

MSSE SOFTWARE, INC.

**Test Plan for  
MSSE-GolfScore**

**Rev. 1.1**

**April 9, 2022**

# Contents

<b>1.0</b>	<b>INTRODUCTION</b>	<b>3</b>
<b>1.1.</b>	<b>Objective</b>	<b>3</b>
<b>1.2.</b>	<b>Project Description</b>	<b>3</b>
<b>1.3.</b>	<b>Process Tailoring</b>	<b>3</b>
<b>1.4.</b>	<b>Referenced Documents</b>	Error! Bookmark not defined.
<b>2.0</b>	<b>ASSUMPTIONS/DEPENDENCIES</b>	<b>4</b>
<b>3.0</b>	<b>TEST REQUIREMENTS</b>	<b>5</b>
<b>4.0</b>	<b>TEST TOOLS</b>	<b>6</b>
<b>5.0</b>	<b>RESOURCE REQUIREMENTS</b>	<b>7</b>
<b>6.0</b>	<b>TEST SCHEDULE</b>	<b>7</b>
<b>7.0</b>	<b>RISKS/MITIGATION</b>	<b>8</b>
<b>8.0</b>	<b>METRICS</b>	<b>8</b>
	<b>APPENDIX A – DETAILED RESOURCE REQUIREMENTS</b>	<b>10</b>
	<b>APPENDIX B – DETAILED TEST SCHEDULE</b>	<b>11</b>
	<b>APPENDIX C - DETAILED TEST CASES</b>	

## **1.0 Introduction**

### **1.1. Objective**

This document specifies the entire Test Plan and activities for GolfScore Release 1.1 which is done by MSSE Software INC. These activities include (unit, development test, system verification test, and Beta), as well as the test methodology will be highlighted. The document will describe the testing schedule, resources required, entry criteria, exit criteria, dependencies, test tools, metrics and the Test Plan Requirements Matrix.

This document serves as a guideline which will constantly adapt and reflect any changes in the requirements and the team as well.

### **1.2. Project Description**

The MSSE software is on showing a GolfScore for golfers. The project is about processing scores from a golf tournament, by showing the reports on who won as well as displaying individual performances of the golfers taking part in the tournament.

### **1.3. Process Tailoring**

The software for the GolfScore will be developed with c or C++ and it intended to run on Windows PC running 2000 or later versions. The program will be a command line interface with no GUI features. Additionally, the program will run as a stand-alone executable.

The following are the various tests planned for this project:

Verification testing – The testing here will be done in parts; this is to ensure that the requirements and the key functionalities of the GolfScore software are tested. To which specification testing, functional testing, Equivalence partitioning and boundary level analysis will be carried. This to ensure wider selection of test cases and coverage.

Entrance Testing – The GolfScore SRS touches on data input and output, as such this test to verify different parameters and file formats are tested.

Regression testing – Here testing will commence as early as possible so every time a new function is developed and added, the previous feature will be regressively tested.

### **Reference Document**

The main reference document for this project will be the Software Requirement Specification (SRS), Revision 1.1 July 18, 2017.

## **2.0 Assumptions/Dependencies**

Assumptions per SRS 1.1

- Golfer are expected to play each course once, for each course play is  $>1$
- Number golf course for the tournament can be from 1 to 5. For each tournament the minimum number of course is  $\geq 1$  and the max is  $<5$ .
- The number of golfers 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 in Section 2.3.2 of the SRS, -The golfer score for a particular golf course is the sum of the score for each of the 18 holes.
- The lower a golfer's stroke count (relative to par), the higher his or her score for that hole.

### **Dependencies:**

- Golfers must complete the course.
- The score is based on the par and at least one stroke.

**Entry Criteria:**

- Setting up test environment with the necessarily tools and devices.
- Checks to ensure all preconditions are met.
- Availability of partial or complete testable code.
- Test cases should be ready.
- All requirements should be defined and approved.

**Exit Criteria:**

- All testing activities should be carried out.
- Closure of all identified defects.
- No critical bug should be left out during the testing process.
- When functional coverage has been achieved.

**3.0 Test Requirements**

Requirement Number	Requirement Name	SRS Section
1	Calling GolfScore	2.2
2	Tournament assumptions	2.3.1
3	Scoring	2.3.2
4	Data Input <ul style="list-style-type: none"><li>• CourseRecords</li><li>• DelimiterRecord</li><li>• GolferRecords</li></ul>	2.4

	<ul style="list-style-type: none"> <li>• DelimiterRecord</li> </ul>	
5	Data Output <ul style="list-style-type: none"> <li>• Tournament Ranking Report</li> <li>• GolferReport</li> <li>• CourseReport</li> </ul>	2.5
6	Error Handling <ul style="list-style-type: none"> <li>• InputParameterErrors InputDataErrors</li> <li>• Errors on Output</li> </ul>	2.6
7	Deliverables	3
8	Performance Testing <ul style="list-style-type: none"> <li>• Processing time &lt;1</li> </ul>	4

#### 4.0 Test Tools

Test management	Tool
Test Design	qTest
Defect Tracking	Confluence, PVCS Tracker
Functional Testing	Manual, JUnit
Performance Testing	JMeter
Test Coverage Monitor	JUnit
Hardware	Monitors

## 5.0 Resource Requirements

### Resource requirements for MSSE software Testers:

- Test manager
- Test Automation engineer
- System Tester
- Test Designer

### Other requirements:

PC running windows 2000 or later for performance testing.

### Test Schedule

Below is the proposed schedule for testing activities for this project.

