SE 3XA3: Test Plan
Title of Project

Team #31, R-DB V2
Jason Tsui tsuij8
Hareem Arif arifh1
Student 3 name and macid

October 26, 2018

Contents

1 General Information **1**
- 1.1 Purpose . . . 1
- 1.2 Scope . . . 1
- 1.3 Acronyms, Abbreviations, and Symbols . . . 1
- 1.4 Overview of Document . . . 1

2 Plan **2**
- 2.1 Software Description . . . 2
- 2.2 Test Team . . . 2
- 2.3 Automated Testing Approach . . . 2
- 2.4 Testing Tools . . . 3
- 2.5 Testing Schedule . . . 3

3 System Test Description **3**
- 3.1 Tests for Functional Requirements . . . 3
  - 3.1.1 Area of Testing1 . . . 3
  - 3.1.2 Area of Testing2 . . . 4
- 3.2 Tests for Nonfunctional Requirements . . . 4
  - 3.2.1 Area of Testing1 . . . 4
  - 3.2.2 Area of Testing2 . . . 4
- 3.3 Traceability Between Test Cases and Requirements . . . 5

4 Tests for Proof of Concept **5**
- 4.1 Area of Testing1 . . . 5
- 4.2 Area of Testing2 . . . 5

5 Comparison to Existing Implementation **5**

6 Unit Testing Plan **5**
- 6.1 Unit testing of internal functions . . . 5
- 6.2 Unit testing of output files . . . 5

7 Appendix **6**
- 7.1 Symbolic Parameters . . . 6
- 7.2 Usability Survey Questions? . . . 6

List of Tables

1 Revision History ii
2 Table of Abbreviations 1
3 Table of Definitions 2

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Oct 25, 2018	1.0	Section 1
Date 2	1.1	Notes

1 General Information

1.1 Purpose

This document is a Test Plan document. This document will discuss and describe the testing, validation, and verification procedures to implement a discord chat bot, R-DB V2. As of document creation, the project is still undergoing major development and the test case procedures and descriptions disscussed are subject to change.

1.2 Scope

This project is a python based server hosted chat bot, which is able to take user input and output into the discord community server. The scope of testing for this program will cover user input, Discord API integration, and expected outputs.

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	**Definition**
API	Aplication Programming Interface
VoIP	Voice over Internet Protocol

1.4 Overview of Document

R-DB V2 will be following the logical structure for testing purposes. Unit testing will primarily focus on handling user input and program logic. Integration testing will be devoted to integrating the bot onto the Discord API platform. System testing will be be done by observing if chat bot commands are correct. Acceptance testing will be done by a 3rd party user to evaluate non-functional requirements.

Table 3: **Table of Definitions**

Term	Definition
Discord	Freeware VoIP application. VoIP service which connects people and creates voice chat communities
VoIP	Methodology for communicating over the internet
R-DB V2	Name of project
Black Box Testing	Functional testing by probing program with inputs. Testing is concered with output of program
White Box Testing	Structural testing by understanding how program processing occurs. Testing is concered with soundess of program
Pytest	Python testing framework
Red Discord Bot	Original program which project is based off of

2 Plan

2.1 Software Description

The software for our product consists of a Discord Bot implemented in python coding language. The environment that the software is being created in is the IDLE environment provided with python by default. The testing that will be done will also be done in the same environment in the same language.

2.2 Test Team

Our group test team is comprised of all members contributing in different aspects. As the files to be implemented have been split up into the group, testing will be done both by the creator of each file of code as well as a level of testing done by other group members to ensure higher coverage.

2.3 Automated Testing Approach

As our implementation is done with python we can make use of the ability to do automated testing through unit testing and integration testing. Unit testing will be done on the individual files that make up the product to ensure that the functions within each file are performing as intended. Integration

testing will be done to ensure that the files are interacting with one another in a cohesive manner and performing their collaborative functionality as intended.

2.4 Testing Tools

The testing tools that being intended for use consist of unittest which is the built in standard library tool that is used for testing python code. Alongside that we are considering using pytest as well which is also a complete testing tool.

2.5 Testing Schedule

Gantt Chart showing the project schedule.

3 System Test Description

3.1 Tests for Functional Requirements

3.1.1 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

Output:

How test will be performed:

3.1.2 Area of Testing2

...

3.2 Tests for Nonfunctional Requirements

3.2.1 Area of Testing1

Title for Test

1. test-id1

   Type:

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

3.2.2 Area of Testing2

...

3.3 Traceability Between Test Cases and Requirements

4 Tests for Proof of Concept

4.1 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

4.2 Area of Testing2

...

5 Comparison to Existing Implementation

With regards to comparing our implementation to the original implementation, we intend on testing the performance comparison by running a series of identical commands on both implementations to see how they perform. As the original implementation does not come with corresponding test files, we do not have a comparison for testing choices, although we have considered

the possibility of running a portion of our test plan on the original implementation to see if the overall functionality is the same. Beyond that the results of testing on our own implementation and the comparison to the original will be updated as the project continues to develop.

6 Unit Testing Plan

6.1 Unit testing of internal functions

6.2 Unit testing of output files

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

This is a section that would be appropriate for some teams.