SOFTWARE, INC

System Verification Test Plan for GolfScore

Version 1.1

CornPump 30/05/2024

Introduction

Objective

This document describes the test plan for the GolfScore release 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.

Description

GolfScore is an executable program that will run from a PC station.
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.

Test Description

Description

The testing plan will be broken into 4 stages:

- Stage one; Unit Testing - Will execute in parallel with the development of the program, each function will be tested according to its functionality and according to SRS's demands.

- Stage two; Module testing – As according to system description by SRS the program will consist of 4 different modules:
  Calling the program module, Input module, output module, Calculations module (this may be revised by Development team).
  Thus, stage two will consist of testing the Modules and making sure each of them is following the specification specified by SRS.

- Stage three; Main Test – Will thoroughly verify the operation of the GolfScore 1.1 the program will be tested as a whole. Various Inputs will be given to the complete program making sure the output is matching the SRS requirements, also verifying the module integrate correctly one with another.

- Stage four; Regression & Version testing – This stage will run after stage three bugs and rejections have been revised and fixed by development.
  Test team will run Main test again with added tests to make sure all critical bugs and problems has been fixed and new bugs haven't been incorporated.
  The program will be tested for all Microsoft Operating system from windows 2000 and forward.

Schedule

Adviced Schedule

The testing defined in this document shall be completed according to the following schedule:

Test Sequence	Start Time	End Time
Unit Testing	31/05/2024	1/7/2024
Module Testing	1/7/2024	15/7/2024
Main Test	1/8/2024	8/8/2024
Regression Testing	22/8/2024	29/8/2024
Version Testing	6/9/2024	10/9/2024

Assumtions

All the timelines are assumed through the lanes that development team can finish the development process within those time intervals.

For Verson Testing there will be a requirement for all those environments to be available for QA team in order to perform and complete the tests, (check dependencies for clarifications)

Real Time Scheduele

Actual testing timeline (to be filled in the future):

Test Sequence	Start Time	End Time
Unit Testing	31/05/2024	
Module Testing		
Main Test		
Regression Testing		
Version Testing		

Resource and Dependencies

- For the completion of creating the tests and the actual testing we'll need the Test team lead by CornPump (2 engineers and 3 testers) for the time between 31/5/2024 up to the scheduled end time 10/9/2024
- For the completion of Stage 4 version testing a Devops engineer will be need for the task of creating environments/VMs for all Windows versions (see Description stage 4).
  The Devops will also help to create a pipeline in order to test on those different VMs. The assumption is for the Devops to work on this between 6/8/2024 - 6/9/2024
- Unit and Module testing will follow the timelines but are dependent on Development team to do so, the schedule may change according to Dev team timeline.

Metrics

The status and progress will be recorded through the collection of various sets of data, and the Test Case Matrices in Appendix B 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.

Definitions and Acronyms

SRS – Software Requirement Specification, refers to GolfScore 1.1 revision paper.

VM- Virtual Machine

Appendix A – Detailed Resource Requirements

Test	No. Personal	Work Time
Unit Testing	2 Engineers 3 Testers	1 Month
Module Testing	2 Engineers 3 Testers	2 Weeks
Main Test	2 Engineers 3 Testers	2 Weeks
Regression Testing	2 Engineers 3 Testers	1 Week
Version Testing	2 Engineers 3 Testers 1 Devops Engineer	5 Days

Additional resources:

* Pc that can run VM in order to test various operating systems per SRS.

* VM software to run the Operating system

Appendix B – Detailed Test Schedule

Test Sequence	Start Time	End Time
Unit Testing	31/05/2024	1/7/2024
Module Testing	1/7/2024	15/7/2024
Main Test	1/8/2024	8/8/2024
Regression Testing	22/8/2024	29/8/2024
Version Testing	6/9/2024	10/9/2024

Test Sequence	Dependencies
Unit Testing	Dev team advancment Testing Team 5 PCs
Module Testing	Dev team finishes Modules Testing Team 5 PCs
Main Test	Dev team finishes Integration of Modules Testing Team 5 PCs Windows Machine
Regression Testing	Main tests bugs fixed Testing Team 5 PCs Windows Machine
Version Testing	Testing Team Regression Testing bug fixed VMs

Appendix C – Test cases & Requirements

This Appendix will detail every Requirement and Test that need to pass in order to determine the GolfScore program as a finished product.
As according to Test Description: Description Stage 2 of Test Plan the modules are separated into 4 categories, as such the Test cases tables are also separated into requirements and categories.

Requirements

No.	Requirement	Relevant Section of SRS	Notes
1	The program shall be written in C / CPP	1.3	
2	The program shall run on operting system Windows 2000 or any later version	1.3	
2.1	Test GolfScore Windows 2000 OS	1.3	
2.2	Test GolfScore Windows XP OS	1.3	
2.3	Test GolfScore Windows VISTA OS	1.3	
2.4	Test GolfScore Windows 7 OS	1.3	
2.5	Test GolfScore Windows 8 OS	1.3	
2.6	Test GolfScore Windows 10 OS	1.3	
3	Program execution time	4	GolfScore will complete its processing within one minute.

