MSSE SOFTWARE, INC.

Test Plan for
GolfScore

Revision 1.1

Aritra Kumar DattaChaudhury

August 13, 2021

Contents

1.0 Introduction

1.1. Objective

This document describes the test plan for GolfScore 1.1 and includes information on what is to be tested, and how the testing is to be accomplished (test methodology). Specifically, this document describes the tests to be performed, the testing schedule, resources required, entry criteria, exit criteria, dependencies, test tools, metrics and the Test Plan Requirements Matrix. This is a living test plan and must be changed to reflect Core Team needs and requirements as they arise.

The main purpose of this test is to verify the software Requirements Specifications(SRS) for GolfScore 1.1 as mentioned in Appendix A.

1.2. Project Description

GolfScore is a program used to generate reports of golfers' results for a golf tournament. The input to the program will consist of a file containing two types of records as described in SRS. The output from the program will consist of up to 3 reports as described in SRS. The program is executed via a command line interface – there is no GUI associated with the application.

1.3. Process Tailoring

The test plan for GolfScore is carried along Functional and Non-Functional testing in the framework of DVT and System Integration testing. The testing procedure has the below mentioned phases:

- **Entrance Test**: To verify that the program can correctly be executed and handle input parameter errors as specified in the SS. See Appendix C for a description of the Entrance Testing test cases, and Appendix A for the SRS.
- **Main Test:** To verify the correctness of program execution. To check if the program accurately processes the input data as specified and produces the required outputs. Further, the program's handling of input data errors and output errors is checked for correctness.

  See Appendix C for a description of the Main Testing test cases.
- **Exit Test:** To verify if the program produced the required outputs, and saved them in the correct format and in the correct location. See Appendix C for Exit Testing test cases.
- **Regression Test:** After defects must have been identified during testing and processed, all tests are run again to ensure proper behaviour.

The following references were used in creating this document:

a. Software Requirements Specification for GolfScore, Revision 1, July 18, 2017.

b. System Verification Test Plan for Advanced Color Module, Revision 2, 22 February, 2000.

2.0 Assumptions/Dependencies

It is assumed that the development team unit test their code while developing the software, and also perform integration testing. Customer validation testing is assumed to be carried out by field personnel together with the customers. For conformation with the set schedule, the program must be made available by the development team by August 26, 2021.

3.0 Test Requirements

- Entrance Tests:
  - The source code is to be written in either C or C++.
  - The program is expected to be run a Machine with Windows 2000 or later version.
  - The program is run with valid input parameters.
  - The program will run as stand-alone executable.
  - The program is run from command line or terminal.
- Main Tests:
  - The number of golf courses specified for the tournament must be from 1 to 5.
  - The number of golfers entered in the tournament can be from 2 to 12.
  - Each golfer is expected to play each course once.
  - Par for holes on each course must be either 3, 4, or 5.
  - Score earned by a golfer for each hole played is between 0 and 6 (inclusive).
  - The first set of records in the input file (course records) exist and follow the specified format for each entry.
  - There is a delimiter record that signals the end of course records.
  - A second set of records (golfer records) exist in the input file and each entry follows the specified format.
  - There is a delimiter record that signals the end of the input file.
- Exit Tests:
  - The program should produce a number of reports corresponding to the specified options.
  - The generated reports should be saved as text files in the specified output directory (or if not specified, in the directory of the input file) with the extension ".rep".
  - The course report should contain a section for each Golf Course listed in the input Course Records in the specified format. It should be saved with an output filename course.rep
  - The golfer report should contain a list of all golfers in the specified format. The list should be alphabetical with respect to the golfers' last name and should be saved with an output filename of golfer.rep.
  - The tournament ranking report should contain a list of all golfers in the specified format. The list should be in descending order of final score and should be saved with an output filename of trank.rep.

4.0 Test Tools

For the testing procedures , following tools are required:

- Defect reporting and tracking software.
- Installation media for multiple Windows versions above 2000.

5.0 Resource Requirements

The required resources are mentioned below:

- GolfScore Program version 1.1
- Three PCs capable of hosting virtual machines
- A virtualization software
- Three Test Group personnel with at least 70% of his/her time available for this effort. See Appendix A for details.

6.0 Test Schedule

No.	Test	Start	Finish
1	Test Development	13.08.21	26.08.21
2	Program Availability	26.08.21	--
3	Entrance Testing	27.08.21	02.09.21
4	Main Testing	03.09.21	15.09.21
5	Exit Testing	15.09.21	21.09.21
6	Regression Testing	22.09.21	28.09.21

7.0 Risks/Mitigation

If the program does not strictly implement the input data structure, there's high probability of input data errors.

8.0 Metrics

The following metrics data will be collected. Some will be collected prior to, and some after product shipment.

Prior to shipment:

Effort expended during DVT, SVT and Regression

# of defects uncovered during DVT, SVT and Regression, and development phase each defect is attributable to

Test tracking S-Curve

PTR S-Curve

After shipment:

# of defects uncovered and development phase each defect is attributable to

Size of software

Appendix A – Detailed Resource Requirements

No.	Test	No of personnel	No of hours
1	Test Development	3	80
2	Entrance Testing	3	40
3	Main Testing	3	80
4	Exit Testing	3	40
5	Regression Testing	2	40

- PCs that are capable of hosting virtual machines are required such that the program can be tested on multiple versions of Windows.
- A virtualization software is required such that multiple versions of Windows can be installed to test the program.

