Factory Service Simulation Software

Test Suite Design Document

Vedant Pate   14CS30021

Shubham Kumar   14CS30033

Table of Contents

Introduction

This document is a high-level overview defining testing strategy for the Factory Service Simulation Software. Its objective is to communicate project-wide quality standards and procedures. This document will address the different standards that will apply to the unit, integration and system testing of the specified application. Testing criteria under the white box, black box, and system-testing paradigm will be utilized. This paradigm will include, but is not limited to the testing criteria, methods, and test cases of the overall design. Throughout the testing process the test documentation specifications described in the IEEE Standard 829-1983 for Software Test Documentation will be applied.

Test Objective

The objective of this test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, a broad range of tests will be exercised to achieve the goal. There will be following functions that can be performed by this application: Add machine, Add Adjuster, view average machine utilization, view average adjuster utilization, and their individual utilization. The user interface to utilize these functions is designed to be user-friendly and provide easy access to all the functions.

Process Overview

The following represents the overall flow of the testing process:

1. Identify the requirements to be tested. All test cases shall be derived using the current program Specification.
2. Identify which particular test(s) will be used to test each module.

3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
4. Identify the expected results for each test.
5. Document the test case configuration, test data, and expected results.
6. Perform the test(s).
7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
8. Successful unit testing is required before the unit is eligible for component integration/system testing.
9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

Testing Strategy

The following outlines the types of testing that will be done for unit, integration, and system testing. It includes what will be tested, the specific use cases that determine how the testing is done.

Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

Black Box Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Error guessing and Boundary Value Analysis testing on our application.

There are three equivalence classes for <u>quantity entity of add machine and add adjuster</u>:

- The set of negative Integers (which is out of range),
- Set of Integers in the range of 1 and 1000,
- Integers greater than 1000 (which is out of range).

Error will be shown for classes I and III.

White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test cases are generated that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once.

If a Machine Id already exists, then if we try to add Machine with existing machine id, it will show error.

System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case we focused only on

function validation and performance. And in both cases we used the black-box method of testing.

Function Validation Testing

The integrated "FSSS" was tested based on the requirements to ensure that we built the right application. In doing this test, we tried to find the errors in the inputs and outputs, that is, we tested each function to ensure that it properly implements the parsing procedures, and that the results are expected. The behavior of each function are contained in the Software Requirement Specification.

Function	Expected Behavior
Add Machine	Machine(s) is(are) added to the Factory.
Add Adjuster	Adjuster(s) is(are) added to the Factory.
Simulation	Average Machine Utilization and Average Adjuster Utilization is calculated and shown.
Get Machine Utilization	Individual Machine Utilization is calculated and shown.
Get Machine Utilization	Individual Machine Utilization is calculated and shown.

In addition, we tested the interfaces to ensure they are functioning as desired (i.e. check if each interface is behaving as expected, specifically verifying the appropriate action is associated with each mouse click event).

Performance testing

This test was conducted to evaluate the fulfillment of a system with specified performance requirements. It was done using black-box testing method. Following things were tested:

- Adding large number of machines and adjusters to see how much time it takes.
- Calculating the average machine and adjuster utilization for a large number of machines and adjuster to test the performance of chart generation.