Muhammad Waqar Ur Rehman

[Email address]

Test Plan

Giphy App

**Table of Contents**

[1 Introduction 1](#_Toc167458943)

[2 Scope 2](#_Toc167458944)

[2.1 In-Scope 2](#_Toc167458945)

[2.2 Out-of-Scope 2](#_Toc167458946)

[3 Testing Strategy 2](#_Toc167458947)

[3.1 Test Objectives 2](#_Toc167458948)

[3.2 Test Assumptions 2](#_Toc167458949)

[3.3 Data Approach 2](#_Toc167458950)

[3.4 Level of Testing 2](#_Toc167458951)

[3.5 Non-Functional Testing 3](#_Toc167458952)

[3.6 User Acceptance Testing 3](#_Toc167458953)

[4 Test Execution Strategy 3](#_Toc167458954)

[4.1 Entry Criteria 3](#_Toc167458955)

[4.2 Exit criteria 4](#_Toc167458956)

[4.3 Validation and Defect Management 4](#_Toc167458957)

[5 Dependencies 4](#_Toc167458958)

[6 Tools 4](#_Toc167458959)

# Introduction

This test plan explains testing approach for the Giphy app that loads GIF photos from the API. It will cover testing for loading trending GIFs, searching GIFs, form feature.

# Scope

## In-Scope

* Testing all functions/features related to loading trending GIFs.
* Testing all functions/features related to searching GIFs.
* Testing the feedback form feature, including form validation and local storage functionality.

## Out-of-Scope

* Integration testing with external services beyond Giphy API.
* Performance testing under heavy load conditions.

# Testing Strategy

## Test Objectives

* Validate the functionality of Giphy app.
* Ensure the search functionality is working.

## Test Assumptions

* The Giphy API is stable and returns expected responses.
* The internet connection is reliable for API calls.

## Level of Testing

Unit test, Functional test.

## Non-Functional Testing

**Performance:**

* Measure loading times for initial GIFs, subsequent loads, and search results.
* Identify and optimize bottlenecks for a smooth user experience.

**Usability:**

* Validate that the Giphy UI is easy to use and the validation messages are easy to understand

## User Acceptance Testing

* Ensure the app meets user requirements for loading and searching GIFs.
* Validate user experience with the feedback form, including ease of use and error handling.
* Obtain feedback from stakeholders regarding overall usability.
* Ensure new changes do not introduce regressions in existing features.
* Re-run tests for loading trending GIFs, searching GIFs, and the feedback form after any modifications.

# Test Execution Strategy

## Entry Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Entry Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *Test environment(s) is available* |  |  |  |
| *Test data is available* |  |  |  |
| *Development has completed unit testing* |  |  |  |

## Exit criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *100% Test Scripts executed* |  |  |  |
| *80% pass rate of Test Scripts* |  |  |  |

## Validation and Defect Management

* Validate test cases against expected results.
* Document defects in a defect tracker or spreadsheet.
* Track defects through resolution, retesting, and closure.

Defects found during the Testing should be categorized as below:

|  |  |
| --- | --- |
| **Severity** | **Impact Level** |
| *1 (Critical)* | * *Functionality is blocked and no testing can proceed* |
| *2 (High)* | * *Functionality is not usable and there is no workaround but testing can proceed* |
| *3 (Medium)* | * *Functionality issues but there is workaround for achieving the desired functionality* |

# Dependencies

* Availability of test items such as APIs, test data, and test environments.
* Availability of testing resources including testers and technical support.

# Tools

* Cypress