MSSE SOFTWARE, INC.

Test Plan for
GolfScore

Rev 1.1

John Doe
May 05,2022

Contents

1.0 Introduction

1.1. Objective

This document is an aggregation of information, which describes the entire test activity for GolfScore Release 1.1. It covers the entire testing effort (unit, development test, system verification test, and Beta). It identifies the product requirements, schedules, resource requirements (people, effort and equipment), quality, assumptions, exclusions, and risks.

A preliminary Test Plan is prepared for the Project Team during the System Phase of PEAQ Process. This Test Plan will be updated in the earliest possible time of the Implementation Phase, so that progress can be tracked during implementation.

1.2. Project Description

GolfScore Release 1.1.

The purpose of the program is to process scores from a golf tournament, and produce reports showing who won the tournament and how the golfers performed on each course played.

The program takes an input file and produces up to three output files based on input options.

1.3. Process Tailoring

Functional and Non-functional tests are planned for the project.

Types of tests:

Design Verification Test

System Verification Test

Tests will be broken into three phases:

Entrance Testing – will be performed to verify that the program can be executed as specified in the SRS. See Appendix A for a detailed description of test cases.

Main Test – will be performed to verify the corectness of the program. Main Test determines that all the requirements in the SRS have been staisfied. See Appendix A for a detailed description of test cases.

Exit Test – will be performed to verify the outputs of the program.
See Appendix A for a detailed description of test cases.

Regression Test – will be performed to verify the integrity of the GolfScore to ensure the integrity of the program.

2.0 Assumptions/Dependencies

It's assumed that unit tests and integration tests were performed by the development team.
The program must be available by 05 Jun, 2022

3.0 Test Requirements

Entrance Test:

- The program will display its title and revision number on the screen at execution time.
- The program can be run from a CLI & IDE as a standalone executable
- The program can be called with valid options and parameters.
- The program runs on a PC running Windows 2000 or any later version.

Main Test:

- The number of golf courses specified for the tournament can be from 1 to 5.
- Each golfer is expected to play each course once.
- The number of golfers entered in the tournament can be from 2 to 12.
- Each golf course has 18 holes, and par for each hole is either 3, 4 or 5 strokes.
- A golfer's score for a each hole is determined as shown in Section 2.3.2 of SRS.
- Thus score and stroke count are not the same.
- A golfer's stroke count for a particular golf course is the sum of the stroke counts for each of the 18 holes.
- A golfer's score for a particular golf course is the sum of the scores for each of the 18 holes.
- A golfer's total tournament score is the sum of his or her scores for all courses played.
- There is a delimiter for the end of Course Records and Golfer Records.

Exit Test:

- The program should generate up to 3 reports based on called options.
- The ouput should be saved in the output location or in the input location (if output location is not specified).

- *Torunament Ranking Report:*
  - Should contain a list of all the golfers with the golfer's name, the score for each course, the total tournament score, and that golfer's final standing.
  - The list will be in descending order of final score (i.e best golfer first). In this case of ties, the golfers will be listed alphabetically.
  - Output filename will be *trank.rep*

- *Golfer Report:*
  - Should contain a list of all the golfers with the golfer's name, the score for each course, the total tournament score, and that golfer's final standing.
  - The list will be in ascending order of final score (i.e best golfer first). In this case of ties, the golfers will be listed alphabetically.
  - Output filename will be *golfer.rep*

- *Course report:*
  - This report will have one section for each Golf Course specified in the input Course Records.
  - For each course, the report will show a list of all the golfers with the golfer's name, the hole-by-hole stroke count for that course, and the total score for that course, listed in descending order of score on that course.
  - The output filename will be *course.rep*

4.0 Test Tools

The following tools are needed to conduct test activities:

- Bug tracking software
- Virtualization software with installed Windows 2000 and later versions.
- IDE's with support of C/C++.

5.0 Resource Requirements

Tests will require the following resources:

- GolfScore version 1.1
- PC's capable of running Virtualization software.
- Testing personnel (at least 2 persons)

6.0 Test Schedule

Test Sequence	**Start**	**Finish**
1. Test Development	05.05.2022	05.06.2022
2. Module Availability	05.06.2022	06.06.2022
3. Entrance Testing	06.06.2022	13.06.2022
4. Main Testing	13.06.2022	27.06.2022
5. Exit Testing	27.06.2022	30.06.2022
6 Regression Testing	30.06.2022	05.07.2022

7.0 Risks/Mitigation

The program is highly dependent on the input data / An input standartization program may be needed in order to avoid future errros.