Appendix B – Detailed Test Schedule

No.	Test	Start	Finish
1	Test Development	13.08.21	26.08.21
2	Program Availability	26.08.21	--
3	Entrance Testing	27.08.21	02.09.21
4	Main Testing	03.09.21	15.09.21
5	Exit Testing	15.09.21	21.09.21
6	Regression Testing	22.09.21	28.09.21

No.	Test	Dependencies
1	Test Development	3 PCs 3 Personnel
2	Program Availability	GolfScore Program
3	Entrance Testing	3 PCs 3 Personnel Virtualization Software
4	Main Testing	3 PCs 3 Personnel Virtualization Software
5	Exit Testing	3 PCs 3 Personnel Virtualization Software
6	Regression Testing	2 PCs 2 Personnel Virtualization Software

Appendix C – Test Cases

Test No.	Test Case	Test Type
1	The program shall be written in C or C++	Non-functional
2	The program shall run on a PC running Windows 2000	Non-functional
3	The program shall run on a PC running Windows XP	Non-functional
4	The program shall run on a PC running Windows vista	Non-functional
5	The program shall run on a PC running Windows 7	Non-functional
6	The program shall run on a PC running Windows 8	Non-functional
7	The program shall run on a PC running Windows 10	Non-functional
8	The program shall run as a stand-alone executable	Non-functional
9	The program shall run from the command line prompt	Non-functional
10	Command line options “-ctg” shall be accepted	Functional
11	Command line option “-c” shall be accepted	Functional
12	Command line option “-t” shall be accepted	Functional
13	Command line option “-g” shall be accepted	Functional
14	Command line options “-c –t -g” shall be accepted	Functional
15	Command line option “-k” shall display an “unrecognizable options” message	Functional
16	Command line option “-j” shall display an “unrecognizable options” message	Functional
17	Command line option “-kj” shall display an “unrecognizable options” message	Functional
18	Command line option “-ckj” shall display an “unrecognizable options” message	Functional
19	Specifying an input filename that does not exist shall display an “input parameter error”	Functional
20	Specifying an output directory that does not exist shall display an “input parameter error”	Functional
21	Command line option “-g” shall be accepted and shall display help information	Functional
22	Calling the program as “golf -ctg in.txt golfout” where “in.txt” exists and is valid and folder “golfout” exists shall be accepted	Functional
23	Calling the program as “golf -ctg in.txt golfout dis” where “in.txt” exists and is valid and folder “golfout” exists shall be accepted	Functional
24	Calling the program as “golf -ctg in.txt golfout” where “in.txt” exists and is valid and folder “golfout” does not exist shall display an “input parameter error”	Functional
25	Calling the program as “golf -ctg in.txt golfout” where	Functional

	“in.txt” does not exist shall display an “input parameter error”	
26	The number of golf course “1” shall be accepted	Functional
27	The number of golf course “5” shall be accepted	Functional
28	The number of golf course “-5” shall return an error	Functional
29	The number of golf course “6” shall return an error	Functional
30	The number of golf course “0” shall return an error	Functional
31	Having multiple records for a golfer on the same golf courses shall be accepted, although a message should be displayed to indicating this. The first record shall be used and processing shall continue.	Functional
32	The number of golfers “0” shall return an error	Functional
33	The number of golfers “1” shall return an error	Functional
34	The number of golfers “2” shall be accepted	Functional
35	The number of golfers “12” shall be accepted	Functional
36	The number of golfers “13” shall return an error	Functional
37	Par for hole “2” shall return an error	Functional
38	Par for hole “6” shall return an error	Functional
39	Par for hole “3” shall be accepted	Functional
40	Par for hole “4” shall be accepted	Functional
41	Par for hole “5” shall be accepted	Functional
42	Golfer score per hole “7” shall return an error	Functional
43	Golfer score per hole “-1” shall return an error	Functional
44	Golfer score per hole “0” shall be accepted	Functional
45	Input data with non-numeric data where numeric data is expected shall return an error	Functional
46	Input data with numeric data where non-numeric data is expected shall return an error	Functional
47	Input data that violates delimiter constraints shall return an error	Functional
48	Input file that does not contain course records shall return an error	Functional
49	Input file that does not contain golfer records shall return an error	Functional
50	Calling the program with command line options “-ctg” shall generate 3 output files: “trank.rep”, “golfer.rep”, “course.rep”. If any of the files already exist, the user shall be prompted with a message that says the file already exists and asking whether to overwrite it or not.	Functional
51	Calling the program with command line option “-c” shall generate an output file: “course.rep”. If the file already exists, the user shall be prompted with a message that says	Functional

	the file already exists and asking whether to overwrite it or not.	
52	Calling the program with command line option “-t” shall generate an output file: “trank.rep”. If the file already exists, the user shall be prompted with a message that says the file already exists and asking whether to overwrite it or not.	Functional
53	Calling the program with command line option “-g” shall generate an output file: “golfer.rep”. If the file already exists, the user shall be prompted with a message that says the file already exists and asking whether to overwrite it or not.	Functional
54	If output cannot be saved due to insufficient permissions, the program shall display an error.	Functional