Calling Module

No.	Test Case	Relevant Section of SRS	Notes
4	Command ‘golf’ shall call the program exe with options as input	2.2	
4.1	Test ‘golf’ for big/small letters		Anything beside ‘golf’ shall not be accepted

5	Test all available input for ‘options’ parameter	2.2	
5.1.1	Hyphen ‘h’ call shall generate help info	2.2	
5.1.2	Test for big/small letters		Anything beside ‘-h’ shall not be accepted
5.2	Test filename and directory are ignored if followed by ‘-h’	2.2	
5.3.1	Hyphen ‘c’ call shall generate the Course Report	2.2	
5.3.2	Test for big/small letters		‘-c’ shall be accepted
5.4.1	Hyphen ‘t’ call shall generate the Tournament Ranking Report	2.2	
5.4.2	Test for big/small letters		‘-t’ shall be accepted
5.5.1	Hyphen ‘g’ call shall generate the Golfer Report	2.2	
5.5.2	Test for big/small letters		‘-g’ shall be accepted
5.6	Test combined options	2.2	‘-gt’, ‘-gtc’, ‘-ct’ … are allowed
6	Test input for input file argument		
6.1	Test missing path in name	2.2	The input file should be taken from the exe directory
6.2	Test full path		Check right input file is used
6.3	Test for buffer overflow from input file name		Max file name should be given
6.4	Test for non ASCI letters in		Undefined letters should not be accepted
7	Test input for output directory argument		
7.1	Test missing directory input	2.2	The output should be taken from the exe directory
7.2	Test full path does not exist		Check the directory is created
7.3	Test full path exist		Test the file created in the directory
7.4	Test for buffer overflow from output directory name		Max file name should be given
7.5	Test for non ASCI letters in		Undefined letters should not be accepted

Input module

No.	Test Case	Relevant Section of SRS	Notes
8	Test output file name is equal to output file argument	2.4	
9	Test Course Record file format is correct	2.4 / 2.4.1	
9.1	Test 1st column is blank	2.4.1	
9.2	Test 2-19 columns are filled with course names	2.4.1	Test against DB valid names
9.3	Test 20th column course identifier	2.4.1	Test against DB valid names
9.4	Test 21-38 colums are filled with pars	2.4.1	
9.5	Test end of line at the end of each Record	2.4.2	
10	Test Golfer Record file format is correct		
10.1	Test 1st column is blank	2.4.3	
10.2	Test 2 is filled with course identifier	2.4.3	Test against DB valid names
10.3	Test 3-9 columns are blank	2.4.3	
10.4	Test 10-29 are filled with golfer names	2.4.3	
10.5	Test 30th column is blank	2.4.3	
10.6	Test columns 31-48 are filled with stroke counts	2.4.3	Test for Integers within possible range (1-100)
10.7	Test end of line at the end of each Record	2.4.3	

Output module

No.	Test Case	Relevant Section of SRS	Notes
11	Each input argument has correct output file with the right format. Files will be located in the output directory argument	2.5	

12	Test Tournament Ranking report format	2.5.1	
12.1	1st column consist of golfer names		Test against input file
12.2.1	Test columns 2-19 consist of course names		Test against input file
12.2.2	Test columns 2-19 data consist of integers		
12.3.1	Test column 20 consist integer		
12.3.2	Test column 20 data consist the score as should be calculated		Test against input file With rules as per SRS 2.3.2
12.4.1	Test column 21 data is ranking		Test integers are from 1 to number of golfers
12.4.2	Test column's 21 ranking is valid		Highest score highest rank
12.5	Test file name is trank.rep	2.5.1	
13	Test Tournament Ranking report format	2.5.2	
13.1	1st column consist of golfer names		Test against input file
13.2.1	Test columns 2-19 consist of course names		Test against input file
13.2.2	Test columns 2-19 data consist of integers		
13.3.1	Test column 20 consist integer		
13.3.2	Test column 20 data consist the score as should be calculated		Test against input file With rules as per SRS 2.3.2
13.4.1	Test column 21 data is ranking		Test integers are from 1 to number of golfers
13.4.2	Test column's 21 ranking is valid		alphabetically by last name
13.5	Test file name is golfer.rep	2.5.2	
14	Test Course report format	2.5.3	
14.1	Test for 18 section	2.5.3	One for each course played
14.2	Test each section name is matching with a crouse from the input	2.5.3	One for each course played
14.3.1	Test 2nd colums for each section is filled with names		
14.3.2	Test each name matching name from input file		
14.4.1	Test colums 3-19 consist of Integers		Range 1-100
14.4.2	Test stroke count is matching		Test against input file
14.5	Test total score has been created		Test against input file With rules as per SRS 2.3.2
14.6	Test list is by score - ascending		
14.7	Test file name is courser.rep	2.5.3	

Calculations/Engine module

No.	Test Case	Relevant Section of SRS	Notes
15.1	Test number of golf courses is accepted	2.3.1	Range 1-5
15.2	Test number of golf courses is rejected with error message	2.3.1	Range != 1-5 Test for appropriate error message
15.3	Test for number of golf courses not an integer	2.3.1	Test for appropriate error message
16.1	Test number of golfers is accepted	2.3.1	Range 2-12
16.2	Test number of golfers is rejected with error message	2.3.1	Range != 2-12 Test for appropriate error message
16.3	Test for number of golfers not an integer	2.3.1	Test for appropriate error message
17	Test each golf course has exactly 18 holes	2.3.1	
18	Test par for each hole is 3-5 integer	2.3.1	
19	Test each course has unique golfer names	2.3.1	
20	Test score per hole per golfer is accurate	2.3.1	Refer to section 2.3.2 SRS calculation method. This should be tested against an oracle/ pre determined score
21	Test every input cell that is expected to be integer with non integer value	2.6.2	Check that correct error message appears
22	Test par values are not in range 3-5	2.6.2	Check that correct error message appears
23.1	Test same course record with multiple runs from the golfers	2.6.2	Check that correct error message appears
23.2	Test same course record with multiple runs from the golfers	2.6.2	Program ignores duplicate runs (the first one is to be calculated)
24	Test engine on multiple file that already exist	2.6.3	

24.1	Test file name already exists pop up message	2.6.3	Check that correct pop up message appears
24.2	Test pop up message ‘Yes’	2.6.3	Check that file is overwritten
24.3	Test pop up message ‘No’	2.6.3	Check that file is not overwritten