MSSE SOFTWARE, INC.

System Test Plan for
GolfScore

Revision 1.1

Silady Nave
November 21, 2023

Contents

1.0 Introduction

1.1. Objective

This Document describes the test plan for GolfScore, Release 1.1, and it includes information on what is and is not to be tested, as well as how the testing is to be executed.

This document also describes the testing schedule, the tests to be performed, dependencies, required resources, test tools and metrics. This document will have changes and any change in the Core Team needs and requirements or the structure of the team must reflect in this document

The main goal of this test is to verify the software requirements specification (SRS) for the GolfScore program as contained in Appendix A.

1.2. Project Description

GolfScore is a program that is used to generate golf tournament results for golfers along each course. The program takes an input, which consists of a text file and produces output, which consists of three text files.

1.3. Process Tailoring

The GolfScore program runs as a stand-alone program only. Therefore, the test plan is adjusted for Functional and Non-Functional testing in the framework of Design Verification and System Validation. Testing is carried out under the following phases:

- Entrance Test: Verify that the program can correctly be executed and handle input parameter errors as specified in the SRS. Refer to Appendix A for the SRS and Appendix C for a description of the Entrance Testing test cases.
- Main Test: Verify the correctness of program execution. Check if the program accurately processes the input data as specified and produces the required outputs. Furthermore, the program's handling of input and output data errors is checked for correctness. Refer to Appendix C for a description of the Main Testing test cases.
- Exit Test: Verify the program produces the required outputs. Make sure they are saved in the correct format and are in the correct locations. Refer to Appendix C for Exit Testing test cases.
- Regression Test: After defects must have been identified during testing and processing, all tests are to run again to ensure proper behavior.

The following references were used in creating this document:

a) Software Requirements Specification/Design for GolfScore, Revision 1.1 | July 18, 2017.

b) System Verification Test Plan for Advanced Color Module, Revision 2.0 | Feb 22 2000.

2.0 Assumptions/Dependencies

It is assumed that the development team unit test their code while developing the software, as well as perform integration testing. It is also assumed that the Customer Validation testing will be done by the field personnel, who will work together with the customers.

Based on the test schedule, the program must be made available by the development team by June 26, 2023.

3.0 Test Requirements

The following test requirements for GolfScore are as such:

Entrance Tests:

- Program must be written in either C or C++.
- Program runs on a Windows 2000 OS computer or any later Windows version.
- Program runs as a stand-alone executable.
- Program can run from the command line prompt.
- Program runs with valid input parameters.

Main Tests:

- The number of golf courses specified for the tournament can be 1 to 5.
- Every golfer is expected to play once for each course.
- The number of golfers entered in the tournament can be from 2 – 12.
- Par for holes on each course must be either 3, 4, or 5.
- Score earned by a golfer for each hole is played is between 0 – 6 (included).
- The course records exists in the input file and follows the specified format for each entry (as specified in the SRS).
- A delimiter record that specifies the end of course records exists.
- The golfer records exists in the input file and follows the specified format for each entry (as specified in the SRS).
- A delimiter record that specifies the end of the input file exists.

Exit Tests:

- Program generates a number of reports corresponding to the SRS specifications, up to 3 reports based on the input options.
- The generated reports are stored as text files in the directory determined from the program call line, the extension of each file should end with ".rep".
- Tournament ranking report contains a list of all golfers in the specified format. The list is in descending order of final score. It should be saved with an output filename of trank.rep.
- Golfer report should contain a list of all golfers in the specified format. The list is alphabetical with respect to the golfers' last name. It should be saved with an output filename of golfer.rep.
- Course report contains one section for each golf course specified in the input course records in the specified format and shows the correct list as specified in the SRS. It should be saved with an output filename of course.rep.

4.0 Test Tools

The following software test tools are required to assist in the testing process:

- Defect control reporting and tracking software.
- Windows OS installation software, for installing windows 2000 and later, including: XP, Vista, 7, 8, 8.1, 10 and 11.

5.0 Resource Requirements

Required resources:

- GolfScore program version 1.1
- Four computers capable of hosting virtual machines
- A virtualization software
- Two personnel from the Testing Group with at least 80% of his/her time available for this effort.

Refer to Appendix A for more details.

