Umaa-love

Test Plan

CS 197
Advanced Software Concepts
Machine Problem

ESGUERRA, Regina Alyssa
TRUELEN, Aaron Dominic

May 27, 2016

Table of Contents

1. Introduction

1.1. Purpose

The Test Plan, created during the Planning Phase, documents the approach used in testing the project. The document is designed to define the scope, type, risks, logistics, objectives, criteria, environment, deliverables, and schedule of the testing activities of project Umaa-love.

1.2. Project Description

Umaa-love, a play on words of Umaalab and Love, is a platform in which users can calculate compatibility between two individuals. It is a Love Calculator that takes two names separated by a comma ‘,’ as input, and outputs the result in the console. Two algorithms are used for calculations, FLAMES and TRUE LOVE. The server picks an algorithm at the beginning of the program; in the case that no algorithm is chosen, the default to be used is FLAMES. It implements a client-server architecture in which the user is the client that connects to the server.

1.3 Audience

The intended audience of this test plan are the instructors and the project developers/team who are also the testing team.

2. Test Strategy

2.1 Test Scope

2.1.1. Features to Test

The features or items to be tested are the following:

- Input - involves input validation
- Flames Algorithm - involves testing each output
- True Love Algorithm - involves testing each output based on number of digits

2.1.2. Features not to Test

The features not to be tested are the following:

- Sockets/Network IO
- Threads

2.2. Test Type

The testing type used in this project is Unit Test, as per the project specifications.

2.3. Tesk Risks

The only risk identified is when the server has exceeded the maximum number of clients that can connect to it. Once the server has reached the limit, an error is most likely to occur.

2.4. Test Logistics

The project uses the project developers as the testers themselves. This is to comply with test-driven development during the process of creating the system. The testing activities will occur once the test specification, testers, and test environment are available. Since the

setup is that the testers are the project developers and that the project specifications are provided by the instructors, only the test environment is needed to start the test execution. However, since the testers are also the project developers, it is assumed that the test environment is readily available to the tester. Hence, test execution may occur continuously during the coding process to implement continuous integration.

2.5 Test Cases

The test cases are as shown below stored in unitTests.py.

For the input verification or validation, the following tests are used:

- Testing for empty string
- Testing for strings that have one comma only “,”
- Testing for strings with more than one comma
- Testing for strings that contain only one name with no comma
- Testing for strings that contain only one name with comma
- Testing for strings with two names separated by comma
- Testing for strings with two names including numbers separated by comma
- Testing for strings with names having non-ASCII characters separated by comma

For testing the FLAMES algorithm, the following tests are used:

- Testing for input that results in output Friendship
- Testing for input that results in output Love
- Testing for input that results in output Affection
- Testing for input that results in output Marriage
- Testing for input that results in output Enemy

- Testing for input that results in output Sister

For testing the TRUE LOVE algorithm, the following tests are used:

- Testing for input that results in output of 0
- Testing for input that results in output of one digit
- Testing for input that results in output of two digits
- Testing for input that results in output of three digits
- Testing for input that results in output of more than three digits

3. Test Objectives

The test objectives are to verify that the project has achieved its functional goals. Carrying out testing seeks to check if the functionality of the program complies with the project specifications. These specifications include input verification, as well as the proper output of both algorithms included in the program.

4. Test Criteria

If the testing results produce a failure of 30% of test cases, fix all the failed cases before proceeding with further testing. Also, once 100% of the test cases have passed, declare a successful testing phrase and end the testing activities.

5. Test Environment

A device the ability to run a Python program or with Python 2.7 and below installed in the system must be available to each tester. The device must be able to handle threads and socket programming.

6. Test Deliverables

The test deliverables provided before testing begins are the:

- Test Plan document; and
- Test cases.

The test deliverables provided during testing are:

- Actual test scripts; and
- Test Data.

Finally, the test deliverables provided after testing are:

- Test Results