.
* **Use Case Testing:** Test based on user scenarios to ensure that the system behaves as expected in real-world usage.

#### 8.2 Test Cases

* **Creation**: Develop test cases based on design techniques and prioritize them according to critical functionality and risk.

#### 8.3 Testing Procedure

* Smoke Testing: Basic functionality checks.
* In-Depth Testing: Detailed testing after stable build.
* Testing Types: Smoke, sanity, regression, retesting, usability, functionality, UI.

### 9. Entry and Exit Criteria

#### 9.1 Requirement Analysis

* Entry Criteria: Receipt of requirements documents.
* Exit Criteria: Understanding of requirements and resolution of any doubts.

#### 9.2 Test Execution

* Entry Criteria: Test scenarios and test cases signed off; application is ready for testing.
* Exit Criteria: Completion of test case reports and defect reports.

#### 9.3 Test Closure

* Entry Criteria: Test case reports and defect reports are available.
* Exit Criteria: Completion and distribution of the test summary report.

### 10. Tools

* For bug tracking: JIRA
* For documentation and reporting: Word/Excel

### 11. Review and Approval

#### 11.1 Project Manager Review Process

* Review of test plan, scenarios, and cases by Project Manager.

#### 11.2 Documents for Approval

* Test Plan
* Test Scenarios
* Test Cases
* Reports

#### 11.3 Approval Process

* Testing will proceed only after all relevant documents have been approved.