Test Sequence	Start Date	End Date
Test Development	2022-05-2	2022-05-8
Module availability	2022-05-9	2022-05-27
SVT Entrance Testing	2022-05-16	2021-05-20
SVT Main Testing	2022-05-23	2021-05-27
Building regression test suite	2022-05-30	2021-06-3
Performance Testing	2022-05-30	2021-06-6

The actual testing proposed for this project was completed as follows:

Test Sequence	Start Date	End Date
Test Development	2022-05-2	2022-05-5

Module availability	2022-05-9	2022-05-20
SVT Entrance Testing	2022-05-16	2021-05-27
SVT Main Testing	2022-05-25	2021-06-3
Building regression test suite	2022-06-15	2021-06-27
Performance Testing	2022-06-30	2021-07-6

## 6.0 Risks/Mitigation

Requirements change Risk

To mitigate this risk the test plan is scheduled six weeks ahead of time, to cater for any change or delays.

Environmental/ System Down time Risk

Requests to fix will be done as a personal responsibility unless there is an escalation. The system down time is catered for within the project timeline and two weeks of buffer to accommodate any delays.

## 7.0 Metrics

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

No.	Metrics
1	Total number of test cases
2	Number of test cases passed



3	Number of test cases blocked/failed
4	Number of defects found.
5	Number of critical defects found.
6	Number of planned test hours.
7	Number of actual test hours.
8	Number of bugs found after launched.

## Appendix A – Detailed Resource Requirements

Role	Name	Responsibility
Test Manager	Mohammed Salah	<ul style="list-style-type: none"><li>• Planning, coordination, and control of test activities</li><li>• Development and definition of test methods and test tools</li></ul>
Test Designer	Diago Jota	<ul style="list-style-type: none"><li>• Identifying and describing test techniques</li><li>• Identifying the appropriate supporting tools</li><li>• Evaluate test effort</li></ul>
System tester	Takumi Minamino	<ul style="list-style-type: none"><li>• Execute tests</li><li>• Report defects</li><li>• Defects follow up</li><li>• Log results</li></ul>
Test automation	Roberto Firmino	<ul style="list-style-type: none"><li>• Set up test data</li><li>• Creates test assets</li><li>• Set up test environment for automation</li></ul>

## Appendix B – Detailed Test Schedule

Milestone Description	Start	No. of Days
Test Development	2022-05-2	7
Module availability	2022-05-9	25
SVT Entrance testing	2022-05-16	7
SVT main testing	2022-05-25	15
Build integration test	2022-06-15	8
Performance testing	2022-06-30	8

From the schedule, the first test will start with the Test Development. It is possible for some test to run concurrently. The Module availability will span for more than three weeks which will continue running with SVT Entrance testing and SVT main testing. This process will not interfere but rather, it will complement each other.

## Appendix C – Test Case

### Tournament Assumptions

ID	Test Scenario	Pre-Condition	Test Script	Expected Result	Expected Result
1	Lower range of number of Golf courses	Input file c:\in.txt	<ol style="list-style-type: none"><li>1. In the in.txt enter no of tournaments 1</li><li>2. Enter command &gt;golf -cgt c:\in.txt golfout</li></ol>		Report generated
2	Upper range of number of Golf courses	Input file c:\in.txt	<ol style="list-style-type: none"><li>1. In the in.txt enter no of tournaments 5</li><li>2. Enter command &gt;golf -cgt c:\in.txt golfout</li></ol>		Report generated
3	Exceed range of number of Golf courses	Input file in.txt	<ol style="list-style-type: none"><li>1. In the in.txt enter no of tournaments 6</li><li>2. Enter command</li></ol>		Report not generated

			>golf -cgt c:\in.txt golfout		
4	Lower range of number of Golf courses	Input file in.txt	<ol style="list-style-type: none"> <li>1. In the in.txt enter no of tournaments 0</li> <li>2. Enter command &gt; golf -cgt c:\in.txt golfout</li> </ol>		Report not generated

#### Data Input

ID	Test Scenario	Pre-Condition	Test Script	Expected Result	Expected Result
1	Lower limit of number of Golf courses	Input file in.txt  Output is to be placed in C:\golfout	<ol style="list-style-type: none"> <li>1. In the file in.txt add course and label C1.</li> <li>2. Save file</li> <li>3. Enter command &gt;golf -cgt</li> </ol>		Course report should be generated with right data.

			c:\in.txt golfout		
2	Upper limit of number of Golf courses	Input file in.txt  Output is to be placed in C:\golfout	1. In the file in.txt add course and label C1.  2. Save file  3. Enter command >golf -c c:\in.txt golfout		Course report should be generated with right data.
3	6 golf courses	Input file in.txt  Output is to be placed in C:\golfout	1. In the file in.txt add course and label C1.  2. Save file  3. Enter command >golf -cgt c:\in.txt golfout		Error should be displayed 'too many course names'
4	No golf courses	Input file in.txt  Output is to be placed in C:\golfout	1. In the file in.txt add course and label C1.  2. Save file  3. Enter		Error should be displayed 'Add a course names'

			command >golf -cgt c:\in.txt golfout		
--	--	--	---	--	--

#### Error handling

ID	Test Scenario	Pre-Condition	Test Script	Expected Result	Expected Result
1	Input parameter Errors	Input file in.txt Output is to be placed in C:\golfout	In the in.txt, enter the characters * & invalid characters Enter the command >golf -c c:\in.txt golfout		Appropriate message should be displayed as report not generated.
2	Input data Errors	Input file in.txt	In the in.txt file enter nonnumeric data.		Appropriate message should be displayed as report not generated.

Performance

ID	Test Scenario	Pre-Condition	Test Script	Expected Result	Expected Result
1	Time to complete		Run the program		Process completed < 1 minutes.