6.0 Test Schedule

Test Sequence	Start	Finish
Test Development	12/6/2023	26/06/2023
Program Availability	26/06/2023	---
Entrance Testing	27/06/2023	04/07/2023
Main Testing	05/07/2023	20/07/2023
Exit Testing	21/07/2023	28/07/2023
Regression Testing	29/07/2023	07/08/2023

Refer to Appendix B for more details

7.0 Risks/Mitigation

- High probability of input data errors. Compliance in the program's input data structure reduces risk.

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:

Test Sequence	Required Personnel	Total Hours
Test Development	2	60
Entrance Testing	2	20
Main Testing	2	60
Exit Testing	2	25
Regression Testing	1	30

- Computers that are capable of hosting virtual machines are required in order to test the GolfScore program on multiple Windows versions.
- A virtualization software is required in order to install multiple versions of windows and to be able to test the GolfScore program.

Appendix B – Detailed Test Schedule:

Test Sequence	Start	Finish
Test Development	12/6/2023	26/06/2023
Program Availability	26/06/2023	---
Entrance Testing	27/06/2023	04/07/2023
Main Testing	05/07/2023	20/07/2023
Exit Testing	21/07/2023	28/07/2023
Regression Testing	29/07/2023	07/08/2023

Test	Requirements
Test Development	2 Computers 2 Personnel
Program Availability	GolfScore Program
Entrance Testing	2 Computers 2 Personnel Virtualization Software
Main Testing	2 Computers 2 Personnel Virtualization Software
Exit Testing	2 Computers 2 Personnel Virtualization Software
Regression Testing	1 Computer 1 Personnel Virtualization Software

Appendix C – Test Cases:

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 on a PC running Windows 11	Non-Functional
9)	The program shall run as a stand-alone executable.	Non-Functional
10)	The program shall run from the command line prompt.	Non-Functional
11)	Command line options “-ctg” shall be accepted.	Functional
12)	Command line options “-c” shall be accepted.	Functional
13)	Command line options “-t” shall be accepted.	Functional
14)	Command line options “-g” shall be accepted.	Functional
15)	Command line options “-c –t -g” shall be accepted.	Functional
16	Command line option “-a” shall display “unrecognizable option” message.	Functional
17	Command line option “-m” shall display “unrecognizable option” message.	Functional
18	Command line option “-gtg” shall display “unrecognizable option” message.	Functional
19	Specifying an input filename that does not exist shall display an “input parameter error”.	Functional
20	Command line option “-g” shall be accepted and shall display help information.	Functional
21)	Calling the program as “golf -ctg in.txt golfout” where “in.txt” exists and is valid and folder “golfout” exists shall be accepted.	Functional
22)	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
23)	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
24)	Calling the program as “golf -ctg in.txt golfout” where “in.txt” does not exist shall display an “input parameter error”.	Functional
25)	The number of golf course “1” shall be accepted.	Functional
26)	The number of golf course “5” shall be accepted.	Functional
27)	The number of golf course “-1” shall return an error.	Functional
28)	The number of golf course “6” shall return an error.	Functional
29)	The number of golf course “0” shall return an error.	Functional
30)	Having multiple records for a golfer on the same golf courses shall be accepted. If so, a message should be displayed indicating this. The first record	Functional

	shall be used and processing shall continue.	
31)	The number of golfers “2” shall be accepted.	Functional
32)	The number of golfers “12” shall be accepted.	Functional
33)	The number of golfers “0” shall return an error.	Functional
34)	The number of golfers “1” shall return an error.	Functional
35)	The number of golfers “13” shall return an error.	Functional
36)	Par for hole “2” shall be accepted.	Functional
37)	Par for hole “3” shall be accepted.	Functional
38)	Par for hole “4” shall be accepted.	Functional
39)	Par for hole “5” shall be accepted.	Functional
40)	Par for hole “6” shall be accepted.	Functional
41)	Golfer score per hole “-1” shall be accepted.	Functional
42)	Golfer score per hole “0” shall return an error.	Functional
43)	Golfer score per hole “7” shall return an error.	Functional
44)	Input data with non-numeric data where numeric data is expected shall return an error.	Functional
45)	Input data with numeric data where non-numeric data is expected shall return an error.	Functional
46)	Input data that violates delimiter constraints shall return an error.	Functional
47)	Input file that does not contain course records shall return an error.	Functional
48)	Input file that does not contain golfer records shall return an error.	Functional
49)	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
50)	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 the file already exists and asking whether to overwrite it or not.	Functional
51)	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
52)	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
53)	If output cannot be saved due to insufficient permissions, the program shall display an error.	Functional