8.0 Metrics

The status and progress of tests will be recorded through the collection of various sets of data, and the Test Case

Matrices in Appendix A will regularly be updated with the status of each test case. Thus, at any time one can see how many test cases have been attempted and, of those, how many have passed.

In addition, effort, size and defect data will be collected prior to and after product shipment. Once data from enough projects has been collected, estimates of testing progress and duration will become more meaningful.

Appendix A – Test Cases

Test No.	Test Case	Test Type
1	The program shall run on a PC running Windows 2000	Non-Functional
2	The program shall run on a PC running Windows XP	Non-Functional
3	The program shall run on a PC running Windows Vista	Non-Functional
4	The program shall run on a PC running Windows 7	Non-Functional
5	The program shall run on a PC running Windows 8	Non-Functional
6	The program shall run on a PC running Windows 10	Non-Functional
7	The program shall run from a CLI	Non-Functional
8	The number of golf course '1' shall be accepted	Functional
9	The number of golf course '5' shall be accepted	Functional
10	The number of golf course '6' shall return an error	Functional
11	The number of golf course '0' shall return an error	Functional
12	The number of golf course '-1' shall return an error	Functional
13	Non-numeric data where numeric data is expected shall return an error	Functional
14	Par value “2” shall return an error	Functional
15	Par value “3” shall be accepted	Functional
16	Par value “4” shall be accepted	Functional
17	Par value “5” shall be accepted	Functional
18	Par value “-1” shall return an error	Functional
19	The number of golfers “13” shall return an error	Functional
20	The number of golfers “1” shall return an error	Functional
21	The number of golfers “2” shall be accepted	Functional
22	The number of golfers “6” shall be accepted	Functional
23	The number of golfers “12” shall be accepted	Functional
24	The number of golfers “5” shall be accepted	Functional
25	The number of golfers “-1” shall return an error	Functional
26	Stroke count “over par” shall have score value 0	Functional
27	Stroke count “par” shall have score value 1	Functional
28	Stroke count “1 under par” shall have score value 2	Functional
29	Stroke count “2 under par” shall have score value 4	Functional
30	Stroke count “3 or more under par” shall have score value 6	Functional
31	Any golfer that has two or more records for the same golf course: the additional records after the first will be ignored, a message will be displayed, and processing will continue.	Functional
32	If the file specified by filename does not exist, an “input parameter error” will be reported.	Functional
33	If the directory specified by output-directory does not exist, an “input parameter error” will be reported	Functional
34	If output-directory is not supplied, the directory that contains filename will be used.	Functional
35	If output cannot be saved due to insufficient permissions, the program shall display an error.	Functional
36	If any of the requested output reports already exists, the program will pause and ask the user if the file should be overwritten. Sample prompt: “File <file> already exists. Do you want to overwrite it? (Y/N)”. There will be a separate user prompt for each output report type requested.	Functional
37	If “Y”, the output file will be overwritten;	Functional

38	if “N”, the generated output will be discarded.	Functional
39	If the specified output report does not exist in the path specified, it will be created.	Functional
40	Calling “golf -ctg filename outuput-directory” shall generate 3 files in output directory: “trank.rep”, “golfer.rep”,”course.rep”.	Functional
41	Calling “golf -c filename output-directory” shall generate an output file: “course.rep”.	Functional
42	Calling the prtgram with command line opti son “-t” shall generate an output file: “trank.rep”. “golf -c filename output-directory”	Functional
43	Calling “golf -g filename output-directory” shall generate an output file: “golfer.rep”.	Functional
44	Command line options “-c –t -g” shall be accepted	Functional
45	Command line option “-k” shall display an “unrecognizable options” message	Functional
46	Command line option “-g” shall be accepted	Functional
47	Command line option “-t” shall be accepted	Functional
48	Command line option “-c” shall be accepted	Functional
49	Command line option “-j” shall display an “unrecognizable options” message	Functional
50	Command line option “-kj” shall display an “unrecognizable options” message	Functional
51	Command line option “-ckj” shall display an “unrecognizable options” message	Functional