Discord

Test plan

| **Project Name** | Discord | **Test Plan Version** | 2.0 |
| --- | --- | --- | --- |
| **Product Version** | 3.0 | **Test Plan Template Version** | 1.0 |

**Table of Contents**

[**1. INTRODUCTION 3**](#_heading=h.gjdgxs)

[**2. TESTING STRATEGY 3**](#_heading=h.30j0zll)

**3**[**. SCHEDULE**](#_heading=h.2et92p0)[**4**](#_heading=h.3as4poj)**. ROLE & RESPONSIBILITY**

[**5**](#_heading=h.1ksv4uv)**. RISK & MITIGATION**

**6. APPROVALS**

## 1. Introduction

**1.1 Purpose of the test plan document**

This document serves as a test plan. It describes the testing approach and automation framework that will test the application.

This document describes:

o Application under test overview

o Testing strategy

o Test management

o Test Risks

o Deliverables

Each section is divided into several subsections that serve to provide more detailed insight on the description and the goals of that section.

This document focuses on the functional testing of the application. Providing details on the strategy and management of performance testing and security testing is not in the scope of

this document.

**1.2 APPLICATION UNDER TEST OVERVIEW - Discord**

Discord is a [VoIP](https://en.wikipedia.org/wiki/Voice_over_IP), [instant messaging](https://en.wikipedia.org/wiki/Instant_messaging) and [digital distribution](https://en.wikipedia.org/wiki/Digital_distribution) platform designed for creating communities. Users communicate with [voice calls](https://en.wikipedia.org/wiki/Voice_over_IP), [video calls](https://en.wikipedia.org/wiki/Videotelephony), [text messaging](https://en.wikipedia.org/wiki/Text_messaging), media and files in private chats or as part of communities called "servers".Servers are a collection of persistent chat rooms and voice chat channels. Discord runs on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [iOS](https://en.wikipedia.org/wiki/IOS), iPad, [Linux](https://en.wikipedia.org/wiki/Linux), and [web browsers](https://en.wikipedia.org/wiki/Web_browser). As of December 2020, the service has over 140 million monthly active users.

**2. TESTING STRATEGY**

The company utilizes the Behavior Driven Testing approach to ensure that automated testing brings business value. The company will work to identify priorities based on the value they bring. Tools like **Cucumber BDD**, Selenium WebDriver will be used to implement the BDD testing approach.

**2.1 TEST SCOPE**

**2.1.1 Functional testing**

Functional testing will be performed to verify if the application features are developed according to the specifications. Functional testing will be done manually and using automation. BA will provide the scenarios for functional tests. However, testers also will need to write scenarios for

functional tests when required.

Functional testing will be carried out by the functional testing team.

**2.1.2. Layers of testing**

Functional testing will be on three layers of the application: **UI, API, and Database**. In each layer, tests can be executed manually or using automation.

**2.1.3. Smoke tests**

Smoke tests will be developed and executed periodically. Smoke testing will be used to identify the general stability of the 4Stay application. Testing scenarios for the smoke test will be approved by the BA or SME. Smoke tests can run against multiple environments. Smoke tests will be integrated into the CI environment and can be executed based on a schedule as well as be triggered after every deployment. end-to-end tests. Smoke tests reports will be emailed to the whole team in case of a failure. Typically, smoke test failure indicates a major issue with an application under test and requires immediate attention.

**2.1.4. Regression tests**

Regression tests will be carried out after a major code change as well as before certain milestones. Functional automated tests and End-To-End which are stable and most value will be added to the Regression. The regression test suite will grow after every sprint iteration. Regression tests will be run as often as possible to keep the Regression suite up to date and find issues early. Maintaining the regression suite will be included in the estimation of the work of test engineers.

**2.2 TEST ENVIRONMENT**

QA1 environment - Functional testing

QA2 environment - Non-Functional testing

Staging environment - UAT testing

A windows environment with Internet Explorer 8, 9, and 10, and with Firefox 27.0, as well as Google Chrome 32.0 and later should be available to each tester

## 3. SCHEDULE / Sprint Iteration

| **Iteration (or) Build No** | **Start Date** | **End Date** |
| --- | --- | --- |
| Iteration#1(sprint#1) | 09-May-2021 | 26-May-2021 |
| Iteration#2(sprint#2) | 27 - May - 2021 | 7 -June - 2021 |
| Iteration#3(sprint#3) | 8 -June - 2021 | 21 - June - 2021 |

| Iteration#4(sprint#1) | 22-June-2021 | 10-July-2021 |
| --- | --- | --- |
| Iteration#2(sprint#2) | 11-july-2021 | 27-July - 2021 |
| Iteration#3(sprint#3) | 28-July - 2021 | 14 - August - 2021 |

## 4. ROLE & RESOURCES

**4.1 Project Management**

Project Manager: reviews the content of the Test Plan, Test Strategy, and Test Estimates signs off on it.

**4.2 Test Lead**

Ensure entrance criteria are used as input before starting the execution.

Develop test plans and guidelines to create test conditions, test cases, expected results, and execute scripts.

Provide guidelines on how to manage defects. Attend status meetings in person or via the conference call line.

Communicate to the test team any changes that need to be made to the test deliverables or application and when they will be completed.

Provide on-premise or telecommute support.

**4.3 Test Team**

Develop test conditions, test cases, expected results, and execute scripts.

Perform execution and validation.

Identify, document, and prioritize defects according to the guidance provided by the Test lead.

Re-test after software modifications have been made according to the schedule.

Prepare testing metrics and provide regular status.

## 5 . RISKS & MITIGATION:

| **NO** | **Risk** | **Mitigation** |
| --- | --- | --- |
| 1 | Resource not enough, schedule | Holiday and vacation estimation is included in original schedule |
| 2 | Hardware failure during testing | Make ready It team, also maintain backup h/w resources. |
| 3 | Too many bugs to handle | Defect management plan is in place to ensure prompt communication and fixing of issues. |

**6. APPROVALS**

The undersigned acknowledge they have reviewed THE TEST PLAN document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

Signature:

Date:

Print Name:

Title:

Role:

Signature:

Date:

Print Name:

Title:

Role:

Signature:

Date:

Print Name:

Title:

Role: