IT Capstone Project 17 – Team 2 – KSU eSports Tournament Bot Test Plan

Prepared by: Patricia Helfrick

Contents

[Document History 3](#_Toc177122397)

[1 Introduction 4](#_Toc177122398)

[2 Scope 4](#_Toc177122399)

[2.1 Functions to be tested (In-Scope Testing): 4](#_Toc177122400)

[2.2 Functions not to be tested (Out-of-Scope Testing): 4](#_Toc177122401)

[3 Objectives 4](#_Toc177122402)

[3.1 Primary Objectives: 4](#_Toc177122403)

[3.2 Secondary Objectives: 5](#_Toc177122404)

[4 Test Approach 5](#_Toc177122405)

[4.1 Testing Types 5](#_Toc177122406)

[5 Test Environment 6](#_Toc177122407)

[5.1 Testing Tools 6](#_Toc177122408)

[5.2 Configuration Management 6](#_Toc177122409)

[5.3 Hardware 6](#_Toc177122410)

[5.4 Software 6](#_Toc177122411)

[5.5 Testing Environment 7](#_Toc177122412)

[6 Test Deliverables 7](#_Toc177122413)

[7 Test Schedule 7](#_Toc177122414)

[8 Test Risks and Assumptions 7](#_Toc177122415)

[8.1 Potential Risks 7](#_Toc177122416)

[8.2 Assumptions made during test planning 7](#_Toc177122417)

[9 Test Execution Criteria 8](#_Toc177122418)

[9.1 Entry Criteria 8](#_Toc177122419)

[9.2 Exit Criteria 8](#_Toc177122420)

[9.3 Suspension Criteria 8](#_Toc177122421)

[9.4 Resumption Criteria 8](#_Toc177122422)

[10 Test Cases 8](#_Toc177122423)

[11 Test Data 9](#_Toc177122424)

[12 Defect Management 9](#_Toc177122425)

[12.1 Severity List: 9](#_Toc177122426)

[12.2 Priority List: 10](#_Toc177122427)

[13 Roles and Responsibilities 10](#_Toc177122428)

[14 Addenda 11](#_Toc177122429)

# Document History

Document Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Version | Description | Author | Reviewer | Approver |
| 09/09/2024 | 1.0 | Test Plan was created | Patricia Helfrick | Jackson Stogsdill, Trinity Miller |  |

# 1 Introduction

This document provides an overview of the testing that will be conducted on the Discord Tournament Bot and will be updated throughout the duration of the project. The Discord bot will be used in the KSU eSports in-house League of Legends intramural tournaments that take place every Friday night. The project sponsor would like a Discord bot that can handle administrative tasks such as checking in players, counting MVP votes, creating teams and brackets, and keeping track of wins, losses, and player participation. This document intends to outline the testing involved with ensuring the Discord bot’s functionality, usability, and quality.

# 2 Scope

This document mainly targets the GUI testing of the Discord bot, usability testing of the bot, and functionality of the bot.

## 2.1 Functions to be tested (In-Scope Testing):

* Discord bot GUI
* Team formation and bracket creation
* Run-time of the Discord bot
* Performance of administrative tasks
* Accuracy in administrative tasks

## 2.2 Functions not to be tested (Out-of-Scope Testing):

* Nothing other than mentioned above

# 3 Objectives

## 3.1 Primary Objectives:

A primary objective of testing is to ensure that the Discord bot meets the full requirements given by the project sponsor, including quality requirements (functional and non-functional requirements) and satisfies the use case scenarios. At the end of the project development cycle, the project sponsor should find that the project has met or exceeded all expectations as detailed in the requirements and guidelines given. Any changes in the requirements given by the project sponsor or to the functionality specifications of the Discord bot will be documented and tested at the highest level of quality possible for the team within the remaining time of the project.

## 3.2 Secondary Objectives:

The secondary objectives of testing will be to identify all issues and associated risks with the testing, communicate all known issues with the project team, and ensure all issues are addressed in an appropriate matter before releasing the Discord bot to the project sponsor and to the KSU eSports Discord server. This requires careful and methodical testing of the Discord bot to ensure all issues and bugs found are dealt with appropriately.

# 4 Test Approach

The team will be testing the Discord bot in a Discord server that was set up by the team. The team performed initial testing of the bot to identify bugs and areas of improvement for the bot. The team will be using a Checklist and Quality Characteristic-Based Test Approach. The team will come up with a list of tasks, criteria, and items that the Discord bot should perform. This list will be defined before testing of the Discord bot begins. The team will perform weekly testing after changes have been made to the Discord bot to ensure quality and functionality.

## 4.1 Testing Types

* GUI Testing
  + GUI testing will include testing the UI part of the Discord bot. This testing will consist of Discord bot appearance, error messages, spelling mistakes, and ease-of-use for players and administrators.
* Integration Testing
  + Integration testing will include testing both the Discord bot and database after they have been constructed to ensure they are communicating properly and working appropriately. This testing will uncover any errors associated with the Discord bot and the database interacting with each other.
* Functional Testing
  + Functional testing will consist of testing the Discord bot and database together to find any unexpected behavior and errors. The main focus of functional testing is to ensure correctness, reliability, and accuracy of the data and output.
* Performance Testing
  + Performance testing will consist of checking the optimal time the Discord bot sends player data to the database in addition to checking the optimal time and accuracy of the Discord bot to create teams and brackets.
* User Acceptance Testing
  + User acceptance testing will consist of ensuring the Discord bot has been completed successfully and meets all the project sponsor’s needs. This testing will confirm to the team and the project sponsor that the Discord bot has been developed successfully and can be deployed to the KSU eSports Discord server for the in-house League of Legends tournaments. User acceptance testing will be done with the project sponsor before being deployed.

# 5 Test Environment

## 5.1 Testing Tools

|  |  |
| --- | --- |
| Process | Tool |
| Test Case Creation | Microsoft Excel |
| Test Case Tracking | Microsoft Excel |
| Test Case Execution | Discord |
| Test Case Management | Microsoft Excel |
| Defect Management | Microsoft Word |
| Test Reporting | PDF |
| Check List Creation | Microsoft Excel |

## 5.2 Configuration Management

Discord has certain dependencies that need to be installed with Python. All the dependencies listed can be installed with “pip3 install [package name]” in a terminal. These dependencies are listed below.

Python libraries/packages required:

* asyncio
* discord
* gspread
* python-dotenv
* Google client library
  + Google client library command: pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib

## 5.3 Hardware

For Discord Bot:

* Mobile devices:
  + iOS devices (at least iOS 14)
  + Android devices (at least Android 5)
* Browsers:
  + Google Chrome
  + FireFox 38+
  + Microsoft Edge 17+
  + Safari 11+

For SQL Server Database:

* 6 GB minimum hard-disk space
* 1 GB minimum of memory
* Minimum of x64 Processor: 1.4 GHz

## 5.4 Software

For Discord Bot:

* Windows 10+

For SQL Server:

* Windows 10+
* Minimum operating system includes minimum .NET framework

## 5.5 Testing Environment

* The testing environment will be on Discord, GitHub, and Microsoft SQL Server Management Studio

# 6 Test Deliverables

All the issues/bugs found during testing will be logged into a Microsoft Word document and will be reported to the team leader. If the team is not able to resolve the issue/bug found during testing, the team will conduct research on why our solutions are not working and will keep a separate document detailing this research. Then, the team will forward the Microsoft Word document to the project sponsor and will discuss other solutions. The team will provide reports containing information about the test cases, test scripts, and test data. The team will also provide a bug logging document to show the project sponsor.

# 7 Test Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Task Name | Start | Finish | Comments |
| Test Planning | 09/09/2024 | 09/12/2024 | Awaiting feedback |
|  |  |  |  |
|  |  |  |  |

# 8 Test Risks and Assumptions

## 8.1 Potential Risks

* Resource constraints
* Technical challenges
* The team gets behind on the task schedule resulting in delayed testing
* Bugs are not able to be fixed within the given schedule resulting in delayed testing

## 8.2 Assumptions made during test planning

* The team has assumed that the first few iterations of testing will yield several bugs and errors. As the team goes through testing iterations, they should encounter fewer and fewer bugs with each iteration. It is assumed that no bugs will be found by the last iteration of testing before deployment to the KSU eSports Discord server.

# 9 Test Execution Criteria

## 9.1 Entry Criteria

* All test hardware platforms must be successfully installed, configured, and functioning properly prior to the beginning of testing.
* All the necessary documentation and requirement information should be available to the testers so they can judge the behavior of the Discord bot during testing appropriately.
* All the standard software tools, including the testing tools, must be successfully installed and function properly.
* Proper test data is available

## 9.2 Exit Criteria

* A certain level of requirements coverage has been achieved
* No high priority or severe bugs have been identified and have been left outstanding
* All high-risk areas have been fully tested and only a minor amount of risk is left outstanding
* The schedule has been achieved

## 9.3 Suspension Criteria

* The Discord bot or the database has many serious defects which seriously limit the testing progress
* Significant change in requirements by the project sponsor
* Software/hardware problems
* Assigned resources are not available when needed by the test team

## 9.4 Resumption Criteria

* Resumption of testing will only occur when the problem(s) that caused the suspension have been resolved.

# 10 Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Name | Test Case ID | Test Step | Description | Test Data | Expected Result | Actual Result | Remarks |
| Launch current bot | 01 | 1 | Perform initial testing of the current bot to find bugs and areas of improvement. |  | Current bugs are found | Current bugs were found | Pass |
|  | 02 | 2 |  |  |  |  |  |
|  | 03 | 3 |  |  |  |  |  |
|  | 04 | 4 |  |  |  |  |  |

# 11 Test Data

The team will be using data from a Google Sheet provided by the project sponsor. The data contained in the Google Sheet provided by the project sponsor will be used as the input data. It is expected that the database will take the player data and store it appropriately. Additionally, it is expected that the Discord bot will be able to form teams and create brackets based off the player information stored in the database. During initial testing of the database to ensure quality and functionality, the team will be using fake data created based off the metrics required by the project sponsor.

# 12 Defect Management

Bugs will be categorized by their priority for when, or if, the bugs will be fixed. The Bug Severity and Priority Levels are defined in the tables below. During testing, a severity level will be assigned to all bugs found for the Test Lead to review. The Test Lead will assign a severity level to each bug to let the team know which bugs to fix first.

A bug logging document will be created for the team to record the bugs encountered, their severity level, and their priority level. During weekly tasks and testing, the team will review the bug logging document and fix the most important bugs first. For each bug, the tester will write a brief description of how they tried to fix the bug and if the issue was resolved or if it is still in progress.

## 12.1 Severity List:

|  |  |  |
| --- | --- | --- |
| Severity ID | Severity | Severity Definition |
| 1 | Critical | The Discord bot crashes or the bug causes non-recoverable conditions. Database corruption, potential data loss, and Discord bot crashes are all examples of a Severity 1 bug. |
| 2 | High | A major system component is unusable due to failure or incorrect functionality. Severity 2 bugs cause problems such as a lack of functionality, insufficient or unclear error messages that can have an impact on the user or prevent other areas of the bot and/or database from being tested. While Severity 2 bugs can have a work around, they are inconvenient or difficult. |
| 3 | Medium | Incorrect functionality of a component or process. There is a simple work around for Severity 3 bugs. |
| 4 | Minor | Documentation errors or signed off Severity 3 bugs. |

## 12.2 Priority List:

|  |  |  |
| --- | --- | --- |
| Priority | Priority Level | Priority Description |
| 1 | Must Fix | This bug must be fixed immediately and the product cannot be deployed with this bug. |
| 2 | Should Fix | There are important problems that should be fixed as soon as possible. |
| 3 | Fix When You Have Time | The problem should be fixed within the time available. If the bug does not delay the deployment of the bug or can be fixed within the project schedule, it should be fixed. |
| 4 | Low Priority | It is not important at the time to fix these bugs. Fix these bugs after all other bugs have been fixed. Currently, Nice-to-Have features and enhancements to the Discord bot are out of the current scope. |

# 13 Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Role | Team Member(s) | Responsibilities |
| Project Sponsor | Kylie Nowokunski | 1. Gives requirements and guidelines for the project 2. Provides necessary data and information pertaining to the project scope |
| Project Manager/Team Leader | Patricia Helfrick | 1. Acts as a primary contact for the project sponsor, instructor, and team members. 2. Reviews documentation and gives final approval 3. Responsible for the Project schedule and the overall success of the project. |
| QA Lead | Trinity Miller | 1. Reviews documentation 2. Ensures testing is done as scheduled |
| Test Lead | Jackson Stogsdill | 1. Reviews bug logs and documentation 2. Assigns severity and priority levels 3. Performs testing |
| Tester | Niranjanaa Jayakumar  Daniel Schroeder | 1. Performs testing and reports bugs |

# 14 Addenda

|  |  |
| --- | --- |
| Term/Acronym | Definition |
| API | Application Program Interface |
| GUI | Graphical User Interface |
| PM | Project Manager |
| QA | Quality Assurance |
| UAT | User Acceptance Testing |
| CM | Configuration Management |
| UI | User Interface |